



ACESSIBILIDADE INTEGRADA NUMA PLATAFORMA INTERNET BANKING

INÊS MARTINS MARQUES

Outubro de 2020

ACESSIBILIDADE INTEGRADA NUMA PLATAFORMA INTERNET BANKING

Inês Martins Marques

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas de Informação e Conhecimento**

Orientador: Professor Doutor Alexandre Bragança

Porto, outubro 2020

Dedicatória

A todos que me apoiaram e me incentivaram a chegar até aqui. Obrigada!

Resumo

A implementação de acessibilidade na web permite que os websites, ferramentas e tecnologias sejam desenhadas e implementadas para que todas as pessoas as consigam utilizar, incluindo as pessoas com deficiência ou algum tipo de incapacidade.

No desenvolvimento deste projeto em contexto empresarial, a acessibilidade não é apenas um requisito ético, como também político, pois existem diversos países onde é obrigatório a implementação da acessibilidade. E, tendo em conta que a plataforma *ebankIT* está na base da maioria dos projetos da empresa, surgiu a necessidade de estar em conformidade com o requisito de acessibilidade.

Neste momento, a solução *ebankIT* não tem em conta a acessibilidade, pelo que surgiu a oportunidade de identificar e documentar boas práticas de implementação de acessibilidade, seguindo leis e padrões, e adaptar e/ou desenvolver novos componentes acessíveis e ainda sugerir melhorias a implementar em soluções futuras.

É esperado que esta análise e desenvolvimento contribua para que a organização consiga estar de acordo com a lei e que seja um estudo para reutilização em futuras implementações e transmissão de conhecimento por toda a equipa que de alguma forma trabalha com esta solução, utilizando boas práticas.

Palavras-chave: Acessibilidade, *Internet Banking*, Acessibilidade na *Web*, WCAG

Abstract

The implementation of accessibility on the web, allows the websites, tools and technologies are designed and implemented so that everybody can use them, including people with disabilities.

In the development of this project in a business context, accessibility is not only an ethical requirement, but also a political one, as there are several countries where it is mandatory to implement accessibility. And, taking into account that the ebankIT platform is at the base of most of the company's projects, the need arose to comply with the accessibility requirement.

At this time, the ebankIT solution does not take into account accessibility, so the opportunity arose to identify and document good practices for implementing accessibility, following laws and standards, and to adapt and / or develop new accessible components and also suggest improvements to be implemented in future solutions.

It is expected that this analysis and development will contribute to the organization being able to comply with the law and that it will be a study for reuse in future implementations and transmission of knowledge by the entire team that somehow works with this solution, using good practices.

Keywords: Accessibility, Internet Banking; Web Accessibility; WCAG

Agradecimentos

Em primeiro lugar, quero agradecer a toda a minha família que sempre me apoiou e nunca me deixou desistir deste objetivo. Em especial aos meus pais por sempre me apoiarem e incentivarem no meu caminho académico.

Muito obrigado ao orientador Professor Doutor Alexandre Bragança por me ter orientado e ajudado continuamente para a realização desta dissertação.

Ao Instituto Superior de Engenharia do Porto e a todos os seus docentes que, durante o Mestrado em Engenharia Informática, contribuíram para o meu conhecimento e formação.

Por último, agradeço à ebankIT e ao meu supervisor João Aguiar por me proporcionar a oportunidade de realizar o presente projeto e a todos os colegas que me apoiaram e passaram conhecimento proporcionando o meu crescimento profissional e académico.

Índice

1	Introdução.....	1
1.1	Contexto	1
1.1.1	Empresa ebankIT	1
1.1.2	Acessibilidade	3
1.2	Problema.....	3
1.3	Objetivos.....	4
1.4	Abordagem.....	4
1.5	Estrutura do documento	5
2	Estado de arte.....	6
2.1	Enquadramento teórico	6
2.1.1	Leis de acessibilidade na <i>web</i> na União Europeia.....	6
2.1.2	Padrões de acessibilidade na <i>web</i>	8
2.1.3	Tecnologias adaptativas/ assistivas.....	12
2.1.4	Usabilidade <i>web</i>	13
2.2	Enquadramento tecnológico	14
2.2.1	Ferramentas de avaliação de acessibilidade	14
2.2.2	Accessible Rich Internet Applications.....	18
2.2.3	Leitores de ecrã.....	19
2.2.4	Aplicações <i>web</i> acessíveis	22
2.3	Resumo.....	25
3	Análise de valor	28
3.1	Novo Modelo de Desenvolvimento de Conceitos.....	28
3.1.1	Identificação de oportunidade	29
3.1.2	Análise de oportunidade.....	30
3.1.3	Geração e aperfeiçoamento de ideias	30
3.1.4	Seleção de ideais.....	30
3.1.5	Desenvolvimento do conceito e tecnologia.....	31
3.2	Modelo Canvas.....	31
4	Análise e Desenho	34
4.1	Análise da solução atual	34
4.1.1	Aplicação <i>web</i>	34
4.1.2	Linguagens de programação e marcação	35
4.1.3	Solução atual.....	36
4.2	Engenharia de Requisitos	39
4.2.1	Requisitos não-funcionais.....	39
4.3	Estudo de soluções	40
4.3.1	Leitor de ecrã	40
4.3.2	Navegação por teclado	41

4.3.3	Headings.....	42
4.3.4	Formulários	43
4.3.5	Elementos.....	46
4.4	Resumo	49
5	Implementação	52
5.1	Contextualização do projeto	52
5.2	Integração de acessibilidade	55
5.3	Componentes.....	56
5.3.1	Skip navigation.....	56
5.3.2	Header.....	57
5.3.3	Menu.....	58
5.3.4	Seletores de conta.....	59
5.3.5	Caixas de texto	59
5.4	Páginas.....	60
5.4.1	Login.....	60
5.4.2	Homepage.....	61
5.4.3	Mensagens.....	62
5.4.4	Meus produtos.....	62
5.4.5	Detalhes de conta	63
5.4.6	Transferências	63
5.5	Tratamento de erros.....	64
6	Avaliação	66
6.1	Abordagem	66
6.2	Identificação de indicadores	67
6.3	Metodologias de avaliação.....	67
6.3.1	Crterios de avaliação	68
6.3.2	Ferramentas de avaliação.....	68
6.4	Avaliação de resultados.....	69
7	Conclusão	72
7.1	Resultados.....	72
7.2	Limitações	73
7.3	Trabalho futuro	73

Lista de Figuras

Figura 1 - Arquitetura do produto ebankIT (ebankIT, 2018b).....	2
Figura 2 - Relação entre componentes de acessibilidade na web	9
Figura 3 - Do estudo “Screen Reader User Survey #8”, a percentagem de utilização de leitores de ecrã (WebAIM, 2019).....	21
Figura 4 - Website acessível: Banco BPI (BPI, 2020).....	24
Figura 5 - Website acessível: Millenium BCP (Milleniumbcp, 2020)	25
Figura 6 - Novo modelo de desenvolvimento de conceitos (NCD) (Koen, 2001).....	29
Figura 7 – Diagrama de componentes da aplicação Internet banking da ebankIT (ebankIT, 2018b).....	36
Figura 8 - Exemplo da fusão de <i>master page</i> com <i>content page</i>	38
Figura 9 - Exemplo da estrutura de <i>headings</i>	43
Figura 10 - <i>Login</i> da aplicação <i>ebankit</i>	53
Figura 11 - Exemplo conteúdo de detalhe de conta	54
Figura 12 - Exemplo conteúdo de transferências	55
Figura 13 – Resultado visual de <i>Skip navigation</i> da solução <i>ebankIT</i>	56
Figura 14 - <i>Skip navigation</i> da solução <i>ebankIT</i>	57
Figura 15 - <i>Header</i> da solução <i>ebankIT</i>	57
Figura 16 - Exemplo de seletor de contas da solução <i>ebankIT</i>	59
Figura 17 - Resultado visual e código de uma caixa de texto	60
Figura 18 - Últimas transações realizadas pelo utilizador presente na <i>homepage</i> da solução <i>ebankIT</i>	61
Figura 19 - Submenu presente na página “Mensagens” da solução <i>ebankIT</i>	62
Figura 20 - Componente de exportação de informação na solução <i>ebankIT</i>	63
Figura 21 - Exemplo de erro de entrada na solução <i>ebankIT</i>	64
Figura 22 - Exemplo de tratamento de erros na solução <i>ebankIT</i>	65

Lista de Tabelas

Tabela 1 - Diretrizes WCAG	10
Tabela 2 - Níveis de critérios de sucesso	12
Tabela 3 - Vantagens e desvantagens da utilização de ferramentas Lighthouse, WAVE e Accessibility Insights for Web	17
Tabela 4 - Comparação de leitores de ecrã.....	22
Tabela 5 - Modelo Canvas.....	31
Tabela 6 - Requisitos não funcionais segundo o modelo de FURPS+	39
Tabela 7 - Resultados de páginas antes da adaptação de acessibilidade através do <i>lighthouse</i>	69
Tabela 8 - Critérios avaliados pelas ferramentas.....	70
Tabela 9 - Critérios avaliados por testes manuais	71

Acrónimos e Símbolos

Lista de Acrónimos

ARIA	Accessible Rich Internet Applications
ATAG	Authoring Tool Accessibility Guidelines
CSS	Cascading Style Sheets
EAA	European Accessibility Act (Lei Europeia da Acessibilidade)
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JAWS	Job Access With Speech
NCD	New Concept Development
NVDA	NonVisual Desktop Access
PDF	Portable Document Format
SVG	Scalable Vector Graphics
UAAG	User Agent Accessibility Guidelines
UE	União Europeia
WAI	Web Accessibility Initiative
WAVE	Web Accessibility Evaluation Tool
WCAG	Web Content Accessibility Guidelines
W3C	World Wide Web
TTS	Text-to-speech

1 Introdução

No âmbito do Mestrado em Engenharia Informática no ISEP, esta dissertação foca-se no estudo da acessibilidade na web e a sua aplicação numa solução em contexto empresarial. Neste capítulo é apresentada a contextualização do problema, os principais objetivos, a abordagem a seguir e a estrutura do documento.

1.1 Contexto

O presente trabalho foi realizado em contexto empresarial, onde é esperado a integração da acessibilidade na solução de *internet banking* da empresa *ebankIT*. Esta necessidade surgiu por cada vez mais ser exigido o requisito da acessibilidade nos projetos da empresa. Além disso, é obrigatório por lei em vários países, como recentemente em Portugal pelo Decreto-Lei n.º 83/2018, de forma a incluir o maior número de utilizadores.

1.1.1 Empresa *ebankIT*

A *ebankIT* é uma empresa internacional fundada em 2014, dedicada a ajudar os negócios bancários a realizar todo o seu potencial e a fortalecer o relacionamento com os clientes, minimizando custos e aumentando as receitas digitais (*ebankIT*, 2018a).

A plataforma *ebankIT* foi criada no intuito de agilizar o processo de desenvolvimento de uma plataforma de múltiplos canais digitais, como podemos observar na Figura 1, apoiando os bancos e instituições bancárias a reduzir o seu tempo de colocação no mercado e aumentar o ritmo da inovação, fornecendo ferramentas configuráveis (*ebankIT*, 2018a).

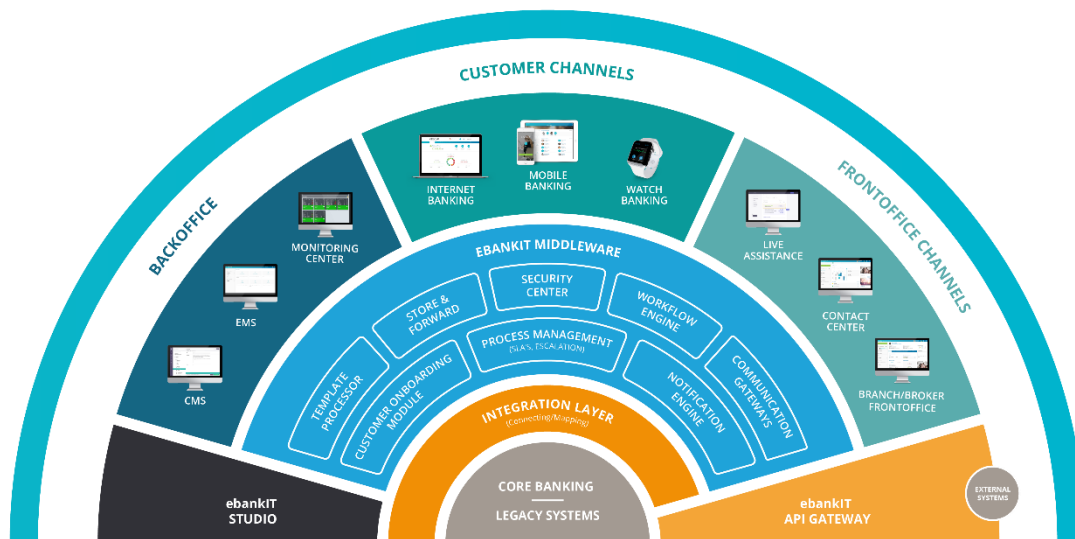


Figura 1 - Arquitetura do produto ebankIT (ebankIT, 2018b)

A solução fornece a flexibilidade de integração a um sistema bancário, disponibilizando as seguintes plataformas aos seus clientes finais:

- *internet banking* - um *website* onde o utilizador consegue realizar diversas transações bancárias, como por exemplo, consulta de saldos, transferências, pagamentos, tal como a abertura de conta;
- *mobile banking* - uma plataforma otimizada para *smartphones* com altos padrões de qualidade de segurança e recursos com as mesmas funcionalidade existentes no *internet banking*.

Para além dos canais públicos, também oferece os seguintes canais para facilitar o negócio, nomeadamente no atendimento ao público e funcionalidades administrativas:

- *backoffice* - oferece as bases necessárias para a personalização e administração de todas as informações do negócio, sem nenhum apoio das equipas técnicas, tal como ferramentas de parametrização;
- *frontoffice* - permite ajudar a organizar as atividades diárias do negócio bancário, como a criação de novas oportunidades para os clientes, e ainda apoia os operadores de calcão na realização das suas funções;
- *contact center* - permite estar em contato a qualquer momento com os clientes, incluindo a gestão de contactos, a execução de transações relevantes e a integração telefónica para um atendimento completo ao cliente.

Possui também uma camada de integração flexível com conetores para muitos sistemas de *core-banking* populares, sendo que o sistema *core-banking* corresponde aos serviços

operacionais que são suportados por um sistema de informação bancário (ebankIT, 2018b).

1.1.2 Acessibilidade

Pessoas com incapacidades pretendem o mesmo que todos os utilizadores digitais: oportunidades para aprender, trabalhar e para participar em atividades online de consumo, social e civil. Assim sendo, a acessibilidade na web significa que os websites, ferramentas e tecnologias são desenhadas e implementadas para que todas as pessoas consigam utilizar. Este conceito tem o objetivo que todas as pessoas consigam perceber, aceder, navegar, interagir com o mundo web e que consigam dar um contributo com a sua experiência para a melhoria constante (W3C, 2019d).

A acessibilidade pode ser implementada por duas maneiras diferentes: através da utilização de tecnologias de acesso ou através de boas práticas no *design* da interface.

Tornar um *website* ou uma aplicação móvel acessível significa garantir que este possa ser utilizado pelo maior número de pessoas, incluindo pessoas com deficiência visuais, motoras, cognitivas e auditivas. Para isso é necessário pensar além da própria utilização e garantir que o conteúdo e *design* disponível seja claro e simples, ainda adaptável para a utilização de ferramentas de apoio, como um leitor de ecrã para pessoas com problemas visuais ou um rato especial para pessoas com dificuldades motoras (GOV.UK, 2019).

A implementação de websites acessíveis apresentam algumas vantagens para a sociedade, como também para o negócio, como o impacto positivo da imagem da empresa e o aumento significativo de utilizadores (Leitner et al., 2016). Para além de trazer vantagens para o negócio, a acessibilidade é obrigatória por lei na União Europeia (UE) mas pode ser um “acréscimo extra” caso se considerada no final do desenvolvimento do projeto, pois pode ser necessário refazer trabalho já implementado (MDN Web Docs, 2020e).

Posto isto, a implementação de acessibilidade na web apresenta várias vantagens para os vários intervenientes, a organização e o utilizador, não necessariamente a pessoas com deficiências, mas também pessoas idosas ou com algum tipo de incapacidade temporária (W3C, Making the Web Accessible, 2019).

1.2 Problema

Cada vez mais é exigido mundialmente a implementação da acessibilidade e a empresa *ebankIT* sentiu necessidade de adaptar as suas soluções para que possuam este requisito. Sendo que a solução *internet banking* tem uma grande importância no negócio foi proposto o estudo da acessibilidade na web e a adaptação da solução para que cumpra o

maior número de requisitos e que seja possível a inclusão deste desenvolvimento em outros projetos.

Para além de uma questão legislativa também se trata de uma questão ética, devido à inclusão do maior número de pessoas. Segundo a Diretiva (UE) 2016/2102 relativa à acessibilidade dos *websites* e das aplicações móveis de organismos do setor público, a Lei Europeia de Acessibilidade e o Decreto-Lei n.º 83/2018, deve-se assegurar a acessibilidade a todos os utilizadores, removendo as barreiras à comunicação e interação que muitos utilizadores enfrentam.

1.3 Objetivos

O objetivo deste estágio é a adaptação do produto *internet banking* da *ebankIT* aos requisitos da acessibilidade na web, cumprindo as diretrizes de acessibilidade.

Neste sentido, para cumprir com o objetivo anteriormente mencionado, foi definido um conjunto de tarefas específicas, tais como:

- Identificar e documentar as boas práticas de implementação de acessibilidade, seguindo as diretivas de acessibilidade;
- Adaptar e/ou desenvolver novos componentes acessíveis para a solução *internet banking*;
- Sugerir melhorias para versões futuras.

1.4 Abordagem

A abordagem adotada para a concretização dos objetivos descritos acima é a definição de tarefas:

- Interpretar o problema de acessibilidade de acordo com as diretrizes da acessibilidade;
- Levantamento de boas práticas e técnicas de acessibilidade;
- Desenvolver novos ou refazer componentes existentes, seguindo as boas práticas de acessibilidade;
- Utilização completa do *internet banking* com o teclado e leitor de ecrã;
- Levantamento de hipóteses de implementação;
- Avaliar, analisar e expor os resultados obtidos.

1.5 Estrutura do documento

O documento está estruturado em sete capítulos diferentes.

O primeiro capítulo é composto pela introdução, com a contextualização da dissertação, o problema que se pretende resolver, os objetivos esperados e a abordagem a seguir.

O segundo capítulo é feito um enquadramento teórico e tecnológico, que permite ao leitor entender melhor os conceitos referidos ao longo do documento e a transmitir conhecimento necessário para a leitura dos capítulos seguintes.

O terceiro capítulo contempla a Análise de valor, através da apresentação do Novo Modelo de Desenvolvimento de Conceitos e o Modelo Canvas, para provar o seu valor para a organização, mercado e para os clientes.

O quarto capítulo é composto pela Análise e Desenho, onde é feita a análise do contexto atual do projeto e o estudo de soluções para este estudo. Ainda contempla os requisitos não funcionais.

O quinto capítulo sintetiza o trabalho da implementação efetuada, com a contextualização do projeto.

O sexto capítulo descreve a abordagem de avaliação utilizada, as metodologias e a avaliação de resultados.

No último capítulo é feita uma análise de todo o trabalho realizado, onde é feita a apresentação dos resultados através de objetivos atingidos, as limitações encontradas e trabalho futuro.

2 Estado de arte

Neste capítulo é apresentado um enquadramento teórico sobre o tema, abordando os conceitos teóricos mais relevantes.

O estado de arte está dividido em secções, onde vai ser feito o enquadramento teórico do tema acessibilidade, desde as leis existentes na União Europeia até aos padrões mais conhecidos e utilizados na implementação da acessibilidade, entre outros temas. Inclui ainda um enquadramento tecnológico, com a exposição de ferramentas e tecnologias para apoiar a implementação da acessibilidade.

2.1 Enquadramento teórico

Nesta secção são abordados os conceitos teóricos adjacentes ao problema do projeto e que são necessários para o entendimento e elaboração do mesmo.

2.1.1 Leis de acessibilidade na *web* na União Europeia

As barreiras encontradas no contexto da acessibilidade na *web* são problemas que impedem que pessoas com algum tipo de deficiência e/ou incapacidade consigam utilizar websites. Por exemplo, o conteúdo de um website com um *design* que contém imagens sem alternativas de descrições em texto pode não chegar a um utilizador com problemas visuais ou um utilizador que dependa da tecnologia de um leitor de ecrã.

Implementar um *website* mais acessível não se trata apenas de padrões técnicos, arquitetura e *design* da Web, mas também é uma questão política e obrigação moral. A Convenção dos direitos das pessoas com deficiência das Nações Unidas reconhece o

acesso à informação e tecnologias de comunicação, como um direito humano básico, sendo parte da lei em alguns países a obrigatoriedade de fornecer serviços acessíveis (EUROPEAN COMMISSION, 2008).

As leis mais relevantes na União Europeia são a Diretiva (UE) 2016/2102 (EUROPEAN PARLIAMENT AND OF THE COUNCIL, 2016) e a Lei Europeia da Acessibilidade (Ferri & Favalli, 2018). A lei em vigor em Portugal é Decreto-Lei n.º 83/2018 que introduz a Diretiva (UE) 2016/2102 (*Decreto-Lei 83/2018*, 2018) .

2.1.1.1 Diretiva de Acessibilidade Web

A Diretiva (UE) 2016/2102 do Parlamento Europeu e do Conselho de 26 de outubro de 2016, relativa à acessibilidade dos *websites* e das aplicações móveis de organismos do setor público, entende o termo acessibilidade como “os princípios e técnicas a observar na conceção, construção, manutenção e atualização de sítios web e aplicações móveis de forma a tornar os seus conteúdos mais acessíveis aos utilizadores, em especial a pessoas com deficiência”. Esta diretiva apresenta quatro princípios de acessibilidade, que estabelecem que os websites e aplicações móveis abrangidos por esta lei devem ser (EUROPEAN PARLIAMENT AND OF THE COUNCIL, 2016):

- Perceíveis – a informação e os componentes de interface com o utilizador devem ser apresentáveis aos utilizadores de forma que eles consigam perceber, por exemplo, texto alternativo para conteúdo não-textual;
- Operáveis – os componentes na interface com o utilizador e navegação têm de ser operáveis, por exemplo, acessíveis através do teclado, sem o uso de rato;
- Compreensíveis – os componentes na interface com o utilizador têm de ser legíveis e compreensíveis para um maior número possível de pessoas, incluindo quando interpretado por uma ferramenta de leitura em voz alta;
- Robustos – significa que os websites e aplicações móveis têm de ser compatíveis com diferentes navegadores, tecnologias assistivas.

2.1.1.2 Lei Europeia de Acessibilidade

A Lei Europeia de Acessibilidade de 2015 pretende melhorar o funcionamento do mercado interno, removendo barreiras à circulação de produtos e serviços acessíveis, e também propõe “harmonizar as medidas nacionais em matéria de acessibilidade como condição indispensável para pôr termo às divergências nas legislações”. Esta lei complementa a Diretiva (EU) 2016/2102 descrita anteriormente, incluindo alguns websites do setor privado e metodologias de controlo da acessibilidade.

Os requisitos direcionados a websites enunciados na Lei Europeia de Acessibilidade no Artigo 3.º indicam que estes devem ser “acessíveis de um modo coerente e adequado que facilite a perceção, a utilização e a compreensão pelos utilizadores, permitindo adaptar a apresentação do conteúdo, incluindo as funções interativas, e prevendo, se necessário, uma alternativa eletrónica acessível; e de um modo que facilite a interoperabilidade com vários agentes utilizadores e tecnologias assistenciais disponíveis na União e a nível internacional”. Este requisito inclui serviços de comunicação social audiovisual, telefonia, transporte de passageiros, serviços bancários, livros e comércio eletrónico (COUNCIL, 2015).

2.1.2 Padrões de acessibilidade na web

Os padrões desempenham um papel vital na definição dos requisitos de acessibilidade para cada componente num website. É fácil atender alguns dos requisitos de acessibilidade, mas entender o básico de como as pessoas com deficiência usam a Web ajuda a implementá-los de maneira mais eficaz e eficiente. Alguns aspetos de acessibilidade exigem mais habilidades técnicas ou conhecimento de como as pessoas usam a Web.

A *World Wide Web* (W3C), uma organização focada no mundo digital, criou uma iniciativa com intuito de se focar na acessibilidade na web para pessoas com deficiência. Esta iniciativa designa-se por *Web Accessibility Initiative* (WAI) e foca-se em cinco áreas de trabalho:

- abordagem de questões tecnológicas de acessibilidade web;
- criar diretrizes para navegadores, ferramentas de avaliação e criação de conteúdo;
- desenvolvimento de ferramentas de avaliação e validação de acessibilidade;
- material de suporte para apoiar na educação e divulgação;
- rastreamento de pesquisa e desenvolvimento.

A WAI publicou diversos documentos de interesse aos desenvolvedores web, de forma a entender como os componentes essenciais da acessibilidade web são dependentes entre si e como interagem entre si entre os utilizadores e os desenvolvedores do conteúdo, como ilustrado na Figura 2 (W3C, 2019d).

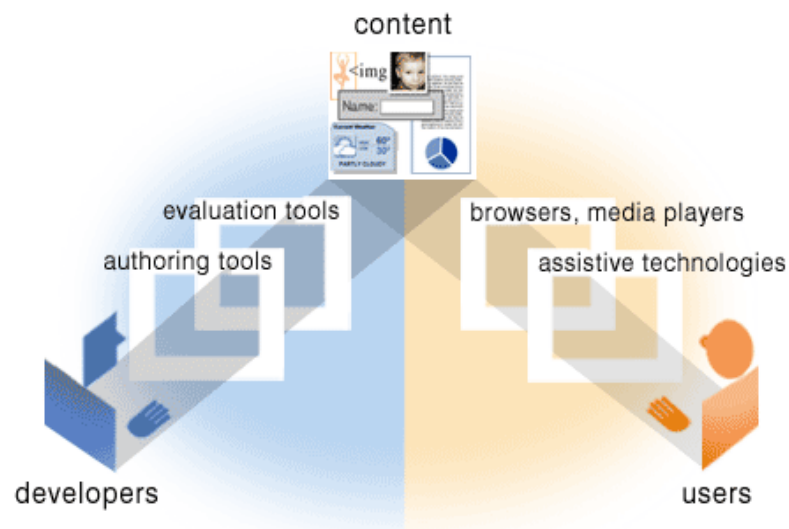


Figura 2 - Relação entre componentes de acessibilidade na web

- Conteúdo da Web (“*content*”), refere-se a qualquer parte do website, incluindo texto, imagens;
- Agentes do utilizador (“*browsers, media players*”) corresponde a software que as pessoas utilizam para aceder ao conteúdo Web;
- Tecnologias assistivas (“*assistive technologies*”), em alguns casos, leitores de ecrã, teclados alternativos;
- Ferramentas de autoria (“*authoring tools*”) refere-se ao software de criação de websites;
- Ferramentas de avaliação (“*evaluation tools*”), como por exemplo, ferramentas de avaliação de acessibilidade, validadores de *HTML*, etc.

Os principais documentos com diretrizes desenvolvidos pela WAI são (W3C, 2019d):

- Diretrizes de acessibilidade do conteúdo web (*Web Content Accessibility Guidelines* (WCAG)): enunciam como tornar o conteúdo mais acessível. Geralmente, o conteúdo Web refere-se à informação apresentada na página web ou na aplicação, incluindo texto, imagens e sons, até código ou marcação que define a estrutura e apresentação do conteúdo;
- Diretrizes de acessibilidade de ferramentas de autoria (*Authoring Tool Accessibility Guidelines* (ATAG)): as ferramentas de autoria são softwares e serviços que os utilizadores usam para produzir conteúdo da web, como por exemplo, sistemas de gestão de conteúdo. Estas diretrizes declaram como tornar as ferramentas de autoria

acessíveis, para que as pessoas com deficiência possam criar conteúdo web e ainda, ajudar os utilizadores a criar conteúdo mais acessível;

- Diretrizes de acessibilidade do agente do utilizador (*User Agent Accessibility Guidelines* (UAAG)): explicam como tornar os agentes de utilizador acessível a pessoas com deficiência. Estes agentes incluem navegadores, extensões de navegador e outros processadores de conteúdo web. Este documento é orientado a pessoas que desejam escolher agentes de utilizador mais acessíveis podem utilizar a UAAG para a avaliação e pessoas que desejam melhorar a acessibilidade do seu agente de utilizador em versões futuras.

Sendo o conteúdo do documento WCAG direcionado para o estudo e implementação do projeto.

2.1.2.1 Diretrizes de acessibilidade do conteúdo web (WCAG)

O intuito da publicação das diretrizes WCAG pela WAI é motivar, coordenar e oferecer uma base de implementação de acessibilidade (W3C, 2019d), existindo as versões 2.0 e 2.1 estáveis e a complementarem-se entre si, isto é, a versão mais recente não substitui nem deprecia a versão anterior.

A WCAG 2.1 é composta por cerca de treze diretrizes diferentes, como se pode verificar na Tabela 1, organizadas pelos princípios: perceptível, operável compreensível e robusto.

Tabela 1 - Diretrizes WCAG

Princípio	Diretriz	Definição
1. Perceptível	1.1 Textos alternativos	Fornecer textos alternativos ao conteúdo não-textual, para ter oportunidade de ser alterado para outros formatos.
	1.2 Multimédia	Oferecer alternativas ao formato áudio e vídeo, como legendas.
	1.3 Adaptável	Desenvolver conteúdo que pode ser apresentado de diferentes maneiras, sem perder a informação e estrutura.
	1.4 Distinguível	Facilitar a visualização dos conteúdos, incluindo a separação do primeiro do segundo plano.

2. Operável	2.1 Teclado	Todo o website tem de estar navegável por teclado.
	2.2 Tempo suficiente	Fornecer ao utilizador o tempo de sessão suficiente para ler, utilizar e perceber o conteúdo do website.
	2.3 Convulsões e reações físicas	Não projetar conteúdo conhecido que possa provocar convulsões ou reações físicas.
	2.4 Navegável	Implementar formas de apoiar o utilizador a navegar, encontrar conteúdo e a se situar no website.
	2.5 Modalidades de entrada	Facilitar a navegação do utilizador através de várias entradas além do teclado.
3. Compreensível	3.1 Legível	O conteúdo do website necessita de ser legível e compreensível.
	3.2 Previsível	Esperado os websites sejam navegáveis de forma previsível.
	3.3 Assistência de entrada	Apoiar os utilizadores a evitarem enganos no website.
4. Robusto	4.1 Compatível	Maximizar a compatibilidade com os agentes de utilizadores atuais e futuros, incluindo as tecnologias assistivas.

Para as diretrizes apresentadas acima existem critérios de sucesso divididos em três níveis de conformidade possíveis de alcançar, variando entre A (mais baixo), AA e AAA (mais alto), como demonstrado na Tabela 2.

Tabela 2 - Níveis de critérios de sucesso

Nível	Controlo de Acessibilidade na web
A - Acessibilidade básica	O website satisfaz os critérios de sucesso de nível A ou é dada uma versão alternativa.
AA - Acessibilidade média	O website satisfaz todos os critérios de sucesso de nível A e nível AA, ou é fornecida uma versão alternativa nível AA.
AAA - Acessibilidade avançada	O website satisfaz todos os critérios de sucesso de nível A, AA e AAA ou fornece uma versão alternativa nível AAA.

Apesar das diretrizes apresentadas, a implementação destas vai depender da respetiva tecnologia e das necessidades especiais de determinados utilizadores, sendo impossível satisfazer todas as necessidades de todas as pessoas.

2.1.3 Tecnologias adaptativas/assistivas

A tecnologia assistiva permite e promove a inclusão e a participação, especialmente de pessoas com deficiência e idosos. O principal objetivo dos produtos assistivos é manter ou melhorar o funcionamento e a independência das pessoas, promovendo o bem-estar e a sua participação no mercado de trabalho, educação e na vida cívica (*Assistive Technology*, 2020). É importante referir que a acessibilidade foca-se na inclusão de todas as pessoas, pois todos temos formas diferentes de navegar e diferentes dificuldades.

Para cada grupo de utilizadores, dependendo da sua utilização da internet ou deficiência ou incapacidade, é essencial ter em atenção as suas necessidades. Os principais tipos de deficiência ou incapacidades a considerar na implementação de um website acessível são: visuais, auditivas, mobilidade e deficiências mentais ou intelectuais. Para cada deficiência ou incapacidade referidas existem tecnologias que permitem remover as barreiras de navegação de um website e apoiar os utilizadores (MDN Web Docs, 2020e).

Alguns exemplos de tecnologias assistivas existentes são:

- Ampliadores de ecrã - podem ser ampliadores físicos ou recursos de zoom de software. Atualmente, a maioria dos navegadores e sistemas operacionais possuem recursos de zoom. Para conseguir facilitar esta utilização é necessário que as páginas não tenham tamanho absoluto, imagens ou texto com pouco contraste, perder informação com o responsivo.
- Leitores de ecrã - um software que lê texto digital em voz alta. Para isso é necessário ter em atenção alguns pontos: as imagens tenham texto alternativo, os vídeos tenham descrição textual ou sonora, documentos formatados seguindo os padrões web,

tabelas com sentido lidas célula por célula e formulários navegados numa sequência lógica. Alguns exemplos grátis são NVDA e ChromeVox.

- Alternativas para utilizadores da web com mobilidade reduzida - podem não conseguir utilizar o rato para a sua utilização da internet e substituírem-no por um rato especial, um teclado alternativo, um dispositivo tipo ponteiro fixo na cabeça ou boca ou até software de reconhecimento de voz. Para isso, é necessário que o website seja acessível a partir do teclado, mas para além disto que o tempo de utilização não seja limitado.

Para finalizar, apesar da necessidade global e dos benefícios reconhecidos das tecnologias assistivas, o acesso a estas tecnologias continua limitado.

2.1.4 Usabilidade web

Acessibilidade e usabilidade são conceitos muito idênticos quando se pensa na criação de um website que seja utilizável por todos.

A acessibilidade aborda os aspetos relacionados com a experiência do utilizador que tenha algum tipo de incapacidade ou deficiência na web, ou seja, significa que todos os utilizadores têm de conseguir perceber, compreender, navegar e interagir num website sem qualquer barreira, contribuindo para a igualdade. Enquanto a usabilidade é direcionada para o *design* de produtos que sejam eficientes e satisfatórios, incluindo apenas o *design* da experiência do utilizador (W3C, 2019d).

Assim sendo, a usabilidade pode incluir aspetos gerais que afetam todos os utilizadores e não afetam desproporcionalmente as pessoas com deficiência, sendo dependente dos seguintes fatores (SAPO, 2020):

- Funcionalidade disponibilizadas, sendo importante que estas sejam úteis e fáceis de uso, independentemente da quantidade;
- Conteúdos úteis que chamem a atenção dos utilizadores a navegar no website;
- Navegabilidade no acesso aos conteúdos, para que os utilizadores consigam chegar rapidamente ao que procuram, melhorando a sua experiência de utilização;
- Acessibilidade, por exemplo, que a informação esteja disponível independentemente do dispositivo utilizado, como um portátil ou um telemóvel.

Medir a usabilidade é possível através de várias maneiras, tais como avaliações heurísticas, avaliações por peritos e testes com utilizadores. As avaliações heurísticas são feitas com uma *checklist* de pontos de verificação e dependendo da heurística utilizada, a lista pode conter diferentes pontos de verificação. A avaliação por perito é uma mistura de avaliação

heurística com testes com utilizadores, pois estes simulam o comportamento dos utilizadores no website. Por fim, testes com utilizadores permite que utilizadores reais utilizem o sistema, realizando algumas tarefas e permite identificar os pontos críticos onde é preciso intervir para melhorar a interação.

2.2 Enquadramento tecnológico

Nesta secção são abordados conceitos tecnológicos adjacentes ao problema do projeto e que são necessários para o entendimento e elaboração do mesmo.

2.2.1 Ferramentas de avaliação de acessibilidade

As ferramentas de avaliação de acessibilidade verificam automaticamente as páginas de um website com o propósito de procurar problemas de acessibilidade, tendo cada ferramenta um objetivo de avaliação dependente dos diferentes públicos, como também podem ser utilizadas no processo de *design* e desenvolvimento web (W3C, 2019d).

Para além da utilização de ferramentas de avaliação, é obrigatório ter a avaliação humana, pois as ferramentas verificam parcialmente através da automação (W3C, 2019d). E apesar de existirem muitas ferramentas de avaliação de acessibilidade do conteúdo na *web*, apenas o ser humano consegue determinar a acessibilidade de uma página *web*.

Para cada tipo de incapacidade existe uma resposta para que fique acessível ao utilizador, seja por meio de uma tecnologia/software ou por meio da implementação de *design*. Os principais aspetos a ter em atenção durante a avaliação de acessibilidade de uma página com o mínimo de acessibilidade presente, são (Ferraz, 2016):

- Verificar se o website é navegável via teclado;
- Apresentar o salto, também conhecido por *skip link*, para o conteúdo visível;
- Verificar se o conteúdo interativo possui um estilo para o estado focado;
- Verificar se o contraste é adequado;
- Vídeos com legendas;
- A ampliação em 200% da página não pode perder informação;
- Verificar estrutura da página marcada adequadamente, por exemplo os cabeçalhos (*headings*);
- Testes com tecnologias assistivas.

As ferramentas de avaliação automatizadas apresentam algumas características que permitem reduzir o número de opções de ferramentas de acordo com as necessidades. Algumas dessas características são (W3C, 2019d):

- Normas e diretrizes: as ferramentas podem seguir diretrizes diferentes, sendo as mais comuns as diretrizes de acessibilidade do conteúdo web (WCAG) desenvolvidas pela W3C;
- Linguagem: podem suportar diferentes idiomas, incluindo a interface do utilizador, tal como o conteúdo do website a avaliar;
- Tipo de ferramenta: podem ser *plugins*, linhas de comando, aplicações ou serviços online;
- Formatos suportados: a maioria das ferramentas verifica a acessibilidade através de HTML, sendo que existem outras que verificam CSS, SVG e PDF;
- Validações automáticas: o limite de verificação de uma ferramenta varia, pode apenas conseguir validar uma única página, como também, por exemplo, existem ferramentas capazes de aceder a conteúdo restrito por palavra-passe;
- Licença: existem diversas ferramentas com, por exemplo, código aberto, comercial, corporativo;
- Informações de acessibilidade: um ponto importante nas ferramentas é que estas também devem ser acessíveis por pessoas com deficiência ou incapacidades temporárias;
- Estilos de relatório: as ferramentas de avaliação geram uma variedade de relatórios com base nos resultados da análise de uma página web, sendo que podem ser baseados em texto e em gráficos/ícones.

Tendo em conta as características descritas acima e apesar que o ideal seria uma avaliação o mais completa possível, existem necessidades/restrições presentes para a utilização de ferramentas neste projeto. Os pontos principais para a escolha de ferramentas são: devem seguir as diretrizes de acessibilidade, de preferência as WCAG 2, não ter custo associado, fácil utilização e perceptível para utilizadores com pouca experiência em acessibilidade.

Assim sendo, foram escolhidas algumas ferramentas para que seja possível uma avaliação mais completa e diferenciada, e estas são WAVE, *Accessibility Insights for Web* e *Lighthouse*.

2.2.1.1 WAVE

Web Accessibility Evaluation Tool (WAVE) criada por WebAIM em 2001, apoia os desenvolvedores web na implementação de conteúdo mais acessível por página, identificando erros por meio de vários padrões (WAVE, 2020).

Esta extensão é muito útil e simples na validação e verificação do conteúdo web, fornecendo um sistema de aviso com cores: vermelho para os erros que precisam de ser corrigidos com urgência, amarelo para o código que precisa de ser analisado e verde para pontos corretos, mas que precisam de ser verificados. Ainda é apresentado na página um resumo dos resultados, com detalhes do aviso: o que ele significa, porque é importante, como corrigir e as normas e diretrizes associadas.

2.2.1.2 Accessibility Insights for Web

Accessibility Insights for Web é uma extensão que apoia os desenvolvedores web a encontrar e corrigir problemas de acessibilidade em aplicações e websites.

A ferramenta suporta dois cenários possíveis, o FastPass e Assessment (Accessibility Insights, 2020):

- *FastPass* é um processo de duas etapas que ajuda os desenvolvedores a identificar problemas comuns de acessibilidade de alto impacto em menos de cinco minutos. São feitas verificações automatizadas conforme aproximadamente 50 requisitos de acessibilidade e ainda fornece instruções claras e um auxiliar visual que facilita a identificação de problemas críticos relacionadas com o acesso por teclado, como os elementos acessíveis por tab (*Tab stops*) e a ordem incorreta de navegação por Tab.
- *Assessment* permite que qualquer pessoa com conhecimento em HTML verifique se a sua aplicação ou website está em conformidade com o Nível AA das diretrizes WCAG 2.1. Realiza verificações automatizadas conforme aproximadamente 50 requisitos de acessibilidade e fornece instruções passo a passo, exemplos e orientações sobre como corrigir aproximadamente 20 testes, sendo muitos testes “assistidos”, ou seja, a ferramenta identifica as instâncias de teste ou fornece um auxiliar visual.

2.2.1.3 Lighthouse

Lighthouse é uma ferramenta, que pode ser usada como extensão no Chrome. Quando realiza a auditoria à página, esta ferramenta corre uma série de testes na página e gera um relatório sobre o desempenho da página, tendo uma secção sobre acessibilidade.

Na secção de acessibilidade é feita a verificação dos cabeçalhos da página, do contraste das cores com o fundo, a tag title do documento, o nome de links e ainda fornece uma

lista de pontos a serem verificados manualmente, pois não é possível verificar todos os pontos relevantes (Google Developers, 2020b).

Foi decidido utilizar todas as ferramentas descritas acima, pois todas elas apresentam um relatório de resultados em formatos diferentes com as suas vantagens e desvantagens, sendo que apesar de apresentarem muitos pontos comuns, complementam-se entre si. A Tabela 3 apresenta vantagens e desvantagens na utilização de cada ferramenta.

Tabela 3 - Vantagens e desvantagens da utilização de ferramentas Lighthouse, WAVE e Accessibility Insights for Web

Ferramenta	Vantagens	Desvantagens
Lighthouse	Apresenta percentagem por página do nível de acessibilidade, considerando os pontos possíveis de avaliar pela ferramenta, excluindo as situações apenas testáveis manualmente. Assim sendo, é possível quantificar uma parte do nível de acessibilidade.	<p>A sinalização dos pontos a melhorar não é feita no elemento diretamente na página mas é devolvida em formato lista.</p> <p>A ferramenta carrega novamente a página e tendo em conta que pode haver elementos que só apareçam com interação do utilizador, estes não vão ser validados.</p>
WAVE	A sinalização dos erros e alertas são feitos diretamente na página junto aos elementos. Para além dos erros e alertas, esta ferramenta sinaliza na página os elementos estruturais na página e os elementos ARIA (secção 2.2.2) presentes, trazendo vantagens para o desenvolvedor identificar visualmente problemas estruturais.	

	Esta ferramenta também apresenta de forma bastante detalhada como corrigir os erros, a importância de os resolver e qual a diretriz WCAG associada.	
Accessibility Insights for Web	Apresenta uma opção que permite identificar os elementos e a ordem da navegação por teclado através da tecla Tab. Para o desenvolvedor e para o <i>tester</i> ¹ é importante, pois podem existir elementos não identificáveis quando estão focados pelo teclado.	

Por fim, na secção 6 Avaliação é possível ver alguns dos relatórios resultantes e pontos avaliados pelo uso das ferramentas descritas acima.

2.2.2 Accessible Rich Internet Applications

Aplicações para a Internet Ricas em Acessibilidade (ARIA) definem formas de tornar o conteúdo web e aplicações mais acessíveis a pessoas com incapacidades.

ARIA permite a marcação da página através de atributos para uma fácil navegação e também para que as interações sejam suportadas por tecnologias assistivas, que podem ser adicionados a qualquer linguagem de marcação, sendo especialmente adequado para HTML. E ainda, fornece funções e estados que descrevem o comportamento dos elementos interativos (*widgets*) da interface do utilizador (MDN Web Docs, 2020b).

A especificação ARIA é dividida em três tipos diferentes de atributos (MDN Web Docs, 2020b)

- As funções que descrevem *widgets*, como barras de menu e caixas de diálogo;
- As propriedades, que descrevem as características do *widgets*, como por exemplo se são clicáveis, têm um elemento obrigatório ou se têm um pop-up associado;
- Os estados, que descrevem o estado atual da interação com o elemento, informando a tecnologia assistiva se estiver ocupada, desativada, selecionada ou oculta. Alguns dos exemplos destes atributos são:

¹ Tester é uma pessoa pertencente à equipa que garante a qualidade de um produto/serviço e que realiza diversos testes para tal feito.

- **aria-checked**: indica que o estado de uma caixa de seleção (*checkbox*).
- **aria-disabled**: indica que um elemento é visível, mas não é editável u operável de outra forma.
- **aria-hidden**: indica que o elemento e os seus descendentes não são visíveis ou perceptíveis para qualquer utilizador, conforme implementado.

Estes atributos são interpretados automaticamente pelo navegador, oferecendo mais consistência na experiência do utilizador. Sendo que, ARIA também oferece atributos de suporte à navegação por teclado.

Tradicionalmente, a navegação por teclado na web é limitada à tecla Tab. Um utilizador carrega no Tab para focar cada link, botão ou formulário numa página em ordem linear e utiliza Shift + Tab para navegar para trás. É uma forma unidimensional de navegação – para frente e para trás, um elemento de cada vez. Em páginas bastante densas, um utilizador pode carregar dezenas de vezes em Tab para chegar à secção que precisa. A implementação de convenções de teclado estilo desktop tem o potencial de acelerar significativamente a navegação para muitos utilizadores, como por exemplo agrupar controlos como o menu ou tabelas.

2.2.3 Leitores de ecrã

Os leitores de ecrã, também conhecidos como *screen readers*, são aplicações de software que apoiam pessoas cegas e pessoas com incapacidades visuais na leitura do texto presente no ecrã com um sintetizador de voz ou em braile. Normalmente, as informações são comunicadas por uma saída auditiva por meio de conversão de texto em fala (TTS) ou em braile (Andrews, 2017).

Há alguns pontos-chave para que o website seja navegável pelo leitor de ecrã, sendo também pontos gerais a serem considerados ao criar ou procurar conteúdo acessível:

- Os leitores de ecrã não vão descrever imagens ou informações visuais e gráficas ao utilizador. Por isso, é essencial existir texto alternativo que forneça uma descrição concisa do item visual. Existem algumas questões em determinar o texto alternativo apropriado e equivalente, pois geralmente é uma questão de interpretação pessoal, e o texto alternativo pode ter várias funções, tais como:
 - Substituir a informação das imagens para que seja lida por leitor de ecrã, permitindo que o conteúdo e a função da imagem seja acessível a pessoas com incapacidades visuais;
 - É exibido o texto alternativo no caso da imagem não ser carregada ou o utilizador optar por não visualizar imagens;

- Fornece significado semântico e uma descrição para imagens que podem ser lidas pelos mecanismos de pesquisa ou utilizadas para determinar posteriormente o conteúdo da imagem somente no contexto da página.
- Deve apresentar ao utilizador o conteúdo e função da imagem no conteúdo web.
- Utilizar títulos descritivos, explicitamente definidos por HTML, para facilitar a estrutura e navegabilidade da página. Assim, os utilizadores podem navegar utilizando o cabeçalho e, em alguns casos, até detalhar um nível de cabeçalho (WebAIM, 2020a).
- Para além dos cabeçalhos, os utilizadores podem navegar na página através de *links*, controlos de formulários, secções de página, entre outros.
- O conteúdo não interativo não deve ser focalizável, pois os leitores de ecrã já vão ler o conteúdo, tornando a utilização por teclado mais eficiente (Google Developers, 2020a).
- O documento tem de possuir o atributo `lang` na *tag* `<html>` para que o leitor de ecrã configure as próprias regras de pronúncia. Para além disto, se uma página possuir palavras de outros idiomas, então deve-se adicionar o atributo `lang` a uma *tag* `` (WebAIM, 2020c).

Segundo um estudo realizado entre agosto e setembro de 2019 pela WebAIM com cerca de 1224 respostas de diferentes utilizadores, os leitores de ecrã mais comuns são o NVDA, JAWS e VoiceOver, como se pode observar na Figura 3.

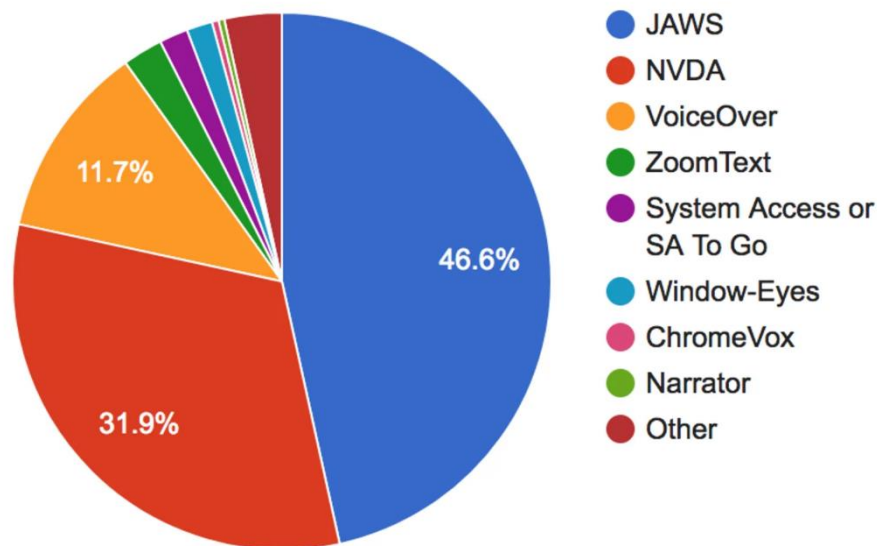


Figura 3 - Do estudo “Screen Reader User Survey #8”, a percentagem de utilização de leitores de ecrã (WebAIM, 2019).

2.2.3.1 *NonVisual Desktop Access (NVDA)*

O NVDA é um leitor de ecrã, grátis e *open-source* para o sistema operativo Microsoft Windows. As informações do programa são fornecidas através de voz sintética e/ou braile e permitem, às pessoas com deficiência visual, em igualdade de custos com os restantes indivíduos, aceder ao computador, que está a executar o *Windows* (NVDA, 2017).

2.2.3.2 *Job Access With Speech (JAWS)*

JAWS é o leitor de ecrã mais conhecido, desenvolvido para utilizadores com perda de visão para que consigam perceber o conteúdo do ecrã e navegar sem o rato (Freedom Scientific, 2020).

Este leitor de ecrã fornece uma série de oportunidades ao utilizador para melhorar a sua navegação, tais como:

- Leitura de documentos, emails, websites e aplicações moveis;
- Fácil navegação com o rato;
- Explora e lê de todos os documentos, incluindo PDF;
- Navegação na internet com teclas de navegação.

2.2.3.3 VoiceOver

O leitor de ecrã VoiceOver foi desenvolvido pela Apple com intuito de apoiar pessoas com deficiência visual e cegueira. Este software descreve exatamente o que está a acontecer no ecrã e está presente em todos os computadores Mac (Apple, 2020).

Possui a funcionalidade de aumentar o tamanho de letra quando o utilizador passa o rato por cima do texto e pressa o comando necessário e ainda oferece a oportunidade ao utilizador escolher o tamanho e as cores do texto. Para além destas funcionalidades, o VoiceOver oferece muito mais, como (Apple, 2020):

- Dá dicas ao longo da navegação do utilizador e apoia a que ações sejam realizadas, quer seja por gestos, com o teclado ou com um ecrã braile;
- Suporta mais de 35 linguagens, incluindo múltiplas opções de voz;
- Capacidade de descrição de imagens para o utilizador;
- Compatível com mais de 70 displays em braile para entrada e saída;
- Modo Escuro, onde os elementos do ecrã são transformados para que em ambientes com pouca luz, o texto claro sobre o fundo escuro melhora a leitura.

Na Tabela 4 são comparados os leitores de ecrã descritos acima.

Tabela 4 - Comparação de leitores de ecrã

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

2.2.4 Aplicações web acessíveis

Uma aplicação web acessível significa que não possui qualquer impedimento no seu acesso por parte de utilizadores com diferentes habilidades para que consigam reconhecer, perceber, navegar e interagir com o conteúdo web (Dongaonkar et al., 2017).

Tal como referido anteriormente na secção 2.1.1, as leis mais relevantes na União Europeia são a Diretiva (UE) 2016/2102 e a Lei Europeia da Acessibilidade, onde são

definidos princípios e algumas regras que devem ser cumpridos para a acessibilidade das aplicações. Assim sendo, é esperado que muitas aplicações *web* dentro da União Europeia sirvam como referência de acessibilidade e que seguem as diretrizes WCAG.

Segundo um estudo realizado pelo PÚBLICO, depois de submeter as páginas oficiais de 124 entidades públicas portuguesas na ferramenta AccessMonitor disponibilizada pelo próprio governo português, apenas três cumprem as regras de acessibilidade. As três páginas são: Centro de Gestão da Rede Informática do Governo, Instituto Nacional de Emergência Médica e Direcção-Geral (DG) do Ensino Superior (Baptista, 2019).

Na Declaração de Acessibilidade e Usabilidade presente no website Instituto Nacional de Emergência Médica é possível verificar que foi feita uma avaliação automática pela ferramenta AccessMonitor que apresenta um resultado de 9.3 na sua escala (1-10), mas não foram feitos testes manuais (INEM, 2017).

Dentro da área bancária em Portugal é possível verificar alguns exemplos de websites acessíveis, como o caso do Millenium BCP e o BPI, que seguem as diretrizes WCAG, ilustrados na Figura 4 e Figura 5.

O BPI apresenta apresenta novas funcionalidades que permitem facilitar a navegação e o acesso aos conteúdos pelos seus clientes com necessidades especiais, com o objetivo de potenciar a interação com todo o tipo de utilizadores. Neste âmbito, as características de acessibilidade do site Banco BPI são (BPI, 2020):

- A acessibilidade de conteúdos tem o nível de conformidade "AA";
- Permite alterar o tamanho da letra de acordo com a preferência do utilizador;
- É acessível através de diferentes tipos de software que efectuem a tradução para braille ou para conteúdos áudio;
- Apresenta imagens e animação não sobrepostas ao texto, mas como complemento visual, evitando constituírem um obstáculo para deficientes visuais;
- Viabiliza a legibilidade da informação por daltónicos, devido à selecção de cores.

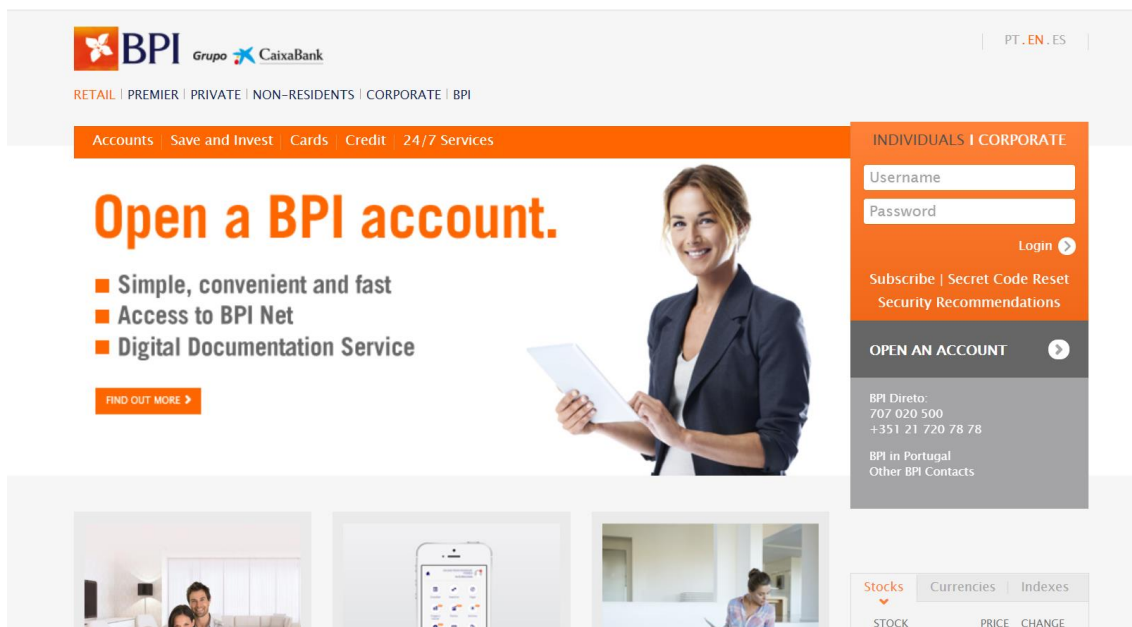


Figura 4 - Website acessível: Banco BPI (BPI, 2020)

O Millenium BCP possui o compromisso de melhorar, cada vez mais, a acessibilidade dos seus clientes com necessidades especiais, através de novas funcionalidades de forma a potenciar este acesso. Por isso, seguindo as recomendações da Web Accessibility Initiative (WAI), é possível observar as seguintes características:

- Compatibilidade com o maior número possível de dispositivos;
- Permite alterar o tamanho da letra, sendo que, a maioria dos browsers permitem alterar o tamanho da letra;
- No topo da página existem links que permitem saltar diretamente para a navegação ou para os conteúdos principais;
- Para a utilização de software de leitura ou de voz, é disponibilizado texto alternativo às imagens.

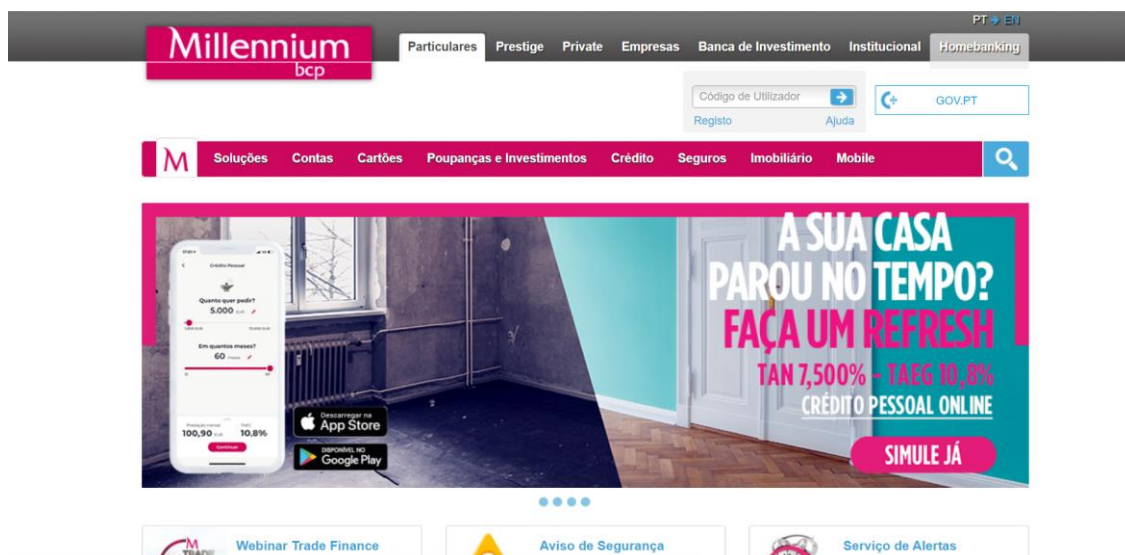


Figura 5 - Website acessível: Millenium BCP (Milleniumbcp, 2020)

Estes websites são uma referência para verificar qual a melhor maneira de adaptar a acessibilidade em determinados elementos.

2.3 Resumo

Todo o conhecimento adquirido através do estudo realizado no estado de arte é uma mais valia na implementação deste projeto, pois permite compreender a importância da acessibilidade e a sua implementação.

A partir do estado de arte é possível verificar que:

- Existem leis que obrigam a implementação da acessibilidade na web e é um direito do utilizador conseguir navegar sem restrições ou impedimentos;
- O conteúdo do documento WCAG é direcionado para o desenvolvimento deste projeto, onde são apresentadas as diretrizes de como implementar a acessibilidade na web;
- A necessidade de implementar uma aplicação web acessível através de tecnologias assistivas, como um leitor de ecrã ou simplesmente, através do teclado;
- Os conceitos usabilidade e acessibilidade estão relacionados, pois têm o objetivo de tornar uma aplicação fácil de utilizar, intuitiva e acessível a todas as pessoas;
- Tendo em conta as características essenciais para a escolha de ferramentas de avaliação, foram escolhidas: WAVE, *Accessibility Insight for Web* e *Lighthouse*;

- Os atributos ARIA permitem que os browsers e as tecnologias assistivas reconheçam elementos e que anunciem as suas mudanças de estado.

Em suma, a acessibilidade na *web* requer muitos cuidados na implementação mas também oferece muitas vantagens ao utilizador e à entidade que disponibiliza a aplicação.

3 Análise de valor

O presente capítulo apresenta a análise de valor do projeto, com base no Novo Modelo de Desenvolvimento de Conceitos e através da apresentação do Modelo Canvas, para provar o seu valor para a organização, mercado e para os clientes.

A análise é feita através 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.

3.1 Novo Modelo de Desenvolvimento de Conceitos

Segundo Koen et al (2001), *front-end* de inovação “envolve as atividades que ocorrem antes do formal e bem estruturado subprocesso de desenvolvimento de novos produtos”. E essas atividades são dependentes de fatores internos e externos à organização, por isso foi desenvolvido o Novo Modelo de Desenvolvimento de Conceitos (NCD) a partir da “necessidade identificada pelos autores de proporcionar maior clareza a essa parte do processo.”

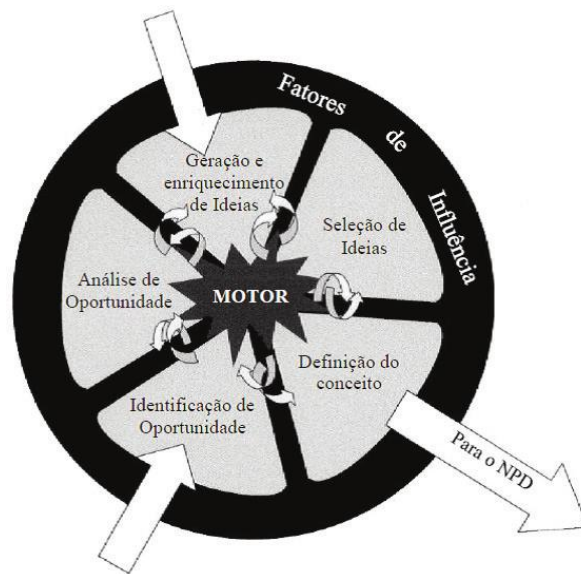


Figura 6 - Novo modelo de desenvolvimento de conceitos (NCD) (Koen, 2001)

O Modelo NCD, como representado na Figura 6, consiste em três diferentes partes:

1. A área interna, onde são definidos os cinco elementos principais que compõem o *front-end* de inovação: Identificação de oportunidade; Análise de oportunidade; Geração e aperfeiçoamento de ideias; Seleção de ideias; Desenvolvimento do conceito e tecnologia;
2. O mecanismo central que conduz os cinco elementos referidos acima e é alimentado pela liderança e cultura da organização;
3. Os fatores de influência, ou ambiente da periferia, que consiste nas capacidades organizacionais, estratégias de negócio, o mundo exterior e a Ciência facilitadora que será utilizada.

A forma circular sugere que as ideias e oportunidades devem fluir e iterar entre todos os elementos da área interna. Tal como as setas apontam na Figura 6, o modelo possui dois pontos de entrada, a geração e enriquecimento de ideias e a identificação de oportunidades, mas só possui apenas uma saída, a definição do conceito, sendo esse ponto a ligação com o processo de desenvolvimento formal (Koen, 2001).

Os cinco elementos principais que compõem o *front-end* de inovação vão ser analisados segundo o tema desta dissertação nos seguintes pontos.

3.1.1 Identificação de oportunidade

As fontes e métodos que uma empresa usa para identificar as oportunidades são a essência deste elemento. Por exemplo, a oportunidade pode ser uma resposta a curto

prazo a uma ameaça competitiva, uma possibilidade inovadora de capturar vantagem competitiva ou um meio de simplificar/ acelerar/reduzir o custo das operações (Koen, 2001).

A identificação da oportunidade deste trabalho ocorreu quando surgiu a necessidade da empresa se adaptar aos novos requisitos legislativos e éticos e neste caso, a implementação da acessibilidade de forma a incluir mais utilizadores e remover barreiras na comunicação e interação com a solução. Para além disto, facilitar a adaptação dos projetos que tiverem o requisito de acessibilidade e utilizarem a solução base *ebankIT*.

3.1.2 Análise de oportunidade

As informações adicionais são necessárias na identificação de oportunidades específicas de negócio e tecnológicas (Koen, 2001).

Existem três principais razões para motivar as organizações a desenvolver websites acessíveis: melhorar a vida das pessoas com deficiência; capitalizar maior audiência, com base no consumidor; e evitar ações jurídicas e/ou má imprensa.

Para a equipa de desenvolvimento em particular não importa a motivação da implementação, pois a acessibilidade na web é mais fácil de alcançar se as pessoas estiverem no centro do processo. Para tal, é necessário entender o ponto de vista do utilizador e as suas necessidades, ultrapassar a acessibilidade técnica e manter o foco nos princípios da acessibilidade.

3.1.3 Geração e aperfeiçoamento de ideias

Processo de desenvolvimento e maturação da oportunidade numa ideia concreta. É um processo evolutivo no qual as ideias são construídas, demolidas, combinadas, remodeladas, modificadas e atualizadas. A ideia pode passar por muitas iterações e mudanças conforme a sua análise (Koen, 2001).

Nesta fase espera-se adaptar a solução existente e melhorá-la com o desenvolvimento baseado nas diretrizes de acessibilidade com intuito de chegar ao nível avançado de acessibilidade.

3.1.4 Seleção de ideais

Na maioria das empresas existem muitas ideias de produtos/processos, em que a parte mais difícil é a escolha da ideia que mais vantagens traz ao valor do negócio (Koen, 2001).

Nesta fase, foram pesados os esforços para a implementação e adaptação da solução existente às regras de acessibilidade e ver qual ideia espelhava mais vantagens a curto

prazo, a ideia selecionada foi a de adaptar as páginas e componentes mais utilizados pela solução e projetos para que a sua utilização seja partilhada em todos os projetos que vão possuir a versão acessível da solução *ebankIT*.

Esta versão deve incluir componentes testados sempre tendo em conta as diretrizes de acessibilidade, passando pela navegação por teclado, o uso de tecnologias assistivas, como leitores de ecrã e perceber se a informação que é lida corresponde ao que a aplicação pretende demonstrar.

3.1.5 Desenvolvimento do conceito e tecnologia

O elemento final do modelo envolve o desenvolvimento de um caso de negócio com base em estimativas de potencial mercado, necessidade do cliente, requisitos de investimento, avaliações concorrentes e risco do projeto (Koen, 2001).

Este trabalho terá como foco a acessibilidade de um utilizador numa aplicação *web* do tipo *internet banking*, tendo o intuito a navegação com o teclado e a utilização de leitor de ecrã, para além das alterações visuais, como o contraste e a responsividade da aplicação à utilização do zoom. Outro ponto importante neste trabalho, é adaptar os componentes existentes e alterar na solução base para melhorar projetos de integração futuras. Ainda, oferecer sugestões de implementação para novas versões com tecnologias diferentes onde a acessibilidade possa ser tida em conta desde o início.

3.2 Modelo Canvas

O modelo Canvas, também conhecido por *Business Model Canvas*, é uma ferramenta de gestão estratégica que permite ver todos os aspetos fundamentais de um modelo de negócio numa única folha.

A Tabela 5 apresenta o Modelo Canvas com os blocos preenchidos consoante o objetivo deste trabalho.

Tabela 5 - Modelo Canvas

Rede de Parceiros	Atividades Chave	Proposta de Valor	Relacionamento com Clientes	Segmentos de Clientes
Organização	Conhecer as diretrizes de acessibilidade	Diferenciação do website	Equipa de desenvolvimento	Organização
Equipa de desenvolvimento	Adaptar e/ou desenvolver novos componentes acessíveis			

	Sugerir melhorias	Aumento de utilizadores		
	Recursos Chave	Imagem positiva transmitida ao cliente	Canais de Distribuição	
	Equipa de desenvolvimento	<i>Design</i> adequado para todos	Partilha de conhecimento com a organização	
	Leitor de ecrã	Aplicação <i>web</i> de qualidade	Aplicação <i>internet banking</i>	
Estrutura de Custos		Fluxo de Receitas		
Desenvolvedores/programadores		Entrega de uma solução <i>internet banking</i> de acordo com as diretrizes de acessibilidade		

O modelo divide-se em nove blocos construtivos:

- Atividades chave - atividades mais importantes para executar a proposta de valor, chegar aos mercados ou obter rendimentos.
 - Conhecer as diretrizes de acessibilidade;
 - Adaptar e/ou desenvolver novos componentes acessíveis;
 - Sugerir melhorias.
- Recursos-chave: recursos necessários para criar valor para o cliente.
 - Equipa de desenvolvimento
 - Leitor de ecrã
- Rede de parceiros: alianças de negócio que complementam outros aspetos do modelo de negócio.
 - Organização
 - Equipa de desenvolvimento
- Proposta de valor: motivo pelo qual os clientes vão escolher o produto da empresa e não de um concorrente, isto é, descreve a forma como a empresa se diferencia dos seus concorrentes.
 - Diferenciação do website
 - Aumento de utilizadores
 - Imagem positiva transmitida ao cliente
 - *Design* adequado para todos
 - Aplicação *web* de qualidade

- Segmentos de clientes: define os diferentes grupos de pessoas ou organizações que a empresa visa alcançar ou/e servir.
 - Organização
 - Clientes interessados em integrar acessibilidade
- Canais: o meio pelo qual a empresa vai fornecer o seu produto/serviço aos clientes.
 - Partilha de conhecimento com a organização
 - Aplicação *internet banking*
- Relações com os clientes: a empresa estabelece ligações entre si e os seus diferentes segmentos de clientes.
 - Equipa de desenvolvimento
- Estrutura de custos: consequências monetárias dos meios utilizados no modelo de negócio.
 - Desenvolvedores/programadores, pois a curva de aprendizagem tem custos por envolver mais tempo e dedicação da equipa de desenvolvimento.
- Fluxo de receita: forma como a empresa ganha dinheiro através da variedade de fluxos de receita.
 - Entrega de uma solução *internet banking* de acordo com as diretrizes de acessibilidade.

4 Análise e Desenho

Este capítulo tem como objetivo contextualizar o estado atual da solução *ebankIT* e analisar soluções que têm como objetivo resolver o problema da dissertação.

4.1 Análise da solução atual

Nesta secção é apresentado uma breve explicação sobre aplicações web e linguagens de programação e marcação, para que seja possível a compreensão das secções seguintes onde é feito um estudo de soluções. Ainda, é contextualizado o estado atual da solução *ebankIT*.

4.1.1 Aplicação web

Aplicação web é uma solução que é executada diretamente no navegador, sem necessitar de qualquer instalação, sendo suportada pelas tecnologias da *World Wide Web* (W3).

A W3 é um sistema interconectado de páginas web públicas acessíveis pela internet e funcionam ligadas a um servidor web, dependentes de pedidos realizados por utilizadores, da utilização de protocolos e métodos e a resposta do protocolo. A funcionalidade do servidor web é receber o pedido do utilizador e devolver uma resposta para a aplicação pelo browser.

Outro componente das aplicações web é o protocolo HTTP, que os clientes e servidores utilizam para comunicar com o servidor e essa comunicação é feita com pedidos (*request*) e respostas (*response*). O conteúdo de um pedido contém o método HTTP, a página que será aberta e os parâmetros do formulário; e o conteúdo da resposta contém o código status de sucesso ou insucesso, o tipo de conteúdo (por exemplo, *HTML*, figuras, texto) e o conteúdo (Noletto, 2020).

Para o desenvolvimento de aplicações web é fundamental ter conhecimento em algumas áreas, como linguagens de marcação e programação, base de dados e padrões de arquitetura.

4.1.2 Linguagens de programação e marcação

Para a estruturação do conteúdo de aplicações web com significado e propósito, existe uma linguagem de marcação designada por HTML. HTML (*Hyper Text Markup Language*) desempenha um papel importante no desenvolvimento web por semanticamente estruturar os conteúdos digitais, utilizando *tags*. Este tipo de código-fonte representa o esqueleto do *website*.

Para facilitar a navegação e interação do utilizador é necessário ter o código HTML o mais semântico possível (MDN Web Docs, 2020c). Sendo que HTML semântico significa utilizar os elementos HTML corretamente, por exemplo se for necessário adicionar um botão que seja pela *tag* `<button>` e não `<div>`, pois o elemento *button* define claramente o conteúdo e o leitor de ecrã identifica-o como tal. Outros pontos importantes para que HTML seja acessível, são (w3schools, 2020):

- **Headings:** Nos mecanismos de pesquisa são utilizados os títulos presentes na página para indexar a estrutura e o conteúdo. Os títulos são definidos em HTML pelas *tags* `<h1>` a `<h6>`, sendo a primeira direcionada para os cabeçalhos principais e seguidos os menos importantes.
- **Texto alternativo:** O atributo `alt` fornece um texto alternativo a uma imagem, que deve ser descritivo, pois vai ser exibido no caso da imagem não ser carregada ou o utilizador não a conseguir visualizar.
- **Declarar o idioma:** A declaração do idioma é importante para os leitores de ecrã e mecanismos de pesquisa, sendo declarada com o atributo `lang`.
- **Linguagem clara:** Evitar utilizar caracteres que não possam ser lidos claramente por um leitor de ecrã e utilizar uma linguagem o mais clara possível.
- **Links:** Um link deve explicar claramente quais as informações que o utilizador vai obter no link, tendo o atributo `title` para fornecer informações extras quando o rato se move sobre o elemento.

Para aprimorar a aplicação utiliza-se as folhas de estilo em cascata, designadas por CSS, são um tipo de padrão de linguagem de formatação para documentos HTML. É utilizado para atribuir regras aos elementos presentes no código HTML e como se devem representar no navegador.

Javascript é uma linguagem de programação que permite dinamizar o conteúdo, avaliando as interações do utilizador e recarrega, atualiza e altera o conteúdo da página.

4.1.3 Solução atual

O projeto enquadrado neste estágio já se encontrava em curso há algum tempo e com diversas versões lançadas. Este projeto é desenvolvido em *ASP.NET Web Forms*, na linguagem C# e desde o início da sua implementação nunca foi pensada a inclusão da acessibilidade, o que tornou os objetivos deste projeto mais desafiantes.

Esta solução em estudo está dividido em três áreas, como apresentado na Figura 7:

- A camada de apresentação corresponde à interface de utilizador, responsável por apresentar os conteúdos aos utilizadores e onde estes realizam interações, que interage com o middleware;
- O middleware, responsável pela lógica de negócio do sistema que inclui a manipulação de dados e serviços.
- A camada de persistência que é responsável pela garantia de persistência de todos os dados da aplicação.

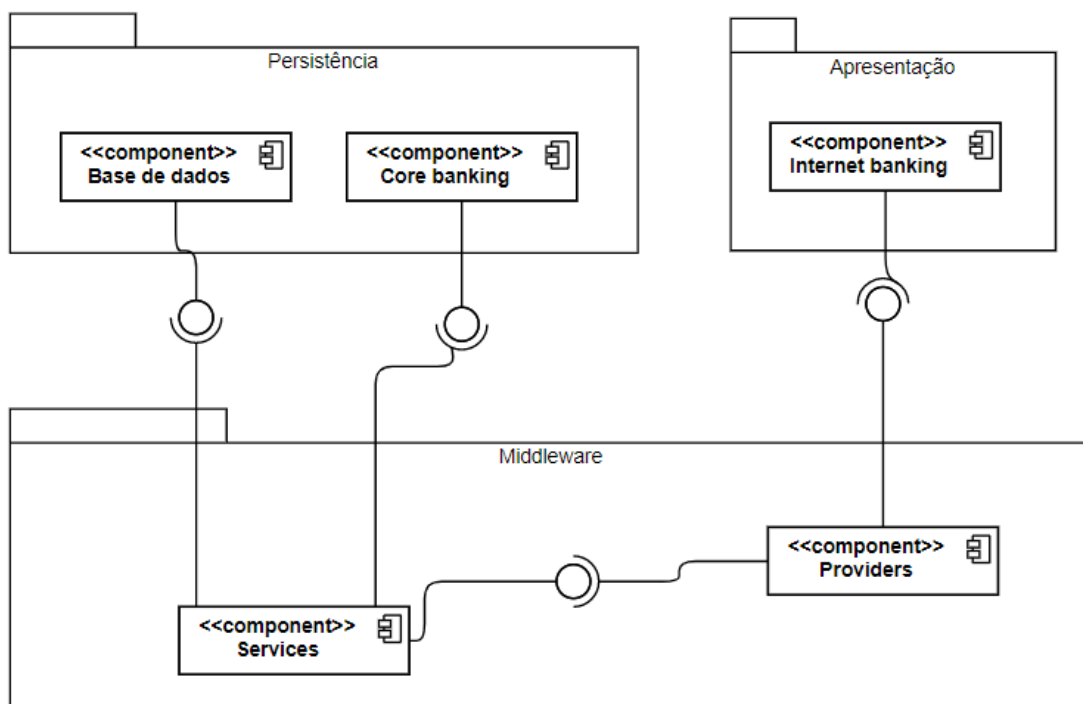


Figura 7 – Diagrama de componentes da aplicação Internet banking da ebankIT (ebankIT, 2018b)

Um diagrama de componentes mostra a relação entre diferentes componentes de um sistema. O termo "componente" refere-se a um módulo de classes que representa sistemas ou subsistemas independentes com capacidade de interagir com o restante do sistema. Na Figura 7 está representado o diagrama de componentes UML da aplicação *internet banking* e é possível visualizar os seguintes componentes:

- Internet banking – responsável por apresentar os conteúdos aos utilizadores e onde estes realizam interações;
- Providers – responsável por manipular as respostas dos serviços de forma legível e de fácil interpretação para a camada de apresentação;
- Services - responsável por realizar todas as chamadas ao core banking e à base de dados;
- Base de dados – componente responsável por guardar algumas informações do sistema, como configurações, histórico de operações;
- Core banking – componente responsável pela comunicação ao sistema de informação bancário.

A equipa envolvente a esta solução é composta por programadores, *designers* e *testers*. Os *designers* desenham o ecrã com o formato e cores a seguir, os programadores executam a interface desenhada pelo *designer* e constroem toda a estrutura necessária para o ecrã funcionar. Por fim, os *testers* verificam a qualidade do que foi desenvolvido, garantindo a qualidade da aplicação.

Neste projeto, a área mais importante que vai sofrer alterações é a camada de interface com o utilizador. Sabendo que, *Web Forms* é um modelo de programação baseado na tecnologia Microsoft ASP.NET em que o código executado no lado do servidor gera dinamicamente a página web para o navegador.

A criação de um modelo de página para toda a aplicação web em ASP.NET é possível através de páginas mestras, mais conhecidas como *master pages*. Esta página define a marcação que é comum em todas as páginas de conteúdo (*content pages*), como também todas as regiões personalizáveis numa página base. Sempre que é feita uma alteração na master page é visualizada em todas as páginas da aplicação e assim, facilita as alterações de formatação em toda a aplicação (Mitchell, 2008).

Na Figura 8 é apresentado um exemplo da fusão de uma *master page* com uma *content page*. A *master page*, para além de toda a estrutura HTML, possui várias regiões como o cabeçalho (*Header Region*) e o rodapé (*Footer Region*), também pode possuir uma região que vai ser preenchida por controlos de conteúdo – na figura designada como *Page-Specific Content*. Quando uma *content page* é visitada, *ASP.NET* renderiza automaticamente *HTML* compatível com o navegador e é retornado ao utilizador a página

com a marcação definida na *master page* e no próprio controlo (Mitchell, 2008). Para além de *HTML*, a solução também integra *jQuery*, *Less* e a *framework Bootstrap*.

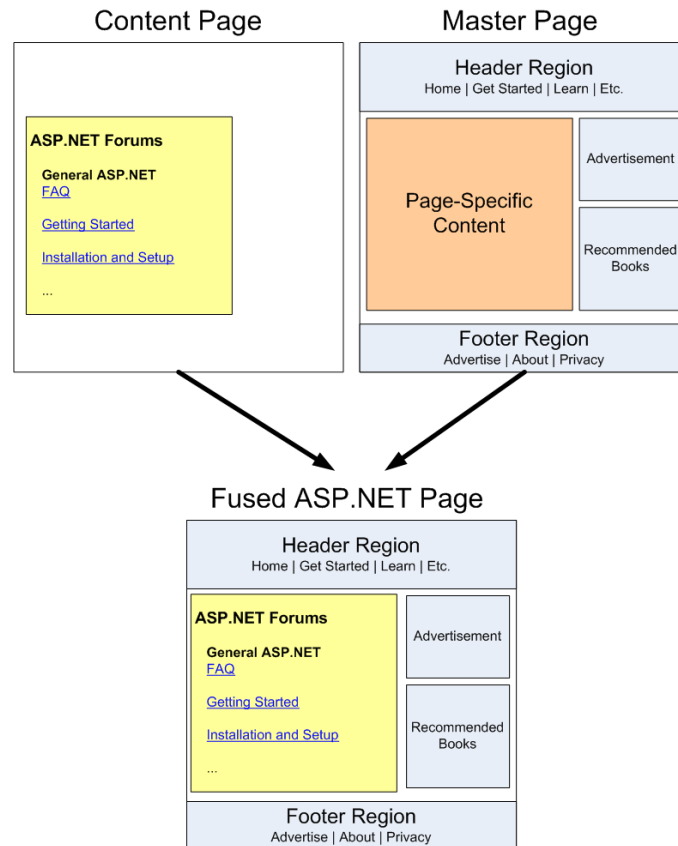


Figura 8 - Exemplo da fusão de *master page* com *content page*

jQuery é uma biblioteca *JavaScript* com o objetivo de adicionar interatividade e dinamismo às páginas web, proporcionando funcionalidades necessárias à criação de scripts (js.foundation, 2020).

Bootstrap é a *framework front-end* gratuita para o desenvolvimento *HTML*, *CSS* e *JavaScript* mais popular para desenvolver websites e aplicações móveis responsivas, focada na simplificação de desenvolvimento de páginas web informativas. *Bootstrap* fornece vários estilos globais e configurações importantes a aplicar, para que seja possível, por exemplo, não ter de desenvolver uma linha de *CSS* (Otto et al., 2020).

Igualmente para facilitar o desenvolvimento é utilizado o *Less*, uma linguagem de estilos de pré-processador dinâmico que pode ser compilada em folhas de estilo *CSS*. *Less* fornece vários mecanismos facilitadores no desenvolvimento, como: variáveis, *mixins* (classe que contém métodos para a utilização por outras classes sem ter de ser a classe pai), operadores e funções (Less, 2020).

Finalmente, o IDE utilizado foi o *Microsoft Visual Studio*.

4.2 Engenharia de Requisitos

Requisitos funcionais definem uma função do sistema ou de um componente, tal como o seu comportamento, enquanto os não-funcionais impõe restrições sobre como o sistema o fará. Tendo isto em conta, neste projeto não existem requisitos funcionais, pois o problema situa-se na implementação de um sistema acessível e não o desenvolvimento de novas funcionalidades.

4.2.1 Requisitos não-funcionais

Os requisitos não-funcionais do sistema passam por cumprir as diretrizes definidas pelos padrões de acessibilidade, como apresentado na secção 2.1.2.1, e estão classificados segundo o modelo de FURPS+ na Tabela 6.

Tabela 6 - Requisitos não funcionais segundo o modelo de FURPS+

Classificação FURPS+	Requisito não funcional
Funcionalidade	Cumprir as diretrizes definidas pelos padrões de acessibilidade, como apresentado na secção 2.1.2.1.
Usabilidade	Disponibilizar componentes de forma acessível. A aplicação deve ser acessível e fácil de usar, de modo a que os utilizadores tenham interfaces intuitivas e a melhor experiência de utilização possível, seguindo os quatro princípios: perceptível, operável, compreensível e robusto (Tabela 1)
Suporte	Adaptação dos componentes para incluir nos projetos a decorrer ou futuros. Compatibilidade com os browsers mais comuns.
+ (Requisitos de implementação)	Tecnologia a adotar: HTML, Less e Jquery

As diretrizes da acessibilidade estão divididas em quatro princípios: percetível, operável, compreensível e robusto, como referido na secção 2.1.2.1. Ou seja, a aplicação deve cumprir estes princípios, onde a informação e os componentes devem ser apresentados ao utilizador de diferentes formas, acessível por teclado ou por outra tecnologia assistiva, compreensível e a aplicação deve ser compatível com tecnologias.

As tecnologias a adotar são as mesmas que a solução *ebankIT* já adota: HTML, Less e Jquery. Os componentes já existentes vão ser adaptados para que sejam acessíveis e compatíveis para utilização nos projetos já existentes que têm como base a solução *ebankIT* e de fácil manutenção, ou seja, que sejam feitas alterações e que seja possível reutilizar.

4.3 Estudo de soluções

Nesta secção são discutidas as soluções que permitem resolver os problemas de acessibilidade existentes na solução.

4.3.1 Leitor de ecrã

A acessibilidade de uma aplicação por um leitor de ecrã é feita normalmente através de uma tecnologia assistiva, mas para que seja possível uma leitura o mais correta e eficiente possível é necessário adicionar elementos, principalmente em HTML.

Existem muitas diretrizes simples de implementar que alteram drasticamente a acessibilidade da aplicação por um utilizador com leitor de ecrã e algumas destas são (OPTASY, 2019):

- **Fornecer texto alternativo para cada imagem** - o texto alternativo pode ser apresentado por duas formas:
 - Dentro do atributo `alt` do elemento `img`;
 - Dentro do contexto ou ambiente da própria imagem, isto é, a informação pode ser fornecida em texto adjacente à imagem ou na página que contém a imagem. Nos casos em que o equivalente não pode ser apresentado de forma sucinta, pode ser fornecido um link para uma página separada que contenha uma descrição mais longa.

O atributo `alt` deve ser normalmente:

- Preciso e equivalente ao conteúdo da imagem;
- Sucinto, normalmente não são necessárias muitas palavras;
- Evitar redundância;
- Não utilizar frases como “imagem de ...” para descrever uma imagem, pois geralmente, não é necessário o utilizador saber que é uma imagem que está a transmitir a informação, excluindo o caso da imagem,

fotografia ou ilustração seja conteúdo importante que tenha de ser incluído no texto alternativo.

As imagens de fundo podem ser utilizadas como imagens decorativas, não sendo necessário incluir a imagem no fluxo de conteúdo da página e o atributo `alt` permanece vazio (W3C, 2019d).

- **Escrever código HTML, utilizando os atributos ARIA** – apresentado na secção 2.2.2.
- **Declarar a linguagem da página** – declarar a linguagem da página é importante porque assim o leitor de ecrã consegue detetar a língua da aplicação e rapidamente traduzir, no caso da aplicação ser acedida numa localização diferente ou que o utilizador tenha configurado outra língua.
- **Utilizar linguagem simples e descritiva** – tanto para links como para os cabeçalhos de uma página é necessário que a linguagem utilizada seja o mais simples para que o utilizador consiga rapidamente e facilmente chegar ao seu destino. Por consequente, os leitores de ecrã possuem funcionalidades como navegar na página através de uma lista de links, dos cabeçalhos, botões, tabelas e muito mais, por isso é importante manter o texto simples e curto, mas de forma a que o utilizador não se descontextualize.
- **Utilizar tags semânticas** - para manter o conteúdo legível e compreensível.

Apesar de existirem diversas diretrizes e tendo em conta que o desenvolvimento da solução *ebankIT* não tem em qualquer atenção a utilização de um leitor de ecrã, as apresentadas acima são as principais a focar numa fase inicial e que permitem de uma forma simples e fácil de implementar melhorar a experiência do utilizador com a aplicação através de um leitor de ecrã, sendo que estas alterações também melhoram a usabilidade da aplicação e a aplicação da acessibilidade é um processo contante.

4.3.2 Navegação por teclado

A navegação por teclado é um dos mais importantes aspetos de acessibilidade web, pois é feita por diversos utilizadores, até os utilizadores com incapacidades visuais. A navegação consiste na passagem por elementos clicáveis, que possuam o estado focado e na sua interação.

Alguns dos pontos importantes a ter durante o desenvolvimento de uma aplicação web para seguir uma navegação por teclado acessível são (WebAIM, 2018):

- O estilo de focos básico é fornecido automaticamente pelo browser e normalmente é feito por uma borda à volta do elemento e é aconselhado evitar esconder estes estilos padrão. Sendo que, os estilos aplicados quando os elementos estão focados são uma oportunidade de *design* e não um compromisso, pois eles podem melhorar

a aparência da aplicação. O objetivo deste estado é chamar a atenção ao utilizador para o elemento. Este estado pode ser acompanhado, para além da mudança de cor, como também por um *outline* ou *highlight* (Black, 2020).

- Um ponto importante em ter em atenção é que o estado do elemento focado não pode ser igual ao estado de *hover*, que é quando o utilizador passa com o rato por cima do elemento.
- A ordem de navegação numa página deve ser lógica e intuitiva. Geralmente, segue o fluxo visual da página: de cima para baixo e da esquerda para a direita. É aconselhado não usar o atributo `tabindex` maior ou igual a 1 para alterar a ordem da navegação, pois pode trazer problemas na ordem navegação da página (King et al., 2019).
- Quando um elemento que nativamente não seja clicável, por exemplo, uma `<div>` é necessário adicionar um atributo para que seja navegável por teclado, por exemplo o atributo `tabindex`, e ainda que os eventos inerentes ao elemento sejam disputados pelo teclado.
- Evitar páginas com muitos elementos navegáveis, pois pode dificultar a interação de utilizadores que apenas usem o teclado, para isso é recomendado utilizar um link no início da página que permite saltar para outras partes da página, como o conteúdo principal e o rodapé.

O atributo `tabindex` indica se um elemento pode receber foco e em que posição da navegação sequencial do teclado é que o elemento faz parte, dependendo do seu valor (MDN Web Docs, 2020d):

- Valor negativo, significa que o elemento não está incluído na navegação sequencial por teclado, mas pode ser incluído através, por exemplo, de JQuery;
- Valor zero (`tabindex="0"`), significa que o elemento é focável e a sua posição é definida pela ordem do código-fonte;
- Valor positivo, significa que o elemento é focável mas a sua posição é definida pelo valor do atributo. Sendo que `tabindex="4"` teria o foco antes de `tabindex="5"`, mas depois de `tabindex="3"`. E se diferentes elementos possuírem o mesmo valor positivo, a sua posição vai ser definida pela ordem do código-fonte.

4.3.3 Headings

Os *headings* correspondem a *tags* HTML h1 a h6 que definem a hierarquia de títulos e subtítulos numa *página web*, como representado na Figura 9. Organizar as páginas com *headings* apoia os utilizadores a terem noção da estrutura da página e ainda traz

vantagens aos utilizadores de leitor de ecrã, tornando possível a navegação na página pelos títulos.

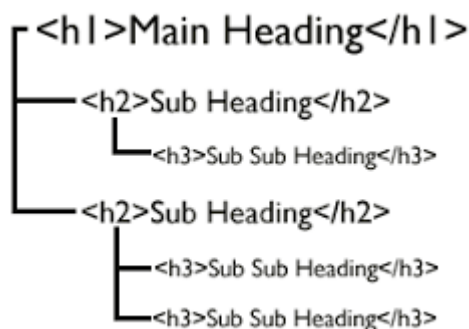


Figura 9 - Exemplo da estrutura de *headings*

Em 2017, o WebAIM questionou como os utilizadores de leitor de ecrã preferiam encontrar informações em páginas web extensas e quase 70% respondeu que preferem utilizar os títulos da página. Para que a utilização seja correta é necessário estruturar corretamente e hierarquicamente os títulos (Yale, 2020).

Outras orientações aconselhadas na implementação de *headings* são:

- Todas as páginas devem ter pelo menos um `<h1>` e deve representar a ideia mais importante na página.
- Não se deve saltar hierarquias, por exemplo, ter numa página `<h1>` e `<h3>`.
- Evitar a utilização de negrito em vez do título. Embora pareça um título, o código subjacente pode não estar definido corretamente e assim os utilizadores de leitor de ecrã não beneficiam.
- Evitar abusar nos títulos para não afetar a navegação.

4.3.4 Formulários

Os formulários são utilizados para recolher interações do utilizador com o website, como fazer *login* ou o registo. Na implementação de formulários, é necessário ter em consideração que os utilizadores normalmente preferem formulários curtos e simples, excluindo dados irrelevantes que possam provocar o abandono do formulário por parte do utilizador (STUDIO, 2020).

Existem aspetos a ter em consideração no caso do desenvolvimento de um formulário, para além dos referidos nos pontos anteriores, navegação por teclado e utilização do leitor de ecrã, e a acessibilidade por elemento presente numa página que vai ser

apresentado no capítulo seguinte. Um formulário acessível deve permitir uma fácil interação por todos utilizadores, incluindo pessoas com incapacidades situacionais e permanentes, trazendo vantagens como:

- Formulários acessíveis têm uma melhor estrutura, instruções de preenchimento e *feedback*, o que possibilita a pessoas com deficiência cognitiva entenderem melhor o preenchimento do formulário;
- As pessoas que utilizam entrada de voz podem utilizar as *labels*² como comandos para ativar controlos e mover o foco para os campos a preencher;
- Áreas grandes clicáveis que incluem *labels*, botões rádio e *checkboxes*, beneficiam pessoas com destreza limitada.
- E a associação de *labels*, conjuntos de campos e outros elementos estruturais permitem que pessoas que utilizam leitor de ecrã possam identificar e entender os formulários com mais facilidade.

Para que seja possível tornar os formulários mais próximos de serem acessíveis a todos, é necessário ter em conta os seguintes pontos:

- Controlos de *labels*

As *labels* representam as legendas específicas para um item numa interface do utilizador e descrevem a finalidade do controlo do formulário.

Sempre que possível deve ser utilizado o elemento `label` e o atributo `for` deve se referenciar ao identificador do controlo. Para o caso que o texto não seja visível é importante que exista uma `label` associado ou que seja adicionado o atributo `aria-label` com a legenda ou o atributo `title` para a legenda ser transmitida a utilizadores visuais (WebAIM, 2018).

- Controlos de agrupamento

O agrupamento de controlos relacionados torna os formulários mais compreensíveis para todos os utilizadores, fazendo com que utilizadores se foquem em grupos menores e mais fáceis de gestão, em vez de tentar compreender de forma inteira.

Uma das abordagens é utilizar o elemento `fieldset` que fornece um contentor para controlos relacionados e o elemento `legend` atua como um cabeçalho identificativo do contentor. Esta abordagem pode ser utilizada para agrupar botões rádio e `checkboxes`. Sendo possível fazer esta associação com WAI-ARIA (W3C, 2020c).

² Etiquetas associadas aos campos.

- Instruções

Fornecer instruções permite que apoiar o utilizador no preenchimento do formulário, sendo essencial sinalizar os campos de entrada obrigatórios e opcionais e outras informações relevantes (W3C, 2019b).

- Validações

Validações correspondem às confirmações feitas aos valores introduzidos no formulário e podem ser feitas tanto do lado do servidor, que acontece quando o formulário é submetido, ou do lado do cliente dinamicamente, através de scripts.

Normalmente, quando valores introduzidos não estão de acordo com os termos de aceitação são apresentados erros na página. Estes erros podem ser apresentados no topo do formulário com a identificação de cada campo com erro, exemplos ou sugestões para o corrigir e a exposição adequada destas informações para tecnologia assistiva. E também pode ser apresentado uma mensagem abaixo no campo de entrada.

Estes erros são apresentados usualmente de cor diferente para chamar a atenção do utilizador, mas esta prática não pode ser aplicada sozinha pois, por exemplo, para pessoas com deficiência visual não vai ter o efeito pretendido. Por isso que a própria mensagem tem de indicar que se trata de um erro, para além de sugestões para corrigir o erro.

No caso da validação de formulários dinâmicos é necessário gerir o foco para direcionar o utilizador ao campo com erro e não esquecer de associar o erro ao campo de entrada. A mensagem geralmente deve ser associada ao campo utilizando o atributo `aria-describedby` (Avila, 2014).

A nível de validações, também deve estar presente um mecanismo para que o utilizador consiga desfazer a ação, ter a oportunidade de rever a informação inserida e que a identificação de erros seja feita antes do utilizador submeta o formulário.

- Notificações ao utilizador

As notificações ao utilizador têm como objetivo fornecer um feedback ao utilizador sobre o resultado da submissão do formulário. Estas notificações devem ser claras e fáceis de entender, para o caso de mensagens de erro, o utilizador consiga compreender como proceder para resolver o erro (W3C, 2019c).

Pode-se concluir que os formulários são uma parte importante para a usabilidade geral das aplicações web, pois são utilizados em diversos tipos de interações na internet. Por isso, a melhoria contante da acessibilidade e usabilidade dos formulários benéfica todos os intervenientes (Abati, 2019).

4.3.5 Elementos

As aplicações web possuem determinados elementos transversais entre si e foi feito o levantamento de algum destes e o seu estudo a nível de acessibilidade, sendo que cada elemento tem uma abordagem específica de como tornar acessível (WebAIM, 2020b).

4.3.5.1 Caixas de texto

As caixas de texto são definidas como um `<input>` em HTML e podem possuir vários atributos que melhoram a sua acessibilidade, como um `placeholder`³, sendo que este atributo não é legível para todos os leitores de ecrã.

Para pessoas com deficiências cognitivas que não utilizam nenhuma tecnologia assistiva, o *placeholder* desaparece quando é inserido um valor na caixa de texto e, por esse motivo, um *placeholder* não é suficiente porque o utilizador pode não se lembrar do rótulo (D. MacDonald, 2020).

Os pontos importantes a ter em consideração no desenvolvimento de uma caixa de texto são os mesmo que no desenvolvimento de um formulário, apresentado na secção 4.3.4.

4.3.5.2 Link

Os *links* têm como função direcionar o utilizador para outro local, para uma página diferente ou para uma localização na página diferente. Existem alguns pontos importantes em ter atenção no desenvolvimento de um link (STVDIO, 2020):

- Texto – o texto do link deve descrever a finalidade e destino. Se o link estiver sob forma de imagem ou ícone, este deve ter o atributo `title` para que as pessoas com leitor de ecrã consigam aceder.
- Estilo – os links devem estar indicados claramente, por exemplo com o texto sublinhado e não terem alterações de cor apenas quando passa o rato ou o elemento é focado.
- Navegação – os utilizadores devem conseguir aceder aos links pelo teclado e ativar com a tecla “Enter”.
- Imagens vinculadas – as imagens devem ter um texto alternativo descritivo, para ser usado como texto do link para leitor de ecrã. O texto da imagem é inserido no atributo `alt` seguido da palavra link.

³ Pequena dica, uma pequena frase, uma palavra, que tem o intuito de ajudar o utilizador a entender como deve preencher a caixa de texto.

- Nova janela ou separador – se o link abrir um novo separador ou o utilizador vai sair do atual formato ou da aplicação, esta ação deve estar explicada no texto do link. Isto deve ser desenvolvido pois pode ser desorientador para os utilizadores, especialmente com problemas cognitivos, visuais ou de mobilidade.

Quando não existem elementos visíveis que descrevam o link, adiciona-se o atributo `aria-label` ao link para fornecer um rótulo de texto descritivo ao link. No caso de existirem elementos descritivos visíveis na página, deve ser utilizado o atributo `aria-labelledby` (W3C, 2014).

4.3.5.3 Botões

Os botões ativam uma ação ou evento, como submeter um formulário, cancelar uma ação ou executar um comando. Sendo que podem ser ativados com as teclas *Enter* e *Space* e não devem ser utilizados para abrir páginas web, sendo os links que devem ter essa função.

Os botões devem manter uma aparência similar ao longo da aplicação web para facilitar a identificação por parte do utilizador e para manter a coerência, para além disto é necessário distinguir pelo *design* a sua prioridade, desde um botão primário ao secundário (Lorensson, 2017).

Ainda existem elementos que não são `<button>` nem `<input type="button">`, adiciona-se o atributo `role="button"` e assim, o leitor de ecrã consegue identificar o elemento como um botão (MDN web docs, 2019).

4.3.5.4 Botões Radio

O botão radio, mais conhecido por *radio button*, é um dos controlos que representa o conjunto de opções mutuamente exclusivas, sendo normalmente representado com um controlo redondo.

Os eventos associados a um radio button são a seleção e a limpeza da seleção. Quando é feita uma seleção é esperado que seja limpo qualquer seleção anterior no conjunto. A interação por teclado para disputar estes eventos são feitos pela tecla *Tab* para focar no conjunto de botão radio e é esperado utilizar as teclas das setas para alterar a seleção entre os botões radio do conjunto. Sempre que é focado o conjunto, o foco vai para elemento selecionado (W3C, 2007).

O estado de seleção deste botão pode ser identificado pela marcação ARIA `aria-checked`, que no caso do botão radio estar selecionado tem valor `"true"`. Ainda existe o papel `radiogroup` que define o conjunto de botões radio e o atributo `aria-labelledby` que define o rótulo do conjunto de botões (W3C, 2020a).

4.3.5.5 Checkbox

Checkbox é um tipo de campo que já contém informações predefinidas e que o utilizador deve selecionar, com a opção de selecionar apenas uma ou mais. É idêntico ao *radio button*, mas com a diferença que é possível selecionar mais que uma opção.

O estilo padrão da *checkbox* é uma caixa quadrada que fica marcada quando selecionada. É possível navegar através da tecla Tab até ao elemento e utilizar o Space para alterar o estado da *checkbox* entre *checked* e *unchecked* (W3C, 2020b).

4.3.5.6 Drop-down

O elemento drop-down permite a seleção de uma ou mais opções de uma lista e é representado em HTML pela tag `<select>`, sendo cada opção representada por `<option>` (MDN Web Docs, 2020a).

Quando uma opção é selecionada esta passa para o estado *selected* e as drop-downs são navegáveis através da tecla Tab até ao elemento, Space para expandir as opções e as setas para navegar entre as opções (WebAIM, 2018).

4.3.5.7 Carrossel

O carrossel, também conhecido por *slider*, consiste em mostrar uma lista de itens de uma forma interativa, existindo várias formas de customizar.

Para tornar um carrossel acessível é necessário que o utilizador seja capaz de parar o movimento do carrossel porque pode estar a passar demasiado rápido e tornar a informação difícil de ler; ainda que seja possível navegar por todos os elementos com o teclado e a posição do foco deve ser gerida de forma compreensível (W3C, 2019a).

Existem vários plugins que permitem através de scripts fazer essa gestão e adicionar os atributos necessários para o componente ser acessível. Um desses plugins designa-se por slick, fácil utilização e possui um atributo que permite habilitar ou desabilitar a acessibilidade do componente. Este atributo quando possui o valor *true*, permite que o utilizador navegue através do teclado pelas setas e pelos slides visíveis e ainda pelos elementos clicáveis dentro dos slides, por exemplo, um slide pode possuir a informação e um link para o detalhe (Wheeler, 2014/2020).

Com o slick é possível adaptar o componente para que seja responsivo através do método *“responsive”* e assim alterar ou adicionar condições com *breakpoints* no tamanho do ecrã e alterar as opções do *slick*.

4.3.5.8 Skip navigation

O objetivo do *link skip navigation* é fornecer um caminho na página que permite saltar conteúdos. O mais comum é situado no topo da página para que seja o primeiro elemento a ser navegado que permite saltar para o conteúdo principal e para o rodapé.

O método mais fácil para implementar este componente é adicionar uma ancora no topo com o destino de outra ancora no início do conteúdo (WebAIM, 2013).

4.3.5.9 Imagens

Existem diferentes tipos de imagens numa aplicação *web*: as informativas, as decorativas e as de texto. As imagens informativas mais importantes não devem ser apresentadas por estilo no formato de *background*, enquanto para imagens decorativas é indiferente (David MacDonald, 2020).

As imagens decorativas não possuem nenhuma função ou não transmitem nenhuma informação e devem ser escondidas da leitura de tecnologias assistivas, isto pode ser feito de diferentes maneiras:

- Adicionar o atributo `alt` a `null`;
- Adicionar o atributo ARIA `role="presentation"`;
- Adicionar a imagem como `background`.

Todas as *tags* `` devem ter o atributo `alt`, mas para as imagens informativas é necessário adicionar conteúdo alternativo, com o atributo `alt` ou com outras formas. As diferentes formas de fornecer um texto alternativo a uma imagem é com a adição do atributo `aria-label` ou `aria-labelledby`.

As imagens de texto devem ser evitadas e são utilizadas normalmente para logotipos. No caso de a imagem conter um pequeno texto, o texto alternativo deve ser exatamente igual ao da imagem, caso contrário, o texto também deve ser apresentado noutra parte da página.

Por fim, é recomendada a implementação de imagem como SVG para resolver problemas relacionados com a redimensão das imagens (Yale University, 2020).

4.4 Resumo

A integração da acessibilidade na solução *ebankIT* nesta fase inicial de adaptação das normas de acessibilidade enquadra-se na implementação da navegação por teclado, da

responsividade da página com zoom, da utilização básica do leitor de ecrã, na estruturação da página e dos estilos de diferenciação do estado focado e correção dos contrastes. Para isso o processo de desenvolvimento deve ter em conta as normas implementadas pela W3C, seguindo as diretrizes WCAG.

5 Implementação

Neste capítulo é apresentada a implementação realizada no projeto, de acordo com o estudo de soluções e recomendações descritas nas secções anteriores, tendo uma contextualização do projeto.

5.1 Contextualização do projeto

Como referido na secção 4.1.3, o projeto enquadrado neste estágio já se encontra em curso há algum tempo e com diversas versões lançadas, sendo um projeto de grande dimensão.

A primeira página a ser apresentada ao utilizador assim que entra na aplicação é a página de *login* (Figura 10), onde é possível visualizar um controlo de alteração de idioma, a caixa de texto para introduzir o nome do utilizador, o botão para continuar, entre outras coisas.



Figura 10 - Login da aplicação ebankit

De seguida ao *login*, é apresentada ao utilizador a *homepage* que é constituída por várias regiões: o header, o menu, a página com o conteúdo e o rodapé.

Dentro do menu verifica-se diversas opções separadas por algumas categorias:

- Meu ebankit – apresenta todas as opções de menu relativas à manutenção de conta, como alterar palavra-passe, contacto, nome de utilizador;
- Contas – As opções relativas às contas, como a lista, o detalhe e a consulta dos movimentos. Ainda tem a abertura de conta, opções de cheques
- Transferências e pagamentos – corresponde a todas as opções relativas a transferências e pagamentos, como pagamento de serviços ou ao Estado; lista, detalhe e criar débito direto; transferências nacionais e visualização das operações agendadas;

As opções de menu de Poupanças, Cartões e Créditos possuem transações idênticas às contas, como a lista de contas, o detalhe e movimentos.

As páginas possuem componentes muito idênticos entre si. No caso das opções de menu de consulta de dados, por exemplo a lista de conta ou o detalhe, possuem componentes e estilos transversais. Na Figura 11 é possível observar a região do conteúdo de um detalhe de conta, possuindo um seletor de contas, um componente de acessos rápidos que direciona para os movimentos, atalhos rápidos e favoritos, o detalhe de conta no formato de *label* e valor, e um componente de extração de informação, através de impressão, enviar por email ou por geração de ficheiro PDF.

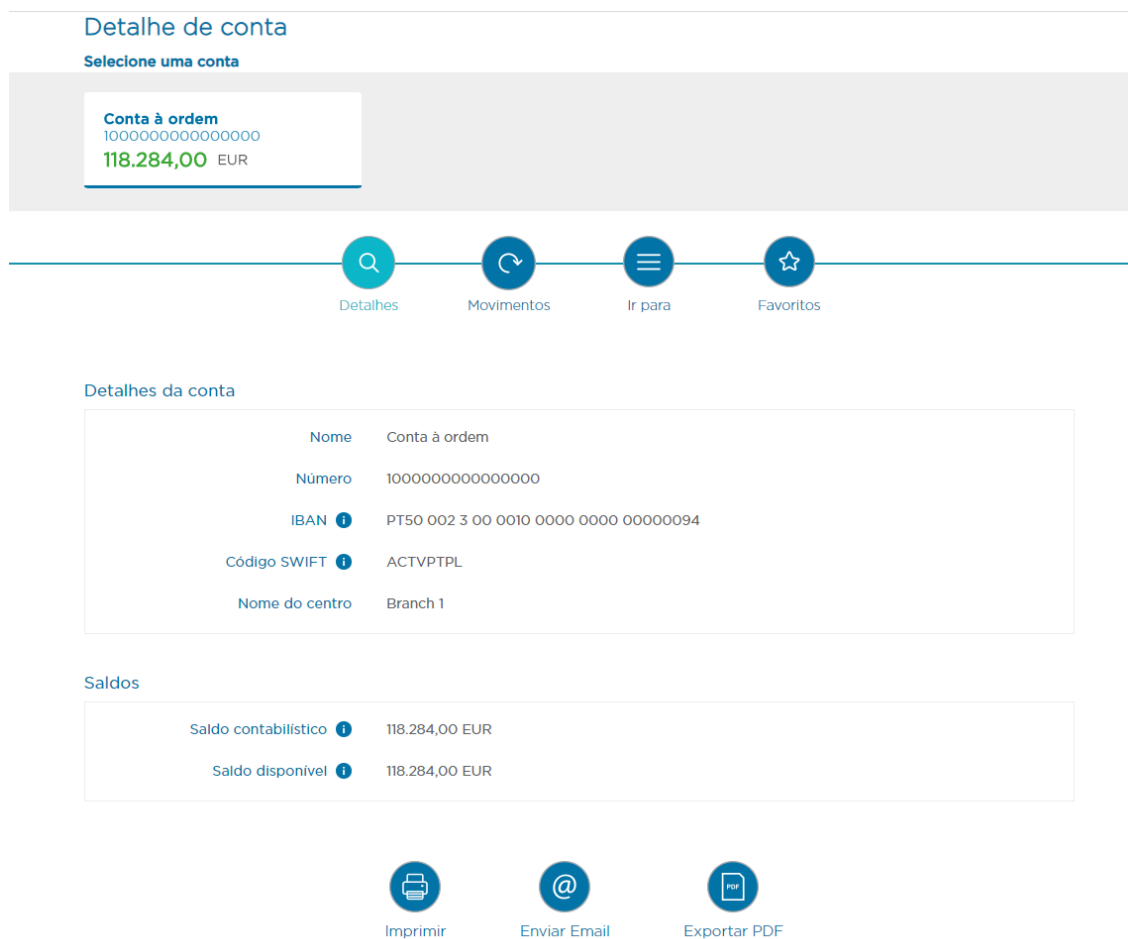


Figura 11 - Exemplo conteúdo de detalhe de conta

Nas opções de menu transacionais, isto é, que há a inserção de dados pelo utilizador e estes dados têm validações, existem componentes e estilos transversais mas são componentes mais complexos na ótica de adaptação a acessibilidade. A razão para serem mais complexos é que estes elementos sofrem interações por parte do utilizador, alteram o seu estado e propriedades. Um exemplo de uma transação transacional é as Transferências, como apresentado na Figura 12. Na região do conteúdo das transferências está presente um seletor de contas, que vai ser a conta origem de transferências; uma região de indicadores da posição da transação: “Dados”, “Confirmação” e “Conclusão”; uma região de inserção de dados onde pode estar presente todo o tipo de elementos, como uma caixa de texto, botões radio, uma dropdown⁴ e por fim, tem os botões para continuar com a submissão dos dados ou para cancelar e voltar para a *homepage*.

⁴ Elemento de interface que é similar a uma lista, que permite que o utilizador escolha um valor de uma lista de opções que "cai para baixo".

Dados | Confirmação | Conclusão

Transferências nacionais

Selecione uma conta

Conta à ordem
10000000000000000000
118.284,00 EUR

Conta destino

Transferências mesmo banco

Conta ebankIT

Transferências para outros bancos

Detalhes da transferência

Valor EUR

Limite disponível para esta transação: 1.000.000,00 EUR

Descrição

Notificação

Email

Agendamento

Tipo Pontual Recorrente

Data de pagamento

Figura 12 - Exemplo conteúdo de transferências

5.2 Integração de acessibilidade

A integração da acessibilidade na solução *ebankIT* nesta fase inicial vai abranger os pontos referidos na secção 4.4. Sendo esperado que a adaptação seja contínua e que seja implementada desde início em todas versões da solução.

É esperado que durante o desenvolvimento sejam consultadas as diretrizes e o estudo para verificar a solução melhor para cada caso encontrado. Ainda, para a integração de acessibilidade são utilizadas ferramentas de apoio e de avaliação para detetar falhas de acessibilidade, para além das falhas já conhecidas.

As alterações realizadas são feitas manualmente no código, não tendo sido utilizada nenhuma ferramenta de geração de código, pois foi necessária a adaptação do código já existente. As ferramentas de avaliação para detetar falhas de acessibilidade foram utilizadas durante a implementação como apoio em encontrar falhas e na validação da correção das alterações testáveis automaticamente.

Para finalizar, o código alterado, depende da correção, passou por alterações de HTML, contrastes por LESS e classes de JQuery.

5.3 Componentes

Nesta secção são abordados todos os componentes adaptados encontrados ao longo da solução. Os componentes descritos foram criados no projeto já partilhados em outras versões utilizadas em projetos ou que ainda vão ser partilhados.

5.3.1 Skip navigation

O *link skip navigation* não existia na solução e para que os utilizadores de leitores de ecrã e os que utilizam o teclado para navegar na aplicação não tenham de percorrer toda a página para chegar ao conteúdo principal foi implementado um componente *skip navigation* para facilitar a navegação na página.

Neste projecto foi adicionado ao topo da página um componente que permite fornecer caminhos rápidos para outras regiões da página e apenas é visível quando a página é navegada por teclado, sendo o primeiro elemento da página a receber o foco como apresentado na Figura 13 .

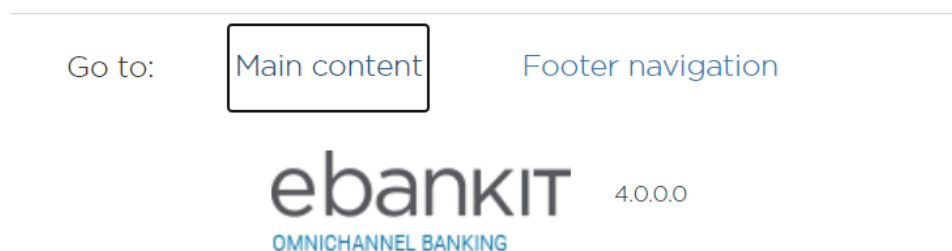


Figura 13 – Resultado visual de *Skip navigation* da solução *ebankIT*

Como se pode verificar este componente permite ao utilizador saltar para o conteúdo principal ou para o rodapé e está presente em todas as páginas da aplicação.

A nível tecnológico foi adicionado à master page duas âncoras que permitem passar diretamente o foco por teclado ou para o conteúdo principal, ignorando o cabeçalho e o menu, ou para o rodapé, o código adicionado está presente na Figura 14.

```

▼<div class="skiplinks mod-skiplinks">
  ▼<div class="skiplinks-container">
    <span id="skiplinks-label">Go to:</span>
    ▼<ul>
      ▼<li>
        <a id="content-link" aria-labelledby="skiplinks-label content-link" href="#skip-nav">Main content</a>
      </li>
      ▼<li>
        <a id="footer-link" aria-labelledby="skiplinks-label footer-link" href="#skip-main">Footer navigation</a>
      </li>
    </ul>
  </div>
</div>

```

Figura 14 - *Skip navigation* da solução *ebankIT*

5.3.2 Header

O *header*⁵ é a secção inicial de qualquer página *web* e neste projeto inclui vários elementos, como o logótipo, o controlo de alteração de linguagem, o atalho para o carrinho de operações, o controlo para alterar o perfil de utilizador e ainda o botão para fechar a sessão iniciada (*logout*), como apresentado na Figura 15.

Verificou-se que existiam problemas na navegação por teclado ao longo do *header*, pois a maioria são elementos visualmente compostos por um ícone.

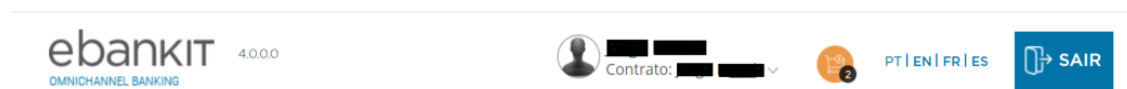


Figura 15 - *Header* da solução *ebankIT*

Na imagem acima está o exemplo do *header* da aplicação *web* e é possível visualizar os seguintes elementos e as suas respetivas melhorias/alterações:

1. O logótipo da empresa é uma imagem clicável com ligação à *homepage* do cliente. A nível de alterações, este elemento sofreu uma alteração para que seja navegável por teclado e ainda que tenha um atributo e ainda o atributo `alt` com a informação descritiva da imagem.
2. O seletor de perfil, no caso de o utilizador possuir mais que um perfil, é apresentado um ícone para abrir uma lista com os outros perfis, sendo cada linha clicável para o utilizador alterar o perfil. Assim sendo, foram adicionadas as alterações para que fosse possível a navegação por teclado através do atributo `tabindex` e ainda scripts para adicionar ações via teclado ao componente para que seja possível abrir, percorrer a lista de perfis e no caso de passar para outro elemento vizinho, a lista de perfis fechar.

⁵ Região/cabeçalho no topo de cada página *web* que contem informações.

3. A nível do carrinho também foi adicionado o atributo `tabindex` para ser encontrado no fluxo de navegação por teclado, a descrição para ser lida por um leitor de ecrã com a informação do número de operações adicionadas e foi alterado visualmente o estado *focus* para a distinção de estados, mudando a cor do ícone.
4. O controlo de linguagem consiste em percorrer todas as abreviaturas de linguagem apresentadas, alterando a cor no caso do estado *hover* e para *focus*. E ainda foi adicionado a descrição da linguagem para o leitor de ecrã ler a descrição da linguagem e não a abreviatura, através do atributo `aria-label`.

O botão de sair foi alterado completamente, pois não seguia as regras de contraste apresentadas pelas diretrizes de acessibilidade. Então, foi refeito para que seja alterada a cor quando está no estado *focus* ou *hover* e ainda que seja manuseável por teclado.

5.3.3 Menu

O menu é um componente da aplicação que é transversal a muitos projetos que já utilizam como base a solução ebankIT, não tendo sofrido muitas alterações ao longo do tempo, e aparece de seguida ao *header*. Tendo em consideração que é um componente já utilizado, isso foi um impedimento para que fossem feitas muitas alterações, pois podia criar problemas quando os projetos incluíssem esta versão acessível.

Por isso, e sabendo que é necessário fazer os mínimos requisitos para tornar o menu acessível, foram feitas as alterações:

- A navegação por teclado não estava funcional, por isso foram adicionados scripts para o utilizador navegar com a tecla *Tab* pelas opções principais do menu e apenas se carregar no *Enter* é que vai ser aberto o submenu e é igualmente utilizada a tecla *Tab* para percorrer o submenu.
- Foram adicionados estilos diferentes para quando a opção do menu e do submenu estiver no estado *hover* e *focus*.
- O submenu apresenta opções com estado *disabled*, sendo que estas opções não são navegáveis por teclado e são lidas como *disabled*, pois é adicionado o atributo `aria-disabled = "true"`.
- Para o ícone de pesquisa foi adicionado o atributo `aria-label` com o texto alternativo respetivo.
- Os submenus não devem ser abertos ao usar a tecla *tab* para navegar pelo menu, pois assim os utilizadores terão de percorrer todos os itens do submenu para chegar ao próximo item de nível superior. A abordagem para resolver este

problema foi utilizar um script em que o submenu abre apenas se o utilizador ativar o item de nível superior e fechado quando o foco sai do submenu.

5.3.4 Seletores de conta

Muitas das páginas existentes nesta aplicação *web* apresentam um seletor de contas que permite escolher a conta e por exemplo, ver o detalhe dessa conta ou selecionar a conta origem de uma transferência. Existe diversos seletores de conta, para os diferentes tipos de conta e para diferentes páginas, existem cerca de oito seletores de conta diferentes.

O seletor é um carrossel que cada slide apresenta uma conta diferente e se não for possível mostrar no ecrã todas as contas são apresentadas setas laterais para a navegação, como apresentado na Figura 16.

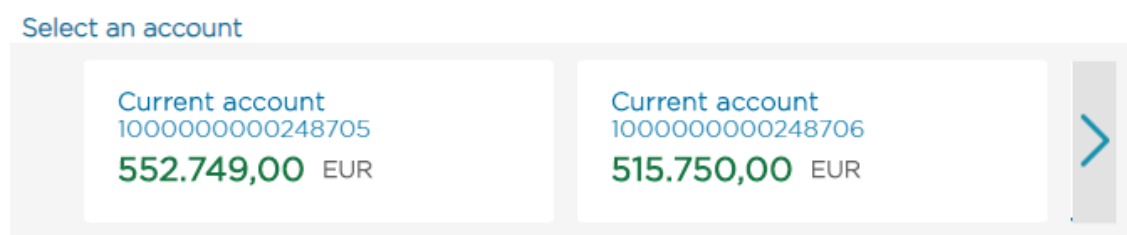


Figura 16 - Exemplo de seletor de contas da solução *ebankIT*.

Este componente foi adaptado para a utilização do *plugin slick* suportado por *jQuery*, pois é totalmente acessível por teclado e facilita a implementação do responsivo do componente, por entre outras coisas. A navegação é feita através do *Tab* para focar o componente e de seguida são utilizadas as setas do teclado para percorrer o carrossel e é também utilizado o *Tab* para percorrer as contas visíveis e para sair do componente e focar no componente seguinte.

O plugin já faz a gestão dos atributos *aria* para que sejam identificados como escondidos com `aria-hidden="true"` e `aria-hidden="false"` para os elementos visíveis.

E ainda, foram adicionados os eventos `onkeyup` e `onkeydown` para que utilizador consiga interagir com o seletor de contas, selecionando uma das contas apresentada.

5.3.5 Caixas de texto

A solução apresenta diferentes tipos de caixas de texto customizadas que foram desenvolvidas ao longo de todo o processo de criação de novas versões da solução com diferentes necessidades e conhecimento. Sendo que, foi feito uma validação nestes componentes de caixas de texto para quando não apresentasse uma `<label>` associada ao `<input>` que adicionasse o atributo `aria-label` com a descrição do campo.

Para além disto, estes componentes possuem uma opção, no caso de serem campos obrigatórios, de apresentarem um * antes da caixa de texto. Como o * não é automaticamente lido como obrigatório por um leitor de ecrã, adicionou-se o atributo `aria-hidden="true"` ao * para que este não seja lido e `aria-required="true"` ao elemento `<input>` para o leitor de ecrã assumir como obrigatório.

Na Figura 17 está apresentado um exemplo de uma caixa de texto com uma *label* associada "IBAN" em que tem um * como um indicativo de campo obrigatório em que tem o atributo `aria-hidden="true"` e o `<input>` possui o atributo `aria-required`.

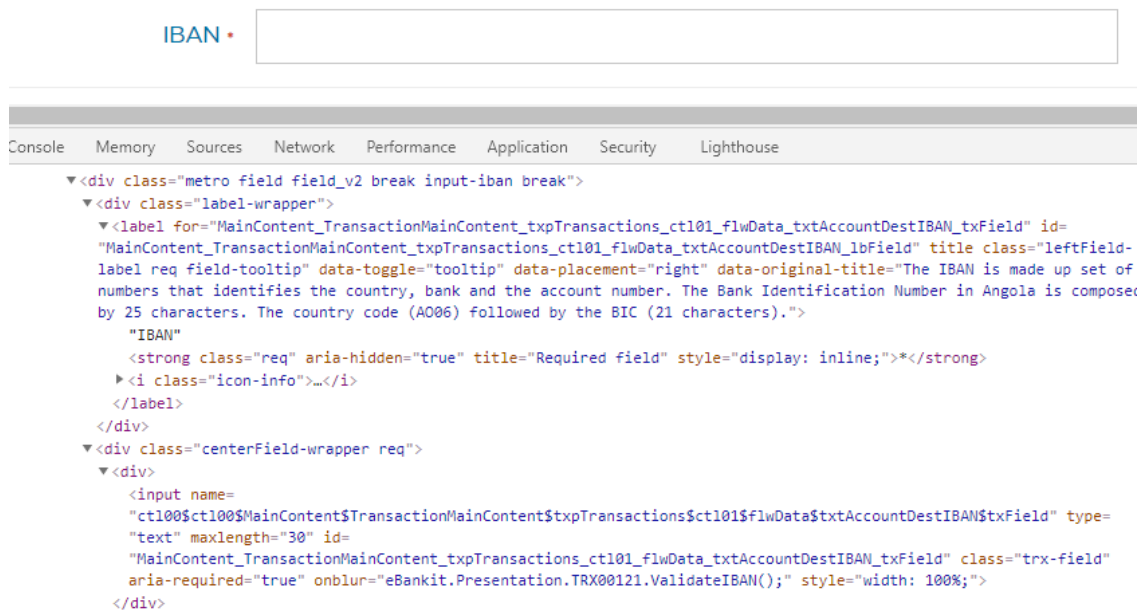


Figura 17 - Resultado visual e código de uma caixa de texto

5.4 Páginas

Foram alteradas as páginas com mais importância na solução, mas ao longo da integração da acessibilidade, os componentes existentes nas páginas vão se repetindo tornando redundante a explicação da sua alteração.

5.4.1 Login

O *login* é efetuado entre dois e quatro passos, dependendo das configurações de utilizador. Possui uma página inicial onde se insere o *username* de utilizador, na página seguinte é inserido a palavra-passe e pode apresentar um ecrã para inserir um código enviado ao utilizador de verificação e ainda um ecrã de seleção de perfis, no caso de o utilizador possuir mais que um perfil.

Na primeira página de *login*, para além de apresentado uma caixa de texto para a inserção do *username*, também mostra o controlo de linguagem, um link para o registo de utilizador no caso de não ser aderente e um link para redirecionar para a página de recuperação de palavra-passe. Nesta página foram feitas alterações para que fosse possível passar via teclado através de todos os elementos navegáveis e ainda foram adicionados scripts para que os componentes ativassem quando a utilização do teclado. E ainda, o link de recuperação de palavra-passe não é um elemento HTML âncora, então foi adicionado `role="link"`.

As páginas de inserção de palavra-passe e do código de verificação são idênticas, possuindo apenas os botões para avançar ou cancelar e um campo de texto.

A página de seleção de perfil tem a opção de apresentar as opções de perfis em lista ou em carrossel. Na opção por carrossel, este componente foi adaptado para utilizar o *plugin slick*, como descrito no componente seletor de contas no capítulo 5.2.1.3.

5.4.2 Homepage

A *homepage* do cliente apresenta vários componentes e dependendo da sessão do utilizador os componentes apresentados serão diferentes.

Neste processo foi garantido que todos os elementos clicáveis eram navegáveis e acionados pelo teclado, que o contraste está dentro das diretrizes de acessibilidade, que os componentes constituintes da página não escondem informação dependendo do tamanho ecrã e que tudo estava dentro das regras para que o leitor de ecrã leia a informação existente na página.

Para isso, foram alterados dois componentes, um deles presente na Figura 18, que possuem um comportamento carrossel para utilizar o slick que já integra a acessibilidade dinamicamente no componente sem qualquer outra alteração.

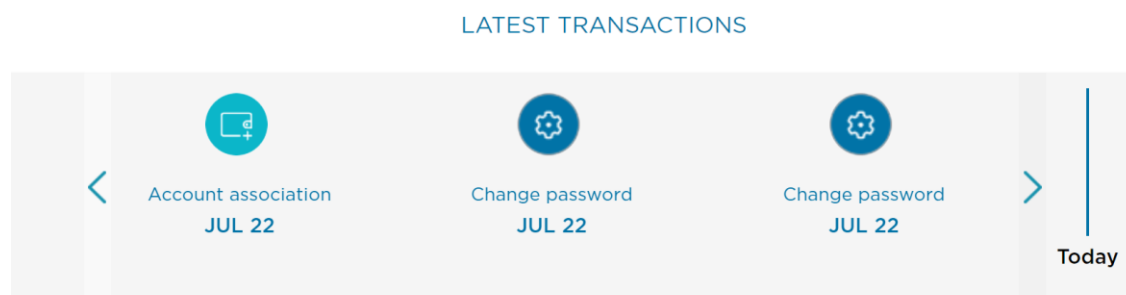


Figura 18 - Últimas transações realizadas pelo utilizador presente na *homepage* da solução *ebankIT*.

Ainda foi adicionado scripts em JQuery para que seja possível acionar e navegar nos componentes por teclado e todos as imagens clicáveis possuem uma descrição para a leitura por parte de tecnologias assistivas direcionadas a esse objetivo.

5.4.3 Mensagens

A solução possui um submenu para mensagens recebidas, enviadas, eliminadas e a opção para escrever uma nova mensagem, como apresentado na Figura 19. Escolhendo qualquer opção, a página possui um seletor para cada as outras opções.

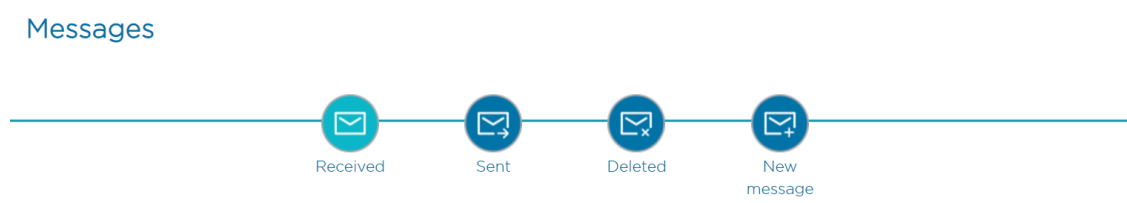


Figura 19 - Submenu presente na página “Mensagens” da solução *ebankIT*.

Nas opções de mensagens recebidas, enviadas e eliminadas, as páginas possuem uma lista de mensagens com um link para o detalhe e uma ação de eliminar a mensagem, esta opção não existe na opção mensagens eliminadas. Para que seja possível navegar pelos elementos clicáveis na página foi adicionado o atributo `tabindex="0"` e um script para disputar a ação do elemento conforme a tecla premida.

Para além disto, foram revistos os contrastes e os estilos para o estado `focus` dos elementos. E ainda, adicionados textos alternativos nas imagens que representam o link do detalhe e de eliminar mensagem.

5.4.4 Meus produtos

Na página designada por Meus produtos são apresentados os produtos do utilizador, como as contas poupança, os empréstimos e as contas corrente.

A listagem de produtos apresenta uma imagem decorativa e atalhos para apresentar os últimos movimentos e para redirecionar para o detalhe do produto.

Para atingir os requisitos de acessibilidade foi realizado:

- Em cada produto é apresentada uma imagem decorativa, não sendo necessário incluir a imagem no fluxo de conteúdo da página, por isso não necessita de texto alternativo.

- Os links de últimos movimentos e detalhe de conta são acessíveis por teclado, apresentam estilo diferenciado quando estão focados e ainda foi adicionado texto alternativo através do atributo `aria-label`. Não esquecendo de alterar o texto para diferenciar quando o ícone está expandido ou colapsado.
- A nível de contraste foi apontado o verde dos valores como baixo contraste e resolvido pela equipa de *design*.
- A nível de cabeçalhos, a página sofreu uma reestruturação pois não apresentava uma lógica correta.

Cada linha apresenta uma imagem decorativa e como tal, foi adicionado um atributo `alt=""` a cada imagem.

5.4.5 Detalhes de conta

A página do detalhe de conta possui um seletor de contas, a informação da conta em texto e no fim um componente que permite fazer uma impressão da informação, exportar para PDF ou enviar por email o conteúdo, como apresentado na Figura 20.

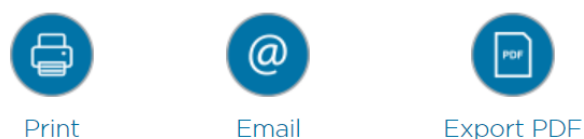


Figura 20 - Componente de exportação de informação na solução *ebankIT*.

As alterações no seletor de conta estão descritas na secção 5.3.4. A informação da conta não sofreu qualquer alteração, pois estava correta a nível de contraste, navegação e leitura por parte do leitor de ecrã.

O componente que possui as três opções de exportar a informação teve que sofrer algumas alterações pois não era navegável por teclado e além disso o componente é apresentado com uma imagem e um texto descritivo, mas a imagem não possui qualquer ligação ao texto. Por isso adicionou-se o atributo `aria-labelledby` a cada imagem.

5.4.6 Transferências

Na página referente às transferências existem diversos componentes diferentes. Primeiramente foi implementado a navegação por teclado em todos os componentes existentes na página e para isso:

- Alterou-se o seletor de favoritos presente no início da transação das transferências para utilizar o plugin slick que facilita e automatiza a acessibilidade.
- Para os botões radio customizados presentes em toda a aplicação web foi alterado uma classe de estilos para que assim, fosse possível navegar por teclado. Esta alteração passou de retirar o `display:none`; para o `<input type="radio">` e adicionar `opacity:0`;
- Adicionou-se `tabindex="0"` e um script JQuery com o evento `onkeypress` para incluir os elementos na navegação a todos os que possuem uma ação e para conseguir acioná-los, mas que não permitem focar ou navegar através de teclado.
- Alteração de contraste de alguns elementos conforme a sugestão da equipa de *design*.

5.5 Tratamento de erros

Pela dimensão desta solução e que este tema de tratamento de erros é transversal a todos os projetos que utilizam a solução *ebankIT*, não foi possível adicionar a lista de erros no início da página como visto na secção 4.3.4 sobre a validação de erros nos formulários, pois altera toda a lógica implementada durante todo o tempo de vida da solução. Ficando esta sugestão para implementações de novas versões.

Os erros de inserção de dados na solução *ebankIT* são apresentados abaixo do elemento com uma borda vermelha e com a mensagem de erro abaixo, também com cor vermelho, como apresentado na Figura 21.

Enter your username to log in

Username

Enter your username

Figura 21 - Exemplo de erro de entrada na solução *ebankIT*.

O tratamento de erros dinâmicos, isto é, os que são verificados por scripts estão a ser apresentados abaixo do componente, por exemplo, de uma caixa de texto, colocando uma borda vermelha à volta do campo. A `<label>` inserida dinamicamente com a descrição do erro vai passar a ter o atributo `role="alert"` e o campo `<input>` inválido a ter o atributo `aria-invalid="true"`, como apresentado na Figura 22.

```
<input name="name" type="text" placeholder="Username" aria-label="Username" class="has-error" aria-invalid="true" aria-  
errormessage="MainContentFull_ebLoginControl_txtUserName_txField_error" id="id">  
<label class="has-error" role="alert" id="errorid">Enter your username</label>
```

Figura 22 - Exemplo de tratamento de erros na solução *ebankIT*.

6 Avaliação

O capítulo de avaliação tem como objetivo descrever o processo de avaliação de resultados deste desenvolvimento para apurar o nível de acessibilidade atingido.

Apenas vai ser possível avaliar tendo em conta o interesse da organização e equipa de desenvolvimento, pois neste projeto o cliente final é a própria organização.

6.1 Abordagem

O interesse da organização é perceber se a adaptação da acessibilidade à solução base já existente foi conseguida e efetuada com sucesso, dentro do prazo e objetivos estipulados. Com o objetivo de verificar se foram realizados com sucesso foram realizados os seguintes procedimentos:

- A alocação da equipa de testes para verificação dos desenvolvimentos e impactos na solução como um todo e a sua aprovação.
- Utilização de diferentes ferramentas de avaliação da acessibilidade nas páginas/componentes onde foram feitas alterações.
- Realização de testes de usabilidade da aplicação, como testar com leitor de ecrã e navegação com teclado, a fim de perceber se a aplicação cumpre os requisitos. Para isso foi realizado um guião para que a equipa de testes consiga seguir e verificar se as etapas cumprem os requisitos.

6.2 Identificação de indicadores

Para a avaliação da implementação realizada é necessário identificar os indicadores para perceber o resultado deste projeto. A integração da acessibilidade na solução *ebankIT* enquadra-se na implementação de:

- Navegação por teclado, onde é expectável que um utilizador consiga navegar e interagir totalmente com a aplicação através do teclado;
- Responsividade da página com zoom, para que o utilizador consiga visualizar o conteúdo e navegar com o zoom que se adapta melhor sem perda de informação;
- Utilização básica do leitor de ecrã, onde é expectável o leitor de ecrã produzir as informações apresentadas ao utilizador;
- Correção na estrutura da página, como os cabeçalhos, pois muitos leitores de ecrã navegam por elementos como os cabeçalhos ou *links*;
- Correção dos contrastes e estilos de diferenciação de estado do elemento, de forma a tornar a aplicação adaptável a pessoas com incapacidades visuais e cognitivas.

Os três primeiros pontos descritos acima são critérios avaliados através de testes manuais, não sendo possível a utilização de ferramentas de avaliação. Assim sendo, a correção da estrutura da página e a correção de contrastes é realizada por testes automatizados pelas ferramentas.

Para além dos pontos descritos acima, também é objetivo do projeto corrigir todos os erros devolvidos pelas ferramentas de avaliação, que por sua vez estão ligados aos pontos acima, como o facto de um elemento não possuir uma label vai afetar a utilização do leitor de ecrã.

Por fim, foram utilizadas as três ferramentas de forma a consolidar a análise de resultados, sendo que cada ferramenta possui as suas vantagens e desvantagens de utilização (secção 2.2.1).

6.3 Metodologias de avaliação

As experiências realizadas para avaliar os desenvolvimentos foram:

- Elaboração de critérios de aceitação para os testes das páginas e componentes.
- Utilização de ferramentas de avaliação para verificar acessibilidade e melhorias.

6.3.1 Critérios de avaliação

Para melhorar a interação e facilitar por parte da restante equipa de desenvolvimento e da equipa de testes, foi disponibilizado ao longo do processo os critérios de avaliação com as instruções sobre pontos a ter em conta durante o desenvolvimento e testes no uso da aplicação para páginas e componentes específicos.

Estes critérios foram realizados depois do estudo das diretrizes de acessibilidade e do levantamento de soluções para adaptar os componentes das páginas à acessibilidade. Este processo teve algumas limitações, como o tempo estipulado para os desenvolvimentos e o facto de serem realizados sobre uma solução com algum tempo e utilizada em diferentes projetos já a decorrer no cliente final.

Por isso, foram descritas nos critérios a melhor solução adaptada a cada página da solução ebankIT e a que melhor cumpre os requisitos da acessibilidade. Estes critérios têm pontos em comum e que alteram o conteúdo dependendo da página, e são:

- A ordem da navegação por teclado e como deve ser feita a interação com os componentes;
- As alterações de estilos quando o elemento está focado;
- O contraste de cores;
- Os textos alternativos para as imagens presentes na página;
- E uma chamada à atenção para o ajustamento do ecrã até 200% e não perder qualquer informação.

6.3.2 Ferramentas de avaliação

As ferramentas utilizadas, como referido na secção 2.2.1, foram *Lighthouse*, *Accessibility Insights for Web* e *WAVE*. Os relatórios gerados apresentam diferentes informações e recomendações.

O *Lighthouse* apresenta uma percentagem que corresponde a quantos pontos da auditoria passou com sucesso. Ainda apresenta os pontos aplicáveis à auditoria, os que não se aplicam, uma lista de recomendações para testes manuais e os pontos que falharam na auditoria e sugestões de correção. Esta percentagem foi útil para quantificar o nível de acessibilidade, apesar de não ser possível pesar com os critérios avaliados por testes manuais.

A ferramenta de auditoria *Accessibility Insights for Web* devolve os erros e dentro de cada erro indica o elemento e como corrigir o erro. Ainda, é indicado que existem muitos problemas de acessibilidade que são apenas detetáveis por testes manuais. Esta

ferramenta foi útil pois apresenta uma opção onde é possível visualizar o percurso da navegação por teclado através da página que apoia nos testes manuais e ainda, é possível correr esta ferramenta em qualquer página da aplicação que não vai ser executado o carregamento da página, podendo perder conteúdo a avaliar.

Outra ferramenta utilizada para auditoria de acessibilidade na aplicação *web* é a WAVE, esta ferramenta disponibiliza um quadro do lado esquerdo do ecrã e sinaliza no ecrã os elementos que possuem um erro, uma melhoria e também sinaliza os *headings* para o utilizador da ferramenta perceber a estrutura da página, sendo uma vantagem durante a implementação de acessibilidade e na avaliação. Para além do sumário de erros, alertas, entre outros, também é possível ver em detalhe. É apresentado o que o erro significa, qual a sua importância, como corrigir e a diretriz WCAG associada.

6.4 Avaliação de resultados

Para a avaliação de resultados foi utilizado uma plataforma de controlo de desenvolvimento, operações, engenharia de qualidade e segurança, designada por *Azure DevOps*, onde foram registados todos os critérios de aceitação por página ou componente e quando lançada as alterações descritas nos critérios, a equipa de testes executa os testes manuais e utilizam as ferramentas de auditoria e se passarem pelos critérios com sucesso, é alterado o estado da tarefa e as alterações são incluídas na solução base.

Os resultados obtidos, antes das alterações realizadas, nas páginas alteradas através da utilização da ferramenta *lighthouse* estão apresentados na Tabela 7, onde é descrito os erros obtidos pela ferramenta e a percentagem gerada. Não esquecer que existem muitas alterações realizadas que não são detetáveis por uma ferramenta e apenas por testes manuais.

Tabela 7 - Resultados de páginas antes da adaptação de acessibilidade através do *lighthouse*

Página	Resultados	Percentagem de acessibilidade
Login	Contraste entre o fundo e o primeiro plano	68
	O elemento <code><html></code> não tem o atributo <code><lang></code>	
	Os elementos do formulário não têm <i>labels</i> associados	
Homepage	Contraste entre o fundo e o primeiro plano	47
	<i>Headings</i> não estão sequencialmente corretos	
	Textos alternativos para imagens	
	Links com um nome discernível	

Mensagens	Contraste entre o fundo e o primeiro plano	61
	O elemento <code><html></code> não tem o atributo <code><lang></code>	
	Textos alternativos para imagens	
	Links com um nome discernível	
Meus produtos	Contraste entre o fundo e o primeiro plano	51
	O elemento <code><html></code> não tem o atributo <code><lang></code>	
	Textos alternativos para imagens	
	Links com um nome discernível	
Detalhes de conta	Contraste entre o fundo e o primeiro plano	56
	O elemento <code><html></code> não tem o atributo <code><lang></code>	
	Textos alternativos para imagens	
	Links com um nome discernível	
	Elemento <code><frame></code> ou <code><iframe></code> não possui um título	
Transferências	Contraste entre o fundo e o primeiro plano	66
	O elemento <code><html></code> não tem o atributo <code><lang></code>	
	Textos alternativos para imagens	
	Links com um nome discernível	

Depois das alterações realizadas, foi possível obter 100% de acessibilidade pela ferramenta *lighthouse* nos critérios avaliados, como se pode observar na Tabela 8, onde é apresentada a ferramenta utilizada para o critério a ser avaliado e o resultado no desenvolvimento efetuado.

Tabela 8 - Critérios avaliados pelas ferramentas

Ferramenta	Critério	Porcentagem
Lighthouse/ Accessibility Insights for Web/WAVE	Contraste entre o fundo e o primeiro plano	100
	Textos alternativos para imagens	100
	Cabeçalhos sequencialmente corretos	100
	Links com um nome discernível	100
	Elementos com nomes acessíveis, por exemplo botões	100
	Existência do atributo <code>lang</code> com um valor válido	100
	Id únicos na página	100
	Elementos com <code>tabindex</code> maior que zero	100

Tendo em conta que algumas das alterações feitas são transversais, como a alteração de cores, a toda a aplicação, pode-se aferir que as páginas não adaptadas a acessibilidade,

apesar de não sofrerem diretamente alterações e testes foram melhoradas a nível do contraste. Os outros pontos não foram desenvolvidos, como os textos alternativos.

Os critérios realizáveis por testes manuais estão apresentados na Tabela 9, onde é apresentado o critério e se a sua implementação e avaliação foi realizada. É de notar que, o projeto nunca foi pensado para ser navegável pelo teclado, nem a utilização de leitor de ecrã, por isso foram realizadas todas as adaptações necessárias para que fosse possível atingir este objetivo.

Tabela 9 - Critérios avaliados por testes manuais

	Critério	Implementado/ Avaliado
Teste manual	Ordem de navegação por teclado	Sim
	Focos nos controlos interativos	Sim
	Elementos interativos indicam o seu estado e propósito	-
	Controlos customizados possuem rótulos associados	Sim
	Controlos customizados possuem papéis ARIA	-
	Elementos de HTML5 são utilizados para melhorar a navegação	Não
	Utilização de zoom sem perda de informação	Sim

Os critérios de ordem da navegação por teclado, o foco nos controlos interativos, controlos customizados possuem rótulos associados e a utilização de zoom em cada página sem perda de informação foram validados e cumpridos em todas as páginas adaptadas seguindo as diretrizes de acessibilidade.

Não foi possível concretizar a implementação de elementos HTML5, sendo uma recomendação da ferramenta Lighthouse, para melhorar a navegação, pois a versão da solução *ebankIT* onde foi feita a implementação não possui esta versão e não faz parte dos requisitos para este projeto.

Os critérios de elementos interativos indicam o seu estado e o seu propósito e os controlos customizados possuem papéis ARIA foram implementados em alguns componentes/páginas, pois nesta fase de desenvolvimento foi tomado em consideração a implementação para uma utilização básica de leitor de ecrã. Sendo estes dois critérios relacionados com a melhoria da navegação de utilizadores com leitor de ecrã.

7 Conclusão

Neste capítulo são apresentadas as conclusões de todo o trabalho desenvolvido ao longo deste projeto, os resultados obtidos e o trabalho futuro.

7.1 Resultados

O principal objetivo deste projeto foi o estudo e a adaptação do produto internet banking da ebankIT aos requisitos da acessibilidade na web, cumprindo as diretrizes de acessibilidade.

Esta implementação focou-se na navegação por teclado, a correção de contrastes, a utilização do zoom e do leitor de ecrã de algumas páginas que vão servir como exemplo para implementação futura de outras páginas idênticas.

Posto isto, o trabalho desenvolvido foi efetuado com sucesso mas existindo a consciência que a implementação de acessibilidade numa aplicação web é um trabalho que tem de estar em constante melhoria, pois é difícil conseguir resolver todas as barreiras de todos os utilizadores. Sendo que a implementação para utilização de leitor de ecrã ficou aquém por falta de conhecimento prático por parte de utilizadores desta tecnologia assistiva e além disso por falta de possibilidade de manuseamento da solução para que a estrutura fosse de encontro ao esperado pela utilização de um leitor de ecrã.

Para além disto, foi feita uma sessão de sensibilização e passagem de algum conhecimento a todas as equipas de testes focadas em outras partes da solução ebankIT para terem noções do trabalho realizado e do tema acessibilidade.

7.2 Limitações

O primeiro caso de estudo sugerido assentava sobre adaptar a plataforma internet banking para um cliente no Reino Unido, tendo falhado por questões de tempo e falta de feedback do lado do cliente. Tendo surgido a oportunidade de trabalhar diretamente com a solução foi proposto a troca do caso de estudo.

Apesar da troca de caso de estudo, este desenvolvimento também teve algumas limitações por questão de tempo e de possibilidade de alterações, pois a solução ebankIT é utilizada nos projetos a decorrer na empresa e a versão desenvolvida vai ser incluída em alguns deles. Para que fosse possível implementar algumas das soluções, era necessário um maior número de testes em toda a solução ou alterar completamente a lógica.

Sabendo que esta implementação vai ser incluída nos projetos a decorrer foi necessário gerir o impacto da alteração, o tempo para o seu desenvolvimento e testes e os erros que poderia mais tarde gerar noutros projetos. Por isso, foi feito o possível para atingir a acessibilidade nesta solução.

Outra limitação encontrada foi a impossibilidade de realizar inquéritos de satisfação pois os desenvolvimentos ainda não estão disponíveis para a inclusão em projetos a decorrer e não foi possível testar com utilizadores que possuem incapacidades para verificar verdadeiramente a usabilidade da aplicação, por exemplo na utilização de um leitor de ecrã.

7.3 Trabalho futuro

Para trabalho futuro, também para novas versões que surjam, seguem algumas indicações:

- Constante melhoria da solução, pois é difícil conseguir resolver todas as barreiras de todos os utilizadores;
- Utilização de linguagens e tecnologias mais recentes que já possuem funcionalidades integradas que facilitam a implementação da acessibilidade, como por exemplo: HTML5 fornece uma gama de funcionalidades embutidas para ajudar a validar entradas como o email;
- Adicionar um link no footer para redirecionar para uma página para as declarações de acessibilidade.

Mesmo não sendo atingido o maior nível de acessibilidade, este trabalho foi um impulsionador para que nas próximas versões seja cada vez mais tido em atenção a acessibilidade e que seja uma mais valia a nível de marketing para angariação de novos projetos.

Referências

Abati, A. (2019). *A practical guide to accessibility for forms—LogRocket Blog*. <https://blog.logrocket.com/a-practical-guide-to-accessibility-for-forms/>

Accessibility Insights. (2020). *Accessibility Insights for Web · Accessibility Insights*. <https://accessibilityinsights.io/>

Andrews, K. (2017). *A Quick Guide to Screen Readers*. 4.

Apple. (2020). *Vision Accessibility—Mac*. Apple. <https://www.apple.com/accessibility/mac/vision/>

Assistive technology. (2020). <https://www.who.int/westernpacific/health-topics/assistive-technology>

Avila, J. (2014). *How to Provide Accessible Form Error Identification—Level Access*. <https://www.levelaccess.com/how-to-provide-accessible-error-identification/>

Baptista, R. (2019). *Apenas três sites da administração pública são acessíveis a deficientes*. PÚBLICO. <https://www.publico.pt/2019/01/02/sociedade/noticia/apenas-tres-sites-administracao-publica-sao-acessiveis-deficientes-1856483>

Black, K. (2020). *Designing for Accessibility: Focus States*. DockYard. <https://dockyard.com/blog/2020/04/28/designing-for-accessibility-focus-states>

BPI. (2020). *Site BPI*. Banco BPI. <https://www.bancobpi.pt/particulares>

COUNCIL, E. P. (2015). DIRECTIVE OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the approximation of the laws, regulations and administrative provisions of the Member States as regards the accessibility requirements for products and services. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52015PC0615>

Decreto-Lei 83/2018. (2018). Diário da República Eletrónico. <https://dre.pt/home/-/dre/116734769/details/maximized>

Dongaonkar, S., Vadali, R., & Dhutadmal, C. (2017). Content Accessibility Evaluation of Government Website using WCAG(Web Content Accessibility Guidelines). 10, 27–31. <https://doi.org/10.15439/2017R79>

ebankIT. (2018a). About Us | ebankIT Omnichannel Digital Banking Solutions. <https://www.ebankit.com/about-us>

ebankIT. (2018b). The ebankIT Digital Platform | Omnichannel Banking Solution. <https://www.ebankit.com/platform>

EUROPEAN COMMISSION. (2008). United Nations Convention on the Rights of Persons with Disabilities—Employment, Social Affairs & Inclusion—European Commission. <https://ec.europa.eu/social/main.jsp?catId=1138&langId=en>

EUROPEAN PARLIAMENT AND OF THE COUNCIL. (2016). DIRECTIVE (EU) 2016/2102 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 26 October 2016 on the accessibility of the websites and mobile applications of public sector bodies. <https://eur-lex.europa.eu/legal-content/PT/TXT/?uri=OJ:L:2016:327:TOC>

Ferraz, R. (2016). *Dicas para verificar a acessibilidade da sua página Web*. Medium. <https://medium.com/revista-web/dicas-para-verificar-a-acessibilidade-da-sua-p%C3%A1gina-web-186acdfb865c>

Ferri, D., & Favalli, S. (2018). Web Accessibility for People with Disabilities in the European Union: Paving the Road to Social Inclusion. *Societies*, 8(2), 40. <https://doi.org/10.3390/soc8020040>

Freedom Scientific. (2020). *JAWS®* – Freedom Scientific. <https://www.freedomscientific.com/products/software/jaws/>

Google Developers. (2020a). *How To Do an Accessibility Review | Web Fundamentals*. Google Developers. <https://developers.google.com/web/fundamentals/accessibility/how-to-review>

Google Developers. (2020b). *Lighthouse | Tools for Web Developers*. Google Developers. <https://developers.google.com/web/tools/lighthouse>

GOV.UK. (2019). *Understanding accessibility requirements for public sector bodies*. GOV.UK. <https://www.gov.uk/guidance/accessibility-requirements-for-public-sector-websites-and-apps>

INEM. (2017). *Acessibilidade*. INEM. <https://www.inem.pt/2019/12/23/acessibilidade/>

js.foundation, J. F.-. (2020). *JQuery*. <https://jquery.com/>

King, M., Jemma Ku, J., Nurthen, J., Bijl, Z., & Cooper, M. (2019). *WAI-ARIA Authoring Practices 1.1*. <https://www.w3.org/TR/wai-aria-practices/>

Koen, P. et al. (2001). Providing Clarity and A Common Language to the “Fuzzy Front End”. *Research-Technology Management*, 46–55.

Leitner, M.-L., Strauss, C., & Stummer, C. (2016). Web accessibility implementation in private sector organizations: Motivations and business impact. *Universal Access in the Information Society*, 15(2), 249–260. <https://doi.org/10.1007/s10209-014-0380-1>

Less. (2020). *Getting started | Less.js*. <http://lesscss.org/>

Lorensson, C. (2017, Março 22). *Designing Usable and Accessible Buttons*. Medium. <https://blog.prototypr.io/designing-usable-and-accessible-buttons-dffb464d9be2>

MacDonald, david. (2020). *Alternate text for background images, alt text accessibility*. <http://www.davidmacd.com/blog/alternate-text-for-css-background-images.html>

MacDonald, D. (2020). Is placeholder text a sufficient accessible label for form fields—David MacDonald Web Accessibility Blog, testing, teaching, and discussion. <http://www.davidmacd.com/blog/is-placeholder-accessible-label.html>

MDN web docs. (2019). *ARIA: button role—Accessibility | MDN*. https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles/button_role

MDN Web Docs. (2020a). *<select>: The HTML Select element*. MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/select>

MDN Web Docs. (2020b). *ARIA*. MDN Web Docs. <https://developer.mozilla.org/pt-BR/docs/Web/Accessibility/ARIA>

MDN Web Docs. (2020c). *HTML basics*. MDN Web Docs. https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics

MDN Web Docs. (2020d). *Tabindex*. MDN Web Docs. https://developer.mozilla.org/pt-BR/docs/Web/HTML/Global_attributes/tabindex

MDN Web Docs. (2020e). *What is accessibility? - Learn web development | MDN*. https://developer.mozilla.org/en-US/docs/Learn/Accessibility/What_is_accessibility

Milleniumbcp. (2020). *Particulares—Millenniumbcp*. <https://ind.millenniumbcp.pt/pt/particulares/Pages/Welcome.aspx>

Mitchell, S. (2008). *Creating a Site-Wide Layout Using Master Pages (C#)*. <https://docs.microsoft.com/en-us/aspnet/web-forms/overview/older-versions-getting-started/master-pages/creating-a-site-wide-layout-using-master-pages-cs>

Noieto, C. (2020, Março 18). *Aplicações web: Entenda o que são e como funcionam!* Blog da Trybe. <https://blog.betrybe.com/desenvolvimento-web/aplicacoes-web/>

NVDA. (2017, Julho 5). About NVDA. *NV Access*. <https://www.nvaccess.org/about-nvda/>

OPTASY. (2019). *Writing HTML Code for Screen Readers: 6 Best Practices*. Medium. <https://medium.com/@OPTASY.com/writing-html-code-for-screen-readers-6-best-practices-bf8f2248318>

Otto, M., Thornton, J., & Bootstrap. (2020). *Acessibilidade*. <http://getbootstrap.com.br/docs/4.1/getting-started/accessibility/>

SAPO. (2020). *Regras de Usabilidade*. SAPO UX. <https://ux.sapo.pt/usabilidade/>

STUDIO. (2020). *Forms | Accessibility Guidelines*. <http://web-accessibility.carnegiemuseums.org/code/forms/>

W3C. (2007). *RadioButton—W3C Wiki*. <https://www.w3.org/wiki/RadioButton>

W3C. (2014). *Using aria-label for link purpose—WCAG WG*. https://www.w3.org/WAI/GL/wiki/Using_aria-label_for_link_purpose

W3C. (2019a). *Carousel Concepts • Carousels • WAI Web Accessibility Tutorials*. <https://www.w3.org/WAI/tutorials/carousels/>

W3C. (2019b). *Form Instructions • Forms • WAI Web Accessibility Tutorials*. <https://www.w3.org/WAI/tutorials/forms/instructions/>

W3C. (2019c). *Forms Concepts • Forms • WAI Web Accessibility Tutorials*. <https://www.w3.org/WAI/tutorials/forms/>

W3C. (2019d). *Making the Web Accessible*. W3C. <https://www.w3.org/WAI/>

W3C. (2020a). *ARIA Radio Group and Radio Example*. <https://www.w3.org/TR/2016/WD-wai-aria-practices-1.1-20160317/examples/radio/radio.html>

W3C. (2020b). *Checkbox Example (Two State) | WAI-ARIA Authoring Practices 1.1*. <https://www.w3.org/TR/wai-aria-practices-1.1/examples/checkbox/checkbox-1/checkbox-1.html>

W3C. (2020c). *Grouping Controls • Forms • WAI Web Accessibility Tutorials*. <https://www.w3.org/WAI/tutorials/forms/grouping/>

w3schools. (2020). *HTML Accessibility*. https://www.w3schools.com/html/html_accessibility.asp

WAVE. (2020). *WAVE Help*. <https://wave.webaim.org/help>

WebAIM. (2013). *WebAIM: «Skip Navigation» Links*. <https://webaim.org/techniques/skipnav/>

WebAIM. (2018). *WebAIM: Keyboard Accessibility*.
<https://webaim.org/techniques/keyboard/>

WebAIM. (2019). *WebAIM: Screen Reader User Survey #8 Results*.
<https://webaim.org/projects/screenreadersurvey8/>

WebAIM. (2020a). *WebAIM: Alternative Text*. <https://webaim.org/techniques/alttext/>

WebAIM. (2020b). *WebAIM: Creating Accessible Forms—Accessible Form Controls*.
<https://webaim.org/techniques/forms/controls>

WebAIM. (2020c). *WebAIM: Designing for Screen Reader Compatibility*.
<https://webaim.org/techniques/screenreader/>

Wheeler, K. (2020). *Kenwheeler/slick* [JavaScript]. <https://github.com/kenwheeler/slick>
(Original work published 2014)

Yale. (2020). *Headings | Usability & Web Accessibility*. <https://usability.yale.edu/web-accessibility/articles/headings>

Yale University. (2020). *Images | Usability & Web Accessibility*.
<https://usability.yale.edu/web-accessibility/articles/images>