

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UM *FRAMEWORK* PARA APOIAR O
DESENVOLVIMENTO DE APLICAÇÕES
ON-LINE ACESSÍVEIS**

PABLO BIZZI MAHMUD

ORIENTADOR: PROF. DR. DELANO MEDEIROS BEDER

São Carlos – SP
Dezembro-2016

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UM *FRAMEWORK* PARA APOIAR O
DESENVOLVIMENTO DE APLICAÇÕES
ON-LINE ACESSÍVEIS**

PABLO BIZZI MAHMUD

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Engenharia de Software

Orientador: Prof. Dr. Delano Medeiros Beder

São Carlos – SP

Dezembro-2016

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária UFSCar
Processamento Técnico
com os dados fornecidos pelo(a) autor(a)

M215f Mahmud, Pablo Bizzi
Um framework para apoiar o desenvolvimento de
aplicações on-line acessíveis / Pablo Bizzi Mahmud. --
São Carlos : UFSCar, 2016.
110 p.

Dissertação (Mestrado) -- Universidade Federal de
São Carlos, 2016.

1. Acessibilidade. 2. On-line. 3. Framework. 4.
Web. I. Título.



UNIVERSIDADE FEDERAL DE SÃO CARLOS
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a defesa de Dissertação de Mestrado do candidato Pablo Bizzi Mahmud, realizada em 21/10/2016.

Prof. Dr. Delano Mezeiros Beder
(UFSCar)

Prof. Dr. Renata Pontin de Mattos Fortes
(USP)

Prof. Dr. Giliane Bernardi
(UFSCar)

Certifico que a sessão de defesa foi realizada com a participação à distância do membro Prof. Dr. Giliane Bernardi. Depois das arguições e deliberações realizadas, o participante à distância está de acordo com o conteúdo do parecer da comissão examinadora redigido no relatório de defesa do aluno Pablo Bizzi Mahmud.

Prof. Dr. Delano Mezeiros Beder
Coordenador da Comissão Examinadora
(UFSCar)

Dedico esta dissertação e o título de Mestre aos meus pais, Karim e Sandra, aos meus familiares, e em especial à minha noiva Giana Lauda, pelo apoio e incentivo em todos os momentos desta jornada.

AGRADECIMENTOS

À Universidade Federal de São Carlos, por ser a instituição que deu toda a estrutura necessária para que eu pudesse trilhar meu caminho rumo ao sonho de obter o título de Mestre.

À Universidade Federal de Santa Maria, que abriu as portas da instituição para que o estudo de caso pudesse ser aplicado com seus alunos.

Ao meu orientador, Delano, por toda a atenção e dedicação depositada em nosso trabalho. Sabíamos desde o início que seria um grande desafio trabalhar com acessibilidade, mas mesmo assim sabíamos que poderíamos produzir um trabalho digno para essa linda área de pesquisa.

À minha mãe, Sandra, que me apoiava a distância e despendia de horas no telefone comigo, sempre me dando orientações e me motivando para que eu concluísse esse objetivo. Mostrando que amor de mãe pode ser sentido de qualquer distância.

Ao meu pai, Karim, que desde o primeiro minuto me apoiou com tudo que foi necessário para a conclusão desta etapa. Lembro como fosse hoje que botamos todas as minhas coisas no carro e viajamos mais de 1500km para ir em busca desse sonho, sendo muito mais do que apenas um parceiro de viagem. Sem falar que quando a saudade de um churrasco estava grande, foi me visitar para fazermos um costelão de 8h.

À minha família que sempre me apoiou em todas as minhas decisões, que de alguma forma me passou algum ensinamento e me ajudou em tempos difíceis, e que hoje está comigo em todos os bons momentos.

Não menos importante, agradeço particularmente à minha noiva, Giana, por ser tão importante na minha vida e ser minha companheira que me poe para cima e me faz acreditar que posso mais que imagino. Devido a seu companheirismo, amizade, paciência, compreensão, apoio, alegria e, principalmente, amor, este trabalho pôde ser concretizado. Obrigado por me aguentar nos momentos difíceis e por ter feito do meu sonho o nosso sonho. Espero poder contar sempre com esse apoio e, como dizem por aí, "ao LADO de um grande homem, sempre há uma grande mulher".

Aos meus amigos e colegas de São Carlos que sempre me acolheram muito bem, levo como grandes amigos para o resto da vida, em especial Fernando Chagas, Arthur Andrade, Rafael Durelli, Bruno Perin, Henrique Velloso e Alex Born.

Às vezes você faz escolhas na vida e às vezes as escolhas fazem você.

Gayle Forman

RESUMO

Com o crescimento dos serviços que hoje são disponibilizados *on-line* e com a diversidade do público que utiliza esses serviços, a *web* começa a ganhar destaque como mecanismo capaz de quebrar barreiras e integrar usuários, conectando pessoas das mais distintas classes econômicas e pessoas com ou sem deficiência. Com isso em mente, devemos repensar nossas diretrizes de desenvolvimento de sistemas, de modo que possam ser acessados pelo maior número de pessoas possível e alcancemos a universalização da *web*. Auxiliando desenvolvedores a confeccionar e prover aplicações *on-line* acessíveis, este trabalho propõe o desenvolvimento de um *framework* que serve como base de desenvolvimento para elas, independentemente do nível de conhecimento dos mesmos. Considerando a existência atual de organizações que padronizam o desenvolvimento de aplicações acessíveis para pessoas com deficiência, este trabalho visa a contemplar todas essas recomendações em uma ferramenta para abstrair todo o trabalho de estudo de diretrizes por parte do programador, focando-o em criar aplicações acessíveis com esforço mínimo. Ainda, este trabalho apresenta a coleta e a validação de diretrizes existentes no que se refere ao desenvolvimento de aplicações acessíveis na plataforma *web* e suas respectivas aplicações no *framework*, já que as diretrizes atualmente disponíveis possuem caráter abrangente no desenvolvimento de jogos acessíveis. A ferramenta, denominada ARIA-ACCESS, atesta em seu estudo de caso que é de fácil manuseio, até mesmo para desenvolvedores iniciantes, e apresenta ganho de produtividade quanto ao tempo de desenvolvimento, resultado da geração automática de código WAI-ARIA.

Palavras-chave: acessibilidade, *on-line*, *framework*, *web*

ABSTRACT

Through the increase of services that are now available online and with the diversity of the public that uses these services, the web begins to gain prominence as a mechanism capable of breaking down barriers and integrating users, connecting people of the most different economic classes and people with or without disabilities. With that in mind, we must rethink systems' development guidelines so that they can be accessed by as many people as possible and reach the universal web. Helping developers make and provide accessible online applications, this work aims to develop a framework that serves as a basis for development, regardless of their skill or knowledge. Considering the current existence of organizations that standardize the development of accessible applications for people with disabilities, this work aims to cover all these recommendations in a tool to abstract all the programmer's work review, creating accessible applications with minimal effort. In addition, this work presents the collection and validation of existing guidelines regarding the development of accessible applications in the web platform and their respective applications in the framework, as currently available guidelines are comprehensive in the development of accessible games. The tool, called ARIA-ACCESS, attests in its case study that it is easy to handle, even for beginner developers, and it presents a productivity gain regarding the development time, a result of the automatic generation of WAI-ARIA code.

Keywords: accessibility, online, framework, web

LISTA DE FIGURAS

2.1	Exemplo de código de um artigo acessível.	25
2.2	Exemplo de código de um menu acessível.	25
2.3	Exemplo de código de uma informação complementar acessível.	25
2.4	Exemplo de código de uma informação complementar acessível.	26
2.5	Exemplo de código de uma informação complementar acessível.	27
2.6	Exemplo da mudança no desenvolvimento com as novas <i>tags</i>	37
3.1	Gráfico de equivalência de diretrizes (CHEIRAN; PIMENTA, 2013).	53
4.1	Arquitetura proposta do <i>framework</i>	61
4.2	Código HTML de texto, botão, <i>link</i> e menu com classes que referenciam o ARIA-ACCESS.	64
4.3	Código gerado de texto, botão, <i>link</i> e menu pelas dependências do ARIA-ACCESS.	64
4.4	Código HTML de imagem e formulário com classes que referenciam o ARIA-ACCESS.	65
4.5	Código gerado de imagem e formulário pelas dependências do ARIA-ACCESS.	65
4.6	Código HTML de tabela com classes que referenciam o ARIA-ACCESS.	66
4.7	Código gerado de tabela pelas dependências do ARIA-ACCESS.	67
5.1	Divisão de grupos para o estudo de caso deste projeto.	87
5.2	Gráfico de colunas que representa o nível de conhecimento de desenvolvimento <i>web</i> dos participantes.	89
5.3	Gráfico de colunas que representa o nível de conhecimento de práticas de desenvolvimento acessível dos participantes.	90

5.4	Gráfico de colunas que representa o nível de facilidade na incorporação do <i>framework</i> no projeto por parte dos participantes.	90
5.5	Gráfico de pizza que mostra a porcentagem de participantes que conseguiram tornar textos e cabeçalhos acessíveis.	91
5.6	Gráfico de pizza que mostra a porcentagem de participantes que conseguiram tornar botões acessíveis.	92
5.7	Gráfico de pizza que mostra a porcentagem de participantes que conseguiram tornar <i>links</i> e menus acessíveis.	93
5.8	Gráfico de pizza que mostra a porcentagem de participantes que conseguiram tornar tabelas acessíveis.	94
5.9	Gráfico de pizza que mostra a porcentagem de participantes que conseguiram tornar imagens acessíveis.	95
5.10	Gráfico de pizza que retrata a porcentagem de participantes que conseguiram tornar formulários acessíveis.	96
5.11	Gráfico de colunas que apresenta a qualidade da documentação do ARIA-ACCESS.	96
5.12	Gráfico de colunas que apresenta a facilidade de uso do ARIA-ACCESS.	97
5.13	Gráfico de colunas que apresenta o impacto no desenvolvimento de aplicações acessíveis do ARIA-ACCESS.	97
5.14	Gráfico de colunas que apresenta a relevância para a sociedade de aplicações que usam o ARIA-ACCESS.	98
5.15	Tempo despendido pelos participantes no estudo do ARIA-ACCESS.	99

LISTA DE TABELAS

2.1	Diretrizes WCAG que se referem ao Princípio 1	30
2.2	Diretrizes WCAG que se referem ao Princípio 2	31
2.3	Diretrizes WCAG que se referem ao Princípio 3	31
2.4	Diretriz WCAG que se refere ao Princípio 4	31
2.5	Resumo das WCAG na versão 1.0 e 2.0	33
3.1	Diretrizes básicas das <i>Game accessibility guidelines</i>	42
3.2	Diretrizes intermediárias das <i>Game accessibility guidelines</i>	43
3.3	Diretrizes avançadas das <i>Game accessibility guidelines</i>	44
3.4	Diretrizes propostas pela IGDA (2012)	46
3.5	Diretrizes de nível 1 de Barlet e Taylor (2012).	50
3.6	Diretrizes de nível 2 de Barlet e Taylor (2012)	51
3.7	Diretrizes de nível 3 de Barlet e Taylor (2012)	51
4.1	Diretrizes selecionadas - <i>Game accessibility guidelines</i>	68
4.2	Diretrizes selecionadas - IGDA	74
4.3	Diretrizes selecionadas - <i>AbleGamers</i>	79

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	16
1.1 Motivação e justificativa	19
1.2 Objetivos	20
1.3 Organização do documento	21
CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA	22
2.1 Tipos de deficiência	22
2.1.1 Cegueira	23
2.1.2 Daltonismo	23
2.1.3 Deficiência motora	23
2.2 WAI-ARIA	24
2.2.1 <i>Roles</i>	25
2.2.1.1 <i>Role document structure</i>	25
2.2.1.2 <i>Role widget</i>	26
2.2.1.3 <i>Role landmarks</i>	26
2.2.2 <i>Properties</i>	26
2.2.3 Como o HTML5 se relaciona com o WAI-ARIA	27
2.2.4 Como funcionam as tecnologias de apoio	27
2.2.5 Interação com o teclado	28
2.3 Diretrizes de acessibilidade ao conteúdo da Web (WCAG)	29

2.3.1	Direcionamento e níveis de abordagem das WCAG	29
2.3.2	Princípios e diretrizes	30
2.3.2.1	Princípio 1 – Perceptível	30
2.3.2.2	Princípio 2 – Operável	30
2.3.2.3	Princípio 3 – Compreensível	31
2.3.2.4	Princípio 4 – Robusto	31
2.3.3	Formato dos documentos	31
2.3.4	Evolução das WCAG	32
2.4	eMAG	32
2.5	HTML5	36
2.6	JavaScript	38
2.7	Considerações finais	38
CAPÍTULO 3 – TRABALHOS CORRELATOS		40
3.1	<i>Game accessibility guidelines</i> (GAG)	40
3.2	Diretrizes da IGDA	44
3.3	<i>A practical guide to game accessibility</i>	49
3.4	Jogos inclusivos: diretrizes de acessibilidade para jogos digitais	52
3.5	<i>Designing universally accessible games</i>	54
3.6	HATEMILE: A biblioteca para gerar páginas <i>web</i> mais acessíveis	56
3.7	Considerações finais	57
CAPÍTULO 4 – ARIA-ACCESS – FRAMEWORK DE DESENVOLVIMENTO ACES-		
SÍVEL		59
4.1	Visão geral	59
4.2	Metodologia	60
4.3	Arquitetura	60
4.4	Linguagem de domínio específico	61

4.5	Documentação	62
4.6	Exemplo de uso	63
4.6.1	Texto, Botão, <i>Link</i> e Menu	63
4.6.2	Formulários e imagens	63
4.6.3	Tabelas	65
4.7	Diretrizes selecionadas e aplicação	66
4.7.1	Diretrizes aplicáveis - <i>Game accessibility guidelines</i>	67
4.7.2	Diretrizes aplicáveis – IGDA	73
4.7.3	Diretrizes aplicáveis – <i>AbleGamers</i>	78
4.7.4	Síntese da coleta	82
4.8	Considerações finais	82
CAPÍTULO 5 – AVALIAÇÃO		83
5.1	Avaliação	83
5.1.1	Componentes do projeto de pesquisa	84
5.1.1.1	Questões de estudo	84
5.1.1.2	Proposições de estudo	84
5.1.1.3	Unidades de análise	85
5.1.1.4	Ligação dos dados à proposição e os critérios para a interpretação dos dados	85
5.1.2	Processo para pesquisa com o uso do método do estudo de caso	85
5.2	Coleta	86
5.3	Avaliação e análise dos dados	88
5.3.1	Nível de conhecimento de desenvolvimento <i>web</i>	88
5.3.2	Nível de conhecimento de <i>tags</i> WAI-ARIA	89
5.3.3	Sucesso na incorporação do <i>framework</i> no projeto	90
5.3.4	Sucesso no processo de tornar acessíveis todos os elementos propostos	91

5.3.5	Relevância da documentação	93
5.3.6	Facilidade de uso	95
5.3.7	Relevância no impacto do desenvolvimento	96
5.3.8	Relevância para a sociedade	97
5.3.9	Dados referentes ao tempo de desenvolvimento	98
5.3.10	Sugestões de melhorias	99
5.4	Considerações finais	100
CAPÍTULO 6 – CONCLUSÃO		102
6.1	Contribuições	102
6.2	Limitações	103
6.3	Trabalhos futuros	103
REFERÊNCIAS		105
GLOSSÁRIO		107
APÊNDICE A – QUESTIONÁRIO APLICADO PARA VALIDAR O ESTUDO DE CASO DO ARIA-ACCESS COM OS DESENVOLVEDORES		108
APÊNDICE B – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO		111

Capítulo 1

INTRODUÇÃO

Existem diversas definições que podemos utilizar quando falamos de acessibilidade. Como se trata de um assunto muito complexo, a acessibilidade ganhou destaque de órgãos federais e instituições de ensino. Reflexo disso é a publicação de um Decreto Federal (número 5.296/2004) cujo oitavo artigo define a acessibilidade como uma “condição para utilização, com segurança e autonomia, total ou assistida, dos espaços, mobiliários e equipamentos urbanos, das edificações, dos serviços de transporte e dos dispositivos, sistemas e meios de comunicação e informação, por pessoa portadora de deficiência ou com mobilidade reduzida” (FEDERAL, 2004).

Esta definição apresenta uma abrangência da aplicabilidade dos mecanismos de acessibilidade, aglomerando desde edificações até meios de comunicação. O Decreto destaca corretamente que a utilização deve prover segurança e autonomia. Observações acerca do Decreto podem ser feitas a fim de ajustar a definição de acessibilidade, já que o texto contempla apenas pessoas com deficiência ou com mobilidade reduzida, em vez de tratar todos os indivíduos como sociedade.

Visando solucionar problemas que tangem à abrangência do Decreto anterior, propôs-se um novo Decreto Federal publicado no dia 25 de agosto de 2009 (número 6.949), com base na Convenção Internacional sobre os Direitos das Pessoas com Deficiência¹, adotada pela Organização das Nações Unidas (ONU)² em 30 de março de 2007. O texto do Decreto estabelece em seu artigo 9 o que se segue:

“A fim de possibilitar às pessoas com deficiência viver com autonomia e participar plenamente de todos os aspectos da vida, os Estados Partes deverão tomar as medidas apropriadas para assegurar-lhes o acesso, em igualdade de oportunidades

¹ A terminologia utilizada é baseada na lei nº 13.146, promulgada no dia 6 de julho de 2015 (BRASIL, 2015)

² <http://nacoesunidas.org/>

com as demais pessoas, ao meio físico, ao transporte, à informação e comunicação, inclusive aos sistemas e tecnologias da informação e comunicação, bem como a outros serviços e instalações abertos ou propiciados ao público, tanto na zona urbana como na rural (FEDERAL, 2009).”

Aprofundando ainda mais os conceitos de acessibilidade dos Decretos Federais e adaptando-os à sua aplicação na *internet*, pode-se afirmar que desenvolver uma aplicação acessível a um usuário é dar-lhe não apenas o direito de acessar a rede de informações mundiais, mas também o direito de eliminar barreiras sociais, que dizem respeito à possibilidade de comunicação, acesso físico, acesso a equipamentos e programas adequados, acesso a conteúdos e apresentações de informações em formatos alternativos (BRASIL, 2012). Essa premissa pode ser justificada baseando-se na evolução da computação com o advento dos computadores pessoais mais robustos, que dispõem de diversos recursos computacionais de processamento de *hardware* e *softwares* e assim auxiliam os usuários independentemente de suas necessidades, sendo necessário apenas o investimento para adquirir tais equipamentos.

Em consequência do destaque e do incentivo que a acessibilidade vem recebendo nos últimos anos, identifica-se um novo cenário para desenvolver soluções positivas no que se refere à acessibilidade na *web*, mais especificamente em jogos *on-line* acessíveis. Com isso em mente, Grammenos, Savidis e Stephanidis (2009) argumentam que jogos de computador costumam ser exigentes em termos de habilidades mentais e equipamentos necessários para interagir com o sistema. Por isso, muitas vezes os jogos tornam-se intangíveis a uma grande parcela de pessoas com deficiência. Os autores expõem que, atualmente, os desenvolvedores têm dado pouca atenção ao desenvolvimento de jogos que podem ser jogados independentemente das características pessoais, requisitos ou habilidades das pessoas. Além disso, por envolverem questões de interação humano-computador, jogos de computador são fundamentalmente diferentes de todos os outros tipos de aplicações de *software*, para os quais as diretrizes de acessibilidade e soluções já estão se tornando amplamente disponíveis, revelando uma necessidade de padronização de desenvolvimento.

Essa necessidade de padronização destacada pelos autores justifica a existência de grupos focados na padronização e de recomendação de desenvolvimento da *web* acessível. Mais especificamente na área de jogos, encontramos a seguinte citação:

“Acessibilidade em jogos pode ser definida como a capacidade de jogar um jogo mesmo quando executado em condições limitantes. Condições limitantes podem ser limitações funcionais ou deficiências, tais como cegueira, surdez ou de limitações de mobilidade (IGDA, 2012).”

Visando uniformizar o desenvolvimento de aplicações para pessoas com necessidades especiais, duas organizações destinam-se a gerenciar padrões com a finalidade de tornar a *web* acessível. A W3C (Consórcio para a WEB) e a WAI (Iniciativa para a Acessibilidade na Rede), que pertence à W3C, fornecem dados que apontam situações e características diversas que o usuário de *websites* com necessidades especiais pode apresentar:

- Incapacidade ou limitação visual;
- Incapacidade ou limitação auditiva;
- Incapacidade ou limitação motora;
- Incapacidade ou limitação de interpretação de informações;
- Incapacidade ou limitação de compreender textos;
- Incapacidade ou indisponibilidade de uso do teclado ou mouse;
- Insuficiência ou limitação do navegador *web*;
- Insuficiência de acesso pela baixa velocidade da *internet*;
- Dificuldade para falar ou compreender, fluentemente, a língua em que o documento foi escrito;
- Ocupação dos olhos, ouvidos ou mãos, por exemplo, ao volante a caminho do emprego, ou no trabalho em ambiente barulhento;
- Desatualização do computador ou dos programas para navegação em *sites*.

Aliado e baseado nas especificação da WAI, justificado pelas leis citadas no início desta Seção e fundamentado na premissa de que o Governo Federal deve promover, como uma de suas principais atribuições, a inclusão social, foi desenvolvido o Modelo de Acessibilidade em Governo Eletrônico, mais conhecido como E-MAG. Modelo este que deve ser seguido por todas as instituições públicas em suas aplicações *web* (ELETRÔNICO, 2016).

1.1 Motivação e justificativa

Para justificar a importância de tornar acessíveis aplicações *web*, uma pesquisa realizada no Brasil pelo Instituto Brasileiro de Geografia e Estatística (IBGE) mostra que existem 24,5 milhões de pessoas (14,5% da população) com alguma deficiência. Destas, 48% possuem deficiência visual, 23% deficiência motora, 17% deficiência auditiva, 8% deficiência intelectual e 4% deficiência física. Deste total, 4,3 milhões (2,5% da população) têm limitações severas, podendo-se concluir que seria possível que a maioria dessas pessoas pudesse estudar, trabalhar, pesquisar na *internet*, usar serviços de bancos e outros serviços de uso básico de comunidades virtuais, se essas oportunidades as incluíssem e tivéssemos maior incentivo à acessibilidade (BRASIL, 2012).

Como complemento à pesquisa anterior, dados divulgados pelo Brasil (2016) mostram que 100 milhões de brasileiros acessam a *internet* regularmente. Tais dados motivam uma mudança de rumo em termos de desenvolvimento das nossas aplicações *web*, e tal mudança poderia impactar um número ainda maior de usuários se tivéssemos ferramentas que auxiliassem desenvolvedores a confeccionar modelos de *software web* utilizáveis por pessoas com necessidades especiais.

Considerando as pesquisas anteriores, pode-se afirmar que o papel da *internet* nas nossas vidas vem tornando-se cada vez mais fulcral, uma vez que a *web* já está presente nos mais diversos serviços do nosso dia a dia, tais como educação à distância, busca e cadastro de empregos, serviços governamentais, comércio eletrônico, serviços bancários, jogos *on-line* e muito mais. Por este motivo, destaca-se a importância de desenvolver ferramentas que possam ser utilizadas por desenvolvedores, de modo que algumas soluções comecem a prover acessibilidade a todos os seus usuários.

Como podemos observar no nosso cenário atual, os *websites* acessíveis propiciam aos seus usuários conhecimentos informativos, ajudando pessoas com deficiência a participar de comunidades virtuais e, desta forma, tornando possível que se sintam mais ativos, o que causa um grande impacto em nível pessoal e também dentro da sociedade. Assim também os jogos *on-line*, que têm como alicerce unir o mais diversificado público em torno de uma atividade, tornando irrelevante o fato de o oponente possuir deficiência ou não, contanto que consiga interagir com o meio.

Partindo do fato de que o mundo está altamente conectado, com acesso fácil e rápido a informações e pessoas, Prensky (2012) conjectura sobre o crescimento do mercado de jogos no Brasil, baseando-se em informações que motivam o investimento de ferramentas para tornar

jogos *on-line* acessíveis — uma vez que 61 milhões de brasileiros têm o costume de brincar com algum jogo eletrônico (ONLINE, 2015) — e ressaltando ainda que houve mudanças no estilo cognitivo da geração que nasceu na era das tecnologias digitais. Segundo o autor, as principais mudanças no estilo cognitivo que podem ser elencadas são: (i) as informações são processadas em uma velocidade mais alta; (ii) atividades paralelas são executadas mais facilmente; (iii) o aprendizado por gráficos é preposto com relação ao textual; (iv) o acesso aleatório é mais utilizado do que o sequencial; (v) os novos usuários são mais conectados, ativos e hábeis com novas tecnologias. Prensky (2012) menciona a necessidade de revisar as teorias e as práticas educacionais para que busquem sempre uma adequação às novas abordagens de aprendizagem, defendendo que essas práticas sejam baseadas em jogos digitais de modo que atenda as necessidades de aprendizagem para a descendência dos filhos dessa geração tecnológica, desenvolvendo uma abordagem que motive-os, que seja versátil e ainda tenha possibilidade de ser adaptada a diferentes disciplinas, informações ou funcionalidades.

Por todos os aspectos abordados nesta Introdução, entende-se que é fundamental a inclusão digital das pessoas com necessidades especiais no ambiente da *internet* por intermédio de aplicações acessíveis. Ademais, a acessibilidade em *sites* é exigida por leis e políticas governamentais na maior parte dos países, como o Brasil, por exemplo.

1.2 Objetivos

O objetivo principal deste trabalho foi apresentar um *framework* que pode ser utilizado tanto no desenvolvimento de jogos *on-line* acessíveis como em *websites* informativos, sistemas *web* ou qualquer outra solução que tenha como base a linguagem HTML.

É importante destacar que originalmente esta pesquisa era voltada para jogos *on-line*. Entretanto, em consequência da abrangência que se pode atingir com uma proposta genérica de ferramenta de apoio que atendesse todas as necessidades dos jogos, foi proposto o desenvolvimento de uma solução que atendesse aplicações *web* mais estáticas, tais como sites e sistemas, não excluindo jogos que se moldem as diretrizes do presente trabalho. Por esse motivo, boa parte da bibliografia encontrada neste trabalho é baseada em jogos *on-line*, servindo como base e sendo adaptada para as funcionalidades do *framework*.

Visando desenvolver as funcionalidades da ferramenta, investigou-se diretrizes presentes atualmente nos projetos de desenvolvimento de aplicações *web* acessíveis, tais como *websites* e jogos *on-line*, e adaptando o que se julgou necessário nas diretrizes encontradas para que seja possível o seu uso na plataforma *web*. Mais especificamente, este trabalho teve como

objetivo coletar diretrizes existentes que são utilizadas no desenvolvimento de jogos acessíveis nas plataformas *desktop* e *console*, aplicá-las no desenvolvimento *web* de jogos, verificando sua utilização, e ainda complementá-las propondo novas abordagens de desenvolvimento com fundamento nos estudos desenvolvidos neste trabalho.

A partir dessa coleta, pôde-se basear o desenvolvimento do *framework* nas especificações obtidas para atender o que é necessário para que o mesmo seja considerado apto no processo de desenvolvimento de aplicações *online* acessíveis. Uma vez desenvolvido e denominado como ARIA-ACCESS, o *framework* possui mecânica similar ao Twitter Bootstrap³. Nele, o processo de tornar o objeto em questão acessível é abstraído do programador no que diz respeito ao conhecimento das *tags* da WAI-ARIA (apresentada na Seção 2.2), sendo necessários apenas o conhecimento de domínio da ferramenta e a devida utilização deste *framework* no projeto, indicando-se via *classes* do HTML que o elemento deve ser acessível.

1.3 Organização do documento

No Capítulo 2, são descritas as fundamentações teóricas necessárias para o entendimento da proposta de pesquisa e sínteses da bibliografia fundamental. No Capítulo 3, são apresentados os trabalhos relacionados ao desenvolvimento do ARIA-ACCESS, tais como as diretrizes coletadas para a validação e desenvolvimento do mesmo. No Capítulo 4, é apresentada a ferramenta obtida a partir dos estudos deste trabalho: o ARIA-ACCESS. Já no Capítulo 5, são apresentados os dados coletados referentes ao estudo de caso realizado neste trabalho, o resultado proveniente deste estudo e as discussões levantadas. Por fim, as contribuições e as limitações deste trabalho, bem como os trabalhos futuros, podem ser encontrados no Capítulo 6.

³<http://getbootstrap.com/>

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são descritos os principais conceitos necessários para o entendimento do projeto, no que se refere ao desenvolvimento que visa a prover acessibilidade em aplicações *on-line*. Os tipos de deficiência abordados neste trabalho são apresentados na Seção 2.1. Na Seção 2.2, encontram-se os principais conceitos do WAI-ARIA, e descrevem-se os principais tópicos necessários para o entendimento do projeto quanto às ferramentas necessárias para desenvolver aplicações *web* que possibilitam o desenvolvimento de componentes acessíveis. Os conceitos da documentação WCAG podem ser vistos na Seção 2.3. Posteriormente, encontramos o Modelo de Acessibilidade em Governo Eletrônico (eMAG) na Seção 2.4. A linguagem HTML na versão cinco é abordada na Seção 2.5. Como complemento, e por ser a linguagem escolhida para desenvolver o *framework*, encontra-se na Seção 2.6 a revisão sobre a linguagem JavaScript. Por fim, na Seção 2.7 são apresentadas as considerações finais.

2.1 Tipos de deficiência

Sabe-se que as deficiências podem ter origem genética, surgir no período de gestação, vir em decorrência do parto ou nos primeiros dias de vida do bebê. Não se limitam porém apenas aos primeiros momentos de vida, podendo ainda ser consequência de doenças transmissíveis ou crônicas, perturbações psiquiátricas, desnutrição, abuso de drogas, traumas e lesões (BRASIL, 2012).

Deficientes físicos e motores podem ser identificados como público-alvo neste projeto de implementação de aplicações *web* acessíveis, uma vez que, atualmente, estão disponíveis diretrizes que visam a auxiliar desenvolvedores a tornar acessível esse tipo de aplicações para usuários portadores de deficiência. Contudo, também vale observar diferentes tipos de deficiências mais corriqueiras e temporárias que produzem impacto na utilização de aplicações em

geral.

A seguir, são descritas as deficiências que desejamos auxiliar seus portadores com este trabalho, bem como seus impactos na acessibilidade.

2.1.1 Cegueira

Para Taleb et al. (2012), o a definição médica de cegueira reúne indivíduos com diversos graus de visão residual e suas respectivas peculiaridades. Segundo os autores, cegueira não significa, necessariamente, total incapacidade para ver, mas o prejuízo dessa aptidão em níveis incapacitantes para o exercício de tarefas rotineiras, alguns casos é possível ter visão a curta distância ou visualizar vultos. Destacam ainda que, mais comum em sua definição, a cegueira total — também conhecida como *amaurose* — pressupõe completa perda de visão. Nela, a visão é nula, isto é, nem a percepção luminosa está presente.

2.1.2 Daltonismo

O daltonismo, também conhecido como cegueira cromática, é uma perturbação da percepção visual caracterizada pela incapacidade de diferenciar alguns tons de cores. A variação mais comum de daltonismo é a incapacidade de distinguir tons das cores vermelha e verde. Neste caso de daltonismo, o que uma pessoa com percepção normal da cor vê na cor verde ou vermelha é visto em tons cinzentos de várias tonalidades. Ainda é possível citar um caso raro de daltonismo, o qual consiste na incapacidade de identificar qualquer tipo de cor: tal limitação também é conhecida como *monocromacia*, e consiste no caso em que tudo é visto em preto, branco e tons de cinza (DALTONISMO, 2014).

2.1.3 Deficiência motora

A deficiência motora é uma limitação do funcionamento físico-motor de um indivíduo. Uma vez identificados, os problemas ocorrem principalmente no cérebro ou no sistema locomotor e, contrariando o senso comum, as deficiências não possuem somente origem pré-natal, podendo ter origem peri ou pós-natal, ser causada por fatores genéticos, fatores virais ou bacterianos ou fatores adquiridos (traumáticos ou por ingestão de drogas). Desta forma, esta disfunção pode afetar o indivíduo no que diz respeito à mobilidade, atividades sensoriais e até mentais (RIBAS, 1985).

2.2 WAI-ARIA

Esta seção foi desenvolvida a partir dos estudos encontrados na seguinte referência WAI-ARIA (2014).

A WAI-ARIA (Accessible Rich Internet Applications) é uma especificação técnica, proposta e gerenciada pelo W3C, e tem como objetivo criar diretrizes e especificações que forneçam uma estrutura para melhorar a acessibilidade e a interoperabilidade dos conteúdos *web* e das aplicações. Esse documento é voltado principalmente aos desenvolvedores, para criação de *widgets* personalizados e de outros componentes de aplicação *web*.

Baseando-se nessa documentação, pode-se afirmar que a relação dessa especificação técnica com o presente trabalho se baseia nas instruções descritas em seus documentos oficiais e nas metodologias empregadas para prover acessibilidade no desenvolvimento *web*, com o propósito de desenvolver um *framework* que siga os padrões definidos pela documentação da WAI-ARIA.

Tais padrões visam a auxiliar desenvolvedores a produzir aplicações acessíveis diante do problema da grande aplicabilidade do HTML, e esta afirmação se justifica na reflexão sobre o uso de *tags* do HTML para representar papéis semânticos completamente diferentes da sua finalidade original.

Como solução aos problemas supracitados, pode-se prover a incorporação da WAI-ARIA no projeto de desenvolvimento, já que esta documentação fornece semânticas adequadas, fornecendo a predisposição ao desenvolvedor com o propósito de fazê-lo implementar aplicações personalizadas, as quais venham a tornar-se acessíveis, utilizáveis e compatíveis com as tecnologias de apoio, como os leitores de tela. Identificando os tipos de aplicações — bem como a sua própria estrutura —, que são normalmente reconhecidas pelos produtos desenvolvidos com as *tags* acessíveis, fornecendo informações sobre a natureza dos papéis correspondentes de cada componente para que eles possam ser indexados na página e fornecer o seu conteúdo para as tecnologias de apoio. Deste modo, pode-se utilizar elementos com um papel diferente do proposto pela concepção do HTML, de maneira que possam ser interpretados como um elemento específico ou tipo estrutural na aplicação desenvolvida.

Consequentemente, pode-se afirmar que a WAI-ARIA define uma abordagem singular para desenvolver aplicações de conteúdo *web*, além de também definir formas de *web* mais acessíveis às pessoas com deficiência, ajudando especialmente com conteúdo dinâmico e controles avançados de interface de usuário. Além disso, possibilita integração com aplicações desenvolvidas nas mais diversas tecnologias, tais como AJAX, HTML, JavaScript, PHP, entre outras.

A WAI-ARIA divide sua semântica em dois componentes distintos: *Roles*, que define com que tipo de elemento o usuário está interagindo, e *States/Properties*, que são suportadas pelas *Roles*, que definem o estado daquele elemento.

2.2.1 Roles

As *Roles* são utilizadas para descrever o tipo de *widget* apresentado, tais como *menu*, *treeitem*, *slider* e *progressmeter*. Podem ainda definir a estrutura da página da *web*, tais como títulos, regiões e tabelas.

2.2.1.1 Role document structure

As *Roles* desta categoria servem para indicar que aquele elemento faz parte da estrutura do leiaute. Um exemplo de sua utilização, da inclusão de um elemento, pode ser visto na Figura 2.1, que indica que um texto está na página. Esta categoria de *Roles* também pode ser usada para a criação de menus, que podem ser vistos na Figura 2.2. Ainda vale destacar a Figura 2.3, onde esse elemento é acrescentado na página complementando a informação dela.

Figura 2.1: Exemplo de código de um artigo acessível.

```
<article role="article">
  <p>Texto de Exemplo</p>
</article>
```

Figura 2.2: Exemplo de código de um menu acessível.

```
<ul role="menubar">
  <li role="menuitem">Menu 1</li>
  <li role="menuitem">Menu 2</li>
  <li role="menuitem">Menu 3</li>
</ul>
```

Figura 2.3: Exemplo de código de uma informação complementar acessível.

```
<aside role="complementary">
  Conteúdo complementar
</aside>
```

2.2.1.2 Role widget

A utilização da tecla *Tab* é um elemento comum nos *websites*, padrão de manipulação de diversos teclados acessíveis, e facilitador da interação de leitores de tela com o *website*. Um exemplo de código que busca facilitar a interação com *Tabs* pode ser visto na Figura 2.4 a seguir.

Figura 2.4: Exemplo de código de uma informação complementar acessível.

```
<ul>
  <li>
    <a href="#tab-panel1" class="active" id="tab1" role="tab" aria-selected="true">Primeiro Link</a>
    <a href="#tab-panel2" id="tab2" role="tab">Segundo Link</a>
  </li>
</ul>

<div id="tab-panel1" role="tabpanel" aria-labelledby="tab1">
  Conteúdo do Primeiro Link
</div>
<div id="tab-panel2" role="tabpanel" aria-labelledby="tab2">
  Conteúdo do Segundo Link
</div>
```

2.2.1.3 Role landmarks

As *landmarks* servem para conduzir a navegação do usuário. Assim como vimos na Seção 2.2.1.2, a condução da leitura feita no *site* dá-se com a movimentação por essas áreas com a tecla *Tab*. Como é apresentado na Figura 2.5, as *roles* de *landmarks* marcam áreas importantes na página para que o usuário encontre os blocos de informações mais importantes.

Apesar de o HTML5 possuir *tags* correspondentes com as *landmarks* apresentadas na Figura 2.5, pode haver eventualidades nas quais o desenvolvedor necessite marcar qualquer elemento da página sem utilizar as *tags* do HTML5. Existem casos onde não haverá *tags* correspondentes, tais como: (i) *Banner*: É uma região da página que contém uma imagem ou um título que faz uma breve introdução do que será apresentado, podendo ainda ser utilizado para publicidades; (ii) *Complementary*: Utilizado para marcar uma região do documento onde uma informação complementar é adicionada ao conteúdo principal da página; (iii) *Content Info*: Tem a finalidade de agregar informações complementares à página toda, mais usado para notas de rodapé e *links* para contratos.

2.2.2 Properties

As *Properties* são usadas para descrever os estados em que os *widgets* estão, como *checked* para uma caixa de seleção, ou *haspopup* para um menu.

Essas *Properties* também podem ser usadas para definir regiões que se modificam dentro de uma página, graças à integração com tecnologias como AJAX e JavaScript, que têm a

Figura 2.5: Exemplo de código de uma informação complementar acessível.

```
<nav role="navigation">
  Conteúdo de Navegação
</nav>
<main role="main">
  Conteúdo Principal
</main>
```

capacidade de atualizar informações da página em tempo real. Essas atualizações podem ser apresentadas em uma caixa de diálogo ou podem ser atualizações de campos dentro da página.

2.2.3 Como o HTML5 se relaciona com o WAI-ARIA

Como se pode verificar mais detalhadamente na Seção 2.5, o HTML5 introduz uma série de novas *tags* semânticas para HTML. Algumas dessas marcações sobrepõem-se diretamente com as funções disponíveis na WAI-ARIA.

A rede de desenvolvedores da Mozilla é uma entusiasta no desenvolvimento de uma "internet acessível". Por isso, sua comunidade auxilia jovens desenvolvedores a se integrar com os principais conceitos de acessibilidade. Destacando que nos casos em que o navegador suporta *tags* do HTML5 que também existem no WAI-ARIA, normalmente não há necessidade de também adicionar papéis WAI-ARIA e estados para o elemento (DEVELOPERS, 2015). Contudo, a WAI-ARIA inclui muitos papéis, estados e propriedades que não estão disponíveis em HTML5, de modo que estes continuarão a ser úteis para desenvolvedores que usam HTML5. No entanto, pode-se perceber no desenvolvimento deste trabalho que é interessante utilizar tanto a *tag* do HTML5 quanto as funções do WAI-ARIA, pelo fato de alguns navegadores utilizados não serem de uma versão mais recente.

2.2.4 Como funcionam as tecnologias de apoio

Tecnologias de apoio utilizam uma API (Application Programming Interface) construída em cada sistema operacional projetado especificamente para descrever papéis, estados e estruturas de interface com o usuário do aplicativo (DEVELOPERS, 2015).

Segundo a Developers (2015), um teclado na tela utiliza uma API para fornecer-lhe uma abordagem mais eficiente na apresentação do leiaute ou no controle da interface do usuário. Tratando-se de leitores de tela, eles fazem uso dessa API para ler a interface do usuário com um

motor *text-to-speech* — sistema que converte texto em linguagem natural de voz —, enquanto um ampliador usa essa API para dar destaque a áreas importantes ou ativas da tela.

Com base na Developers (2015), as tecnologias assistivas frequentemente acessam uma página através dessa API, a fim de compreender a semântica e os atributos da página para que a integração da *web* convencional possa agregar funcionalidades acessíveis.

2.2.5 Interação com o teclado

Todas as interações com WAI-ARIA devem ser capazes de serem manuseadas com o teclado (WAI-ARIA, 2014; DEVELOPERS, 2015).

Conforme os padrões de desenvolvimento encontrados na Developers (2015), um *widget* desenvolvido para que o usuário possa clicar, fazer *drag and drop*, *slide*, *scroll* ou qualquer outra interação com o *mouse*, deve ser feito para que um usuário interaja e seja capaz de manipulá-lo com uma ação equivalente ao uso do teclado.

Todos os *widgets* devem responder aos comandos e às combinações padrões de teclas dos sistemas operacionais. Consoante à Developers (2015), ao desenhar um botão com uma *tag span* e ao atribuir um *role=“button”*, esse elemento é identificado na página e deve apresentar o comportamento de um botão, ou seja, se o usuário der foco a esse elemento e apertar ENTER, ele deve agir como um botão.

Com a padronização da interação de teclados com as páginas orientada pela tecla *Tab* (para ir ao próximo menu) e *Shift + Tab* (para voltar ao menu anterior), surge um novo problema: desenvolvedores da Developers (2015) verificaram que, em páginas bastante densas, um usuário que utiliza o teclado para navegar muitas vezes tem de pressionar inúmeras vezes a tecla *Tab* até acessar a seção que deseja. Com isso, a Developers (2015) propõe a implementação de convenções de teclado de estilo *desktop* na *web*, nas quais se verifica o potencial dessa técnica para acelerar significativamente a navegação a muitos usuários. Um resumo de como essa navegação deveria funcionar pode ser visto a seguir:

- A tecla *Tab* deve fornecer o foco para o *widget* como um todo;
- As teclas de seta devem permitir a seleção ou a navegação dentro do *widget*;
- Dentro de um formulário, a *Space Bar* deverá selecionar ou ativar o controle, enquanto a tecla *Enter* deverá apresentar a ação padrão do formulário;

- Em caso de dúvida, imitar o comportamento da área de trabalho padrão do controle que se está criando.

2.3 Diretrizes de acessibilidade ao conteúdo da Web (WCAG)

Os estudos que compõem esta Seção foram baseados na referência do WCAG (2012).

A documentação oficial da WCAG se trata de um conjunto de diretrizes publicadas pela WAI do W3C que visa orientar os desenvolvedores na criação de páginas com conteúdo acessível aos portadores de deficiência, independentemente do meio físico onde ele é acessado (*desktops, notebook, smartphones* ou *tablets*).

Também é possível encontrar na documentação diretrizes que explicam como produzir conteúdos para a *web* acessíveis às pessoas portadoras de deficiência. Encontra-se também a definição formal de “conteúdo”, definido como qualquer informação contida em uma página *web* ou em uma aplicação *web*, incluindo-se aí textos, imagens, formulários, sons e componentes semelhantes.

A WCAG é parte de uma série de diretrizes para acessibilidade, como as Diretrizes de Acessibilidade para as Ferramentas de Produção (ATAG) e as Diretrizes de Acessibilidade para Agentes de Usuários (UAAG).

A versão mais recente é a WCAG 2.0, a qual foi publicada em dezembro de 2008 e é também padrão ISO/IEC 40500:2012.

2.3.1 Direcionamento e níveis de abordagem das WCAG

Intrinsecamente, as diretrizes WCAG são direcionadas a desenvolvedores de conteúdo para a *web*, desenvolvedores de *widgets* para a *web* e desenvolvedores de ferramentas de avaliação de acessibilidade. Sua documentação tem como finalidade suprir as necessidades de diferentes tipos de audiência, incluindo legisladores para políticas e linhas de ação em geral, administradores e outros.

Considerando a grande diversidade do público que utiliza as WCAG, foram elaborados diversos níveis de abordagem, e são eles: (i) *Princípios*: constituem a fundação da acessibilidade da *web*, considerados como perceptível, operável, compreensível ou robusto; (ii) *Diretrizes*: complementam os princípios. Atualmente, 12 diretrizes fornecem os objetivos básicos que os autores devem atingir para produzir conteúdo mais acessível a utilizadores com diferentes in-

capacidades; (iii) *CrITÉrios de Sucesso*: para cada diretriz, são fornecidos critérios de sucesso para que elas possam ser testadas e utilizadas onde os requisitos e os testes de conformidade sejam aplicáveis. Para isso, foram definidos três níveis de conformidade: A, AA e AAA; (iv) *Técnicas de tipo Suficiente e de tipo Aconselhada*: para validar os critérios de sucesso, foram definidas duas categorias de técnicas: as que são de tipo suficiente, para satisfazer os critérios de sucesso, e as que são de tipo aconselhada.

2.3.2 Princípios e diretrizes

A versão 2.0 das WCAG trouxe em forma de instruções normativas um conjunto de princípios, bem como suas diretrizes, que as aplicações deveriam apresentar para ser consideradas acessíveis. Pode-se visualizar a seguir os princípios defendidos pela documentação oficial da WCAG e suas respectivas diretrizes.

2.3.2.1 Princípio 1 – Perceptível

O conteúdo e como ele é posicionado na interface do usuário deve ser apresentado de forma que os utilizadores possam percebê-lo e possam interagir com ele sem dificuldade. Podemos ver as diretrizes na Tabela 2.1.

Tabela 2.1: Diretrizes WCAG que se referem ao Princípio 1

Diretriz	Descrição
Diretriz 1.1 – Alternativas em Texto	Fornecer alternativas em texto para todo o conteúdo não textual, de modo que este possa ser apresentado de outras formas, de acordo com as necessidades dos utilizadores. Por exemplo: caracteres ampliados, braille, fala, símbolos ou uma linguagem mais simples.
Diretriz 1.2 – Média Dinâmica ou Contínua	Fornecer alternativas para conteúdo em multimídia dinâmica ou temporal.
Diretriz 1.3 – Adaptável	Criar conteúdo que possa ser apresentado de diferentes formas (por ex., um esquema de página mais simples) sem perder informação ou estrutura.
Diretriz 1.4 – Distinguível	Facilitar aos utilizadores a audição e a visão dos conteúdos nomeadamente através da separação entre o primeiro plano e o plano de fundo.

2.3.2.2 Princípio 2 – Operável

O conteúdo da aplicação e a navegação da interação entre a interface de usuário e seus componentes têm de ser operáveis. Podemos ver as diretrizes na Tabela 2.2.

Tabela 2.2: Diretrizes WCAG que se referem ao Princípio 2

Diretriz	Descrição
Diretriz 2.1 – Acessível por Teclado	Fazer com que toda a funcionalidade fique disponível a partir do teclado.
Diretriz 2.2 – Tempo Suficiente	Proporcionar aos utilizadores tempo suficiente para ler e utilizar o conteúdo.
Diretriz 2.3 – Convulsões	Não criar conteúdo de uma forma que sabemos poder causar convulsões.

2.3.2.3 Princípio 3 – Compreensível

O conteúdo presente na aplicação e a manipulação da interface do usuário têm de ser compreensíveis. Podemos ver as diretrizes na Tabela 2.3.

Tabela 2.3: Diretrizes WCAG que se referem ao Princípio 3

Diretriz	Descrição
Diretriz 3.1 – Legível	Tornar o conteúdo textual legível e compreensível.
Diretriz 3.2 – Previsível	Fazer com que as páginas <i>web</i> apareçam e funcionem de forma previsível.
Diretriz 3.3 – Assistência na Inserção de Dados	Ajudar os utilizadores a evitar e a corrigir os erros.

2.3.2.4 Princípio 4 – Robusto

O conteúdo deve prezar pela capacidade de ser robusto, de forma que seja interpretado o mais fidedignamente possível por uma ampla variedade de agentes de usuário, incluindo as tecnologias de apoio. Podemos ver as diretrizes na Tabela 2.4.

Tabela 2.4: Diretriz WCAG que se refere ao Princípio 4

Diretriz	Descrição
Diretriz 4.1 – Compatível	Maximizar a compatibilidade com os agentes de usuário atuais e futuros, incluindo as tecnologias de apoio.

2.3.3 Formato dos documentos

As diretrizes da WCAG são documentadas e divididas em forma de uma lista, padrão adotado pelo W3C. Essa lista serve para validar uma página ou *site* de acessibilidade indicando, para cada ponto de verificação, se o princípio foi satisfeito, não foi satisfeito ou não é aplicável.

2.3.4 Evolução das WCAG

A primeira versão das diretrizes para acessibilidade ao conteúdo da *web* foi aprovada em maio de 1999 e, apesar de ter evoluído com o tempo, ainda constitui uma versão confiável de referência para um ponto de partida no que tange ao desenvolvimento de aplicações acessíveis.

Partindo do princípio de que a evolução é algo comum no desenvolvimento de *software*, as especificações da W3C para o processo de desenvolvimento preveem a inclusão de períodos nos quais o público faz revisões e comentários sobre os documentos. A partir dessas revisões é feita uma síntese que contém os principais conceitos para desenvolver aplicações acessíveis. Na Tabela 2.5 podemos ver como as WCAG foram evoluindo ao longo dos anos, comparando item por item as diretrizes presentes nas versões 1.0 e 2.0.

2.4 eMAG

Esta Seção foi desenvolvida conforme os estudos apresentados no Eletrônico (2014).

Conforme se verifica no Federal (2004), a acessibilidade em *sites* da administração pública passou a ser obrigatória no Brasil (vide o decreto nº 5.296 de 2004) e estabeleceu-se que o acesso e a navegação de pessoas com necessidades especiais deveriam ser garantidos. Com isso em mente, o governo federal comprometeu-se em prover o acesso completo a qualquer pessoa que buscasse informações em *sites* de administração pública, elaborando o Modelo de Acessibilidade do Governo Eletrônico (eMAG).

O eMAG (2014) é um documento criado em 2005 que contém recomendações e diretrizes sobre a acessibilidade. À medida que a *web* ia se desenvolvendo, verificou-se a necessidade de adaptar a documentação para que acompanhasse os avanços que a *internet* solicitava. Sendo assim, a versão mais recente é o eMAG 3.1, que se baseia em diretrizes internacionais, como a WCAG 2.0, que possibilita a criação e a adaptações de conteúdos para *web* acessíveis de forma padronizada e de fácil implementação.

O documento surgiu inicialmente no ano 2000, da parceria firmada entre o Ministério do Planejamento, Orçamento e Gestão (MP) e a Secretaria de Logística e Tecnologia da Informação (SLTI). Posteriormente, com o lançamento da WCAG (2008), houve a revisão do modelo 2.0 realizada através da parceria entre o Departamento de Governo Eletrônico e o Projeto de Acessibilidade Virtual da RENAPI (Rede de Pesquisa e Inovação em Tecnologias Digitais), resultando na versão 3.0 do documento.

Os autores reconhecem que, para desenvolver um portal verdadeiramente acessível, é cru-

Tabela 2.5: Resumo das WCAG na versão 1.0 e 2.0

WCAG 1.0	WCAG 2.0
Forneça alternativas equivalentes ao conteúdo de forma sonora e visual.	Garanta que as informações e os componentes da interface sejam apresentáveis e perceptíveis para os usuários.
Não confie somente na cor.	Forneça alternativas multimídias com temporizador.
Use corretamente as CSS's.	Crie conteúdos que possam ser apresentados de diversas formas sem perder estrutura e informação.
Use linguagem natural.	Torne fácil para o usuário ver ou ouvir conteúdos.
Garanta a recepção de novas tecnologias às novas páginas.	Crie componentes de navegação operáveis.
Garanta que o usuário terá controle de alterações de acessibilidade.	Permita que todas as funcionalidades da página sejam acionadas por teclados.
Garanta acessibilidade rápida às páginas ocultas.	Não crie conteúdos visuais (3D ou animações muito rápidas, por exemplo) que tendem a causar convulsões.
Desenhe de forma que as páginas se adaptem às telas de qualquer dispositivo.	Ajude o usuário a encontrar os conteúdos.
Use o W3C para validação.	Garanta que a navegação e operação das páginas sejam compreensíveis.
Garanta navegações claras.	Garanta que o funcionamento das páginas seja previsível.
Garanta conteúdo simples e de vocabulário claro.	Ajude os usuários a evitar e corrigir erros.
Crie tabelas que se transformem graciosamente.	
Use soluções provisórias.	
Fornecer informações de contexto e orientação.	

cial, inicialmente, que o código desenvolvido esteja em conformidade com os padrões acessíveis definidos pelo W3C, sendo utilizada como referência a documentação oficial da WCAG.

No eMAG, de forma distinta da WCAG, as recomendações de acessibilidade são apresentadas por áreas de aplicabilidade e não por níveis de prioridade. Como a sua aplicação é voltada para páginas governamentais, todos os quesitos, quando aplicáveis, devem ser seguidos.

Atualmente, o eMAG 3.1 (2014) trouxe novas seções em relação às versões mais antigas. O documento está dividido nas seis seguintes categorias:

1. Recomendações de acessibilidade – MARCAÇÃO

- (a) Respeitar os padrões *web*;
- (b) Organizar o código HTML de forma lógica e semântica;
- (c) Utilizar corretamente os níveis de cabeçalho;
- (d) Ordenar de forma lógica e intuitiva a leitura e a tabulação;
- (e) Fornecer âncoras para ir direto a um bloco de conteúdo;
- (f) Não utilizar tabelas para diagramação;
- (g) Separar *links* adjacentes;
- (h) Dividir as áreas de informação;
- (i) Não abrir novas instâncias sem a solicitação do usuário.

2. Recomendações de acessibilidade – COMPORTAMENTO

- (a) Disponibilizar todas as funções da página via teclado;
- (b) Garantir que os objetos programáveis sejam acessíveis;
- (c) Não criar páginas com atualização automática;
- (d) Não utilizar redirecionamento automático de páginas;
- (e) Fornecer alternativa para modificar limite de tempo;
- (f) Não incluir situações com intermitência de tela;
- (g) Assegurar o controle do usuário sobre as alterações temporais do conteúdo.

3. Recomendações de acessibilidade – CONTEÚDO/INFORMAÇÃO

- (a) Identificar o idioma principal da página;
- (b) Informar mudança de idioma no conteúdo;
- (c) Oferecer um título descritivo e informativo à página;
- (d) Disponibilizar informação sobre a localização do usuário na página;
- (e) Descrever *links* clara e sucintamente;
- (f) Fornecer alternativa em texto para as imagens;
- (g) Utilizar mapas de imagem de forma acessível;
- (h) Disponibilizar documentos em formatos acessíveis;
- (i) Em tabelas, usar títulos e resumos de forma apropriada;

- (j) Associar células de dados a células de cabeçalho;
- (k) Garantir a leitura e a compreensão das informações;
- (l) Disponibilizar uma explicação para siglas, abreviaturas e palavras incomuns.

4. Recomendações de acessibilidade – APRESENTAÇÃO/DESIGN

- (a) Oferecer contraste mínimo entre plano de fundo e primeiro plano;
- (b) Não utilizar apenas cores ou outras características sensoriais para diferenciar elementos;
- (c) Permitir redimensionamento de texto sem perda de funcionalidade;
- (d) Possibilitar que o elemento com foco seja visualmente evidente.

5. Recomendações de acessibilidade – MULTIMÍDIA

- (a) Fornecer alternativas para vídeo;
- (b) Fornecer alternativas para áudio;
- (c) Oferecer audiodescrição para vídeo pré-gravado;
- (d) Fornecer controles de áudio para som;
- (e) Fornecer controles de animação.

6. Recomendações de acessibilidade – FORMULÁRIOS

- (a) Fornecer alternativas em texto para os botões de imagem de formulários;
- (b) Associar etiquetas aos seus campos;
- (c) Estabelecer uma ordem lógica de navegação;
- (d) Não provocar automaticamente alteração no contexto;
- (e) Fornecer instruções para entrada de dados;
- (f) Identificar e descrever erros de entrada de dados;
- (g) Agrupar campos de formulário;
- (h) Fornecer estratégias de segurança específicas ao invés de CAPTCHA.

2.5 HTML5

HTML é a sigla em inglês para *HyperText Markup Language*, que, em português, significa linguagem para marcação de hipertexto (SILVA, 2011; MEYER, 2010).

Não existe um proprietário da linguagem HTML, costuma-se dizer que HTML é o resultado do empenho de diversas pessoas que trabalham em muitos países e em muitas organizações para definir as características da linguagem. Um documento HTML é um documento de texto que pode ser produzido utilizando-se qualquer editor de texto. Documentos HTML contêm elementos baseados em *tags*, que por sua vez são usadas na criação de todos os elementos visuais e não visuais de uma página na *web*, tais como: imagens, *links*, parágrafos, sons, etc. (MEYER, 2010).

Consoante Meyer (2010), o HTML básico pode incluir diretivas para a formatação de layout em uma linguagem chamada Folhas de Estilo em Cascata (CSS) e de *widgets* para a interação com o usuário em tempo de execução, podendo ser utilizada a linguagem JavaScript.

O que diferencia intrinsecamente o HTML5 e o HTML4 é o fato de o HTML5 estar sendo desenvolvida com o propósito de substituir tanto o HTML, criada nos anos '90, quanto o XHTML, que foi uma tentativa frustrada de reformular o HTML4 como uma aplicação XML (SILVA, 2011).

Usar os novos elementos disponíveis no HTML5 torna o processo de demonstração dos benefícios da linguagem no quesito de acessibilidade mais fácil, uma vez que a linguagem faz com que seja mais simples para programas, como leitores de tela, consumir facilmente o conteúdo. Esse ganho com relação à acessibilidade dá-se com o fato de o HTML5 ter incluído diversas *tags* que auxiliam o usuário a interagir com o *website* (HOGAN, 2011). Ademais, o HTML5 apresenta um cenário promissor: ele está causando considerável excitação por causa de recursos tais como *canvas* (telas) na exibição de vídeo e áudio (MEYER, 2010), recurso que atualmente a Google vem utilizando para substituir o Flash nos recursos do YouTube¹.

Conforme Meyer (2010) e Hogan (2011), é possível criar um *site* sofisticado e interativo com o HTML5, aliado à integração com diversas bibliotecas disponíveis na linguagem JavaScript e à criação *templates* com a linguagem CSS. Essa afirmação justifica-se graças à criação de novas *tags* disponibilizadas na linguagem, as quais podem ser verificadas a seguir:

- *<header>*: Elemento que comporta as informações internas da página. Uma vantagem interessante de sua utilização é que podemos setar as *tags* h1 a h6 utilizadas, considerando

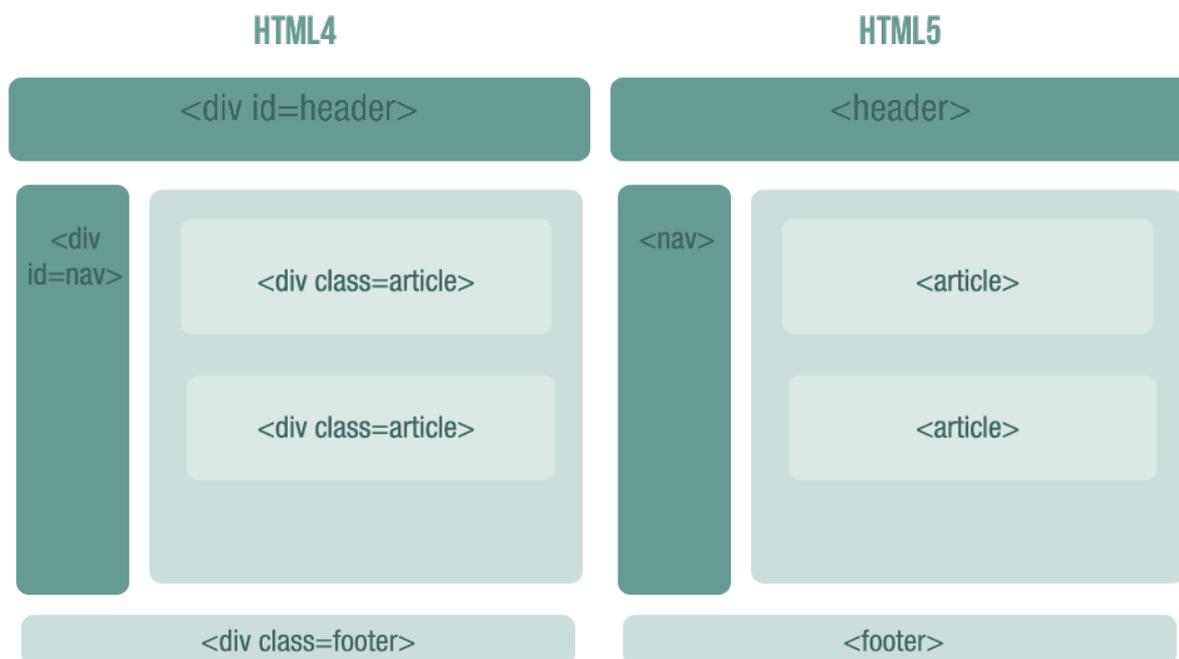
¹<http://www.youtube.com/>

apenas as hierarquias dos elementos;

- `<footer>`: Elemento que indica o rodapé de uma página;
- `<article>`: Elemento que indica o conteúdo de um artigo;
- `<nav>`: Elemento que indica a área de navegação do *website*;
- `<aside>`: Elemento semelhante ao `<sidebar>`, porém com possibilidade de inserção de outros elementos, tais como propagandas e menus de navegação.

Imagina-se que, com o lançamento de uma nova versão, se tenha uma versão melhorada da versão anterior. Com o HTML5 não foi diferente. Estima-se também que a longo prazo a utilização das novas *tags* auxiliarão na manutenção de código, na integração com novas tecnologias e até mesmo na facilidade de desenvolvimento. Pode-se observar na Figura 2.6 como costumava-se desenvolver um aplicação *web* na antiga versão do HTML, demonstrando as novas *tags* com suas funções equivalentes.

Figura 2.6: Exemplo da mudança no desenvolvimento com as novas *tags*.



Por fim, é importante constatar que as melhorias no HTML5 influenciam na semântica da linguagem. Nas versões anteriores, não é possível distinguir um cabeçalho de um rodapé, ou um menu de uma simples lista. Em outras palavras, as novas *tags* foram desenvolvidas para identificar o maior número de elementos possível dentro de uma aplicação, sem utilização de nomes genéricos.

2.6 JavaScript

JavaScript é a linguagem de programação usada para desenvolver o *framework* de desenvolvimento acessível.

A linguagem JavaScript foi criada por Brendan Eich quando este trabalhava para a Netscape, empresa de serviços de computadores, em parceria com a Sun Microsystems, com a finalidade de fornecer um meio de adicionar interatividade a uma página *web*. No ano de 1995, foi lançada a sua primeira versão, denominada JavaScript 1.0, e implementada em março de 1996 no navegador Netscape Navigator 2.0, navegador *web* mais utilizado na época da criação da linguagem (DOUGLAS, 2014).

O diferencial dessa linguagem é ter sido desenvolvida visando à interatividade com o usuário em tempo de execução, focada basicamente em interagir no lado do cliente, isto é, que a interpretação e o funcionamento da linguagem dependam de funcionalidades hospedadas no navegador do usuário (SILVA, 2010). Essa interação é possível porque as empresas incluem interpretadores JavaScript hospedados em seus navegadores.

De acordo com Silva (2010), JavaScript é uma linguagem capaz de acessar campos e valores digitados em um formulário, acessar e alterar informações de elementos da página em tempo de execução. Ademais, pode ser utilizada para interagir com outras linguagens de programação, tais como PHP, Java ou Python, para cumprir tarefas complementares relacionadas ao fluxo de eventos planejados no desenvolvimento. Tais capacidades tornam JavaScript a linguagem ideal para o desenvolvimento deste projeto, já que pode acessar todos os elementos HTML da página, interagir com eles e, ainda, ser utilizada com qualquer linguagem *web* com suporte a JavaScript.

2.7 Considerações finais

Os temas apresentados neste capítulo possuem relação direta com o presente trabalho, desde os estudos dos tipos de deficiências abordados até as tecnologias necessárias para adaptar aplicações *web* para pessoas com deficiência. Além disso, é importante destacar que essa revisão se faz importante porque, a partir dela, pode-se desenvolver um *framework* que seja de fácil utilização por desenvolvedores e que concomitantemente conceba conteúdos acessíveis de qualidade para os usuários com necessidades especiais.

É importante enfatizar que desenvolver um sistema acessível vai muito além de *tags* e abordagens, tornando as pessoas com necessidades especiais capazes de usufruir os benefícios de uma vida em sociedade. Técnicas de desenvolvimento acessível tornam viável a participa-

ção dessas pessoas nas atividades *on-line* de qualquer pessoa sem deficiência, generalizando o acesso de sistemas computacionais que incluem desde *games* até sistemas bancários.

No próximo capítulo, será possível ver os trabalhos relacionados que fundamentaram a proposta de desenvolvimento de *framework* apresentada neste projeto.

Capítulo 3

TRABALHOS CORRELATOS

Neste capítulo são apresentados os trabalhos encontrados na literatura que foram considerados diretamente relacionados ao tema deste trabalho. Portanto, cada seção do capítulo apresenta um trabalho relacionado ao projeto de pesquisa proposto, além de uma revisão das diretrizes para jogos, considerando que jogos têm mais referências em termos bibliográficos e têm diretrizes interativas. Na Seção 3.1, pode-se encontrar as diretrizes propostas por Ellis et al. (2013); Na Seção 3.2 são apresentadas as diretrizes desenvolvidas pela IGDA (2012); As diretrizes propostas por Barlet e Taylor (2012) podem ser vistas na Seção 3.3; o trabalho de Cheiran e Pimenta (2013), que apresenta uma reestruturação das diretrizes, encontra-se na Seção 3.4; a Seção 3.5 apresenta uma ênfase da disciplina de desenvolvimento, visando a superar as limitações das abordagens existentes. Por fim, na Seção 3.7 são apresentadas as considerações finais.

3.1 *Game accessibility guidelines (GAG)*

As *Game accessibility guidelines* são resultados de um esforço de colaboração entre um grupo de estúdios de desenvolvimento, especialistas e acadêmicos. Essa colaboração visava a produzir uma simples referência de abordagens de desenvolvimento acessível, para evitar que jogadores com necessidades específicas fossem excluídos desnecessariamente e com isso, assegurar que os jogos sejam divertidos para a mais ampla diversidade de pessoas possível.

Em seu trabalho, Ellis et al. (2013) afirmam que a acessibilidade pode ser vista como o ato de evitar barreiras desnecessárias que impedem as pessoas com deficiência de acessar ou desfrutar de uma aplicação.

Foram diversos os motivos que levaram os autores a desenvolver seu trabalho, mas eles

destacam na sua documentação que os principais fatores são: inclusão social, dados da demanda de pessoas com deficiência que necessitam desse investimento em acessibilidade, e pessoas com impedimentos temporários.

Os autores destacam ainda que, além do benefício financeiro com o desenvolvimento de jogos acessíveis, há o benefício humano, uma vez que os jogos são entretenimento, cultura e socialização. Indo mais além, os jogos acessíveis podem ser vistos como uma terapia, alívio da dor, fuga da rotina de limitações e independência.

Antes de apresentar as diretrizes propostas por Ellis et al. (2013), é necessário apresentar o processo para desenvolver aplicações acessíveis, composto de seis passos estabelecidos pelos autores para tirar o melhor proveito das diretrizes. São eles:

1. Familiarizar-se: Examinar as diretrizes antes de iniciar o trabalho de implementação. Os autores defendem que este processo reduz o trabalho se for feito na etapa do *design*, uma vez que as alterações serão mais fáceis de tratar;
2. Avaliar e Planejar: É importante definir uma mecânica base para ser aplicada. Muitas vezes, essa mesma mecânica pode beneficiar um público mais amplo apenas por fornecer funcionalidades extras;
3. Programar-se: Depois de definir as diretrizes que irão compor o projeto, é importante verificar quais diretrizes possuem maior impacto de produção se forem implementadas depois da etapa de desenvolvimento e colocá-las no início da etapa de desenvolvimento;
4. Implementar: As diretrizes são um bom começo, mas para validar o desenvolvimento é importante que os protótipos sejam testados por jogadores com deficiência e estes forneçam um retorno sobre as experiências obtidas;
5. Informar: É importante que a comunidade tenha conhecimento do trabalho desenvolvido. Listar os recursos de acessibilidade empregados é de extrema importância para essa comunidade, além de ganhar um impulsionamento de *SEO* (otimização para motores de busca, do inglês, *search engine optimization*);
6. Revisar e Aprender: Recolher dados sobre os recursos aplicados no projeto permite que o analista compare o custo de desenvolvimento com o número de jogadores impactados, podendo, assim, obter números precisos sobre a rentabilidade. Isto serve para avaliar custos de futuros projetos.

As diretrizes não têm relação direta com a WAI-ARIA, uma vez que são amplamente utilizadas em outras plataformas de desenvolvimento, não se limitando à *web*. Ellis et al. (2013) dividem suas diretrizes em três categorias: básico, intermediário e avançado. Estas categorias podem ser visualizadas como níveis definidos a partir da mensuração de três variáveis: alcance (número de pessoas beneficiadas), impacto (a diferença na vida das pessoas) e valor (custos de implementação). As diretrizes básicas estão na Tabela 3.1, as diretrizes intermediárias são apresentadas na Tabela 3.2, e na Tabela 3.3 encontramos as diretrizes avançadas para o desenvolvimento de jogos acessíveis.

Tabela 3.1: Diretrizes básicas das *Game accessibility guidelines*

Prática	Justificativa
Garantir que não haja informação essencial transmitida por uma só cor	As cores são os meios mais comuns para comunicar-se com significados já estabelecidos. No entanto, esses significados podem variar em algumas culturas ou podem perder-se com pessoas que não conseguem distinguir certas cores.
Se o jogo usa campo de visão, definir um padrão apropriado para o ambiente de visualização esperado	Cada jogo 3D tem um campo de visão ou perspectiva. Se esse campo de visão é significativamente diferente do que estamos acostumados, a experiência do jogo pode causar náuseas e desorientação nos jogadores.
Utilizar um tamanho padrão de fonte que seja de fácil leitura	Fontes pequenas são uma queixa muito comum entre <i>gamers</i> , isso afeta a experiência de pessoas com tela pequena ou pessoas com problemas médicos (como a dislexia).
Usar uma simples e clara formatação de texto	Utilizar uma fonte de fácil leitura sobre um fundo não exigente tem um impacto muito positivo para a legibilidade.
Fornecer alto contraste entre texto e fundo	Baixo contraste é uma queixa comum entre usuários mais velhos e pessoas que tem problemas situacionais, como luz solar direta na tela ou tela de má qualidade.
Desenvolver elementos interativos grandes e bem espaçados	Um problema comumente encontrado em jogos é que os botões são muito pequenos e pouco espaçados. Pessoas com insuficiência de precisão encontram muitas dificuldades em interagir com esses sistemas.

Tabela 3.2: Diretrizes intermediárias das *Game accessibility guidelines*

Prática	Justificativa
Evitar qualquer diferença entre movimento do controlador e movimento da câmera	Assim como vimos no campo de visão, qualquer informação diferente do que estamos esperando pode causar certo desconforto.
Usar o som <i>surround</i>	Para pessoas com certa dificuldade de visão, o som <i>surround</i> pode auxiliar na ambientação do jogador com o cenário.
Possibilitar a desabilitação da movimentação de fundo	Em algumas situações, as animações de fundo podem distrair o jogador e afetar as suas condições cognitivas.
Garantir a leitura de tela para dispositivos móveis	O leitor de tela é o meio que possibilita a interação de pessoas com pouca ou nenhuma visão com o jogo.
Permitir o ajuste do contraste	Assim como vimos na Tabela 3.1, a função de ajustar o contraste impactaria positivamente a experiência dos <i>gamers</i> .
Certificar-se que cada objeto-chave tenha um som bem definido e distinto de outros	Esse recurso faz-se necessário para pessoas com certa dificuldade em diferenciar objetos através outros meios não sonoros.
Permitir a troca de estilo de cursor, cores e mira de objetos	Os <i>gamers</i> frequentemente encontram problemas com o modelo do cursor e de suas cores, muitas vezes causados pelo contraste do cenário. Permitir a troca desses elementos torna o jogo mais agradável ao usuário.
Dar uma clara indicação de quais elementos são interativos	Jogadores com deficiências cognitivas ou visuais podem ter dificuldade na distinção de quais elementos ou itens do jogo são interativos.
Garantir que o manual fornecido possa ser lido por leitores de tela	Garantir que os manuais, FAQs e listas de recursos possam ser acessados por leitores de tela é muito mais importante do que disponibilizar, por exemplo, um PDF.
Fornecer controles de volume separados para efeitos, fala e fundo	Muitas fontes de informação diferentes podem dificultar a concentração em qualquer um deles. Há inclusive uma condição específica em que os sons simultâneos podem ser impossíveis de distinguir, causando muito desconforto ao jogador.
Evitar colocar informações temporárias importantes fora do campo de visão do jogador	Muitos problemas de visão podem resultar na perda da visão periférica. É importante organizar os elementos da interface ao longo das bordas de uma tela, assim os jogadores irão familiarizar-se com o leiaute e saber intuitivamente para onde olhar.
Permitir o redimensionamento das interfaces	Além de ser uma forma importante para atender os diferentes tamanhos de tela, o redimensionamento manual também é útil para pessoas com dificuldade de enxergar pequenos elementos ou sem a capacidade motora necessária para interagir com eles.

Tabela 3.3: Diretrizes avançadas das *Game accessibility guidelines*

Prática	Justificativa
Permitir que o tamanho da fonte seja ajustado	Embora definir um tamanho padrão seja benéfico, não há um tamanho único que todos possam ler, devido à diferenças nos níveis de deficiência visual, às diferenças de tamanho da tela ou à distância de visualização.
Fornecer um mapa que informe objetos importantes	Um mapa que pode ser acionado para alertar sobre objetos importantes ou inimigos através de um pulso estilo sonar.
Disponibilizar dublagens para todo o texto, incluindo menus e instaladores	É uma opção mais complexa, uma alternativa para o apoio à leitura de tela.
Apresentar um GPS orientado por voz	Mapas orientados por áudio podem ser usados como uma ferramenta para as pessoas com pouca ou nenhuma visão residual para navegar em ambientes 3D complexos.
Permitir fácil orientação de movimento ao longo dos pontos cardeais	Permitir aos jogadores saltar diretamente para um ponto cardinal da bússola auxilia jogadores com pouca ou nenhuma visão.
Certificar-se de que todas as ações podem ser realizadas por meio de comandos digitais, sendo possível adicionar outros controladores de modo que não sejam obrigatórios	Dispositivos de entrada analógicos, como <i>joysticks</i> , são difíceis de usar por um <i>gamer</i> de baixa visão.
Desenvolver menus e instaladores que possam interagir com leitores de tela	Entende-se que, quanto mais elementos nos jogos possam ser lidos por leitores de telas, mais independentes os jogadores poderão ser.
Usar sons e músicas distintas para todos os objetos e eventos	Um bom <i>design</i> de som, garantindo que cada ação distinta tenha um som igualmente distinto, pode tornar jogos até mesmo tradicionais completamente acessíveis a jogadores com pouca ou nenhuma visão.
Simular gravação binaural	Gravação binaural significa a construção de uma cabeça artificial na qual se colocam microfones na posição dos tímpanos. O áudio resultante registra com precisão o efeito do som que viaja através do crânio e das orelhas, resultando em um som incrivelmente preciso que pode ser reproduzidas através de qualquer fone de ouvido estéreo padrão.

3.2 Diretrizes da IGDA

Fundada no ano de 2003, a International Game Developers Association (IGDA) é um grupo especial de voluntários interessados em dedicar seu tempo em prol da acessibilidade em jogos. Os focos iniciais do grupo, que se mantêm até hoje, são a capacidade de fazer jogos que possam ser reproduzidos para todos e o cuidado especial para jogadores com deficiências de qualquer

tipo.

O grupo atualmente trabalha em conjunto com comunidades para desenvolver jogos acessíveis, criando métodos de desenvolvimento, compartilhando o conhecimento dentro da comunidade e, com isso, definindo as necessidades levantadas por diferentes deficiências e diferentes gêneros de jogos.

“Jogos de computadores e de consoles são uma cultura importante e questões de qualidade de vida. Ao colaborar com o resto da comunidade de desenvolvimento de jogos, a IGDA pretende desenvolver métodos de preparação de todos os gêneros de jogos universalmente acessível a todos, independentemente da deficiência. A fim de fazer isso, vamos promover a educação de desenvolvedores de jogos em *design* de acessibilidade, incentivos fiscais para os desenvolvedores de jogos acessíveis, patrocínio de empresas e as avaliações de acessibilidade (IGDA, 2012).”

Dentre os motivos que levaram os autores pesquisar sobre acessibilidade em jogos, os principais que podemos destacar são:

- Satisfação do Usuário: Jogadores buscam os jogos por entretenimento, não para ter experiências frustrantes. Se um jogador perder sua partida por uma deficiência do jogo em prover a interação com o ambiente, causará frustração;
- Atingir o maior público possível: Vários países têm normas relativas à acessibilidade dos produtos para os deficientes, destacando-se os Estados Unidos da América. Recentemente, as Nações Unidas propuseram “Normas sobre a Igualdade de Oportunidades para as Pessoas com Deficiência”;
- Proporcionar a oportunidade de aprender novas habilidades: Os jogos podem permitir que pessoas com deficiência possam aprender novas habilidades ou podem proporcionar benefícios terapêuticos, destacando-se o crescimento dos jogos para auxiliar a recuperação de movimentos;
- Jogos educacionais: Este é um novo e emergente campo, que resulta nos jogos que podem ser utilizados para aprendizagem – inclusive nas escolas públicas –, garantindo que sejam acessíveis a todos os alunos.

Depois de ter estudado o público beneficiado com jogos acessíveis e as razões para proporcionar acessibilidade, os autores desenvolveram diretrizes que visam à universalização de jogos.

Tendo, porém, um foco em deficiência visual, apresentaremos apenas as diretrizes focadas em melhorias para os deficientes visuais. A IGDA (2012) definiu diversas abordagens possíveis que os desenvolvedores podem utilizar para proporcionar a acessibilidade. A Tabela 3.4 apresenta as diretrizes a serem consideradas para tornar os jogos acessíveis para deficientes visuais. Os autores ainda destacam que essas recomendações podem ser aplicadas a jogos de PC e console.

Tabela 3.4: Diretrizes propostas pela IGDA (2012)

Prática	Justificativa
Alta visibilidade gráfica	É uma alternativa para fontes pequenas ou de difícil distinção que disponibiliza esquemas de cores de alto contraste ou os disponibiliza como uma opção, caso não seja padrão. Essa técnica destaca o que é importante no cenário para pessoas com baixa visão e pode, ainda, oferecer a desativação ou o escurecimento de fundo nos jogos 2D.
<i>Design</i> harmônico para daltônicos	Evitar combinações de cores que são difíceis ou impossíveis de distinguir a daltônicos (por exemplo, vermelho em cinza ou verde), ou oferecer opções alternativas no jogo onde essas cores não sejam utilizadas.
Fornecer diversos níveis de dificuldade e ajuste de velocidade	Permitir que os jogadores possam escolher entre uma vasta gama de dificuldades e velocidades do jogo, deixando que ele tenha o discernimento de quando está muito fácil ou muito lento.
Disponibilizar modos de tutorial	Oferecer um modo no qual o jogador é capaz de aprender a jogar sem se importar com a falha, fornecendo informações sobre como jogar o jogo. Isso ajuda a compreensão do jogo, o ajuste do controle, o desenvolvimento de habilidades, e também, simplesmente, oferece uma forma divertida para aquelas pessoas que têm dificuldades com o jogo padrão.
Menus acessíveis	Disponibilizar modos de início rápido, prezar pela navegação e fornecer textos alternativos de entrada, como <i>text-to-speech</i> e símbolos. Em um jogo com uma interface complexa, é relevante desenvolver uma interface simplificada que exiba apenas os controles comuns.

Listagem de recursos de acessibilidade, opções e requisitos do jogo	Fazer esforços para assegurar que essas informações sejam livres e fáceis de obter e entender. Elas podem ser disponibilizadas no <i>site</i> ou na embalagem do jogo. É importante enviá-las para revisão de <i>sites</i> de avaliação de jogos acessíveis.
Apresentação de texto padronizado	Para os jogos voltados para o sistema operacional Windows, existem técnicas de implementação que são compatíveis com os leitores de tela desse sistema.
Capacidade de auto-vocalização	É a capacidade de fornecer uma leitura do texto que está sendo exibido na tela. Já há uma variedade de ferramentas de <i>software</i> que fornecem esse recurso e podem ser integradas aos jogos. A solução <i>text-to-speech</i> deve trabalhar com APIs, como o API Speech da Microsoft. Esse texto pode estar escrito em diversas línguas e pode ser lido corretamente por sintetizadores de voz.
Navegação pelo teclado de todos os controles, com <i>feedback</i> visual e falado	Permitir que todos os comandos possam ser inseridos através do teclado. Também fornecer uma mensagem visual e auditiva para indicar o que foi feito.
Melhores tutoriais em tempo-real	Orientar o jogador além do esperado, fornecendo <i>feedbacks</i> e ensinando-lhe melhores abordagens. Fornecer ao jogador indicadores de objetivos ou seta direcional, evitando que jogadores se sintam perdidos.
Possibilidade de definir a cor da unidade	É a capacidade de controlar a cor das diferentes unidades, como inimigos, companheiros de equipe e outras unidades importantes dentro de um jogo.
Áudio GPS	Em um jogo acessível para cegos, um sistema de posicionamento global pode ser usado para obter as posições exatas dos objetos num espaço, bem como a posição do avatar. Um sistema de menu sonoro pode fornecer uma visão geral de objetos próximos.

Compasso de Som	Um som 3D original ou comentários de voz e uma tecla no teclado representam cada sentido em um período de oito direções no mundo do jogo. Quando o jogador aperta a tecla que representa a direção, o áudio toca para mostrar onde está direção, em relação ao avatar, que ele selecionou.
Orientação direta	Usar o teclado numérico para orientar o avatar em oito direções. É uma forma bem mais completa de orientar o avatar do que o padrão de alguns jogos (W,A,S,D).
Modo sem gráficos 3D	Dar ao jogador a capacidade de desligar a renderização 3D.
Mira automática	Ajudar o jogador, oferecendo a opção por mira automática, ou autocentralização. Isto permite que os jogadores com baixa visibilidade possam encontrar rapidamente as metas e os alvos.

3.3 A practical guide to game accessibility

A AbleGamers é uma fundação criada em 2013 e composta por desenvolvedores e jogadores com deficiência. Seus pesquisadores disponibilizaram um livro e um *website*¹ com suas diretrizes, com a finalidade de que desenvolvedores as utilizem como um recurso para responder a perguntas que eles venham a ter sobre a adição de acessibilidade para o jogo.

A informação que se encontra no *site*, segundo Barlet e Taylor (2012), é a coroação de quase uma década de pesquisas, avaliações e experiências em primeira mão, feitas por jogadores com deficiência que vivem com esses problemas todos os dias.

Para determinar se o jogo é acessível, os autores disponibilizam uma *checklist* que é utilizada como base no processo de revisão. Cada seção do documento cobre as áreas mais importantes de acessibilidade, e as pontuações são determinadas pela gravidade da infração. Um exemplo dessa revisão nos jogos é o controle das cores que possam prejudicar os daltônicos: se um jogo usa vermelho e verde para elementos importantes da interface do usuário, eles perdem mais pontos do que um jogo que usa vermelho e verde com outros indicadores visuais. Cada subcategoria é comparada do mesmo modo.

De acordo com Barlet e Taylor (2012), há uma observação muito importante a fazer sobre a proposta de suas diretrizes: partir do princípio de que todas as pessoas têm uma maneira diferente de jogar, ou seja, de que não existe um jeito certo ou errado de jogar quando se trata de alguém com necessidades especiais. Os autores destacam que os jogadores com deficiência querem apenas ser capazes de jogar o jogo, e, por vezes, isso significa que as opções de acessibilidade são uma obrigação pela qual zelar.

Nenhuma das sugestões que o desenvolvedor encontra nas diretrizes apresentadas a seguir poderá afetar negativamente, de alguma forma, o *design* do jogo. As orientações têm como finalidade habilitar pessoas com deficiência a aproveitar o conteúdo que foi disponibilizado. É importante entender que as opções de acessibilidade são apenas isto, opções.

É igualmente importante verificar as opções disponíveis para aqueles que delas necessitam; aqueles que não precisam não necessariamente terão conhecimento dessas opções implementadas. Tratar a acessibilidade como uma prioridade vai impactar um público que realmente aprecia o esforço extra. No mundo da deficiência, pequenas mudanças podem fazer uma grande diferença (BARLET; TAYLOR, 2012). Os autores dividiram as diretrizes em três níveis: a Tabela 3.5 apresenta as diretrizes de nível 1 (mínimo de acessibilidade que deve ter no jogo), as diretrizes de nível 2 (melhor compromisso entre a necessidade acessibilidade e a facilidade

¹<http://www.ablegamers.com/>

Tabela 3.5: Diretrizes de nível 1 de Barlet e Taylor (2012).

Prática	Justificativa
Cores de texto alteráveis	Esta diretriz é voltada para pessoas que têm dificuldade em distinguir algumas cores e busca melhorar a funcionalidade do jogo de modo geral, gerando um conforto do jogador.
Tamanho de fontes alteráveis	<i>Gamers</i> com dificuldades visuais, incluindo aqueles que são quase totalmente cegos, podem participar de muitos jogos, mas a leitura do texto nas legendas, instruções e <i>chats</i> pode ser frustrante se o tamanho do texto é pequeno demais para ler.
Opção para daltônicos	É o recurso mais popular para quem tem problemas visuais. Idealmente, todos os jogos devem ter opções para daltônicos, que podem ser habilitadas para compensar uma variedade de problemas de deficiência de cor. Por curiosidade, uma abordagem utilizada baseia-se em mudar a cor verde para azul ao indicar informações, como companheiros de grupo ou barras de saúde.
Miras em alto contraste	Para jogos de tiro em primeira pessoa, é extremamente importante verificar se o jogador não tem dificuldade em usar a mira e se ela não é confundida com elementos do cenário. Um exemplo disto é quando alvo e inimigo são vermelhos, o que causa uma confusão na visão do jogador.
Marcação do inimigo	A capacidade de dizer amigavelmente sobre o inimigo é um dos diferenciais mais fundamentais em todos os jogos de 3D. Alguns jogos usam nomes, barras visuais, símbolos ou marcadores de cor.

de implementação) podem ser vistas na Tabela 3.6, e, por fim, podemos ver na Tabela 3.7 as diretrizes de nível 3 (demonstra como seria a acessibilidade num mundo ideal).

Tabela 3.6: Diretrizes de nível 2 de Barlet e Taylor (2012)

Prática	Justificativa
Fontes personalizáveis	Personalização da fonte não é uma nova tendência no desenvolvimento do jogo, mas o uso de fontes muito estilizadas tem um efeito negativo sobre os jogadores de baixa visão. A capacidade de trocar as fontes mais elaboradas por outra mais simples, como Arial ou Times New Roman, permite que o <i>gamer</i> de baixa visão interprete o texto mais facilmente.
<i>Head Up Display</i> personalizável	Para aqueles com problemas de visão, tais como deficiências de cor, baixa visão, visão em túnel e dificuldade de ver rapidamente as informações em movimento, a capacidade de organizar a interface do usuário conforme a preferência dos usuários finais ajuda a mitigar os problemas com a identificação de informações importantes em um relance. Geralmente é utilizado em jogos de simuladores (avião e carro).
Mapa de alteração de cores no mapa	Classificado como a progressão lógica a partir das opções para daltônicos estabelecidos no nível 1. Permite aos usuários colorir novamente um mapa, deixando de lado o tradicional vermelho e verde e dando espaço a novas cores, como o azul e o laranja. Isto permite que as pessoas com deficiências de visualização de cores possam fazer melhor uso da interface.

Tabela 3.7: Diretrizes de nível 3 de Barlet e Taylor (2012)

Prática	Justificativa
Ajustes de velocidade	Esta é uma opção de primeira linha para pessoas com baixa visão e distúrbios cognitivos, pois oferece possibilidade de retardar o relógio do jogo, permitindo que jogadores necessitados de mais tempo para processar o que está acontecendo possam reagir de maneira adequada em cada ação do jogo.
<i>Inputs Text-to-Speech</i>	Outra opção de acessibilidade bastante relevante é tornar o jogo capaz de ler o texto na tela e reproduzi-lo em forma de áudio. Muitos programas atuais fazem isso para os usuários na <i>web</i> com a inclusão de <i>tags</i> da WAI-ARIA.

3.4 Jogos inclusivos: diretrizes de acessibilidade para jogos digitais

O trabalho de mestrado desenvolvido por Cheiran e Pimenta (2013) na Universidade Federal do Rio Grande do Sul (UFRGS), Diretrizes de Acessibilidade para Jogos, por meio de uma análise de conteúdos preexistentes, apresentou uma unificação de diretrizes de acessibilidade em jogos digitais, gerando uma reestruturação delas. O autor menciona que o foco da sua abordagem é apresentar um conteúdo que possa ser classificado como “norte” na avaliação de acessibilidade em jogos.

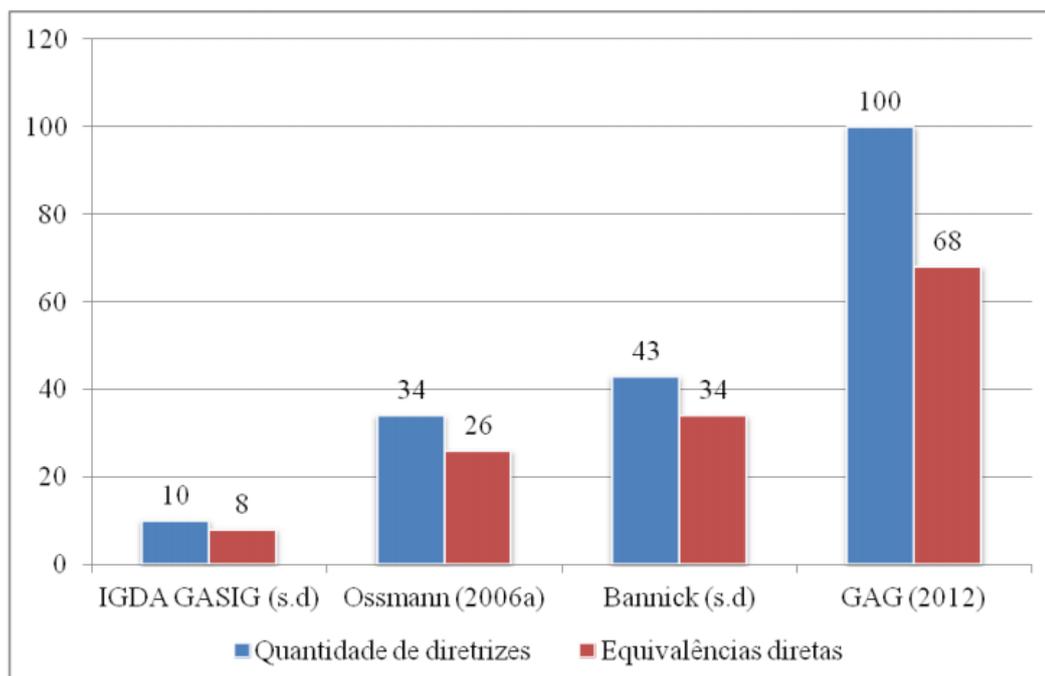
O autor buscou manter a mesma estrutura do documento WCAG 2.0, mantendo-se as diretrizes em quatro princípios fundamentais de acessibilidade. São elas:

1. Perceptível: Capacidade de apresentar as informações, o conteúdo e as interfaces de modos que os jogadores consigam identificar a presença da diretriz;
2. Operável: Garantir que todos os componentes presentes na interface, na interação e na navegação sejam facilmente operáveis;
3. Compreensível: Verificar onde as informações, o conteúdo e as interfaces com o jogador devem ser compreensíveis;
4. Robusto: Abordagem na qual o conteúdo apresentado na interface deve ser interpretado confiavelmente de uma forma simples e disponível em uma ampla variedade de dispositivos de saída, tais como tecnologias assistivas.

Com a extensão das doze diretrizes originais no documento WCAG 2.0, podem-se contemplar diversos aspectos em relação à interatividade e à plataforma dos jogos. Por formarem apenas a estrutura geral para compreensão dos critérios testáveis, esses aspectos tiveram ainda a finalidade de apresentar metas básicas para a implantação de acessibilidade sem o englobamento de testes.

Fazendo uma breve comparação entre as diretrizes propostas por Cheiran e Pimenta (2013) e as demais diretrizes revisadas neste Capítulo, podemos verificar que existem diretrizes semelhantes em sua base, mas com uma pequena alteração na abordagem. O autor também verificou isso, o que pode ser visto na Figura 3.1.

Para cada diretriz proposta pelo autor, foram mantidos e propostos critérios de teste que indicam como deve ser realizada a validação da diretriz segundo a finalidade para a qual ela é

Figura 3.1: Gráfico de equivalência de diretrizes (CHEIRAN; PIMENTA, 2013).

proposta, bem como o nível de conformidade com essa diretriz. O autor ainda dividiu os níveis de conformidade três categorias, medidas pelo esforço e pela dificuldade técnica de implantação. São elas: A (mais baixo), AA (intermediário) e AAA (nível mais alto).

Dentre as limitações do trabalho de Cheiran e Pimenta (2013), e apesar da vasta análise bibliográfica, é importante destacar a consistência das conclusões do autor perante uma necessidade real de acessibilidade, uma vez que os sujeitos da pesquisa foram seis alunos do curso de desenvolvimento de sistemas que realizaram a disciplina de Interface Homem-Máquina. Acreditamos que seria necessário essas diretrizes serem avaliadas por pessoas com deficiência ou por desenvolvedores com conhecimentos avançados em acessibilidade. Uma abordagem mais consistente para essa avaliação poderia acontecer em um ambiente onde o *player* jogasse com o acompanhamento do autor, e assim pudesse coletar o *feedback* em tempo-real, não aplicando questionários posteriores em que muitas informações do momento são perdidas.

As diretrizes podem ser encontradas *on-line* na página pessoal do autor².

²http://www.inf.ufrgs.br/~jfpcheiran/arquivos/diretrizes_de_acessibilidade_para_jogos.pdf

3.5 *Designing universally accessible games*

Grammenos, Savidis e Stephanidis (2009) em seu trabalho mencionam, a partir de seu ponto de vista, que foram adotadas duas abordagens principais à questão da acessibilidade jogos de computadores:

1. Jogos que intrinsecamente não são acessíveis, mas são considerados operacionalmente acessíveis com o uso de tecnologias assistivas de terceiros. Na prática, existem sérias barreiras inerentes e gargalos, porque não se empregam esforços suficientes para garantir a compatibilidade durante o desenvolvimento de jogos de computador e sistemas de tecnologia assistiva. No entanto, mesmo quando algum tipo de compatibilidade é alcançado, trata-se normalmente ou do resultado de adaptações de baixo nível personalizados ou de pura coincidência, e não do resultado de considerações de *design* apropriadas;
2. Jogos acessíveis que são desenvolvidos a partir do zero, mas direcionados a pessoas com uma deficiência específica, tais como jogos baseados em áudio para pessoas cegas ou jogos para pessoas com limitações motoras graves nos membros superiores.

Grammenos, Savidis e Stephanidis (2009) afirmam que, embora a segunda abordagem seja muito mais promissora, ela apresenta dois fatores que afetam negativamente essa prática: (i) há um custo significativo associado ao desenvolvimento de jogos acessíveis de alta qualidade, enquanto o retorno esperado do investimento é bastante baixo, uma vez que o grupo de jogadores representa um mercado limitado; (ii) e existe um risco de exclusão social devido ao potencial para a segregação entre jogadores com ou sem deficiência.

Por conta destes fatores, o trabalho apresentado pelos autores tem como objetivo enfatizar uma disciplina de desenvolvimento que pudesse superar as limitações das abordagens existentes. A fim de alcançar este objetivo, são necessárias duas novas características: (i) os jogos devem ser inerentemente acessíveis a todos os grupos de usuários potenciais, sem a necessidade de novos ajustes ou de aplicativos de *software* de apoio de terceiros; (ii) o mesmo jogo deve ser simultaneamente acessível e jogável, cooperativamente ou competitivamente, para pessoas com diferentes capacidades.

Os autores fizeram uso de quatro jogos como estudo de caso, para validarem suas diferentes propostas, abordagens e diretrizes. A experiência adquirida pelos autores no desenvolvimento dos jogos descritos em seu trabalho mostra que a concepção de jogos universalmente acessíveis pode ser uma tarefa exigente, mas ainda administrável e exequível, implicando um conjunto de instruções para seguir, como se vê abaixo:

- Desenvolver visando à independência de contexto em um nível abstrato, sem considerar modalidades específicas de interação, metáforas, técnicas ou dispositivos. Separar o conteúdo e os mecanismos relacionados de modo que possam ser acessados e apresentados ao usuário;
- Mapear os elementos de *design* abstratos para projetos coerentes, utilizáveis e de interação acessível, que são baseados nas características individuais dos usuários;
- Criar interfaces de usuário que possam apoiar métodos alternativos de interação e modalidades que possam coexistir e cooperar;
- Criar interfaces de usuário capazes de adaptar-se a perfis de usuário alternativos, ou seja, a conjuntos de preferências, requisitos, necessidades e contextos de uso;
- Prover um *design* inclusivo e participativo, ou seja, considerar a mais ampla população possível durante o projeto, contando com representantes de todas as categorias de acessibilidade e trabalhando juntamente com eles para colher *feedbacks* de todas as fases de desenvolvimento;
- Projetar baseando-se em conhecimento incompleto, como jogos podem atingir uma diversidade de público;
- Desenvolver uma interação aberta e extensível para que, mais tarde, ainda seja possível ampliar o projeto para atender a mais categorias de usuários e contextos de uso;
- Atender grupos de usuários atípicos, que podem não ter nada em comum com o desenvolvedor.

Embora a maioria dos exemplos apresentados pelos autores representem jogos 2D, o conceito e os princípios de jogos universais não estão vinculados a nenhuma tecnologia particular e a nenhum gênero de jogos. O processo de projeto proposto pelos autores pode auxiliar o projeto de qualquer tipo de jogo acessível, e as diretrizes de acessibilidade apresentadas podem ser aplicadas a todos os gêneros de jogos. Um exemplo disto é a variabilidade com que elas podem ser empregadas, adaptadas para o devido contexto, sendo possível fazer jogos acessíveis em primeira pessoa em 3D ou jogos acessíveis casuais no *smartphone*.

Os resultados obtidos pelos estudos de caso mostram que se podem obter melhorias no que se refere à acessibilidade e à usabilidade de jogos, com o emprego de um processo participativo de desenvolvimento centrado no usuário, processo esse que integra a avaliação de usabilidade. Considerando a acessibilidade formal e a avaliação de usabilidade, os autores concluíram que as

ferramentas disponíveis até a data do estudo eram inadequadas ou insuficientes. Os resultados que essas ferramentas fornecem são questionáveis e têm valor prático limitado. Contudo, essas ferramentas podem ser muito úteis como um ponto de partida da entrevista, ao solicitar-se aos participantes que justifiquem seus julgamentos. Considerando todos os fatores, como o domínio da acessibilidade nos jogos e a avaliação de usabilidade, trata-se atualmente de uma direção de pesquisa emergente ainda sub-explorada com reais necessidades, o que, no futuro próximo, deverá ganhar uma importância considerável.

3.6 HATEMILE: A biblioteca para gerar páginas *web* mais acessíveis

O HATEMILE (CRUZ; ESTOMBELO-MONTESCO, 2015) é uma biblioteca de código-fonte aberto que busca melhorar a acessibilidade dos *sites* e promete converter um código HTML em outro mais acessível.

Segundo Cruz e Estombelo-Montesco (2015), o HATEMILE foi desenvolvido utilizando o paradigma da Orientação a Objetos (OO), por ser ele suportado pelas linguagens em que a biblioteca foi feita. De acordo com os autores, seu trabalho possui versão para PHP, CoffeeScript, JavaScript, Java, Ruby e Python.

Os autores também destacam que a maioria dos problemas identificados pelo e-Mag 3.1 e pelo WCAG 2.0 não são solucionados pelo HATEMILE. Sendo assim, apresentam como soluções as seguintes funcionalidades:

- Disponibilizar todas as funções da página via teclado;
- Fornecer descrições para os campos;
- Fornecer instruções para a entrada de dados;
- Web Semântica e WAI-ARIA;
- Fornecer âncoras para ir direto a um bloco de conteúdo;
- Associar células de dados às células de cabeçalho;
- Exibir a lista de atalhos da página;
- Prover uma forma de navegação para a descrição longa de uma imagem;
- Promover a navegação entre os cabeçalhos da página.

Apesar de o artigo não fazer nenhuma referência a onde as bibliotecas são disponibilizadas, algumas versões delas podem ser encontradas no GitHub.

Para testes de concordância com este trabalho, utilizamos as versões em PHP e JavaScript que estavam disponíveis. O motivo da escolha dessas versões se dá pela familiaridade com essas linguagens de programação.

Lidando com os testes propriamente ditos, não houve como verificar se a biblioteca cumpre o que promete, uma vez que a documentação é ineficaz para terceiros, criando uma barreira para a sua utilização. Tanto na versão PHP quanto na JavaScript, é necessária uma configuração não trivial para que o HATEMILE interaja com a página, deixando pessoas inexperientes totalmente incapazes de utilizar a biblioteca, o que foi classificado como o principal problema do trabalho dos autores.

Este trabalho, no entanto, busca eliminar esse problema identificado, de modo a tornar o processo de integração trivial e simples, e não um empecilho para o desenvolvedor.

Ainda em relação ao desenvolvimento, a diversificação de linguagens disponíveis para o uso da biblioteca se mostra ineficaz devido a dificuldade de integração. Relacionando-a com o trabalho aqui apresentado, não é necessário disponibilizar diversas linguagens de programação, pois o JavaScript interage com as principais linguagens de programação *web*.

3.7 Considerações finais

Estudar, revisar e questionar as diretrizes de *websites* e de jogos acessíveis é uma abordagem muito importante na adaptação do desenvolvedor com um novo paradigma de desenvolvimento. Essa interação com as diretrizes é fulcral no projeto, porque elas fornecem um ponto de partida para o complexo processo de desenvolvimento acessível.

É relevante observar a individualidade de cada usuário e como ele interage com as aplicações *on-line*. As diretrizes não são apenas ideias lançadas a partir do que se acredita melhorar a interação, mas são resultados obtidos por estudos focados na melhora da interação de um usuário com o ambiente. Independentemente de como as diretrizes são obtidas e de como são consideradas obrigatórias ou opcionais, não se devem restringir outras abordagens que visam a incluir o maior número possível de pessoas com a possibilidade de experimentar uma aplicação *on-line* acessível.

Ainda, é importante destacar que nenhuma das diretrizes, se for implementada, poderá de alguma forma afetar negativamente a mecânica ou o *layout* da aplicação. Elas têm como

fundamento apenas a integração de pessoas com deficiência, fornecendo opções para que os usuários interajam com o jogo da forma que lhes for mais agradável.

O capítulo seguinte destina-se ao resultado deste trabalho, batizado como ARIA-ACCESS, e apresenta a visão geral do *framework*, sua propriedades, a documentação e exemplos de uso.

Capítulo 4

ARIA-ACCESS – FRAMEWORK DE DESENVOLVIMENTO ACESSÍVEL

Neste capítulo, apresentamos todo o contexto necessário para o processamento automático da aplicação, para que possa ser usada pelo público com necessidades especiais. Na Seção 4.1, apresentamos a visão geral do ARIA-ACCESS; na Seção 4.3, a arquitetura deste trabalho; na Seção 4.4, uma definição da linguagem de domínio; na Seção 4.5, a descrição dos conceitos da documentação do *framework*; na Seção 4.6, exemplos de utilização. Ainda, as diretrizes aplicáveis para este trabalho e a metodologia de aplicação das mesmas podem ser visualizadas na Seção 4.7 e, por fim, na Seção 4.8 as considerações finais.

4.1 Visão geral

A abordagem empregada para o processamento de uma aplicação *on-line* a fim de torná-la acessível produziu um *framework* chamado ARIA-ACCESS. Este *framework* tem como principais objetivos abstrair as *tags* e as marcações da WAI-ARIA no código HTML do desenvolvedor, sendo necessário apenas que ele importe o código no projeto e indique os elementos que julgar devam ser acessíveis. Essa ferramenta busca tornar acessível uma aplicação sem que o desenvolvedor tenha de estudar a documentação WCAG e as marcações da WAI-ARIA, visando poupar tempo para os desenvolvedores.

Ao analisar ferramentas de desenvolvimento de aplicações *web* acessíveis, encontramos basicamente validadores de acessibilidade que se baseiam na documentação da WCAG 2.0, apresentada na Seção 2.3. Essas ferramentas têm como finalidade validar o *website* nos padrões da documentação e, conseqüentemente, fornecer o selo de acessibilidade.

Tais selos são fornecidos por validadores existentes, que são ferramentas válidas para programadores com experiência em desenvolvimento acessível e programadores que querem verificar se sua abordagem está sendo fidedigna com a documentação WCAG 2.0. Atualmente, não se encontra disponível nenhuma ferramenta que forneça um ambiente de desenvolvimento, nenhuma ferramenta de recomendação ou um *framework* de abstração de código.

O ARIA-ACCESS pode ser acessado no *link*: http://github.com/pablobizzi/plugin_acessivel/.

4.2 Metodologia

A metodologia aplicada neste trabalho buscou inicialmente analisar o perfil do desenvolvedor que utilizará o ARIA-ACCESS, baseando-se em conhecimentos empíricos de desenvolvimento e recomendações advindas dos trabalhos apresentados no Capítulo 3, e então, foi desenvolvido o *framework*.

Uma vez desenvolvido o *framework*, foi realizada a validação com desenvolvedores e a coleta do *feedback* dos participantes acerca da eficácia do ARIA-ACCESS. A partir das definições citadas por Sommerville (2007), o método mais relevante que permite uma análise cuidadosa e abrangente para isso, é a utilização de questionários ou entrevistas.

A técnica de aplicação de questionários utilizada foi desmembrada em dois grupos, uma para os desenvolvedores que receberam instrução de uso do ARIA-ACCESS, e outra para o grupo que não recebeu instruções. O questionário foi elaborado com perguntas subjetivas, onde procura-se captar informações relevantes para a boa funcionalidade e utilização do *framework*. Os questionários podem ser visualizados no Apêndice A.

4.3 Arquitetura

O princípio da aplicação consiste em fornecer um ambiente automatizado para o desenvolvedor, abstraindo conhecimentos específicos e linhas de código repetitivas, para tornar uma aplicação acessível. Esse processo, apresentado na Figura 4.1, é composto pelos passos (i), (ii) e (iii).

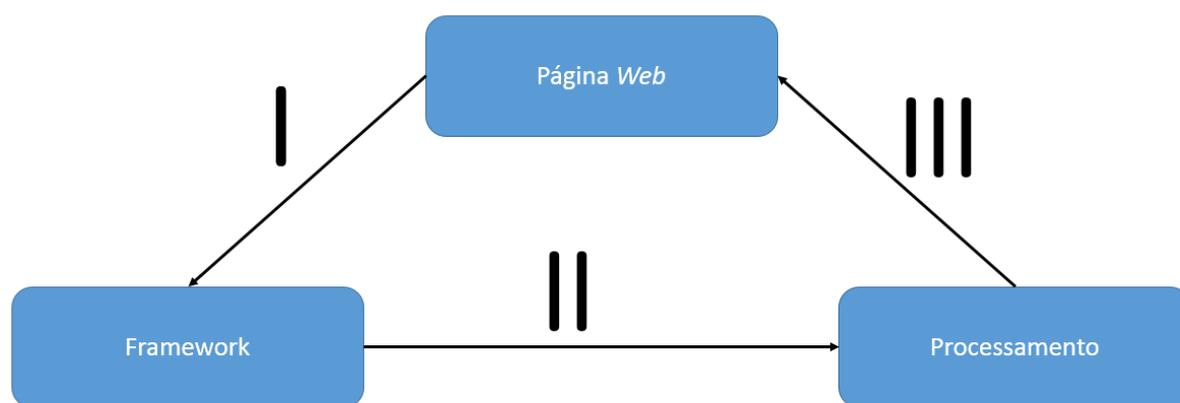
O passo (i) consiste na importação do *framework* ao projeto HTML. Este passo é o que identifica quais elementos deverão ser processados pelo *plugin*.

O passo (ii) é acionado no carregamento da página HTML e fundamenta-se em processar

os elementos sinalizados como acessíveis pelo desenvolvedor. Os elementos identificados são processados e adaptado para o padrão do WAI-ARIA.

Por fim, o passo (iii) manifesta-se a partir do passo anterior, onde o HTML gerado pelo ARIA-ACCESS substitui os elementos da página em tempo de execução. Em outras palavras, via JavaScript, o *framework* identifica o elemento sinalizado pelo desenvolvedor, classifica-o no comportamento correspondente e substitui o HTML para a visualização do usuário.

Figura 4.1: Arquitetura proposta do *framework*.



A utilização do ARIA-ACCESS é semelhante à do Twitter Bootstrap, e funciona como um *plugin* no qual não é necessário o desenvolvedor ter conhecimentos de *templates* responsivos, bastando apenas que ele conheça as funcionalidades do *framework* e utilize as marcações corretas para que seu *site* seja automaticamente convertido para um *template* responsivo.

4.4 Linguagem de domínio específico

Uma linguagem de domínio específico são os fundamentos e funções, ou códigos específicos da plataforma, de uma linguagem de programação ou linguagem de domínio para a utilização de um determinado software. Esta linguagem se dedica a um domínio de problema particular, uma técnica de representação de problema ou uma técnica de solução particular (FOWLER, 2010).

Para garantir a funcionalidade do ARIA-ACCESS, foi estabelecida uma linguagem de domínio específica que serve para indicar os elementos que podem ser tornados acessíveis em tempo de execução. Essa linguagem é utilizada para sinalizar, via *tag class* do HTML, o elemento a ser processado.

Outra funcionalidade dessa linguagem é diferenciar nas *tags* o comportamento específico do elemento acessível. Podem-se verificar a seguir as *tags* que compõem o ARIA-ACCESS,

bem como as suas aplicações:

- *class=“menu-acessivel”*: Classe utilizada para identificar menus, ordenados ou não, da aplicação;
- *class=“botao-acessivel”*: Serve para identificar botões, independentemente de estarem sinalizados como elemento *button* ou *div*;
- *class=“texto-acessivel”*: Empregado para a identificação de textos e cabeçalhos, podendo ser identificado dentro de qualquer elemento HTML que modifique o comportamento textual, como *p*, *div*, *span*, *h1*, [...], *h6*, etc.;
- *class=“imagem-acessivel”*: Empregada para a identificação de imagens. Neste modelo, a *tag alt* é usada para apresentar ao leitor de texto acessível o conteúdo da imagem;
- *class=“tabela-acessivel”*: As tabelas também foram inseridas neste trabalho. Com a utilização desta *tag*, o conteúdo textual das tabelas pode ser acessado, manipulado e lido via teclado;
- *class=“form-acessivel”*: Os formulários, com seus respectivos elementos *inputs* (*text*, *password*, *e-mail*, rádio, etc.), *textareas*, *select*, *buttons* são suportados por este trabalho mediante a utilização da sua *tag*;
- *class=“acessivel”*: Esta *tag* é utilizada para identificar elementos genéricos que não estão presentes na lista de elementos suportados na primeira versão deste trabalho.

Todos os elementos apresentados nesta Seção devem ser usados mediante a padronização necessária que compõe a documentação oficial do ARIA-ACCESS.

4.5 Documentação

Sempre que tratamos de apresentar novas ferramentas e novas abordagens de desenvolvimentos em trabalhos acadêmicos, deparamo-nos com a necessidade de justificar o porquê dessa nova abordagem e como deve ser realizada a integração dessa ferramenta com as práticas de desenvolvimento que estamos habituados a praticar.

No caso específico do ARIA-ACCESS, considera-se necessário disponibilizar a documentação de forma precisa, por tratar-se de uma ferramenta adicional no desenvolvimento de aplicações acessíveis, e não obrigatória.

Busca-se, com a documentação, justificar os motivos que levam desenvolvedores a preferir as *tags* convencionais do WAI-ARIA às do ARIA-ACCESS, seja em busca de agilidade, de praticidade ou de facilidade de uso.

Além disso, a documentação tentou ser clara, usando exemplos de código-fonte para tornar a integração o mais visual possível, e indicando ainda os passos de importação de bibliotecas adicionais para o funcionamento do *framework*.

Também é possível encontrar os elementos incorporados ao *plugin* que são classificados como acessíveis, bem como exemplos de código de cada um.

Tal documentação pode ser verificada no *link*: http://github.com/pablobizzi/plugin_acessivel/wiki/Documentation.

4.6 Exemplo de uso

Como o ARIA-ACCESS trabalha com geração dinâmica de código sob o HTML da página, executando suas dependências após o carregamento da página e em tempo de execução, esclarecemos nas Seções a seguir a diferença entre o código-fonte do projeto com a linguagem e o código gerado pelo *framework*.

4.6.1 Texto, Botão, *Link* e Menu

Na Figura 4.2, ilustramos como usamos os elementos textuais, de botões, *links* e menus — com as classes referenciando as dependências do ARIA-ACCESS —, os quais são os elementos mais básicos de uma página HTML.

Considerando que não ocorram problemas de programação por parte do desenvolvedor ou erros na importação da biblioteca auxiliar, obteremos um código acessível semelhante ao encontrado na Figura 4.3.

4.6.2 Formulários e imagens

Quando refletimos sobre formulários e imagens em uma aplicação *web*, tratamos de elementos complexos e essenciais em qualquer interação mais robusta, principalmente se considerarmos uma aplicação dinâmica com *backend*. Utilizando os elementos supracitados com as dependências do ARIA-ACCESS, teremos um código semelhante ao que pode ser visto na Figura 4.4.

Figura 4.2: Código HTML de texto, botão, *link* e menu com classes que referenciam o ARIA-ACCESS.

```

<div class="acessivel"> Componente Genérico Acessível </div>
<div class="teste acessivel"> Teste com mais de uma propriedade no Class </div>
<button class="botao-acessivel">Teste de Botão</button>

<ul class="menu-acessivel">
  <li>
    <a href="#tab-panel1">Primeira Aba</a>
  </li>
  <li class="item-menu-acessivel">
    <a href="#tab-panel2">Segunda Aba</a>
  </li>
  <li class="item-menu-acessivel">
    <a href="#tab-panel3">Terceira Aba</a>
  </li>
</ul>

<ul class="menu-acessivel">
  <a href="#tab-panel1">Ítem 1</a>
  <a href="#tab-panel1">Ítem 2</a>
  <a href="#tab-panel1">Ítem 3</a>
</ul>

<div class="texto-acessivel">
  Teste de texto com <a href="#" class="link-acessivel">Link acessível</a>
</div>

```

Figura 4.3: Código gerado de texto, botão, *link* e menu pelas dependências do ARIA-ACCESS.

```

<div class="acessivel" aria-label=" Componente Genérico Acessível " tabindex="1"> Componente Genérico Acessível </div>
<div class="teste acessivel" aria-label=" Teste com mais de uma propriedade no Class " tabindex="1"> Teste com mais de uma propriedade no Class </div>
<button class="botao-acessivel" aria-label="Teste de Botão" tabindex="1" role="button">Teste de Botão</button>

<ul class="menu-acessivel" aria-label="Ítems listados em um menu acessível" tabindex="1" role="menubar">
  <li>
    <a href="#tab-panel1" aria-label="Primeira Aba" tabindex="1" role="menuitem">Primeira Aba</a>
  </li>
  <li class="item-menu-acessivel">
    <a href="#tab-panel2" aria-label="Segunda Aba" tabindex="1" role="menuitem">Segunda Aba</a>
  </li>
  <li class="item-menu-acessivel">
    <a href="#tab-panel3" aria-label="Terceira Aba" tabindex="1" role="menuitem">Terceira Aba</a>
  </li>
</ul>

<ul class="menu-acessivel" aria-label="Ítems listados em um menu acessível" tabindex="1" role="menubar">
  <a href="#tab-panel1" aria-label="Ítem 1" tabindex="1" role="menuitem">Ítem 1</a>
  <a href="#tab-panel1" aria-label="Ítem 2" tabindex="1" role="menuitem">Ítem 2</a>
  <a href="#tab-panel1" aria-label="Ítem 3" tabindex="1" role="menuitem">Ítem 3</a>
</ul>

<div class="texto-acessivel" tabindex="1" role="article" aria-label="Teste de texto com Link acessível">
  Teste de texto com <a href="#" class="link-acessivel" aria-label="Link acessível" tabindex="1" role="article">Link acessível</a>
</div>

```

Novamente considerando que não ocorram nem problemas de programação por parte do desenvolvedor nem erros na importação da biblioteca auxiliar, obteremos um código acessível como o da Figura 4.5.

Figura 4.4: Código HTML de imagem e formulário com classes que referenciam o ARIA-ACCESS.

```



<form class="form-acessivel" action="">
  <input name="teste" type="text" placeholder="Nome">
  <input name="teste1" type="email" placeholder="E-mail">
  <input name="teste2" type="password" placeholder="Senha">
  <textarea name="textarea" placeholder="Digite sua Mensagem"></textarea>
  <select name="select">
    <option value="1">Opção 1</option>
    <option value="2">Opção 2</option>
    <option value="3">Opção 3</option>
  </select>
  <input name="teste3" type="radio" placeholder="Homem">
  <input name="teste4" type="radio" placeholder="Mulher">
  <input name="teste5" type="checkbox" placeholder="Homem">
  <input name="teste6" type="checkbox" placeholder="Mulher">
  <button type="reset">Resetar</button>
  <button type="submit">Enviar</button>
</form>

```

Figura 4.5: Código gerado de imagem e formulário pelas dependências do ARIA-ACCESS.

```



<form class="form-acessivel" action="">
  <input name="teste" type="text" placeholder="Nome" aria-label="Nome" tabindex="1">
  <input name="teste1" type="email" placeholder="E-mail" aria-label="E-mail" tabindex="1">
  <input name="teste2" type="password" placeholder="Senha" aria-label="Senha" tabindex="1">
  <textarea name="textarea" placeholder="Digite sua Mensagem" aria-label="Digite sua Mensagem" tabindex="1"></textarea>
  <select name="select" aria-label="Selecione uma opção" tabindex="1">
    <option value="1" aria-label="Opção 1" tabindex="1">Opção 1</option>
    <option value="2" aria-label="Opção 2" tabindex="1">Opção 2</option>
    <option value="3" aria-label="Opção 3" tabindex="1">Opção 3</option>
  </select>
  <input name="teste3" type="radio" placeholder="Homem" aria-label="Homem" tabindex="1">
  <input name="teste4" type="radio" placeholder="Mulher" aria-label="Mulher" tabindex="1">
  <input name="teste5" type="checkbox" placeholder="Homem" aria-label="Homem" tabindex="1">
  <input name="teste6" type="checkbox" placeholder="Mulher" aria-label="Mulher" tabindex="1">
  <button type="reset" aria-label="Resetar" tabindex="1" role="button">Resetar</button>
  <button type="submit" aria-label="Enviar" tabindex="1" role="button">Enviar</button>
</form>

```

4.6.3 Tabelas

Talvez o elemento HTML mais complexo de tornar acessível — segundo as recomendações dos autores citados neste trabalho e considerando a recomendação de que o elemento deve ser acessível e manipulado pelo teclado — sejam as tabelas do ARIA-ACCESS, que então receberam o suporte das setas direcionais, e estas, por sua vez, cores destacadas para a sua manipulação. Utilizando o elemento com as dependências do ARIA-ACCESS, teremos um código semelhante ao da Figura 4.6.

Sempre considerando a ausência de problemas de programação por parte do desenvolvedor e de erros na importação da biblioteca auxiliar, obtém-se um código acessível como o da Figura 4.7

Figura 4.6: Código HTML de tabela com classes que referenciam o ARIA-ACCESS.

```
<table class="tabela-acessivel">
  <thead>
    <tr>
      <th>Título da Tabela 1</th>
      <th>Título da Tabela 2</th>
      <th>Título da Tabela 3</th>
      <th>Título da Tabela 4</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Linha 1 coluna 1</td>
      <td>Linha 1 coluna 2</td>
      <td>Linha 1 coluna 3</td>
      <td>Linha 1 coluna 4</td>
    </tr>
    <tr>
      <td>Linha 2 coluna 1</td>
      <td>Linha 2 coluna 2</td>
      <td>Linha 2 coluna 3</td>
      <td>Linha 2 coluna 4</td>
    </tr>
    <tr>
      <td>Linha 3 coluna 1</td>
      <td>Linha 3 coluna 2</td>
      <td>Linha 3 coluna 3</td>
      <td>Linha 3 coluna 4</td>
    </tr>
    <tr>
      <td>Linha 4 coluna 1</td>
      <td>Linha 4 coluna 2</td>
      <td>Linha 4 coluna 3</td>
      <td>Linha 4 coluna 4</td>
    </tr>
  </tbody>
</table>
```

4.7 Diretrizes selecionadas e aplicação

Baseando-nos nos estudos de diretrizes de acessibilidade — que podem ser encontrados no Capítulo 3 — e nas recomendações da WCAG — Seção 2.3 —, e considerando que as diretrizes apresentadas são abrangentes no que se refere ao desenvolvimento de conteúdos acessíveis, identificou-se a necessidade de filtrar as mesmas de modo que fosse possível a aplicabilidade e implementabilidade neste trabalho. Além da coleta de diretrizes, este trabalho propõe a abstração de linhas de código por meio da utilização do ARIA-ACCESS com o desenvolvimento do ARIA-ACCESS.

Considerando que o *framework* tem como objetivo principal atender diretrizes de acessibilidade que possam ser adequadas neste trabalho que se justifiquem sua utilização no escopo do trabalho, que é abstrair a utilização de *tags* WAI-ARIA no código HTML e fornecer a integração necessária com leitores de tela, e discorrendo ainda sobre as diretrizes que são aplicáveis no escopo de desenvolvimento acessível na *web*, mas que necessitam de conhecimento específico

Figura 4.7: Código gerado de tabela pelas dependências do ARIA-ACCESS.

```
<table class="tabela-acessivel">
  <thead>
    <tr>
      <th aria-label="Título da Tabela 1" tabindex="1">Título da Tabela 1</th>
      <th aria-label="Título da Tabela 2" tabindex="1">Título da Tabela 2</th>
      <th aria-label="Título da Tabela 3" tabindex="1">Título da Tabela 3</th>
      <th aria-label="Título da Tabela 4" tabindex="1">Título da Tabela 4</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td aria-label="Linha 1 coluna 1" tabindex="1">Linha 1 coluna 1</td>
      <td aria-label="Linha 1 coluna 2" tabindex="1">Linha 1 coluna 2</td>
      <td aria-label="Linha 1 coluna 3" tabindex="1">Linha 1 coluna 3</td>
      <td aria-label="Linha 1 coluna 4" tabindex="1">Linha 1 coluna 4</td>
    </tr>
    <tr>
      <td aria-label="Linha 2 coluna 1" tabindex="1">Linha 2 coluna 1</td>
      <td aria-label="Linha 2 coluna 2" tabindex="1">Linha 2 coluna 2</td>
      <td aria-label="Linha 2 coluna 3" tabindex="1">Linha 2 coluna 3</td>
      <td aria-label="Linha 2 coluna 4" tabindex="1">Linha 2 coluna 4</td>
    </tr>
    <tr>
      <td aria-label="Linha 3 coluna 1" tabindex="1">Linha 3 coluna 1</td>
      <td aria-label="Linha 3 coluna 2" tabindex="1">Linha 3 coluna 2</td>
      <td aria-label="Linha 3 coluna 3" tabindex="1">Linha 3 coluna 3</td>
      <td aria-label="Linha 3 coluna 4" tabindex="1">Linha 3 coluna 4</td>
    </tr>
    <tr>
      <td aria-label="Linha 4 coluna 1" tabindex="1">Linha 4 coluna 1</td>
      <td aria-label="Linha 4 coluna 2" tabindex="1">Linha 4 coluna 2</td>
      <td aria-label="Linha 4 coluna 3" tabindex="1">Linha 4 coluna 3</td>
      <td aria-label="Linha 4 coluna 4" tabindex="1">Linha 4 coluna 4</td>
    </tr>
  </tbody>
</table>
```

de outras linguagens, como JavaScript, visamos apenas indicar que é possível fornecer a funcionalidade descrevendo brevemente o desenvolvimento, não excluindo a possibilidade dessas funcionalidades compor o ARIA-ACCESS em trabalhos futuros.

Para apresentar essas diretrizes, utilizaremos tabelas indicando a diretriz selecionada, sua descrição, sua aplicabilidade e justificativa neste trabalho. Pode-se verificar as diretrizes presentes na versão atual do ARIA-ACCESS a seguir.

4.7.1 Diretrizes aplicáveis - *Game accessibility guidelines*

Nesta Seção, encontra-se a seleção das diretrizes da Seção 3.1 para incorporar este trabalho.

A metodologia usada para selecionar as diretrizes é entender se podem ser empregadas em código fonte JavaScript, HTML ou CSS equivalente. Podem-se conferir na Tabela 4.1 as diretrizes selecionadas.

Tabela 4.1: Diretrizes selecionadas - *Game accessibility guidelines*

Código	Prática	Descrição	Prática	Justificativa
1.1	Garantir que não haja informação essencial transmitida por uma só cor	As cores são os meios mais comuns para comunicar-se com significados já estabelecidos. No entanto, esses significados podem variar em algumas culturas ou podem perder-se com pessoas que não conseguem distinguir certas cores.	Não atendido.	Pode-se desenvolver uma função em JavaScript que capture o elemento pelo seu ID e o modifique para a cor selecionada.
1.2	Utilizar um tamanho padrão de fonte que seja de fácil leitura	Fontes pequenas são uma queixa muito comum entre <i>gamers</i> , isso afeta a experiência de pessoas com tela pequena ou pessoas com problemas médicos (como a dislexia).	Atendido.	Todas as fontes têm o padrão de tamanho de 14px.
1.3	Usar uma simples e clara formatação de texto	Utilizar uma fonte de fácil leitura sobre um fundo não exigente tem um impacto muito positivo para a legibilidade.	Atendido.	Utiliza-se a fonte Courier por ter uma largura fixa.
1.4	Fornecer alto contraste entre texto e fundo	Baixo contraste é uma queixa comum entre usuários mais velhos e pessoas que tem problemas situacionais, como luz solar direta na tela ou tela de má qualidade.	Não atendido.	Esta é uma tarefa do desenvolvedor, que deve verificar a cor de fonte que melhor se aplica ao <i>background</i> .
1.5	Desenvolver elementos interativos grandes e bem espaçados	Um problema comumente encontrado em jogos é que os botões são muito pequenos e pouco espaçados. Pessoas com insuficiência de precisão encontram muitas dificuldades em interagir com esses sistemas.	Atendido.	Botões, tabelas, formulários e outros elementos presentes no ARIA-ACCESS têm espaçamento padrão que facilita a leitura da fonte.

1.6	Possibilitar a desabilitação da movimentação de fundo	Em algumas situações, as animações de fundo podem distrair o jogador e afetar as suas condições cognitivas.	Não atendido.	Basta o desenvolvedor usar a <i>tag disabled</i> .
1.7	Garantir a leitura de tela para dispositivos móveis	O leitor de tela é o meio que possibilita a interação de pessoas com pouca ou nenhuma visão com o jogo.	Atendido.	Utilizando os padrões WAI-ARIA, é possível que a aplicação garanta a interação com dispositivos móveis.
1.8	Permitir o ajuste do contraste	Assim como vimos na Tabela 3.1, a função de ajustar o contraste impactaria positivamente a experiência dos <i>gamers</i> .	Não atendido.	Esta é uma tarefa do desenvolvedor, que deve fornecer uma aplicação que permita alterar as cores dos elementos na tela, podendo usar eventos do JavaScript.
1.9	Certificar-se que cada objeto-chave tenha um som bem definido e distinto de outros	Esse recurso faz-se necessário para pessoas com certa dificuldade em diferenciar objetos através outros meios não sonoros.	Não atendido.	Tarefa do desenvolvedor, que deve configurar o som devidamente nos eventos da aplicação.
1.10	Permitir a troca de estilo de cursor, cores e mira de objetos	Os <i>gamers</i> frequentemente encontram problemas com o modelo do cursor e de suas cores, muitas vezes causados pelo contraste do cenário. Permitir a troca desses elementos torna o jogo mais agradável ao usuário.	Parcialmente atendido.	É possível trocar o cursor usando <i>tags</i> do ARIA-ACCESS.

1.11	Dar uma clara indicação de quais elementos são interativos	Jogadores com deficiências cognitivas ou visuais podem ter dificuldade em distinguir quais elementos ou itens do jogo são interativos.	Atendido.	Botões, tabelas, formulários e outros elementos presentes no ARIA-ACCESS ganham destaque na tela, facilitando a interação.
1.12	Garantir que o manual fornecido possa ser lido por leitores de tela	Garantir que os manuais, FAQs e listas de recursos, são fornecidos de uma forma que pode ser acessado por leitores de tela é muito mais importante do que disponibilizar um PDF, por exemplo.	Atendido.	Utilizando-se os padrões WAI-ARIA, é possível que a aplicação garanta a interação com qualquer texto identificado com a <i>tag</i> “ <i>texto-acessível</i> ”.
1.13	Fornecer controles de volume separados para efeitos, fala e fundo	Muitas fontes de informação diferentes podem dificultar a concentração em qualquer um deles. Há inclusive uma condição específica em que os sons simultâneos podem ser impossíveis de distinguir, causando muito desconforto ao jogador.	Não atendido.	Tarefa do desenvolvedor, que deve configurar os sons e as cores devidamente nos eventos da aplicação.
1.14	Permitir o redimensionamento das interfaces	Além de ser uma forma importante para atender os diferentes tamanhos de tela, o redimensionamento manual também é útil para pessoas com dificuldade de enxergar pequenos elementos ou sem a capacidade motora necessária para interagir com eles.	Não atendido.	Esta é uma tarefa do desenvolvedor, que deve fornecer uma aplicação que permita alterar as dimensões dos elementos na tela, podendo utilizar eventos do JavaScript.

1.15	Permitir que o tamanho da fonte seja ajustado	Embora definir um tamanho padrão seja benéfico, não há um tamanho único que todos possam ler, devido à diferenças nos níveis de deficiência visual, às diferenças de tamanho da tela ou à distância de visualização.	Não atendido.	Esta é uma tarefa do desenvolvedor, que deve fornecer uma aplicação que permita alterar o tamanho da fonte dos elementos na tela, podendo utilizar eventos do JavaScript.
1.16	Fornecer um mapa que informe objetos importantes	Um mapa que pode ser acionado para alertar sobre objetos importantes ou inimigos através de um pulso estilo sonar.	Parcialmente atendido.	Esta é uma tarefa do desenvolvedor, que deve fornecer o mapa para seus usuários. Contudo, as <i>tags</i> do ARIA-ACCESS dão acesso aos usuários para fazerem a leitura da tela.
1.17	Disponibilizar du- blagens para todo o texto, incluindo menus e instaladores	É uma opção mais complexa, uma alternativa para o apoio à leitura de tela.	Não atendido.	Esta é uma tarefa do desenvolvedor, que deve fornecer uma aplicação que faça a dublagem dos elementos na tela, podendo utilizar eventos do JavaScript.

1.18	Apresentar um <i>GPS</i> orientado por voz	Mapas orientados por áudio, podem ser usados como uma ferramenta para as pessoas com pouca ou nenhuma visão residual para navegar em ambientes 3D complexas.	Atendido.	O desenvolvedor deve fornecer o mapa para seus usuários. Assim, as <i>tags</i> do ARIA-ACCESS dão acesso aos usuários para fazerem a leitura da tela.
1.19	Permitir fácil orientação de movimento ao longo dos pontos cardeais	Permitindo aos jogadores saltar diretamente para um ponto cardeal da bússola, auxilia jogadores com pouca ou nenhuma visão.	Atendido.	No ARIA-ACCESS, o usuário pode movimentar uma tabela, por exemplo, com as setas direcionais do teclado, o que facilita a iteração.
1.20	Desenvolver menus e instaladores que possam interagir com leitores de tela	Entende-se que, quanto mais elementos nos jogos possam ser lidos por leitores de telas, mais independentes os jogadores poderão ser.	Atendido.	As <i>tags</i> do ARIA-ACCESS dão acesso aos usuários para fazerem a leitura da tela.
1.21	Usar sons e músicas distintas para todos os objetos e eventos	Um bom <i>design</i> de som, garantindo que cada ação distinta tenha um som igualmente distinto, pode tornar jogos até mesmo tradicionais completamente acessíveis a jogadores com pouca ou nenhuma visão.	Não atendido.	Tarefa do desenvolvedor, que deve configurar os sons e as cores devidamente nos eventos da aplicação.

É importante destacar que algumas diretrizes não selecionadas podem ser aplicadas no desenvolvimento de jogos e *websites* acessíveis; o que as exclui dessa seleção é o fato de serem mais características de *game design* ou de produção de conteúdo (como sons e imagens) do que propriamente linhas de código do desenvolvedor.

4.7.2 Diretrizes aplicáveis – IGDA

Nesta Seção, encontram-se as diretrizes da Seção 3.2 que foram selecionadas para incorporar este trabalho. A metodologia de escolha das diretrizes foi a mesma indicada anteriormente, a saber, entender se podem ser empregadas em código fonte JavaScript, HTML ou CSS equivalente. Podem-se conferir na Tabela 4.2 as diretrizes selecionadas.

Tabela 4.2: Diretrizes selecionadas - IGDA

Código	Prática	Descrição	Prática	Justificativa
2.1	Alta visibilidade gráfica	É uma alternativa para fontes pequenas ou de difícil distinção que disponibiliza esquemas de cores de alto contraste ou os disponibiliza como uma opção, caso não seja padrão. Essa técnica destaca o que é importante no cenário para pessoas com baixa visão e pode, ainda, oferecer a desativação ou o escurecimento de fundo nos jogos 2D.	Atendido.	O mesmo dos itens 1.2 e 1.3.
2.2	Design harmônico para daltônicos	Evitar combinações de cores que são difíceis ou impossíveis de distinguir a daltônicos (por exemplo, vermelho em cinza ou verde), ou oferecer opções alternativas no jogo onde essas cores não sejam utilizadas.	Não atendido.	O mesmo do item 1.4.
2.3	Fornecer diversos níveis de dificuldade e ajuste de velocidade	Permitir que os jogadores possam escolher entre uma vasta gama de dificuldades e velocidades do jogo, deixando que ele tenha o discernimento de quando está muito fácil ou muito lento.	Não atendido.	Esta é uma tarefa do desenvolvedor, que deve fornecer uma aplicação que permita alterar as funcionalidades da aplicação, podendo utilizar eventos do JavaScript.

2.4	Menus acessíveis	Disponibilizar modos de início rápido, prezar pela navegação e fornecer textos alternativos de entrada, como <i>text-to-speech</i> e símbolos. Em um jogo com uma interface complexa, é relevante desenvolver uma interface simplificada que exiba apenas os controles comuns.	Atendido.	O mesmo do item 1.20.
2.5	Listagem de recursos de acessibilidade, opções e requisitos do jogo	Fazer esforços para assegurar que essas informações sejam livres e fáceis de obter e entender. Elas podem ser disponibilizadas no <i>site</i> ou na embalagem do jogo. É importante enviá-las para revisão de <i>sites</i> de avaliação de jogos acessíveis.	Atendido.	O desenvolvedor deve fornecer a lista de acessibilidade para seus usuários. Assim, as <i>tags</i> do ARIA-ACCESS dão acesso para os usuários fazerem a leitura da tela.
2.6	Capacidade de auto-vocalização	É a capacidade de fornecer uma leitura do texto que está sendo exibido na tela. Já há uma variedade de ferramentas de <i>software</i> que fornecem esse recurso e podem ser integradas aos jogos. A solução <i>text-to-speech</i> deve trabalhar com APIs, como o API Speech da Microsoft. Esse texto pode estar escrito em diversas línguas e pode ser lido corretamente por sintetizadores de voz.	Não atendido.	O mesmo do item 1.17.
2.7	Navegação pelo teclado de todos os controles, com <i>feedback</i> visual e falado	Permitir que todos os comandos possam ser inseridos através do teclado. Também fornecer uma mensagem visual e auditiva para indicar o que foi feito.	Atendido.	O mesmo do item 1.19.

2.8	Melhores tutoriais em tempo-real	Orientar o jogador além do esperado, fornecendo <i>feedbacks</i> e ensinando-lhe melhores abordagens. Fornecer ao jogador indicadores de objetivos ou seta direcional, evitando que jogadores se sintam perdidos.	Não atendido.	O mesmo do item 1.17.
2.9	Possibilidade de definir a cor da unidade	É a capacidade de controlar a cor das diferentes unidades, como inimigos, companheiros de equipe e outras unidades importantes dentro de um jogo.	Atendido.	O desenvolvedor pode utilizar <i>tags</i> já desenvolvidas no ARIA-ACCESS para destacar os elementos em tela, pode alterar as informações como desejar ou pode desenvolver uma aplicação que permita que o usuário faça as alterações.
2.10	Áudio GPS	Em um jogo acessível para cegos, um sistema de posicionamento global pode ser usado para obter as posições exatas dos objetos num espaço, bem como a posição do avatar. Um sistema de menu sonoro pode fornecer uma visão geral de objetos próximos.	Não atendido.	Tarefa do desenvolvedor, que deve fornecer em tempo real as informações de onde o usuário está e o que pode ser feito.

2.11	Compasso de Som	Um som 3D original ou comentários de voz e uma tecla no teclado representam cada sentido em um período de oito direções no mundo do jogo. Quando o jogador aperta a tecla que representa a direção, o áudio toca para mostrar onde está direção, em relação ao avatar, que ele selecionou.	Não atendido.	O mesmo do item 1.21.
2.12	Orientação direta	Usar o teclado numérico para orientar o avatar em oito direções. É uma forma bem mais completa de orientar o avatar do que o padrão de alguns jogos (W,A,S,D).	Não atendido.	Tarefa do desenvolvedor, que deve fornecer iteração de acordo com os propósitos da aplicação em seus eventos.
2.13	Mira automática	Ajudar o jogador, oferecendo a opção por mira automática, ou autocentralização. Isto permite que os jogadores com baixa visibilidade possam encontrar rapidamente as metas e os alvos.	Não atendido.	O mesmo do item 2.12.

Também neste caso, e repetimos, é importante destacar que algumas diretrizes não selecionadas podem ser aplicadas no desenvolvimento de jogos e *websites* acessíveis; o que as exclui dessa seleção é o fato de serem mais características de *game design* ou de produção de conteúdo (como sons e imagens) do que propriamente linhas de código do desenvolvedor.

4.7.3 Diretrizes aplicáveis – *AbleGamers*

Nesta Seção, encontram-se as diretrizes da Seção 3.3 que foram selecionadas para incorporar este trabalho, seguindo o mesmo critério de escolha apresentado anteriormente. A Tabela 4.3 apresenta as diretrizes selecionadas.

Tabela 4.3: Diretrizes selecionadas - *AbleGamers*

Código	Prática	Descrição	Prática	Justificativa
3.1	Cores de texto alteráveis	Esta diretriz é voltada para pessoas que têm dificuldade em distinguir algumas cores e busca melhorar a funcionalidade do jogo de modo geral, gerando um conforto do jogador.	Não atendido.	O mesmo do item 1.8.
3.2	Tamanho de fontes alteráveis	<i>Gamers</i> com dificuldades visuais, incluindo aqueles que são quase totalmente cegos, podem participar de muitos jogos, mas a leitura do texto nas legendas, instruções e <i>chats</i> pode ser frustrante se o tamanho do texto é pequeno demais para ler.	Não atendido.	O mesmo do item 1.8.
3.3	Opção para daltônicos	É o recurso mais popular para quem tem problemas visuais. Idealmente, todos os jogos devem ter opções para daltônicos, que podem ser habilitadas para compensar uma variedade de problemas de deficiência de cor. Por curiosidade, uma abordagem utilizada baseia-se em mudar a cor verde para azul ao indicar informações, como companheiros de grupo ou barras de saúde.	Não atendido.	O mesmo do item 1.8.

3.4	Miras em alto contraste	<p>Para jogos de tiro em primeira pessoa, é extremamente importante verificar se o jogador não tem dificuldade em usar a mira e se ela não é confundida com elementos do cenário. Um exemplo disto é quando alvo e inimigo são vermelhos, o que causa uma confusão na visão do jogador.</p>	Não atendido.	O mesmo do item 1.21.
3.5	Marcação do inimigo	<p>A capacidade de indicar visualmente o inimigo é um dos diferenciais mais fundamentais em todos os jogos de 3D. Alguns jogos usam nomes, barras visuais, símbolos ou marcadores de cor.</p>	Não atendido.	O mesmo do item 1.21.
3.6	Fontes personalizadas	<p>Personalização da fonte não é uma nova tendência no desenvolvimento do jogo, mas o uso de fontes muito estilizadas tem um efeito negativo sobre os jogadores de baixa visão. A capacidade de trocar as fontes mais elaboradas por outra mais simples, como Arial ou Times New Roman, permite que o <i>gamer</i> de baixa visão interprete o texto mais facilmente.</p>	Não atendido.	O mesmo do item 1.8.

3.7	Mapa de alteração de cores no mapa	Classificado como a progressão lógica a partir das opções para daltônicos estabelecidos no nível 1. Permite aos usuários colorir novamente um mapa, deixando de lado o traço de cor original e dando espaço a novas cores, como o azul e o laranja. Isto permite que as pessoas com deficiências de visualização de cores possam fazer melhor uso da interface.	Não atendido.	O mesmo do item 1.8.
3.8	Ajustes de velocidade	Esta é uma opção de primeira linha para pessoas com baixa visão e distúrbios cognitivos, pois oferece possibilidade de retardar o relógio do jogo, permitindo que jogadores necessitados de mais tempo para processar o que está acontecendo possam reagir de maneira adequada em cada ação do jogo.	Não atendido.	O mesmo do item 2.3.
3.9	<i>Inputs Text-to-Speech</i>	Outra opção de acessibilidade bastante relevante é tornar o jogo capaz de ler o texto na tela e reproduzi-lo em forma de áudio. Muitos programas atuais fazem isso para os usuários na <i>web</i> com a inclusão de <i>tags</i> da WAI-ARIA.	Atendido.	O mesmo do item 1.20.

Vale aqui a mesma observação feita anteriormente sobre a exclusão das diretrizes não selecionadas.

4.7.4 Síntese da coleta

Ao todo foram selecionadas 43 (quarenta e três) diretrizes. Destas, 26 (vinte e seis) não foram atendidas por se caracterizar como trabalho do programador, sendo de responsabilidade deste trabalho indicar como devem ser implementadas estas diretrizes. As demais diretrizes foram parcialmente ou totalmente atendidas.

4.8 Considerações finais

Neste capítulo abordamos os principais aspectos que caracterizam a validade deste trabalho, partindo da visão geral da ferramenta, na qual encontramos os pilares que guiaram o desenvolvimento do que foi visto aqui.

Também vimos neste capítulo a arquitetura do ARIA-ACCESS, disponibilizando um breve tutorial de como se dá incorporação da ferramenta em projetos de desenvolvedores, que vislumbram a disponibilização de aplicações acessíveis, juntamente com o *link* do projeto na plataforma *GitHub*.

No próximo capítulo, veremos a avaliação deste trabalho, a coleta e a análise dos dados obtidos.

Capítulo 5

AVALIAÇÃO

Neste capítulo é apresentada a avaliação da abordagem proposta neste trabalho. O objetivo deste capítulo é apresentar a análise do impacto no desenvolvimento de jogos e aplicações *web* acessíveis, verificando não apenas se a ferramenta atende o objetivo proposto, mas também se cumpre seu papel em tempo inferior ao desenvolvimento sem a sua implementação. A Seção 5.1 apresenta a estratégia avaliativa aplicada na validação deste trabalho; a Seção 5.2 apresenta informações sobre a coleta de informações com desenvolvedores; na Seção 5.3, a avaliação e a análise dos dados obtidos neste estudo de caso. Por fim, encontram-se na Seção 5.4 as considerações finais deste capítulo.

5.1 Avaliação

A estratégia avaliativa aplicada na validação deste trabalho, para averiguar se é possível desenvolver uma aplicação *web* acessível utilizando o ARIA-ACCESS, é o estudo de caso. É importante destacar que não houve controle algum sobre eventos comportamentais dos participantes da pesquisa.

Além disso, a estratégia aqui escolhida focaliza acontecimentos contemporâneos, utilizando questionários para coletar o *feedback* necessário na validação do uso do ARIA-ACCESS e satisfazendo as diretrizes do estudo no que se refere às séries sistemáticas de entrevistas.

O planejamento deste estudo foi realizado de acordo com as recomendações propostas por Yin (2015).

5.1.1 Componentes do projeto de pesquisa

A necessidade da escolha de um estudo de caso justifica-se por visarmos a que a avaliação não separe o fenômeno do seu devido contexto, já que os resultados obtidos afetam diretamente o seu propósito. Não foi possível, desta forma, justificar sua eficácia com um estudo no qual fenômeno e contexto não estejam interligados.

5.1.1.1 Questões de estudo

O experimento realizado por meio de um estudo de caso tem o intuito de averiguar se é possível desenvolver uma aplicação *web* acessíveis utilizando o *framework* ARIA-ACCESS, sem a necessidade de nenhum conhecimento prévio da linguagem do WAI-ARIA ou das recomendações da WCAG.

Além disso, o experimento também visa a mensurar se o tempo de desenvolvimento com o *framework* é satisfatório se comparado ao desenvolvimento que usa as *tags* do WAI-ARIA, desconsiderando ainda o tempo de estudo das diretrizes WCAG e de boas práticas com WAI-ARIA.

Buscando atender todos os objetivos dessa avaliação e garantir que a caracterização do estudo seja mantida, observou-se que a contribuição de um público específico da área de desenvolvimento de aplicações *web* seria necessária. Desta forma, a partir desta conclusão, o público da pesquisa escolhido foi a turma da disciplina “Desenvolvimento de *Software* para a *Web*”, dos cursos de Ciência da Computação e Sistemas de Informação da Universidade Federal de Santa Maria, disciplina essa que é oferecida no penúltimo semestre de ambos os cursos.

Com o foco do experimento e o público definidos, um conjunto de documentos como instrução foram disponibilizados para a melhor interação do desenvolvedor com a ferramenta. Juntamente com a documentação, foi disponibilizado um *template* com os elementos que deverão ser acessíveis. Essa prática justifica-se porque o foco do experimento é apenas a integração de uma página com o ARIA-ACCESS, desconsiderando-se o tempo que um desenvolvedor despenderia para desenvolver um *template* completo.

5.1.1.2 Proposições de estudo

As proposições de estudo são as variáveis que serão analisadas dentro do escopo deste experimento. Como regra, englobam-se as proposições mais específicas a fim de formar um grupo que inclua todas as variáveis examinadas. Assim, duas proposições ganham destaque

nesse estudo:

- A possibilidade de utilizar o *framework* ARIA-ACCESS para desenvolver jogos e aplicações *web* acessíveis, tendo pouco ou nenhum conhecimento prévio dos conceitos de acessibilidade WAI-ARIA ou das diretrizes de desenvolvimento acessível da WCAG;
- A eficácia do *framework* ARIA-ACCESS no que tange ao desenvolvimento de jogos e aplicações *web* acessíveis, considerando diferentes variáveis, dentre as quais podemos citar: tempo da curva de aprendizado, abrangência e relevância da ferramenta.

5.1.1.3 Unidades de análise

Segundo Yin (2015), as unidades de análise são os objetos ou eventos aos quais a análise se relaciona, o que ou quem será descrito, analisado ou comparado, para definir o que é um “caso”. No presente trabalho, o caso em questão é o *framework* ARIA-ACCESS.

5.1.1.4 Ligação dos dados à proposição e os critérios para a interpretação dos dados

Estes dois componentes do experimento representam a análise no estudo de caso, e o projeto de pesquisa é a base sobre a qual esta análise será feita. Nesta abordagem, as informações coletadas a partir do experimento são relacionadas às proposições estabelecidas no início da elaboração do trabalho.

Para isso, é necessário que se definam as teorias iniciais relativas ao estudo a ser avaliado. No nosso caso, as teorias foram desenvolvidas antes da coleta de dados, ajudando a cobrir de forma incremental as questões, as proposições ou o propósito do estudo e as unidades de análise; assim possibilitaremos a ligação dos dados às proposições, fornecendo, com isso, os critérios para a análise dos dados.

Seguindo este processo, a afirmação de que “o ARIA-ACCESS reduz o tempo de desenvolvimento de aplicações acessíveis” pode ser validada a partir da coleta de dados, já que a pesquisa segue um roteiro definido, possibilitando que os dados coletados contribuam com a validação ou refutação da afirmação base, que assim se torna mais precisa em relação ao fato.

5.1.2 Processo para pesquisa com o uso do método do estudo de caso

Segundo o processo definido por Yin (2015), quando um investigador decide usar o método do estudo de caso para avaliar seus conceitos de pesquisa, ele deve ter conhecimento e domínio

do processo a ser utilizado para tal.

O planejamento da avaliação, que é caracterizado em estágio inicial, é feito a partir da análise do problema que o trabalho deseja solucionar, assim como das questões iniciais que servirão como base da pesquisa a ser aplicada nos passos subsequentes.

As teorias iniciais foram definidas em formato de questões para responder. São elas:

1. O ARIA-ACCESS reduz o tempo de desenvolvimento de aplicações acessíveis?
2. É possível aprender a utilizar o ARIA-ACCESS tendo pouco ou nenhum conhecimento de diretrizes de acessibilidade ou de recomendações WCAG? A documentação disponível é suficiente?
3. O ARIA-ACCESS satisfaz o desenvolvedor em termos de produtividade, seguindo uma curva de aprendizado satisfatória?

Para responder a essas questões de pesquisa, foram coletados os seguintes dados:

- Dados referentes ao tempo de desenvolvimento;
- Nível de conhecimento de desenvolvimento *web*;
- Nível de conhecimento de *tags* WAI-ARIA;
- Sucesso na incorporação do *framework* no projeto;
- Sucesso no processo de tornar acessíveis todos os elementos propostos;
- Relevância da documentação;
- Relevância no impacto do desenvolvimento;
- Sugestões de melhorias.

5.2 Coleta

O estudo corroborativo do ARIA-ACCESS, como dito, foi aplicado na turma da disciplina “Desenvolvimento de *Software* para a *Web*”, dos cursos de Ciência da Computação e Sistemas de Informação da Universidade Federal de Santa Maria, no dia 14/10/2015.

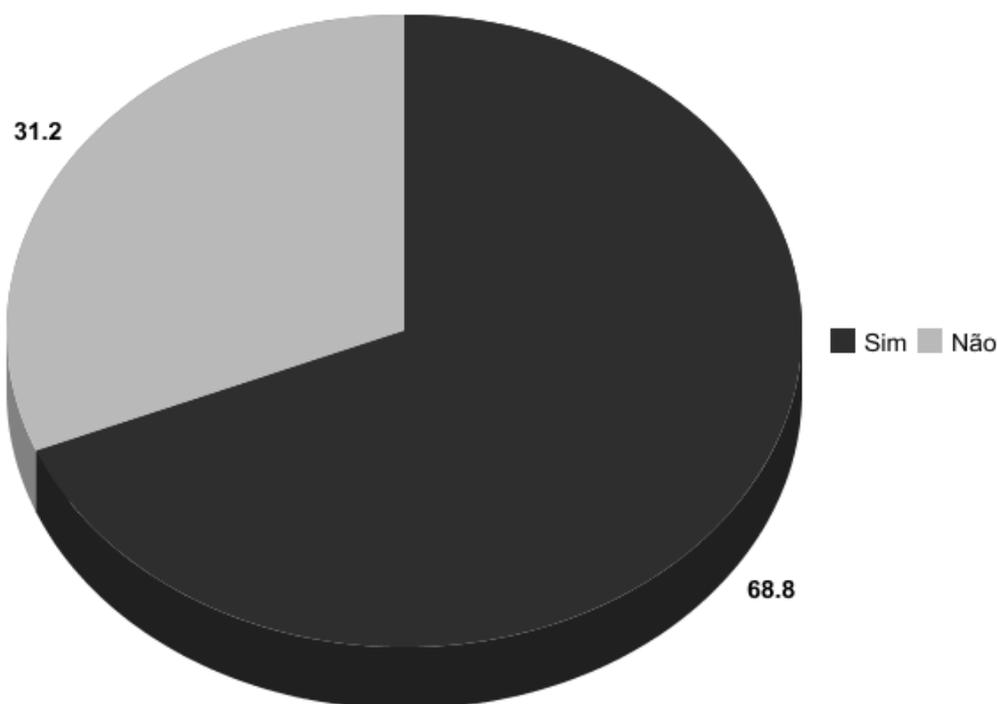
O questionário aplicado neste estudo pode ser verificado no Apêndice A. O termo de consentimento livre e esclarecido está disposto no Apêndice B.

Para a aplicação do estudo, disponibilizou-se aos estudantes a documentação do *framework*, disponível no *link* a seguir:

https://github.com/pablobizzi/plugin_acessivel/wiki/Documentation.

Além disso, a turma foi dividida em dois grupos, de modo que um dos grupos teria uma aula introdutória sobre o ARIA-ACCESS (grupo I), ministrada após a divisão dos grupos, e o outro teria como base apenas a documentação previamente disponibilizada (grupo II); a divisão é apresentada na Figura 5.1. A participação no experimento foi facultativa e a divisão foi realizada antes da explicação do estudo.

Figura 5.1: Divisão de grupos para o estudo de caso deste projeto.



Dissertando sobre a aula introdutória, foram elencados conceitos de acessibilidade *web* e melhores práticas de utilização e como é a mecânica do ARIA-ACCESS.

Estendendo sobre a divisão dos grupos, a separação inicial dos participantes foi realizada de forma igual em número de participantes para ambos os grupos, sendo assim, houve desistência da participação por parte de alguns integrantes, sendo de maior número os integrantes do grupo II.

A documentação e o *template* da aplicação estiveram disponíveis durante todo o tempo do estudo, e foi solicitado aos participantes que fornecessem o *feedback* em até sete dias corridos após a disponibilização do material e o envio dos arquivos-fonte para verificar possíveis problemas e verificar veracidade das respostas. Ao todo, dezesseis participantes contribuíram com dados para a análise deste trabalho.

5.3 Avaliação e análise dos dados

No contexto do desenvolvimento de aplicações *web* acessíveis, considerando especialmente o *framework* desenvolvido nesse trabalho, os dados são avaliados e analisados a partir das seguintes variáveis:

1. Nível de conhecimento de desenvolvimento *web*;
2. Nível de conhecimento de *tags* WAI-ARIA;
3. Sucesso na incorporação do *framework* no projeto;
4. Sucesso no processo de tornar acessível todos os elementos propostos;
5. Relevância da documentação;
6. Relevância no impacto do desenvolvimento;
7. Dados referentes ao tempo de desenvolvimento;
8. Sugestões de melhorias.

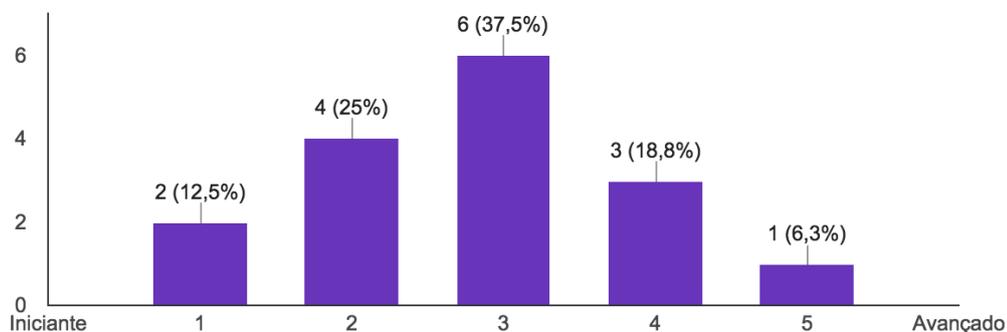
A validação de cada uma dessas variáveis se dará por meio da análise de características específicas dos dados coletados e serão abordadas nas seções seguintes.

5.3.1 Nível de conhecimento de desenvolvimento *web*

Esta variável é destacada neste estudo para medir se o contato com o ARIA-ACCESS é dificultado pela inabilidade dos participantes em manipular aplicações *web*.

A Figura 5.2 apresenta um gráfico de colunas que ilustra as informações sobre o nível de conhecimento de desenvolvimento *web* dos participantes. No caso, 1 (um) é nível iniciante e 5 (cinco) é desenvolvedor avançado.

Figura 5.2: Gráfico de colunas que representa o nível de conhecimento de desenvolvimento *web* dos participantes.



A partir dos dados coletados, é possível afirmar que os participantes têm a capacidade de testar a ferramenta fidedignamente, visto que os elementos HTML utilizados no *framework* para a sua validação são de caráter básico — tais como parágrafos, cabeçalhos, imagens, tabelas, menus não ordenados, etc. Vale destacar que não se exigiu nenhuma manipulação *backend* dos participantes, mas apenas a manipulação dos elementos em tela, não influenciando diretamente a manipulação do ARIA-ACCESS por falta de conhecimento *web* mais avançado.

5.3.2 Nível de conhecimento de *tags* WAI-ARIA

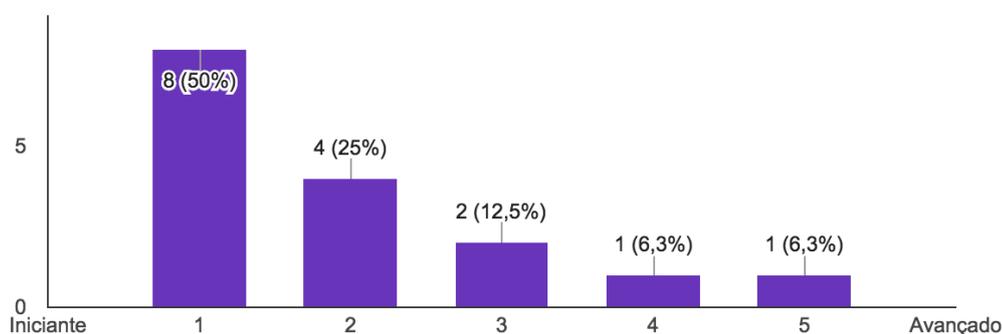
Assim como o conhecimento de desenvolvimento *web*, o conhecimento de práticas de desenvolvimento acessível é relevante para nós, dado que participantes que já trabalharam com acessibilidade — como *tags* WAI-ARIA — podem relatar com precisão quais os pontos positivos e negativos da abstração deste trabalho por parte do desenvolvedor.

Outro aspecto positivo do *feedback* de participantes com conhecimento avançado é a possível detecção de problemas na ferramenta, visto que o desenvolvedor pode prever o comportamento necessário daquele elemento em que foram aplicados os conceitos de acessibilidade.

A Figura 5.3 apresenta um gráfico de colunas que mostra informações a respeito do nível de conhecimento de práticas de desenvolvimento acessível dos participantes; 1 (um) é nível iniciante e 5 (cinco) é desenvolvedor avançado.

A partir do *feedback* recebido, é possível afirmar que os dados colhidos representam opiniões e percepções de participantes dos mais diversos níveis de conhecimento no que se refere a prática de desenvolvimento *web* acessível. Houve desde participantes com pouco ou nenhum conhecimento — representando a maioria deste estudo — até pessoas com conhecimento bom ou avançado, que podem dar *feedbacks* técnicos precisos advindos dos conhecimentos que ul-

Figura 5.3: Gráfico de colunas que representa o nível de conhecimento de práticas de desenvolvimento acessível dos participantes.



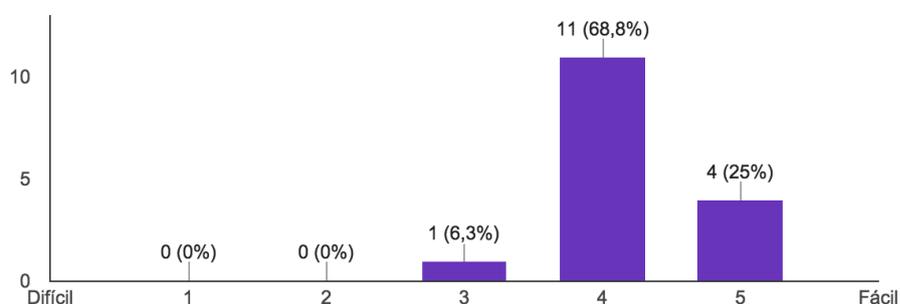
trapassam a utilização do ARIA-ACCESS.

5.3.3 Sucesso na incorporação do *framework* no projeto

Com o objetivo de desenvolver uma ferramenta que agregue qualidade no que tange ao desenvolvimento de aplicações *web* acessíveis, é necessário que ela, além de ser fácil de utilizar, não seja considerada um empecilho na incorporação ao projeto.

A Figura 5.4 retrata, por meio de um gráfico de colunas, informações coletadas a respeito da facilidade de incorporar o ARIA-ACCESS e suas dependências na aplicação *web* em questão. Novamente, 1 (um) é nível iniciante e 5 (cinco) é desenvolvedor avançado.

Figura 5.4: Gráfico de colunas que representa o nível de facilidade na incorporação do *framework* no projeto por parte dos participantes.



A Figura 5.4 aponta que não houve problemas por parte dos desenvolvedores na importação do ARIA-ACCESS e suas dependências ao projeto. É possível fazer esta afirmação, pois todos os participantes classificaram a facilidade de incorporação da ferramenta entre notas 3 (três) e 5 (cinco).

5.3.4 Sucesso no processo de tornar acessíveis todos os elementos propostos

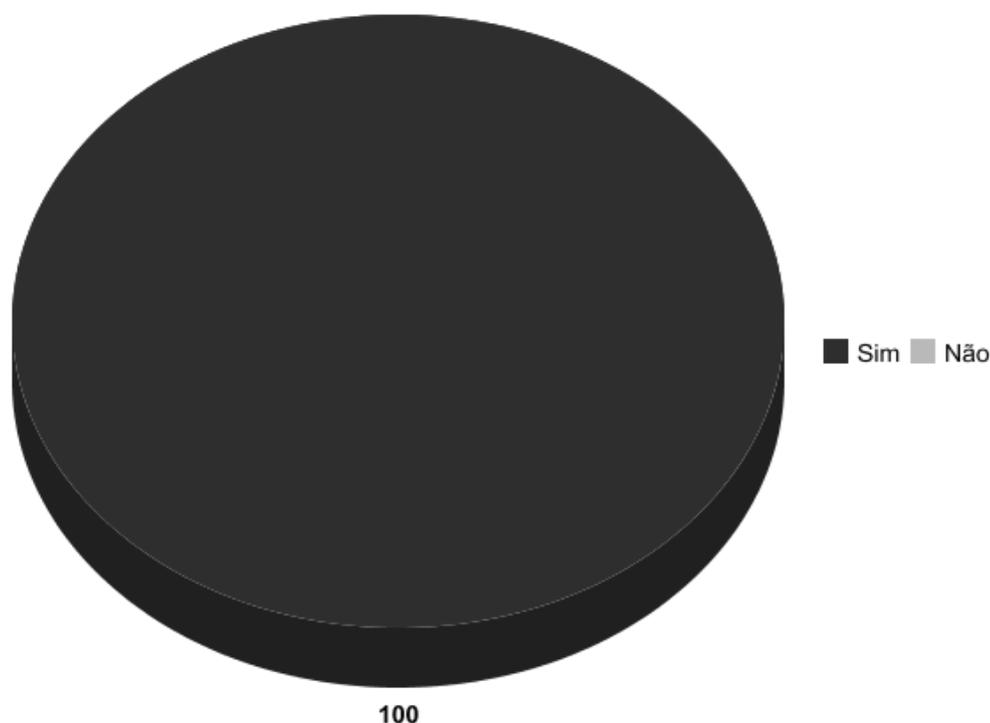
Talvez a principal variável deste estudo, o sucesso no processo de tornar todos os elementos propostos acessíveis avalia a abordagem proposta como um todo, uma vez que o sucesso do ARIA-ACCESS é basicamente o êxito em tornar elementos *web* acessíveis sem ser necessário ter conhecimento de *tags* WAI-ARIA ou das diretrizes WCAG.

Nesta Seção vamos expor, por meio de gráficos de pizza, a porcentagem de participantes que obtiveram êxito ou não em determinada tarefa.

Para que os participantes pudessem realizar a verificação de sucesso no processo, foi recomendada a utilização do leitor de tela NVDA¹.

A Figura 5.5 retrata a porcentagem de participantes que conseguiram tornar textos e cabeçalhos acessíveis, independentemente do grupo a que pertencem (com ou sem instrução).

Figura 5.5: Gráfico de pizza que mostra a porcentagem de participantes que conseguiram tornar textos e cabeçalhos acessíveis.

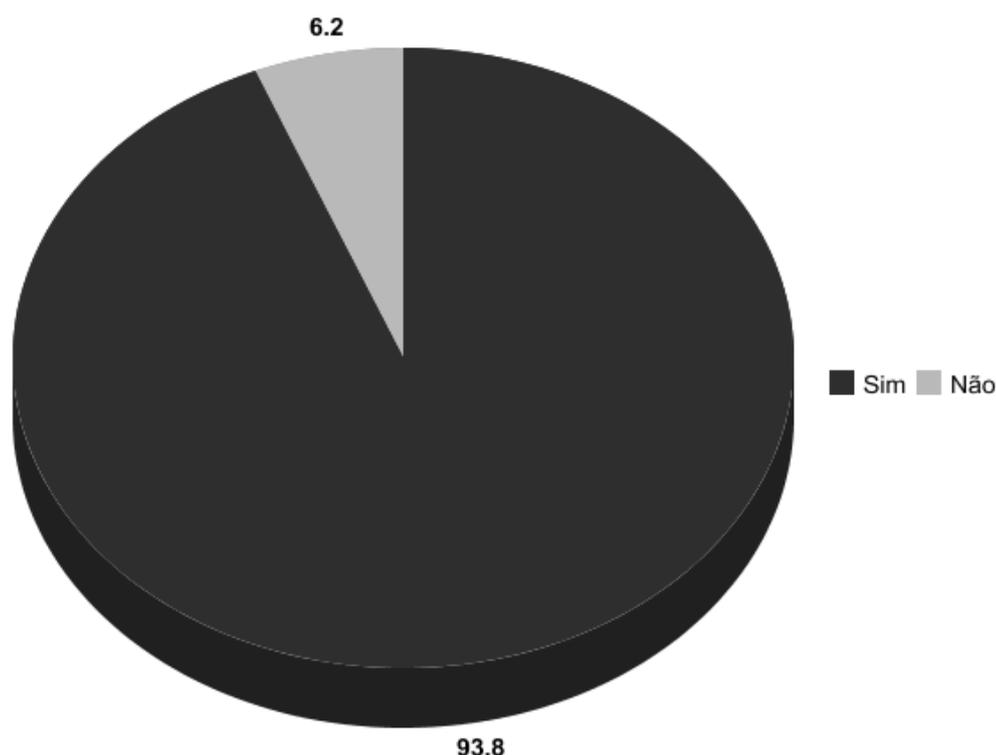


Acompanhando ainda a série de elementos gráficos mais simples, o gráfico de pizza presente na Figura 5.6 nos mostra a porcentagem de participantes que conseguiram tornar botões

¹<http://www.nvaccess.org/>

acessíveis, independentemente do grupo a que pertencem. É importante ressaltar que se pode verificar o êxito do processo analisando se o elemento pode ser acessado pelo teclado e se o leitor de tela acessível indica que o elemento é de fato um botão. Pode-se ainda transformar quaisquer elementos em botões com essa funcionalidade.

Figura 5.6: Gráfico de pizza que mostra a porcentagem de participantes que conseguiram tornar botões acessíveis.

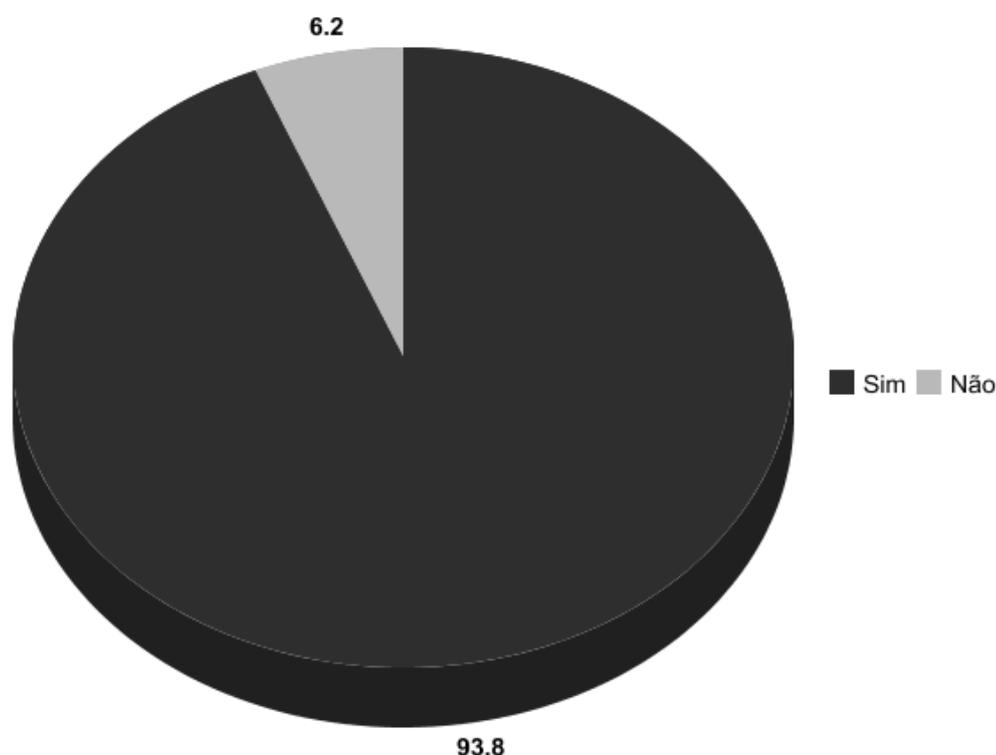


Também se destaca neste contexto a porcentagem de participantes que conseguiram tornar *links* e menus acessíveis; a Figura 5.7 exibe o gráfico de pizza que representa o êxito dessa variável. Para uma validação precisa do desenvolvedor, é importante verificar se o leitor de tela acessível indica que o elemento é um “*link*” ou um “menu”.

Pode-se ver, ademais, no gráfico de pizza presente na Figura 5.8, a porcentagem de participantes que conseguiram tornar tabelas acessíveis, independentemente do grupo a que pertencem. É importante frisar que o processo de tornar uma tabela acessível vai além de apenas fornecer a leitura do texto, permitindo que o usuário possa navegar nos nós da tabela pelo teclado.

Ainda, o gráfico de pizza presente na Figura 5.9 indica-nos a porcentagem de participantes que conseguiram tornar imagens acessíveis, independentemente do grupo a que pertencem. Vale destacar que é possível verificar o êxito do processo se o que está presente dentro da *tag alt* do

Figura 5.7: Gráfico de pizza que mostra a porcentagem de participantes que conseguiram tornar *links* e menus acessíveis.



HTML é lido pelo leitor de tela acessível.

Por fim, a variável mais complexa mapeada pelo ARIA-ACCESS, devido à quantidade de elementos englobados na abstração, é evidenciada na Figura 5.10, que apresenta a porcentagem dos participantes que obtiveram êxito em tornar formulários acessíveis. É válido destacar que essa funcionalidade do ARIA-ACCESS torna acessíveis elementos como *inputs* de texto ou senha, *radio button*, *check button*, *buttons*, *textarea*, etc.

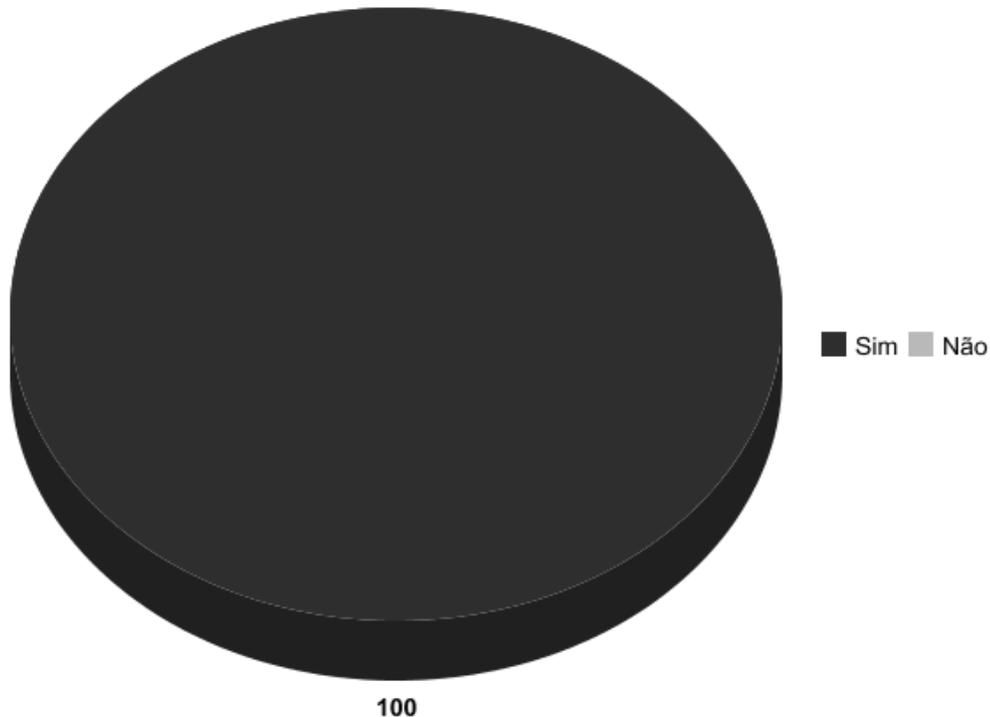
Baseando-se em todos os gráficos de pizza apresentados nesta Seção, pode-se afirmar que, no pior caso de desenvolvimento, apenas dois participantes não obtiveram êxito no processo de tornar acessível um elemento. Obteve-se, assim, uma média aritmética simples de 94,817% de sucesso na utilização de todos os elementos do ARIA-ACCESS.

5.3.5 Relevância da documentação

Já destacamos previamente, na Seção 4.5, a importância da documentação de uma ferramenta.

Buscou-se, no estudo, coletar informações do público a fim de avaliar a relevância da docu-

Figura 5.8: Gráfico de pizza que mostra a porcentagem de participantes que conseguiram tornar tabelas acessíveis.

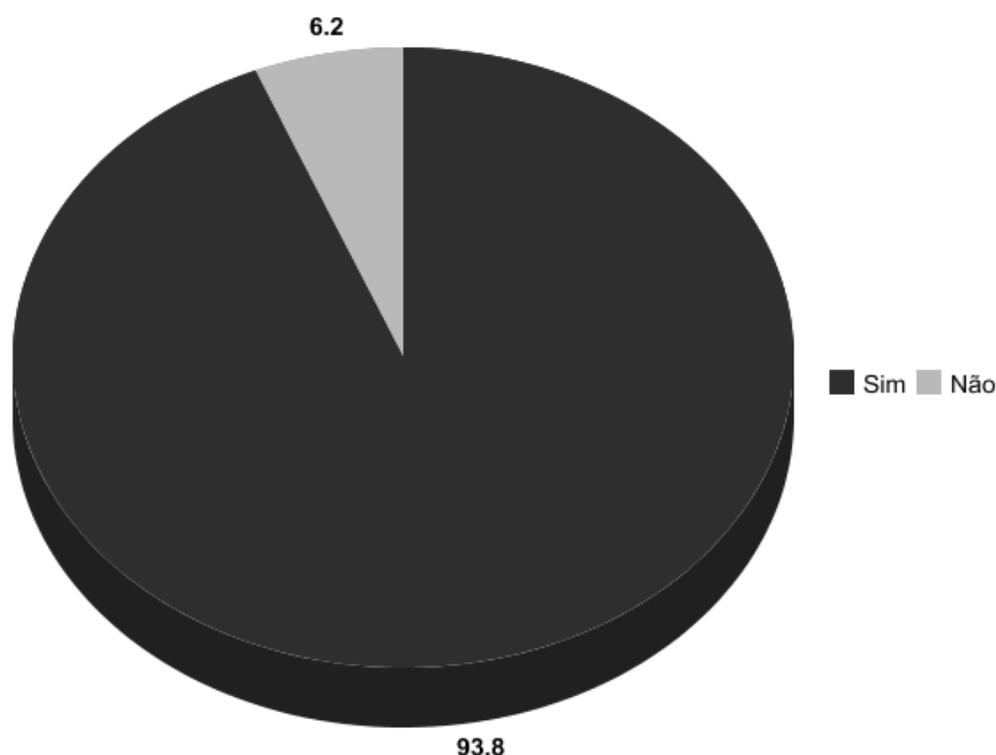


mentação, ou seja, o quanto ela pode ajudar desenvolvedores que nunca tiveram contato prévio com a ferramenta, com *tags* WAI-ARIA ou conceitos de acessibilidade WCAG.

A Figura 5.11 apresenta um gráfico de colunas com as informações coletadas a respeito da qualidade e da relevância da documentação disponibilizada para o estudo de caso. No caso, 1 (um) indica que a documentação é insatisfatória, e 5 (cinco) indica que a documentação é completa e de fácil manuseio.

A partir do *feedback* recebido, pode-se concluir que a documentação produzida para o estudo foi satisfatória, ficando com uma avaliação intermediária de 3,57. É importante destacar que, analisando individualmente as respostas, participantes com maior conhecimento de desenvolvimento *web* classificaram a documentação com uma nota superior, e com isso se pode inferir que a documentação deve ser mais clara, principalmente para desenvolvedores com conhecimento de desenvolvimento *web* básico.

Figura 5.9: Gráfico de pizza que mostra a porcentagem de participantes que conseguiram tornar imagens acessíveis.



5.3.6 Facilidade de uso

Não menos importantes que as demais variáveis, sendo uma peça fulcral para a validação deste estudo, a facilidade de uso foi avaliada com o intuito de averiguar como se deu a interação dos desenvolvedores com o ARIA-ACCESS.

É impossível captar exatamente o que cada desenvolvedor considera como fácil de usar ou não, por se tratar de uma avaliação completamente subjetiva baseada em preferências pessoais. Contudo, podemos avaliá-lo qualitativamente, a partir de uma abordagem genérica, como é apresentado na Figura 5.12. Aqui, coletamos a opinião de cada desenvolvedor sobre a facilidade do uso do ARIA-ACCESS no projeto, a partir de uma abordagem numérica e disposta em um gráfico de colunas. Classificamos como 1 (um) o muito difícil ou inaplicável no projeto, e 5 (cinco) o muito fácil ou com curva de aprendizado rápida com manipulação facilitada.

O *feedback* desta variável revela que a abordagem do ARIA-ACCESS obteve sucesso, sendo de simples manuseio e de simples incorporação no projeto, surgindo como uma real alternativa à abordagem tradicional de desenvolvimento acessível. A média aritmética simples dos dados, que comprova esta nossa conclusão, tem o valor de 4,19.

Figura 5.10: Gráfico de pizza que retrata a porcentagem de participantes que conseguiram tornar formulários acessíveis.

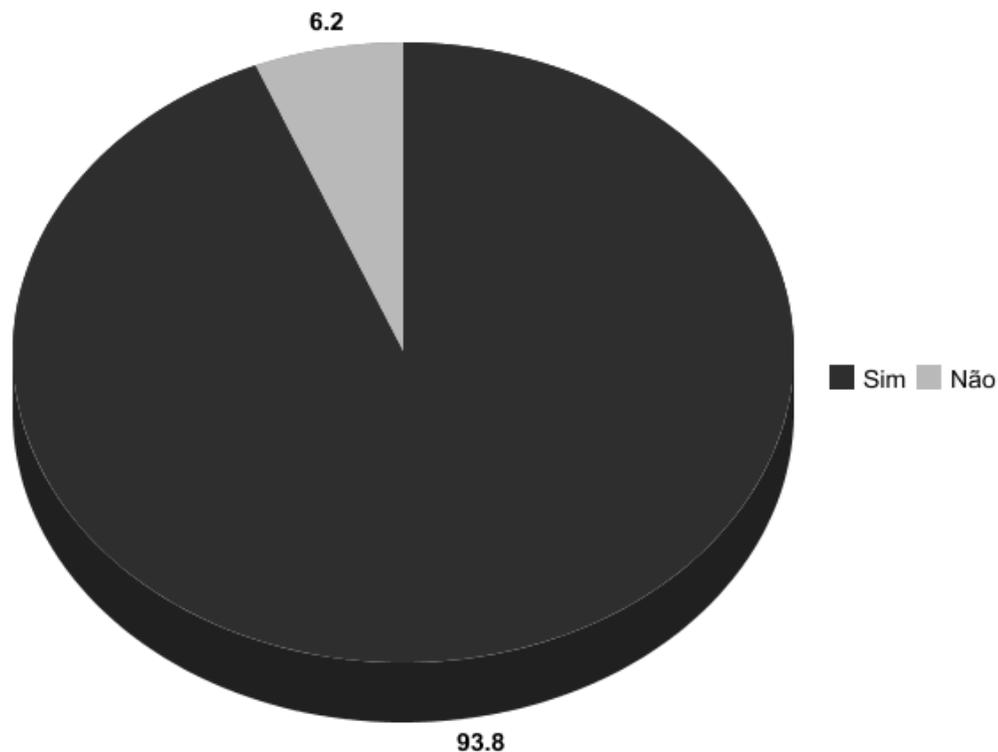
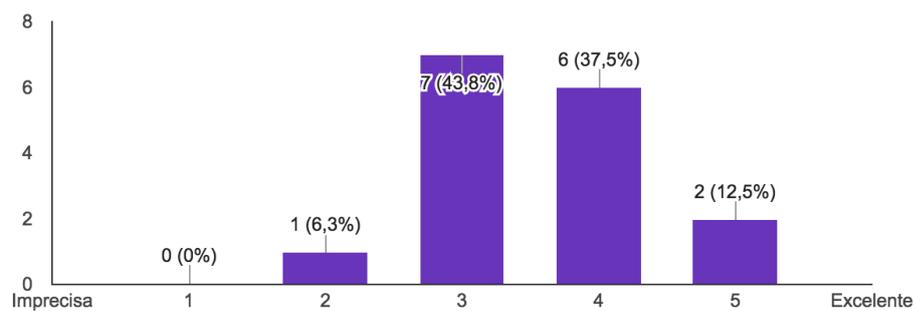


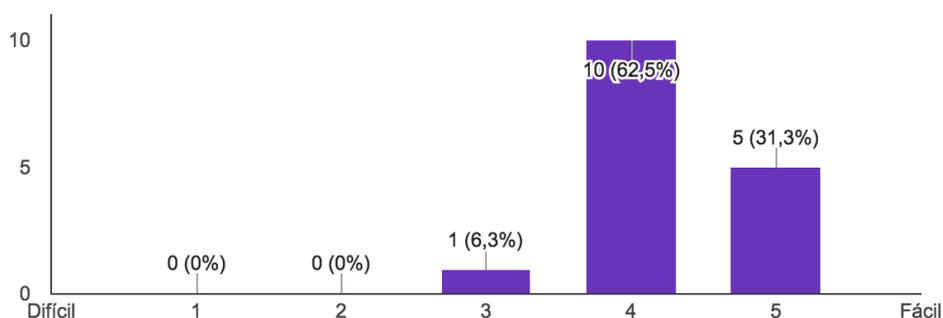
Figura 5.11: Gráfico de colunas que apresenta a qualidade da documentação do ARIA-ACCESS.



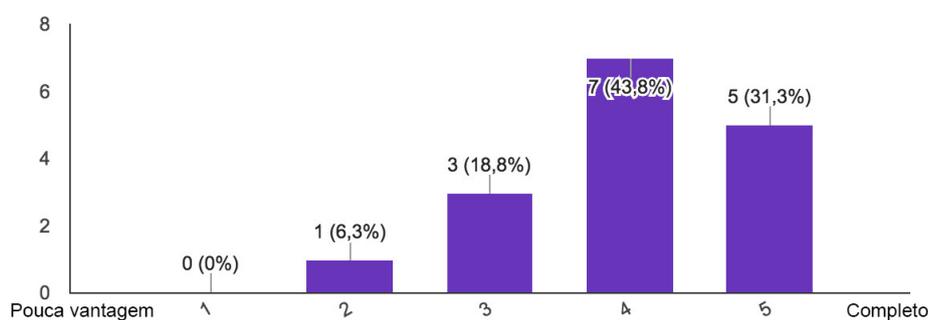
5.3.7 Relevância no impacto do desenvolvimento

Esta variável do estudo buscou coletar informações referentes ao impacto no desenvolvimento de aplicações *web* acessíveis, comparando as práticas tradicionais de desenvolvimento acessível com a abordagem proposta pelo ARIA-ACCESS.

Observemos a Figura 5.13, que exibe as informações coletadas a respeito do impacto no desenvolvimento de aplicações acessíveis com a utilização do ARIA-ACCESS, ao invés das práticas tradicionais de desenvolvimento acessível. O qualificador 1 (um) indica que o impacto

Figura 5.12: Gráfico de colunas que apresenta a facilidade de uso do ARIA-ACCESS.

é insatisfatório, ou seja, não é vantajoso utilizar o ARIA-ACCESS, e sim utilizar as *tags* WAI-ARIA; e 5 (cinco) significa que o *framework* é completo e satisfaz o desenvolvedor, por ter tornado acessíveis os elementos do seu projeto.

Figura 5.13: Gráfico de colunas que apresenta o impacto no desenvolvimento de aplicações acessíveis do ARIA-ACCESS.

Refletindo sobre os dados coletados, pode-se afirmar que o ARIA-ACCESS está apto para ser incorporado em projetos de desenvolvimento acessível, o que é sustentado pela média aritmética simples dos resultados: 4 (quatro).

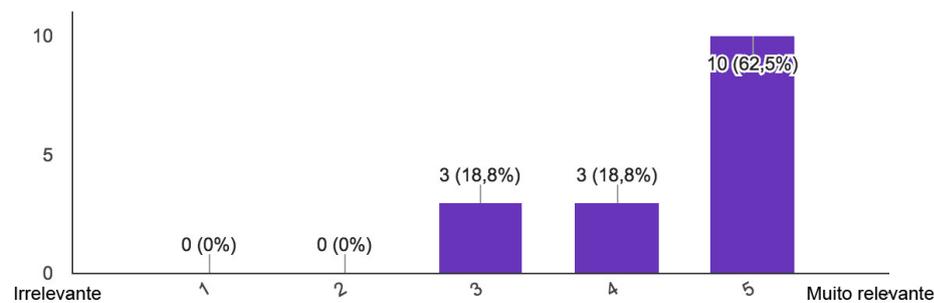
5.3.8 Relevância para a sociedade

Diversas vezes tomamos decisões baseados no benefício comum que tal ato geraria para toda a sociedade. No caso deste projeto não foi diferente, e frequentemente mencionamos os benefícios para a sociedade quando uma abordagem de desenvolvimento acessível é aplicada.

Esta variável foi incluída neste estudo pois se caracteriza como potencial motivador para que o desenvolvedor torne acessível sua aplicação *web*. A Figura 5.14 revela a avaliação pessoal

dos participantes do estudo acerca da relevância para a sociedade que um sistema acessível pode trazer, utilizando o ARIA-ACCESS no projeto.

Figura 5.14: Gráfico de colunas que apresenta a relevância para a sociedade de aplicações que usam o ARIA-ACCESS.



É visível que os desenvolvedores demonstraram um interesse por essa nova abordagem e reconheceram seus benefícios para toda a sociedade, por ser possível, a partir do ARIA-ACCESS, disponibilizar aplicações *web* acessíveis. A média aritmética simples de 4,44 dá suporte às afirmações que dizem respeito aos benefícios para a sociedade que o *framework* proposto pode trazer.

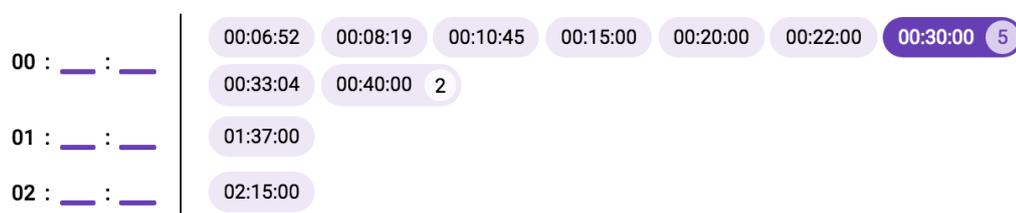
5.3.9 Dados referentes ao tempo de desenvolvimento

Tratar do tempo de desenvolvimento de *software* engloba diversas questões que transcendem o objetivo central deste projeto. Em razão disso, procurou-se disponibilizar um *template* pronto em HTML para os desenvolvedores aplicarem apenas a abordagem proposta pelo ARIA-ACCESS.

Tal decisão justifica-se pelo fato de o objetivo do trabalho ser medir apenas o tempo gasto para tornar a aplicação acessível, desconsiderando o tempo de programação de conceitos que envolvem lógica de programação e conhecimento de *leiaute*.

Pode-se visualizar na Figura 5.15 o tempo utilizado pelos desenvolvedores para realizar por completo o experimento proposto, considerando participantes que obtiveram instrução de uso e participantes que receberam apenas a documentação.

Examinando os tempos a partir de uma análise macro, obtemos um tempo médio aproximado de 36 (trinta e seis) minutos. Analisando individualmente os tempos dos grupos, pode-se afirmar que quando os desenvolvedores recebem instrução sobre o *framework* ou eventualmente podem estar utilizando o mesmo com conhecimento prévio, o tempo investido aproximado no

Figura 5.15: Tempo despendido pelos participantes no estudo do ARIA-ACCESS.

desenvolvimento foi de 29 (vinte e nove) minutos, contra aproximados 51 (cinquenta e um) minutos por parte dos desenvolvedores que utilizaram o ARIA-ACCESS pela primeira vez sem nenhum auxílio além da documentação.

Com isso, pode-se afirmar que em ambos os casos os tempos obtidos ficaram dentro da margem esperada, não prejudicando o andamento do desenvolvimento de software como um todo. Acredita-se que o ARIA-ACCESS agrega ao desenvolvimento dos interessados, já que no pior caso de teste, que foi de 2h15 (duas horas e quinze minutos), não seria possível ler a documentação básica do WAI-ARIA ou da WCAG.

5.3.10 Sugestões de melhorias

Finalmente, foram solicitadas aos participantes do estudo sugestões sobre os objetos do estudo, que compreendem todas as variáveis citadas anteriormente, e sobre possíveis funcionalidades que poderiam complementar o ARIA-ACCESS.

Entre as principais sugestões apresentadas, verificou-se que a documentação precisa de explicações mais claras que incorporem cada detalhe do ARIA-ACCESS, principalmente quando se trata de integrar o *framework* a aplicações *web* existentes e de como adaptar isso ao leiaute. Alguns exemplos das sugestões de melhoria na documentação podem ser lidas a seguir:

Sugestão: colocar uma explicação mais detalhada de como funciona o *plugin* (na página da documentação), para facilitar a escolha das *tags* que o programador deixará acessível. – Identidade preservada de participante do Grupo II.

A partir do *feedback* anterior, vê-se certa carência na lista de *tags* e funcionalidades apresentadas na documentação do ARIA-ACCESS.

Melhorar a documentação. Pouca orientação. Apesar de ser tranquila a implementação, não aborda detalhes como por exemplo inserir a *class* como a última das

tags para não dar problema no layout. – Identidade preservada de participante do Grupo I.

A sugestão anterior, de um desenvolvedor do grupo I *web* com experiência, reflete um problema apresentado por um desenvolvedor iniciante que não recebeu instruções sobre o experimento, apresentado a seguir:

Não sei se porque sou muito iniciante com CSS e HTML, mas quando coloquei a imagem acessível ela saiu da formatação, o mesmo com a tabela. – Identidade preservada de participante do Grupo II.

Por fim, destaca-se a partir das sugestões que o ARIA-ACCESS tem o que é preciso para adequar aplicações *web* a fim de que sejam classificadas como acessíveis. É necessário apenas um esforço para não tornar a documentação um empecilho no uso da ferramenta.

5.4 Considerações finais

Neste capítulo, descrevemos a compilação de todos os dados coletados pelo estudo de caso, analisando e discutindo cada resultado com base nos conceitos adotados e nas perspectivas dos desenvolvedores, visando a identificar o impacto do ARIA-ACCESS na produção de aplicações acessíveis.

Os resultados analisados e discutidos neste capítulo apontam avanços relativos ao desenvolvimento de aplicações *on-line* acessíveis, independentemente do nível de conhecimento em linguagens de desenvolvimento *web*, WCAG ou WAI-ARIA.

Retomando as teorias iniciais definidas na Seção 5.1.2, pode-se responder essas questões levantadas com o *feedback* recebido dos participantes das pesquisas.

1. O ARIA-ACCESS reduz o tempo de desenvolvimento de aplicações acessíveis? R: Sim, devido ao fato que o maior tempo registrado, de duas horas e quinze minutos, ser insuficiente para ler toda a documentação da WAI-ARIA e WCAG, dada a vasta bibliografia das mesmas.
2. É possível aprender a utilizar o ARIA-ACCESS tendo pouco ou nenhum conhecimento de diretrizes de acessibilidade ou de recomendações WCAG? A documentação disponível é suficiente? R: A partir do *feedback* coletado, é possível afirmar que foi possível aprender

a utilizar o ARIA-ACCESS tendo pouco ou nenhum conhecimento de acessibilidade *web* e que a documentação satisfizes a necessidade dos participantes.

3. O ARIA-ACCESS satisfaz o desenvolvedor em termos de produtividade, seguindo uma curva de aprendizado satisfatória? R: A resposta dos participantes indicou que o ARIA-ACCESS satisfizes o desenvolvimento de aplicações acessíveis, indicado pela facilidade de uso e relevância no impacto do desenvolvimento.

Tomando como base a exposição dos resultados apresentados neste capítulo, o capítulo seguinte apresenta as contribuições deste trabalho, suas limitações e os trabalhos futuros.

Capítulo 6

CONCLUSÃO

Neste trabalho, apresentamos uma nova abordagem ao desenvolvimento de aplicações *web*, disponibilizando recursos desenvolvidos neste projeto que permitem a disponibilização de aplicações *web* acessíveis, a fim de evitar ao máximo algum esforço da parte dos desenvolvedores. Para que a abordagem fosse aplicada, construímos uma ferramenta — ARIA-ACCESS — para validar o estudo de caso proposto.

Em vista disso, disponibilizamos na Seção 6.1 as contribuições obtidas nas atividades realizadas aqui. Também identificamos as limitações do trabalho, apresentadas na Seção 6.2. Por fim, algumas sugestões para a continuidade desta dissertação foram descritas na Seção 6.3.

6.1 Contribuições

O cerne principal deste trabalho foi não modificar a abordagem atual dos desenvolvedores de aplicações *web*, de modo que não fosse visto como um empecilho à rotina de incorporar a acessibilidade em seus projetos. Seu trunfo, como a consideramos, é a facilidade de integração do ARIA-ACCESS, para que ele seja usado nas mais diversas aplicações *web*.

Deste modo, o ARIA-ACCESS utiliza uma abordagem semelhante à de diversas bibliotecas Javascript, JQuery e até mesmo à do Bootstrap, sendo apenas necessário que a nossa ferramenta seja incorporada ao projeto para que as dependências do *framework* processem a página e gerem código HTML acessível.

É relevante ressaltar que este trabalho não apenas contribui com a facilidade de integração do *framework* ao projeto, mas é também um pilar no ganho de produtividade quanto ao tempo de desenvolvimento. Isto porque o ARIA-ACCESS gera o código WAI-ARIA automaticamente, abstraindo o trabalho do desenvolvedor de replicar diversos trechos de código para

que a página cumpra o mínimo necessário de acessibilidade, independentemente do seu nível de conhecimento em técnicas de desenvolvimento acessível.

Ainda, destaca-se a coleta de diretrizes existentes que são utilizadas no desenvolvimento de jogos acessíveis nas plataformas *desktop* e *console* que podem ser aplicadas no desenvolvimento de aplicações *on-line* acessíveis. Não se limitando a coleta, este trabalho complementa algumas diretrizes, propondo novas abordagens de desenvolvimento com fundamento nos estudos desenvolvidos neste trabalho e disponibilizando recomendações para os desenvolvedores.

Por fim, a ferramenta possibilita obter uma aplicação que cumpre com os requisitos básicos de acessibilidade em um tempo médio aproximado de 36 (trinta e seis) minutos, como vimos na Seção 5.3.9, o que justifica uma das principais premissas de validade deste trabalho.

6.2 Limitações

Um problema identificado no desenvolvimento foi a limitação dos recursos da plataforma *web*, tendo em vista a vasta aplicabilidade de diretrizes em outras plataformas, como *consoles* e PCs, o que tornou a aplicabilidade das diretrizes no ARIA-ACCESS um processo complexo. Em complemento à essa limitação, é importante destacar que não há possibilidade de tornar acessível um dos principais componentes para desenvolver jogos para a *web*, o *canvas*.

Ademais, é considerável a dificuldade em gerar verificações nas mais diferentes componentes e bibliotecas utilizadas em aplicações *web*, considerando-se impraticável o desenvolvimento de uma ferramenta que consiga atender todas as ferramentas atualmente disponíveis.

Por fim, desenvolver o trabalho com foco em funcionalidades cria uma lacuna no que tange à manipulação de folha de estilo (CSS), que foi tratada bem superficialmente no auxílio a fontes de leitura e a manipulação de componentes, quando utilizados pelo teclado.

6.3 Trabalhos futuros

Vislumbra-se que este trabalho possa contemplar mais funcionalidades de interação com as folhas de estilo (CSS) de um projeto, possibilitando que o desenvolvedor incorpore elementos do ARIA-ACCESS. Desta forma, a nossa ferramenta pode alterar em tempo de execução as mais distintas abordagens de apresentação do conteúdo, tais como alteração de estilo e tamanho de fonte, personalização do contraste, troca de cursor, modificação de espaçamento de elementos, entre outros.

Este trabalho, ademais, pode ser adaptado para tornar hábil a interação com componentes mais complexos, tais como *slideshow*, *datepicker*, *gallery*, etc.

REFERÊNCIAS

BARLET, M.; TAYLOR, M. J. *A practical guide to Game Accessibility*. [S.l.]: Includification, 2012.

BRASIL. *Lei no 13.146/15*. 2015. http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/Lei/L13146.htm. Acesso em 05/11/2016.

BRASIL, P. *Tipos de deficiência*. 2012. <http://www.brasil.gov.br/saude/2012/04/tipos-de-deficiencia/>. Acesso em 03/02/2015.

BRASIL, P. *Pesquisa revela que mais de 100 milhões de brasileiros acessam a internet*. 2016. <http://www.brasil.gov.br/ciencia-e-tecnologia/2016/09/pesquisa-revela-que-mais-de-100-milhoes-de-brasileiros-acessam-a-internet>. Acesso em 10/11/2016.

BRASIL, P. A. *O que é acessibilidade*. 2012. http://www.acessibilidadebrasil.org.br/versao_anterior/index.php?itemid=45/. Acesso em 03/02/2015.

CHEIRAN, J. F. P.; PIMENTA, M. S. *Jogos inclusivos: diretrizes de acessibilidade para jogos digitais*. Disserta (Mestrado) — Universidade Federal do Rio Grande do Sul, Instituto de Informática, Curso de Pós-Graduação em Computação, Porto Alegre, 2013.

CRUZ, C. S.; ESTOMBELO-MONTESCO, C. A. *HATEMILE: A biblioteca para gerar páginas web mais acessíveis*. Disserta (Mestrado) — Universidade Federal de Sergipe, Departamento de Computação, Curso de Pós-Graduação em Ciência Computação, São Cristóvão, 2015.

DALTONISMO, T. de. *Daltonismo*. 2014. <http://emag.governoeletronico.gov.br/cursoconteudista/introducao/deficiencia-daltonismo.html/>. Acesso em 04/02/2015.

DEVELOPERS, M. N. *Aplicações Web e ARIA FAQ*. 2015. https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Web_applications_and_ARIA_FAQ/. Acesso em 05/02/2015.

DOUGLAS, E. *Javascript Furtivo*. [S.l.]: Leanpub, 2014.

ELETRÔNICO, G. *eMAG - Modelo de Acessibilidade em Governo Eletrônico*. 2014. <http://emag.governoeletronico.gov.br/>. Acesso em 20/08/2016.

ELETRÔNICO, G. *Governo Eletrônico*. 2016. <http://www.governoeletronico.gov.br/>. Acesso em 20/08/2016.

- ELLIS, B. et al. *Game Accessibility Guidelines: A straightforward reference for inclusive game design*. 2013. <http://gameaccessibilityguidelines.com/>. Acesso em 01/03/2015.
- FEDERAL, C. *Decreto no 5.296, de 2 de dezembro de 2004*. 2004. http://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2004/Decreto/D5296.htm. Acesso em 05/11/2015.
- FEDERAL, C. *Decreto no 6.949, de 25 de agosto de 2009*. 2009. http://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2009/Decreto/D6949.htm. Acesso em 05/11/2015.
- FOWLER, M. *Domain-specific languages*. [S.l.]: Pearson Education, 2010.
- GRAMMENOS, D.; SAVIDIS, A.; STEPHANIDIS, C. Designing universally accessible games. *Computers in Entertainment (CIE)*, ACM, v. 7, n. 1, 2009.
- HOGAN, B. P. *HTML 5 & CSS 3*. [S.l.]: O'Reilly Germany, 2011.
- IGDA. *Accessibility in games: Motivations and approaches*. 2012. <http://igda-gasig.org/>. Acesso em 01/03/2015.
- MEYER, J. *The Essential Guide to HTML5*. [S.l.]: Springer, 2010.
- ONLINE, G. *G1*. 2015. <http://g1.globo.com/pernambuco/noticia/2015/02/mercado-de-games-movimenta-r-44-mi-em-pe-e-quer-crescer-em-2015.html>. Acesso em 06/11/2015.
- PRENSKY, M. *Aprendizagem baseada em jogos digitais*. [S.l.]: São Paulo: SENAC, 2012.
- RIBAS, J. B. C. *O que são pessoas deficientes*. [S.l.]: Brasiliense, 1985. 10-17 p.
- SILVA, M. S. *JavaScript: guia do programador*. [S.l.: s.n.], 2010.
- SILVA, M. S. *HTML5: a linguagem de marcação que revolucionou a web*. [S.l.]: São Paulo: Novatec, 2011.
- SOMMERVILLE, I. *Engenharia de Software, 8ª edição, Tradução: Selma Shin Shimizu Mel-nikoff, Reginaldo Arakaki, Edilson de Andrade Barbosa*. [S.l.]: São Paulo: Pearson Addison-Wesley, 2007.
- TALEB, A. et al. *As condições de saúde ocular no Brasil*. [S.l.]: Walprint Gráfica e Editora, 2012. 10 p.
- WAI-ARIA. *Site oficial da WAI-ARIA*. 2014. <http://www.w3.org/TR/wai-aria/>. Acesso em 05/02/2015.
- WCAG. *Site oficial da WCAG*. 2012. <http://www.w3.org/WAI/intro/wcag.php/>. Acesso em 05/02/2015.
- YIN, R. K. *Estudo de Caso-: Planejamento e Métodos*. [S.l.]: Bookman editora, 2015.

GLOSSÁRIO

AJAX – *Asynchronous Javascript and XML*

API – *Application Programming Interface*

CSS3 – *Cascading Style Sheets Version 3*

CSS – *Cascading Style Sheets*

HTML5 – *HyperText Markup Language Version 5*

HTML – *HyperText Markup Language*

PHP – *Hypertext Preprocessor*

RWD – *Responsive Web Design*

WAI-ARIA – *Web Accessibility Initiative - Accessible Rich Internet Applications*

WCAG – *Web Content Accessibility Guidelines*

Apendice A

QUESTIONÁRIO APLICADO PARA VALIDAR O ESTUDO DE CASO DO ARIA-ACCESS COM OS DESENVOLVEDORES

1.Nome?

2.E-mail?

3.Qual é o seu grupo? – Grupo 1: Recebeu explicação | Grupo 2: Liberado mais cedo

(a)Grupo 1

(b)Grupo 2

4.Conhecimento prévio de desenvolvimento WEB? – 1: Iniciante | 5: Avançado

(a)1

(b)2

(c)3

(d)4

(e)5

5.Conhecimento prévio de Acessibilidade WEB – 1: Iniciante | 5: Avançado

(a)1

(b)2

(c)3

(d)4

(e)5

6. Conseguiu tornar um texto acessível?

(a) Sim

(b) Não

7. Conseguiu tornar uma tabela acessível?

(a) Sim

(b) Não

8. Conseguiu tornar uma imagem acessível?

(a) Sim

(b) Não

9. Conseguiu tornar um botão acessível?

(a) Sim

(b) Não

10. Conseguiu tornar um formulário acessível?

(a) Sim

(b) Não

11. Conseguiu tornar um *link* acessível?

(a) Sim

(b) Não

12. Classifique a facilidade de uso do *plugin* – 1: Difícil | 5: Fácil

(a) 1

(b) 2

(c) 3

(d) 4

(e) 5

13. Classifique a qualidade e relevância da documentação – 1: Imprecisa | 5: Excelente

- (a)1
- (b)2
- (c)3
- (d)4
- (e)5

14. Como você avalia a utilização do *plugin* – 1: Difícil | 5: Fácil

- (a)1
- (b)2
- (c)3
- (d)4
- (e)5

15. Como você avalia a relevância do *plugin* e seu impacto no desenvolvimento – 1: Irrelevante | 5: Muito Relevante

- (a)1
- (b)2
- (c)3
- (d)4
- (e)5

16. Como você avalia a relevância para a sociedade – 1: Irrelevante | 5: Muito Relevante

- (a)1
- (b)2
- (c)3
- (d)4
- (e)5

17. Tempo gasto para tornar o *site* acessível?

Apendice B

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

1. Você está sendo convidado para participar da pesquisa "Um framework para apoiar o desenvolvimento web de jogos acessíveis".
2. Considera-se necessário propor um framework para o desenvolvimento de sites web, uma vez que não existe incentivo para este desenvolvimento. Espera-se que com este framework, os desenvolvedores possam disponibilizar websites acessíveis com pouco ou nenhum esforço.
 - a. Você foi selecionado e sua participação não é obrigatória
 - b. Os objetivos deste estudo são validar a utilização do framework.
 - c. Sua participação nesta pesquisa consistirá em utilizar o framework e responder a um questionário.
3. Os participantes são escolhidos aleatoriamente. Único requisito requerido para este estudo é que o participante deve estar matriculado na disciplina de Desenvolvimento de Software para a Web.
4. A pesquisa baseia-se no teste do Framework com desenvolvedores visando medir o ganho de produtividade dos mesmos no desenvolvimento web.
5. A sua participação neste estudo envolve riscos como desconforto moral e ético pela privacidade, ou físico por ficar exposto(a) a radiação, em contato com o computador.
 - a. Essa pesquisa será feita em um ambiente familiar, no DC-UFSCar o qual você está com grande frequência, com o intuito de diminuir os riscos de desconfortos.
 - b. Os dados que serão publicados desse estudo não te identificarão devido ao uso de siglas e números para se referir a você, quando necessário.
 - c. Há o risco da ferramenta não otimizar consideravelmente o tempo de desenvolvimento dos desenvolvedores.
6. Não aplica-se nenhum risco ao participante por utilizar a ferramenta em seus projetos ou de participar neste estudo.
7. Será ministrada uma aula sobre a ferramenta, então será proposta um estudo de caso visando a utilização da ferramenta. Após isso será solicitado informações sobre a utilização da ferramenta e sugestões.
8. A qualquer momento você pode desistir de participar e retirar seu consentimento. Sua recusa não trará nenhum prejuízo em sua relação com o pesquisador ou com a instituição.
9. As informações obtidas através dessa pesquisa serão confidenciais e asseguramos o sigilo sobre sua participação. Os dados não serão divulgados de forma a possibilitar sua identificação.
10. Caso necessite uma cópia deste termo, ele pode ser solicitado junto ao pesquisador Pablo Bizzi Mahmud pelo e-mail pablo.mahmud@dc.ufscar.br.

Pablo Bizzi Mahmud
pablo.mahmud@dc.ufscar.br (55) 9962-4468

Declaro que entendi os objetivos, riscos e benefícios de minha participação na pesquisa e concordo em participar. O pesquisador me informou que o projeto foi aprovado pelo Comitê de Ética em Pesquisa em Seres Humanos da UFSCar que funciona na Pró-Reitoria de Pós-Graduação e Pesquisa da Universidade Federal de São Carlos, localizada na Rodovia Washington Luiz, Km. 235 - Caixa Postal 676 - CEP 13.565-905 - São Carlos - SP - Brasil. Fone (16) 3351-8110. Endereço eletrônico: cephumanos@power.ufscar.br

Santa Maria, ____ de _____ de _____.

Sujeito da pesquisa