

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

LUCAS MIGLIORINI DE ARAÚJO

**DESENVOLVIMENTO DA FERRAMENTA AMA: AVALIADOR
MÚLTIPLO DE ACESSIBILIDADE DE PÁGINAS WEB**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA
2017

LUCAS MIGLIORINI DE ARAÚJO

**DESENVOLVIMENTO DA FERRAMENTA AMA: AVALIADOR
MÚLTIPLO DE ACESSIBILIDADE DE PÁGINAS WEB**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. MSc. Geraldo Ranthum

PONTA GROSSA

2017



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa
Diretoria de Pesquisa e Pós-Graduação
Departamento Acadêmico de Informática
Tecnologia em Análise e Desenvolvimento de Sistemas



TERMO DE APROVAÇÃO

DESENVOLVIMENTO DA FERRAMENTA AMA: AVALIADOR MÚLTIPLO DE ACESSIBILIDADE DE PÁGINAS WEB

por

LUCAS MIGLIORINI DE ARAÚJO

Este(a) Trabalho de Conclusão de Curso (TCC) foi apresentado em 23 outubro de 2017 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas. O(a) candidato(a) foi arguido(a) pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Geraldo Ranthum
Prof.(a) Orientador(a)

Clayton Kossoski
Membro titular

Rafael dos Passos Canteri
Membro titular

Prof^a. Dra. Helyane B Borges
Responsável pelo Trabalho de Conclusão
de Curso

Prof^a. Dra. Mauren Louise Sguario
Coordenadora do curso

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

RESUMO

ARAÚJO, Lucas Migliorini de. *Desenvolvimento da Ferramenta AMA: Avaliador Múltiplo de Acessibilidade de Páginas Web*. 2017. 36f. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2017.

Este trabalho tem como objetivo central propor uma melhoria na utilização das ferramentas já disponíveis para avaliar a acessibilidade de um *site* bem como incentivar os desenvolvedores a fazer o uso das regras e das normas estipuladas pela W3C (*World Wide Web Consortium*). Busca também resposta para o seguinte questionamento: Por que apenas pouco mais de 2% de todos os sites disponíveis na Internet, possuem o mínimo necessário para se dizer acessível? A pesquisa contemplou oito etapas metodológicas, a saber: Estudo bibliográfico; Análise das ferramentas e linguagens disponíveis; Comparação do uso das ferramentas utilizadas com outras linguagens; Estudo sobre a documentação dessas ferramentas; Criação do diagrama do processo de funcionamento da aplicação; Desenvolvimento da parte *server-side* do projeto; Tratamento dos dados coletados; Desenvolvimento da parte *front-end* da aplicação. O estudo se caracteriza como explicativo, de abordagem qualitativa e o procedimento adotado é a pesquisa experimental. Os resultados demonstram que há divergências na forma como os *sites* são avaliados entre um *software* avaliador e outro. Adicionalmente, devido à não padronização na forma como os dados são exibidos, dificulta tanto a leitura quanto o entendimento dos resultados. Por fim, espera-se que a visualização desses resultados pode tornar-se mais efetiva e amigável se incorporada as recomendações deste trabalho junto com o possível aumento no percentual de *sites* acessíveis disponíveis na Internet.

Palavras-chaves: Acessibilidade. Interface Humano-Computador. IHC. *Web Design*. Acessibilidade na *Web*. Avaliador de Acessibilidade.

ABSTRACT

ARAÚJO, Lucas Migliorini de. *Development of the MAV Tool: Multiple web Accessibility Validator*. 2017. 36f. Course Completion Assignment (Graduation of Technology in System Analysis and Development) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2017.

This work proposes an improvement in the use of the tools available to evaluate the accessibility of a website as well as to encourage developers to make use of the rules and standards stipulated by the World Wide Web Consortium (W3C). It intends to answer the following question: What is the reason why just a little more than 2% of all sites available on the Internet have the minimum necessary to be considered accessible? The research contemplated eight methodological steps, as follows: Bibliographic study; Analysis of available tools and languages; Comparison between the use of the tools used and other languages; Study on the documentation of these tools; Creation of a diagram of the application functioning process; Development of the back-end part of the application; Treatment of collected data; and Development of the front-end part of the application. The study is characterized as explanatory, with a qualitative approach and the adopted procedure is the experimental research. The results demonstrate that there are differences in the way sites are evaluated when used different types of evaluating softwares. Additionally, due to non-standardization of the way the data is displayed, it makes both reading and understanding the results a difficult task. Finally, it is expected that the visualization of these results can become more effective and friendly if the recommendations of this work are incorporated, resulting in a possible increase in the percentage of accessible sites available in the Internet.

Key-words: Accessibility. Human-Computer Interface. HCI. Web Design. Accessibility on the Web. Accessibility Avaliator.

LISTA DE ILUSTRAÇÕES

Figura 1	– Exemplo diferença de acessibilidade do mundo físico pro mundo virtual ...	12
Figura 2	– Navegação utilizando as setas direcionais, a tecla TAB e H, respectivamente	15
Figura 3	– Teclado de linha braile	16
Figura 4	– Lupa do Windows, ampliador de tela	16
Figura 5	– <i>Mouses</i> e teclados adaptados para pessoas com deficiência motora	17
Figura 6	– Etapas do projeto	18
Figura 7	– Modelo do processo	20
Figura 8	– Árvore de objetos do HTML DOM	21
Figura 9	– Tratamento do endereço fornecido pelo usuário	24
Figura 10	– Definição do cabeçalho do protocolo HTTP.....	25
Figura 11	– Função que realiza a requisição a partir do cabeçalho	25
Figura 12	– Exemplo de Web Scraping do DOM com a biblioteca Cheerio	26
Figura 13	– Resultado da avaliação por prioridades do avaliador <i>AccessMonitor</i>	26
Figura 14	– Resultado da avaliação da ferramenta AMA	26
Figura 15	– Página para o usuário informar o endereço	28
Figura 16	– Página de resposta do servidor após a requisição	29
Figura 17	– Resultado da análise do validador <i>AccessMonitor</i>	29
Figura 18	– Resultado da análise do validador <i>Hera</i>	30

LISTA DE ABREVIATURAS E SIGLAS

AMA	Avaliador Múltiplo de Acessibilidade
CSS	<i>Cascading Style Sheets</i>
CSRF	<i>Cross-site Request Forgery</i>
DGE	Departamento de Governo Eletrônico
DOM	<i>Document Object Model</i>
EMAG	Modelo de Acessibilidade em Governo Eletrônico
HTML	<i>HyperText Markup Language</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IFRS	Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
UTFPR	Universidade Tecnológica Federal do Paraná
XHTML	<i>eXtensible Hypertext Markup Language</i>
XML	<i>eXtensible Markup Language</i>
W3C	<i>World Wide Web Consortium</i>
WAI	<i>Web Accessibility Initiative</i>
WCAG	<i>Web Content Accessibility Guidelines</i>

SUMÁRIO

1 INTRODUÇÃO	8
1.1 CONSIDERAÇÕES INICIAIS	8
1.2 JUSTIFICATIVA	9
1.3 ESCOPO DO TRABALHO	9
1.4 OBJETIVOS	9
1.4.1 Objetivo Geral	9
1.4.2 Objetivos Específicos.....	9
1.5 ESTRUTURA DO TRABALHO	10
2 REFERENCIAL TEÓRICO	11
2.1 ACESSIBILIDADE	11
2.1.1 Acessibilidade na <i>Web</i>	11
2.1.2 Importância da Acessibilidade na <i>Web</i>	12
2.1.3 Prioridades e Diretrizes WCAG	13
2.2 AVALIADORES DE ACESSIBILIDADE	13
2.2.1 Importância dos Validadores de Acessibilidade.....	14
2.2.2 Funcionamento dos Avaliadores	14
2.3 TECNOLOGIAS ASSISTIVAS	15
2.3.1 Exemplos de Tecnologias Assistivas	15
3 METODOLOGIA DE DESENVOLVIMENTO	18
3.1 ETAPAS DO PROJETO	18
4 DESENVOLVIMENTO DA FERRAMENTA	23
4.1 TRAJETÓRIA DO DESENVOLVIMENTO DO SISTEMA	23
4.1.1 Funcionalidades da Ferramenta	23
4.1.2 Funcionamento do <i>Web Scraping</i>	24
4.2 DIFICULDADES ENCONTRADAS DURANTE O PROJETO.....	26
5 RESULTADOS GERAIS	28
5.1 RESULTADOS OBTIDOS	28
5.1.1 Facilidade com o Uso do Sistema.....	28
5.1.2 Layout do Sistema	29
6 CONCLUSÃO	31
6.1 CONSIDERAÇÕES FINAIS	31
6.2 TRABALHOS FUTUROS	31
REFERÊNCIAS	35

1 INTRODUÇÃO

Este capítulo contempla os aspectos introdutórios do presente estudo, abrangendo os seguintes tópicos: considerações iniciais, justificativa, escopo, objetivos, introdução à metodologia utilizada e, por fim, organização estrutural do projeto.

1.1 CONSIDERAÇÕES INICIAIS

Com o objetivo de tornar a Web mais acessível a qualquer pessoa, em 1997 o W3C (*World Wide Web Consortium*) (W3C, 2005), comitê que regula os assuntos ligados à Internet, criou um grupo de trabalho, o WAI (*Web Accessibility Initiative*) — responsável por elaborar um conjunto de diretrizes WCAG (*Web Content Accessibility Guidelines - 1.0*) (W3C, 1999) com o intuito de tornar os *sites* acessíveis às pessoas com deficiência e às que usam a Internet em condições especiais de ambiente, equipamento, navegador e outras ferramentas *Web*.

Com base nas recomendações do WCAG, foram desenvolvidos *softwares* chamados “programas avaliadores” que analisam o nível de acessibilidade de *sites*. Em relação ao Brasil, quando um *site* é aprovado pelo programa avaliador, existe uma orientação governamental para se colocar um selo certificando o nível de acessibilidade alcançado (MELO *et al.*, 2009).

No caso de *sites* brasileiros, o Modelo de Acessibilidade para o desenvolvimento e a adaptação de conteúdo do governo na Internet (eMAG) (EMAG, 2014) determina que estes sejam avaliados pelo programa avaliador ASES (desenvolvida através de uma parceria entre a DGE e a IFRS) (EMAG, 2016).

O problema é que o selo em si não garante que o *site* é acessível a todas as pessoas, pois 23,9% da população brasileira possui algum tipo de deficiência enquanto apenas 2% dos *sites* da Internet são acessíveis a esse público (IBGE, 2010). Toda alteração no *site* deve ser feita cuidadosamente de modo a preservar o nível de acessibilidade já alcançado. A simples preocupação com acessibilidade no momento do desenvolvimento ou da reforma (*redesign*) de um *site* não garante sua manutenção durante as atualizações.

Existe um grande número de programas que analisam o nível de acessibilidade de *sites* cadastrados na W3C/WAI são eles: ASES (EMAG, 2016), DaSilva (DASILVA, 2014), HERA (HERA, 2005), eXaminator (EXAMINATOR, 2009), Taw (TAW, 1999), Cynthia (CINTHYA-SAYS, 2003), Delorie (LYNX, 2010), Wave (WAVE, 2001), entre outros, que realizam tarefas específicas diferentes (W3C, 2016). Eles variam de acordo com a necessidade particular de quem vai usá-lo, e para tal, existe um guia no *site* da W3C que ajuda a decidir qual ferramenta é mais apropriada em cada caso (W3C, 2005). Além dessa questão, cada um desses avaliadores possui seus próprios padrões para exibição dos resultados, em alguns casos, tornando o uso dessas ferramentas de difícil manuseio mesmo para usuários avançados.

1.2 JUSTIFICATIVA

Com esta ferramenta, os desenvolvedores poderão usufruir dos programas avaliadores com maior frequência e agilidade devido a facilidade em se comparar os resultados de todos os avaliadores listados e, desta forma, aumentar o percentual dos *sites* acessíveis na Internet, visto que mais de 23% da população brasileira considera possuir algum tipo de deficiência.

1.3 ESCOPO DO TRABALHO

O escopo deste trabalho está no desenvolvimento da ferramenta AMA (Avaliador Múltiplo de Acessibilidade) por meio de um sistema *Web* baseado nas normas da WCAG visando favorecer tanto aos desenvolvedores na visão de *back-end*¹, na visão *front-end*, oferecendo facilidade de uso, além de eficiência, qualidade, autonomia e segurança para os usuários. Inicialmente o foco deste projeto é realizar a comparação apenas entre dois avaliadores, avaliador AccessMonitor (ACCESSMONITOR, 2013) e o avaliador Hera (HERA, 2005).

1.4 OBJETIVOS

Nesta seção serão apresentados o objetivo geral e os objetivos específicos deste trabalho.

1.4.1 Objetivo Geral

Desenvolver uma ferramenta que apresente o resultado de dois *sites* avaliadores de acessibilidade em uma tabela comparativa.

1.4.2 Objetivos Específicos

- a) Analisar o funcionamento dos *sites* avaliadores de acessibilidade;
- b) Fazer um levantamento de quais ferramentas serão utilizadas para realizar a extração do conteúdo das páginas *Web*;

¹ O *front-end* é responsável pela interface e interação do usuário com a aplicação enquanto *back-end* está relacionado a regra de negócio da aplicação. (TREINAWEB, 2017)

- c) Examinar o funcionamento dos *softwares* leitores de tela;
- d) Construir e desenvolver uma página *Web* para instalar a ferramenta;
- e) Extrair os resultados analisados de cada site selecionando-os com maior relevância;
- f) Reexibir os resultados da extração por ordem de relevância com as normas e regras do W3C/WAI.

1.5 ESTRUTURA DO TRABALHO

O segundo capítulo compreende a Revisão da Literatura, que está subdividida em: Acessibilidade, Avaliadores de Acessibilidade e Tecnologias Assistivas.

No terceiro capítulo é descrita a Metodologia de Desenvolvimento e são apresentadas as Etapas do Projeto.

O quarto capítulo apresenta o desenvolvimento do trabalho. Ele está subdividido em Trajetória do Desenvolvimento e Dificuldades Enfrentadas.

Por fim, o quinto capítulo se refere à Organização dos Resultados Obtidos.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta algumas ferramentas avaliadoras de acessibilidade, bem como seu processo de funcionamento no que diz respeito a acessibilidade Web.

2.1 ACESSIBILIDADE

Acessibilidade, de um modo geral, está associado à eliminação de barreiras arquitetônicas. Conforme o artigo 2 da Lei 10.098:

Acessibilidade: possibilidade e condição de alcance para utilização, com segurança e autonomia, de espaços, mobiliários, equipamentos urbanos, edificações, transportes, informação e comunicação, inclusive seus sistemas e tecnologias, bem como de outros serviços e instalações abertos ao público, de uso público ou privados de uso coletivo, tanto na zona urbana como na rural, por pessoa com deficiência ou com mobilidade reduzida.

(BRASIL, 2000)

As pessoas podem ter necessidades especiais porque estão temporariamente em uma situação, por exemplo, que torna as mãos incapazes de trabalhar com o teclado, ou com nível de ruído tão alto que não pode ouvir um leitor de tela ou outra forma de necessidade. A fim de ajudar as pessoas com necessidades especiais, é necessário identificar suas necessidades e requisitos (NEVILE, 2002).

A pessoa com deficiência física por lesão medular apresenta perdas ou reduções em sua estrutura física, entretanto sua personalidade e modo de agir permanecem os mesmos. Prover acessibilidade para essas pessoas é uma preocupação constante sendo promovida pelos órgãos públicos e privados a fim de fornecer condições às pessoas com necessidades especiais, para utilização dos meios físicos e de comunicação com segurança e autonomia (BRASIL, 2000).

2.1.1 Acessibilidade na *Web*

De acordo com a Cartilha de Acessibilidade na *Web*:

Acessibilidade na Web significa que pessoas com deficiência podem usar a Web. Mais especificamente, a acessibilidade na Web significa que pessoas com deficiência podem perceber, entender, navegar, interagir e contribuir para a Web. Ela também beneficia outras pessoas, incluindo pessoas idosas com capacidades em mudança devido ao envelhecimento.

(W3CBRASIL, 2013)

A Figura 1 apresenta uma dessas diferenças entre o mundo real e o virtual.

Mundo físico - Nos prédios há rampas para acesso e a pessoa com deficiência física utiliza a cadeira de rodas para entrar.

Mundo Virtual - Os sites são desenvolvidos utilizando-se código acessível e a pessoa cega utiliza um leitor de tela para navegar.

Figura 1 – Exemplo diferença de acessibilidade do mundo físico pro mundo virtual



Fonte: Manual eMAG para desenvolvedor módulo 1 - (EMAG, 2015)

Acessibilidade na *Web* significa garantir que todas as pessoas tenham direito de acesso à informação e comunicação através de qualquer tecnologia. “Pressupõe que os *sites* e portais sejam projetados de modo que seja possível perceber, entender, navegar e interagir de maneira efetiva com as páginas” (EMAG, 2015).

2.1.2 Importância da Acessibilidade na *Web*

Com o mundo cada vez mais conectado, é visível a influência que a Internet tem na vida das pessoas (MACEDO; CARVALHO, 2010) seja a nível de usuário cotidiano seja como de pessoas que dependem deste meio como forma de trabalho. A Internet também é utilizada como meio de comunicação e informativo, e como qualquer meio informativo, é necessário adequar de uma maneira que todos tenham acesso.

De acordo com a W3C Brasil: A influência da *Web* abrange cada vez mais áreas da atividade humana e torna-se imprescindível a sua disponibilidade de utilização, seja nos meios comerciais ou domésticos (W3CBRASIL, 2013). Logo, é necessário dar destaque especial à essa parcela da população que acaba por não ter acesso a maioria do conteúdo disponível na *Web*.

As pessoas com deficiência também trabalham, estudam, pesquisam e vivem como qualquer outra, ou seja, tornar a *Web* mais acessível a nível de empresas, significa ter mais consumidores e clientes, mais alunos de ensino superior a distância, mais trabalhadores em sites virtuais, ou seja, mais usuários e conseqüentemente maiores lucros. Quase um quarto da população brasileira é ignorada no que diz respeito a *Web*. Os benefícios de se ter uma *Web* mais acessível são para ambas as partes, tanto para as pessoas com deficiência ou para quem desenvolve os *sites* (W3CBRASIL, 2013).

2.1.3 Prioridades e Diretrizes WCAG

De acordo com WCAG 1.0 (W3C, 1999), os conteúdos de uma página *Web* podem ser classificados em três níveis de prioridades:

Prioridade 1: Os desenvolvedores *Web* **precisam** satisfazer estes requerimentos, caso contrário será impossível para um ou mais grupos acessarem o conteúdo *Web*. Nível de conformidade *Level* “A” é atendido quando todos os requisitos para a prioridade 1 são atendidos.

Prioridade 2: Os desenvolvedores *Web* **deveriam** satisfazer estes requerimentos, caso contrário alguns grupos podem não conseguir ou terão dificuldade em acessar o conteúdo *Web*. Nível de conformidade *Level* “AA” é atendido quando todos os requisitos para a prioridade 1 e prioridade 2 são atendidos.

Prioridade 3: Os desenvolvedores *Web* **podem** satisfazer estes requerimentos para facilitar que alguns grupos acessem o conteúdo *Web*. Nível de conformidade *Level* “AAA” é atendido quando todos os requisitos para a prioridade 1, prioridade 2 e prioridade 3 são atendidos.

2.2 AVALIADORES DE ACESSIBILIDADE

Com uso das normas definidas pela WAI, desenvolvedores criaram ferramentas que validam a acessibilidade dos *websites* de forma automática. Porém, por não serem capazes de validar todas as vertentes da acessibilidade é importante que sejam utilizadas essas ferramentas nas fases iniciais do desenvolvimento (QUEIROZ, 2008).

Conforme explica Marco Antônio Queiroz, consultor especialista em acessibilidade digital: “Os avaliadores de acessibilidade são ferramentas automáticas que após analisarem uma página *Web*, emitem relatórios indicando erros e avisos com bases nas normas e diretrizes da WCAG” (QUEIROZ, 2008).

Essas ferramentas permitem ao desenvolvedor da página verificar os eventuais erros no desenvolvimento das páginas *Web* para então corrigi-los e, principalmente, auxilia no sentido de tornar o site mais acessível para as pessoas com deficiência e necessitam equipamentos especiais

para navegar por essas páginas.

2.2.1 Importância dos Validadores de Acessibilidade

Proporcionar um *site* acessível manualmente durante sua construção requer muito tempo. É necessário que o desenvolvedor do *site* revise todo o código em busca de cada parâmetro, *tags*² e *links* que possam não estar em conformidade com as diretrizes da WCAG. Essa revisão deve ser feita em cada documento, página e folha criados, sejam esses arquivos (X)HTML, XML, folhas de estilo como CSS ou mesmo no que diz respeito as imagens, vídeos, áudios e inclusive o contraste e conjunto de cores utilizadas no desenvolvimento da página.

As ferramentas chamadas de validadores de acessibilidade fazem todo esse processo descrito acima de uma forma automática, alertando o desenvolvedor sobre os erros e avisos encontrados em questão de segundos. Cabe ao criador verificar e corrigir os erros encontrados manualmente de uma forma mais eficaz (QUEIROZ, 2008).

A importância da aplicação de acessibilidade em um *site* também reflete no cumprimento legal do Decreto nº 5.296 de 2 de dezembro de 2004, que, em seu artigo 47 trata da acessibilidade virtual (BRASIL, 2004).

2.2.2 Funcionamento dos Avaliadores

Os validadores percorrem os códigos desenvolvidos em busca de determinadas *tags*, (como por exemplo a *tag* ‘h’ em busca de títulos e subtítulos de notícias, e a *tag* ‘p’ para navegar por entre os parágrafos de um texto), as quais algumas tecnologias assistivas utilizam para traduzir os significados, verificando se o conteúdo e a construção estão de acordo com as normas e diretrizes estabelecidas pela WCAG.

Ao realizar a análise do *site*, é gerado um relatório o qual contém erros e avisos que o validador aponta ao ler o código proposto. Em geral, a quantidade de avisos é maior que a quantidade de erros apontados, isso devido ao fato da limitação desses validadores em testar as regras aplicadas. Regras essas que são necessárias a interferência humana para decidir determinados conteúdos de *tags* tal como se a descrição de uma imagem condiz efetivamente com seu conteúdo, ou elementos que possuem um posicionamento absoluto no *site* não interferem na legibilidade caso seja necessário ampliar ou reduzir tamanho de fonte, garantindo assim a clareza da linguagem e facilidade de navegação (W3C, 2017).

Existem diferentes tipos de validadores, que analisam a sintaxe (HTML, XML, etc.), folhas de estilo (CSS), verificação de contrastes, etc., e para cada tipo, existem dezenas de valida-

² *Tag* é um elemento que compõe o DOM de uma página HTML.

dores diferentes com critérios que diferenciam um do outro, por isso é interessante utilizar mais de um desses avaliadores para garantir um bom resultado em termos de acessibilidade (W3C, 2016).

2.3 TECNOLOGIAS ASSISTIVAS

Tecnologia Assistiva é um termo relativamente novo no Brasil e refere-se aos recursos que contribuem para que pessoas com deficiência possam ter uma vida mais independente e autônoma (EMAG, 2015).

2.3.1 Exemplos de Tecnologias Assistivas

Como exemplos de recursos de Tecnologia Assistiva voltados para *Web* podemos citar:

- **Softwares Leitores de Tela** – Programas que auxiliam na leitura dos elementos exibidos na tela do computador.

Conforme especifica o eMAG:

Geralmente usado por pessoas cegas, fornece a informação através da síntese de voz através dos elementos exibidos na tela. Transcreve o conteúdo escrito para uma resposta falada usando um sintetizador de voz. (EMAG, 2015)

A Figura 2 apresenta um exemplo do funcionamento de um *software* leitor de tela. Na sequência, a Figura “a” apresenta a navegação com utilização das setas direcionais, a Figura “b” mostra a navegação com a utilização da tecla TAB e a Figura “c” mostra a navegação utilizando-se a tecla H.

Figura 2 – Navegação utilizando as setas direcionais, a tecla TAB e H, respectivamente



Fonte: Manual eMAG para desenvolvedor módulo 1 (EMAG, 2015)

- **Linha Braille** – Instrumento que, quando conectado ao computador, é capaz de traduzir linhas de textos para o Braille.

Conforme especifica o eMAG:

Equipamento eletrônico que transforma o conteúdo em informação tátil, exibida dinamicamente em Braille. É ligado ao computador por cabo, que possui uma linha régua de células Braille, cujos pinos se movem para cima e para baixo, representando uma linha de texto da tela do computador.

(EMAG, 2015)

A Figura 3 mostra o equipamento de Linha Braille.

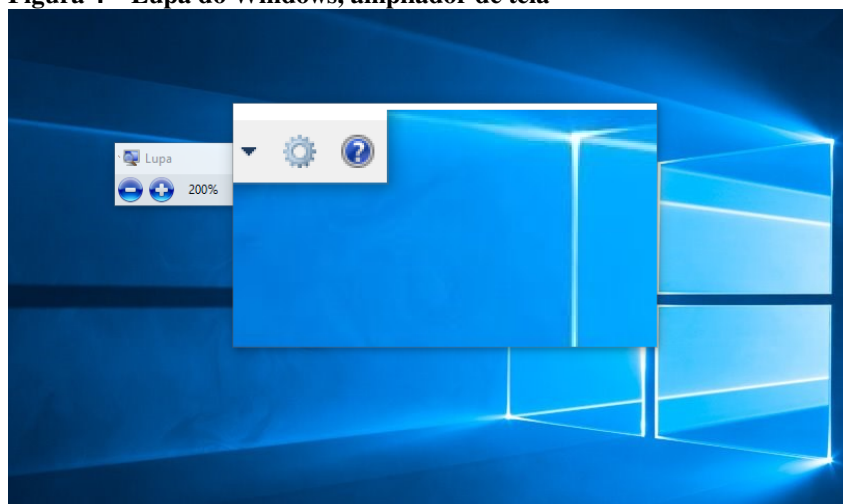
Figura 3 – Teclado de linha braille



Fonte: VisionAware - (VISIONAWARE, 2015)

- **Amplificadores de Tela** – *Softwares* utilizados por pessoas com baixa visão que ampliam toda ou uma parte da tela. A Figura 4 mostra um exemplo de amplificador de tela utilizando a ferramenta do sistema operacional *Windows*, a lupa:

Figura 4 – Lupa do Windows, amplificador de tela



Fonte: *Print screen* do sistema operacional Windows 10

- Teclado ou *mouses* adaptados – Equipamentos convencionais de teclado e *mouse* podem não suprir as necessidades de quem possui algum tipo de deficiência, logo, equipamentos e programas alternativos foram criados a fim de atender tais necessidades.

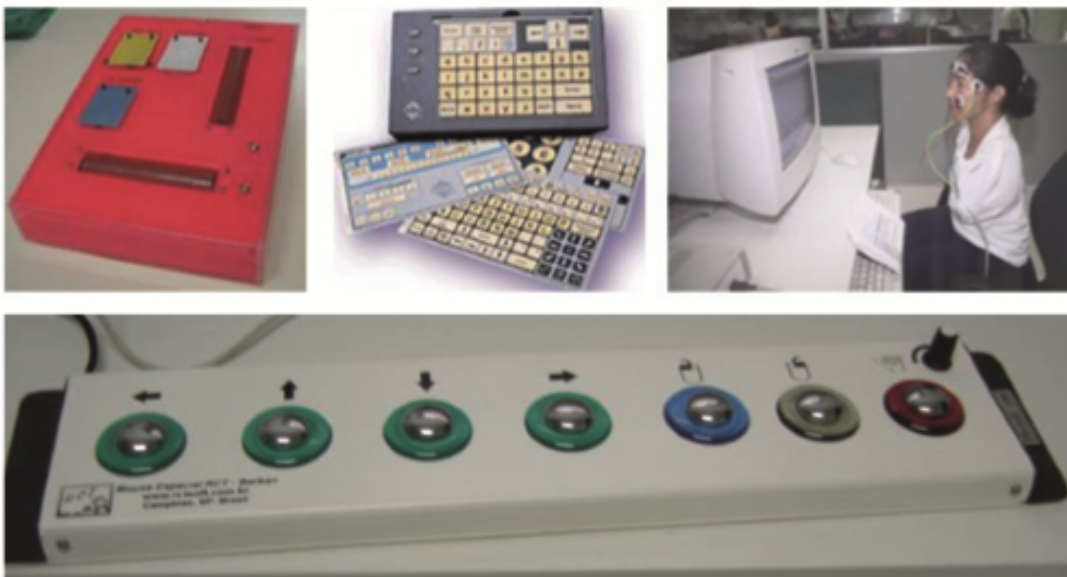
Conforme especifica o eMAG:

Representam recursos de hardware ou software utilizados por pessoas com deficiência física ou mobilidade reduzida, que proporcionam uma maneira alternativa de utilizar as teclas. Exemplos incluem teclados que possuem um espaçamento maior ou menor entre as teclas, teclado de lâminas, teclado virtual, acionadores que realizam a função dos botões do mouse, mouse de sopro, mouse ocular, etc. Apesar da sua importância, eles não garantem o acesso se as páginas Web não forem projetadas visando a acessibilidade.

(EMAG, 2015)

A Figura 5 mostra exemplos de tecnologias alternativas:

Figura 5 – Mouses e teclados adaptados para pessoas com deficiência motora



Fonte: Manual eMAG para desenvolvedor módulo 1 - (EMAG, 2015)

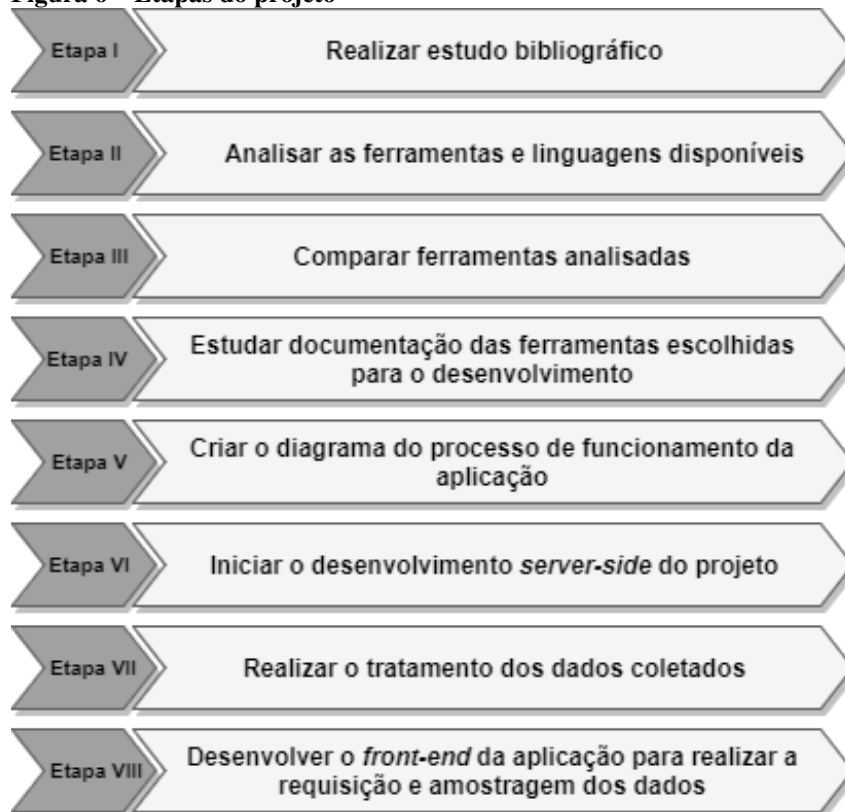
3 METODOLOGIA DE DESENVOLVIMENTO

O presente capítulo apresenta os procedimentos metodológicos utilizados para o desenvolvimento da proposta de avaliador múltiplo de acessibilidade de páginas *Web*.

3.1 ETAPAS DO PROJETO

Para o desenvolvimento deste trabalho de conclusão de curso foram realizadas oito etapas de desenvolvimento, que são apresentadas na Figura 6 e descritas de forma detalhada nas páginas que seguem.

Figura 6 – Etapas do projeto



Fonte: Autoria própria

Etapa I – Realizar Estudo Bibliográfico

A fim de direcionar esforços para a compreensão dos temas diretamente relacionados ao presente trabalho, o estudo contemplou revisão de literatura em bibliografia especializada com foco nos seguintes temas:

- Acessibilidade;
- Avaliadores de acessibilidade;

- Tecnologias assistivas.

Etapa II – Analisar as Ferramentas e Linguagens Disponíveis

No intuito de proporcionar o desenvolvimento ideal da aplicação, foram analisadas e estudadas linguagens de programação *server-side* para desenvolvimento *Web*. Dentre as linguagens estudadas, estão: *PHP*, *Java*, *Python* e *JavaScript (Node.js)* como algumas das mais utilizadas linguagens no mundo (TIOBE, 2016).

Etapa III – Comparar Ferramentas Analisadas

As ferramentas *Node.js* juntamente com a linguagem *JavaScript* foram as escolhidas para o desenvolvimento do projeto. Por conseguir trabalhar com múltiplas requisições HTTP sem necessitar do uso de *threads*, a linguagem *Node.js* economiza os recursos físicos do computador, como processamento e memória, proporcionando alto grau de escalabilidade (NODEBR, 2016).

Node.js, diferentemente das demais linguagens *server-side*, trabalha sobre um modelo *single thread* e utiliza um recurso de entrada e saída orientado a eventos não bloqueantes, permitindo a execução assíncrona das funções em tempo real (NODEBR, 2016).

Uma vez que o intuito da ferramenta a ser desenvolvida é de enviar requisições simultâneas para diversos avaliadores de forma assíncrona, foi escolhido junto ao *Node.js*, a linguagem *JavaScript* para manipular os eventos tanto do *front-end* como do *back-end* por consumir apenas uma *thread* em comparação às linguagens *PHP*, *Java* e *Python*, que utilizam uma *thread* para cada requisição feita de forma simultânea.

Etapa IV – Estudar Documentação das Ferramentas Escolhidas para o Desenvolvimento

Buscando responder aos objetivos específicos deste projeto, foram estudadas as documentações (funções e módulos) que nortearam todo o desenvolvimento da aplicação. Estes foram extraídos do site oficial da ferramenta *Node.js* (NODEJS, 2017).

A seleção dos módulos foi realizada considerando a pertinência de aplicação na construção do projeto. Nesse sentido, foram identificados 4 módulos, a saber:

- I. *Express* – Módulo que permite a geração automática dos recursos necessários para criar uma aplicação *Web*. Será utilizado como *framework* do projeto em questão (EXPRESS, 2017).
- II. *HTTP* – Módulo que permite iniciar um servidor localmente. Será utilizado para criar o servidor em questão e, conseqüentemente, alocar a aplicação a ser desenvolvida (NODEJS, 2017).

- III. *Request* – Módulo que permite o servidor realizar uma requisição a qualquer *Website*. Será utilizado para realizar o preenchimento de formulários digitados anteriormente pelo usuário (REQUEST, 2017).
- IV. *Cheerio* – Módulo que permite percorrer o HTML DOM de uma página. Será utilizado para realizar o tratamento de dados, detalhado na etapa VIII deste documento (CHEERIO, 2016).

Etapa V – Criar o Diagrama do Processo de Funcionamento da Aplicação

Com intuito de facilitar a compreensão do funcionamento do projeto, foi modelado o processo desde a interação do usuário com a aplicação até a resposta emitida pelo servidor após a coleta e o tratamento dos dados.

O modelo está apresentado na Figura 7:

Figura 7 – Modelo do processo



Como é possível observar na Figura 7, o usuário acessa a ferramenta desenvolvida neste projeto, insere um endereço de uma página *Web* a ser analisada e envia a solicitação. O servidor da aplicação fica encarregado de realizar essa mesma solicitação para ambos os avaliadores de forma simultânea, aguardar o processamento da análise, para então retornar os resultados já comparados para o usuário.

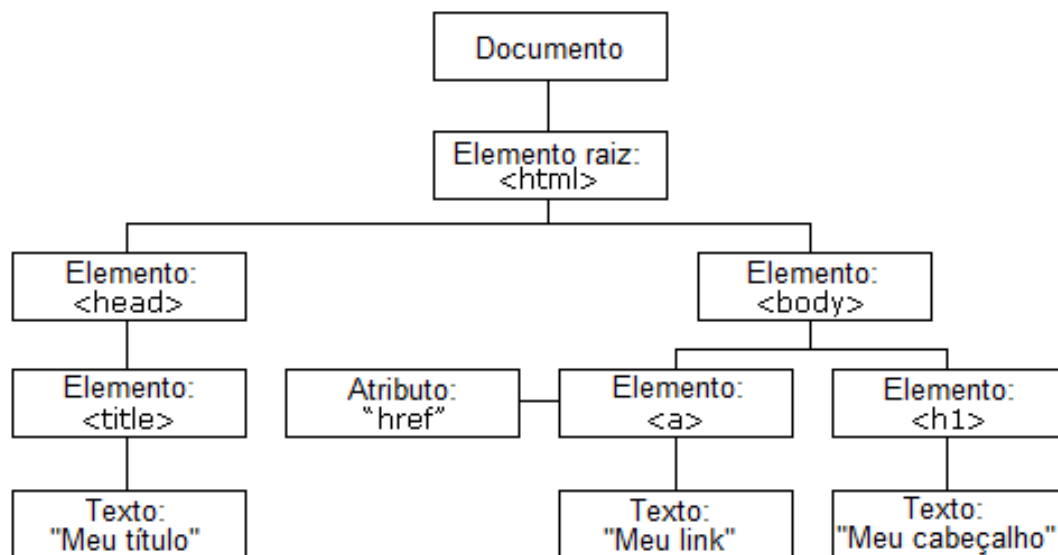
Etapa VI – Iniciar o Desenvolvimento *Server-Side* do Projeto

Para dar início, foi desenvolvido o *software* do lado do servidor. Foram incluídos os métodos para estruturar o projeto e iniciar o servidor numa determinada porta, utilizando dos módulos *express* e *HTTP* respectivamente.

Etapa VII – Realizar o Tratamento dos Dados Coletados

Quando uma página *Web* é carregada, o navegador de Internet gera o HTML DOM (*Document Object Model*) dessa página. O modelo HTML DOM é composto por uma árvore de objetos, sendo eles os elementos estruturais de qualquer site. A seguir, é apresentado a Figura 8, exemplificando essa árvore:

Figura 8 – Árvore de objetos do HTML DOM



Fonte: Adaptado de W3Schools - (W3SCHOOLS, 2017)

Como a linguagem *JavaScript* é capaz de manipular os elementos DOM durante a requisição do servidor, o HTML DOM dessas páginas foi percorrido e então foram extraídos apenas os elementos relevantes para a aplicação.

Cada página *Web* possui sua própria estrutura DOM, então, cada uma dessas páginas possuem uma regra e função específica para o tratamento desses dados propriamente desenvolvidas no servidor.

Etapa VIII – Desenvolver o *Front-End* da Aplicação para Realizar a Requisição e Amostragem dos Dados

Para concluir o desenvolvimento do projeto, foi criada a interface para o usuário realizar a requisição das páginas. Essa interface dispõe essencialmente de um campo para inserir a página a ser analisada e um botão para realizar a solicitação. Como resposta, uma nova página é gerada com os resultados das prioridades devidamente tratadas na etapa VII.

4 DESENVOLVIMENTO DA FERRAMENTA

Neste capítulo, são descritas a trajetória de desenvolvimento do sistema bem como as dificuldades encontradas durante esta fase.

4.1 TRAJETÓRIA DO DESENVOLVIMENTO DO SISTEMA

Com base na revisão da literatura sobre acessibilidade e validadores de acessibilidade, foram realizados procedimentos de desenvolvimento visando inovação e melhoria para os desenvolvedores de *Website* no momento de analisar as páginas desenvolvidas. São eles:

- I. Funcionalidades da Ferramenta;
- II. Funcionamento do *Web Scraping*³.

4.1.1 Funcionalidades da Ferramenta

Com base no modelo de processo realizado, o projeto foi dividido em dois módulos, sendo estes definidos como rotas e servidor.

Módulo de rotas: nesta etapa, foram criadas as rotas de acesso ao servidor, ou seja, as devidas requisições que o usuário pode fazer para o servidor uma vez dentro da página. Foram construídas três páginas (*Views*) as quais o usuário consegue o acesso, sendo elas:

Página principal (nomeada '*Home*'), página para realizar a requisição e consequente análise do site (nomeada '*Requesti*') e a página com o resultado da requisição (nomeada '*Dados*').

Na página *Home*, há uma breve descrição do funcionamento sobre a ferramenta e um menu para acesso a página de requisição.

Na página *Requesti*, há um campo para inserir o endereço da página a ser analisada com um botão para realizar a requisição.

Na página *Dados*, há o resultado da requisição com duas tabelas respectivas aos avaliadores *AccessMonitor* e *Hera*, tabela na qual estão o total de acertos e erros por prioridades de avaliação coletadas e tratadas dos avaliadores em questão; a pontuação parcial de cada avaliador; a pontuação AMA; o *link* para acesso ao resultado completo na página dos respectivos avaliadores.

³ *Web Scraping* é o termo utilizado quando um programa percorre pelo HTML de uma página a fim de coletar dados (TECHOPEDIA, 2017).

Módulo do servidor: neste módulo, são feitos os apuramentos dos dados recebidos pelos usuários; tratamento das rotas; requisição aos avaliadores; tratamento dos resultados; redirecionamento dos usuários.

4.1.2 Funcionamento do *Web Scraping*

Dentre as funcionalidades do sistema, destaca-se: Requisição aos avaliadores; Coleta e tratamento dos resultados.

Requisição aos avaliadores: nesta funcionalidade foi explorado o pacote *Request* para *JavaScript*. Após a solicitação, o servidor trata o endereço fornecido pelo usuário, define o cabeçalho da requisição e envia a solicitação para o avaliador.

Como os avaliadores possuem suas próprias particularidades quanto à forma de entrada do endereço de URL, é feito o tratamento para padronizar e evitar erros nas solicitações. Esse tratamento verifica o protocolo HTTP/HTTPS e se o endereço possui o WWW da URL para então atribuir a variável, conforme a Figura 9:

Figura 9 – Tratamento do endereço fornecido pelo usuário

```

1   if (req.body.siteurl.indexOf('http://') !== -1) {
2       if (req.body.siteurl.indexOf('www.') !== -1) {
3           request_url = req.body.siteurl.split('http://www.')[1];
4       } else {
5           request_url = req.body.siteurl.split('http://')[1];
6       }
7   } else if (req.body.siteurl.indexOf('https://') !== -1){
8       if (req.body.siteurl.indexOf('www.') !== -1) {
9           request_url = req.body.siteurl.split('https://www.')[1];
10      } else {
11          request_url = req.body.siteurl.split('https://')[1];
12      }
13  } else if (req.body.siteurl.indexOf('www.') !== -1){
14      request_url = req.body.siteurl.split('www.')[1];
15  } else {
16      request_url = req.body.siteurl;
17  }

```

Fonte: Autoria própria

Após esse processo, é feita a configuração do cabeçalho da requisição, contendo o protocolo de comunicação entre o servidor da aplicação e o avaliador em questão. Neste cabeçalho, é inserido o endereço da solicitação (do avaliador), a forma como será feita esta requisição (neste caso será acessado o formulário do avaliador), o tipo de requisição trabalhada (via *POST*), o dado que será passado ao formulário (o endereço fornecido pelo usuário devidamente tratado)

e, por final, utilizado o parâmetro para seguir o redirecionamento de página caso o avaliador o faça, conforme a Figura 10:

Figura 10 – Definição do cabeçalho do protocolo HTTP

```

1   var headers          = {
2     'User-Agent':      'Super Agent/0.0.1',
3     'Content-Type':    'application/x-www-form-urlencoded'
4   }
5
6   var options_acessmonitor = {
7     url: 'http://www.acessibilidade.gov.pt/accessmonitor/',
8     method: 'POST',
9     followAllRedirects: true,
10    jar: true,
11    headers: headers,
12    form: {'uri': request_url}
13  };

```

Fonte: Autoria própria

Por fim, é realizada a requisição ao avaliador com o devido cabeçalho, conforme a Figura 11:

Figura 11 – Função que realiza a requisição a partir do cabeçalho

```

1 | request(options_acessmonitor, function (error, response, body)

```

Fonte: Autoria própria

Todo esse processo descrito acima é replicado para cada avaliador que deseje realizar a requisição. Como a aplicação utiliza a plataforma *Node.js*, as requisições ocorrem de forma independente, ou seja, simultaneamente realizam as solicitações aos avaliadores, tendo como tempo de resposta total, o avaliador que tiver o maior tempo de resposta.

Coleta e tratamento dos resultados: nesta funcionalidade, é verificada a resposta da solicitação conforme descrita acima. Foi utilizado o pacote da biblioteca *Cheerio* para *JavaScript* para auxiliar no *Web Scraping* dos dados. O DOM da página de resposta do avaliador é carregado por essa biblioteca e então salvo dentro uma variável. A partir desta variável (que contém todo o conteúdo pré-selecionado da página requisitada), será percorrida seu conteúdo para localizar as informações relevantes para então tratá-las e reexibir para o usuário. A Figura 12 demonstra como esse processo é feito, usando como exemplo, a localização da prioridade “A” (pela classe ‘lev_A’) enquanto a Figura 13 mostra o respectivo dado no *site* do avaliador.

Figura 12 – Exemplo de Web Scraping do DOM com a biblioteca Cheerio

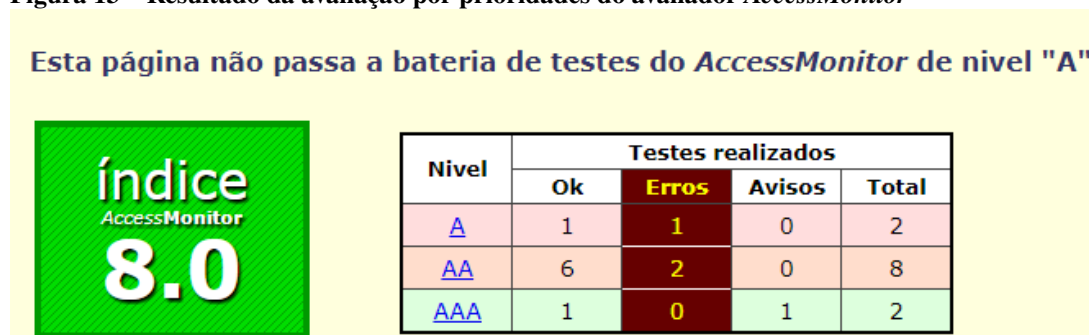
```

1 let $ = cheerio.load(body);
2 let prioridade_p1 = ('table.levels').children().children('.
   lev_A').children('.txfail').prev();
3 let prioridade_p2 = $('table.levels').children().children('.
   lev_AA').children('.txfail').prev();
4 let prioridade_p3 = ('table.levels').children().children('.
   lev_AAA').children('.txfail').prev();
5 parsedResults.push({'monitor_p': prioridade_p1.text()}, {'
   monitor_p': prioridade_p2.text()}, {'monitor_p':
   prioridade_p3.text()});
6 parcial_monitor += parseInt(prioridade_p1.text()+parseInt(
   prioridade_p2.text()+parseInt(prioridade_p3.text()));

```

Fonte: Autoria própria

Figura 13 – Resultado da avaliação por prioridades do avaliador AccessMonitor



Fonte: Print screen do site AccessMonitor

Repetido esse processo de *Web Scraping* para as demais prioridades da tabela, o servidor redireciona o usuário para a página Dados com todas essas informações distribuídas em tabelas, para uma padronização desses dados, conforme ilustra a Figura 14:

Figura 14 – Resultado da avaliação da ferramenta AMA

	Prioridade P1: A	Prioridade P2: AA	Prioridade P3: AAA	Total
Erros:	1	1	0	
Acertos:	1	0	0	
Avisos:	9	1	1	
Nota PARCIAL:				86

Fonte: Print screen da aplicação AMA

4.2 DIFICULDADES ENCONTRADAS DURANTE O PROJETO

Durante o desenvolvimento do trabalho, algumas dificuldades foram enfrentadas. Logo no início do projeto, decidiu-se por criar uma aplicação na linguagem *JavaScript* com a plataforma *Node.js*, ambiente ausente em quaisquer disciplinas do curso e sem experiência do pesqui-

sador do projeto em questão, que melhor responderiam as solicitações e recursos que a aplicação exigiria.

A primeira dificuldade ocorreu no momento de realizar as requisições nos *sites* validadores os quais possuem em seu formulário, campos do tipo *'hidden'*, ou seja, campos que visualmente não aparecem para o usuário comum que, porém, são enviados nessa mesma requisição.

A questão é que esses campos especiais, possuem em seu atributo um gerador de *token*, isto é, um autenticador que verifica se o usuário em questão é o mesmo que está realizando a requisição no *site*, esse *token* é gerado tanto no *client-side* quanto no *server-side* para serem comparados na sessão atual do usuário quando este acessa a página.

É uma forma de garantir a segurança do usuário contra *Cross-Site Request Forgery* (CSRF), um tipo de ataque ocorre quando um *site*, *email*, *blog*, mensagem instantânea ou programa malicioso faz com que o navegador da *Web* do usuário execute uma ação indesejada em um *site* confiável para o qual o usuário está atualmente autenticado (OWASP, 2017).

Outra dificuldade foi durante o tratamento dos dados analisados pelo HTML DOM. Como cada página *Web* possui padrões e objetos diferentes em sua composição, a forma os quais esses dados eram exibidos variavam em vários aspectos:

- As *tags* poderiam não conter *id's*⁴ ou classes⁵ vinculadas como atributos, dificultando o *Web Scraping* destes dados.
- Os resultados poderiam ser gerados dinamicamente com outros tipos de *token* para visualização, o que exigia realizar mais uma requisição aninhada na função programada no projeto.

⁴ *Id* é um nome que identifica unicamente em todo o contexto do DOM um determinado elemento.

⁵ Classes é o nome que pode identificar um ou mais elementos do DOM de uma página HTML.

5 RESULTADOS GERAIS

Este capítulo apresenta os resultados obtidos através das requisições realizadas por esta aplicação.

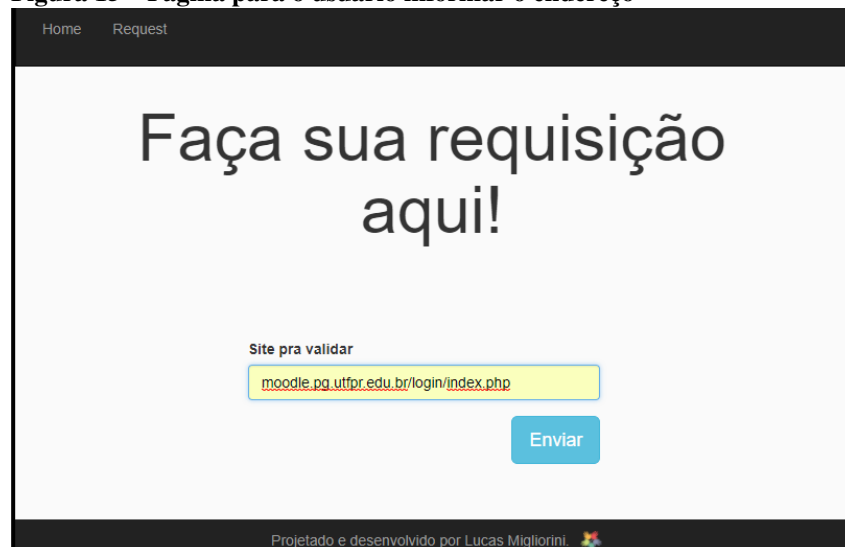
5.1 RESULTADOS OBTIDOS

Com a finalização da ferramenta, foi disponibilizado de forma gratuita e online através do seguinte endereço: <<https://mav-app.herokuapp.com/>>.

5.1.1 Facilidade com o Uso do Sistema

A ferramenta desenvolvida apresenta em sua página inicial, um campo para inserir o endereço do *site* que será analisado e um botão para efetuar a requisição. Após concluída esta análise, a ferramenta atualiza a página e as respectivas classificações das normas para cada um dos avaliadores de forma padronizada, buscando facilitar a legibilidade, conforme apresentado nas Figuras 15 e 16 a seguir:

Figura 15 – Página para o usuário informar o endereço

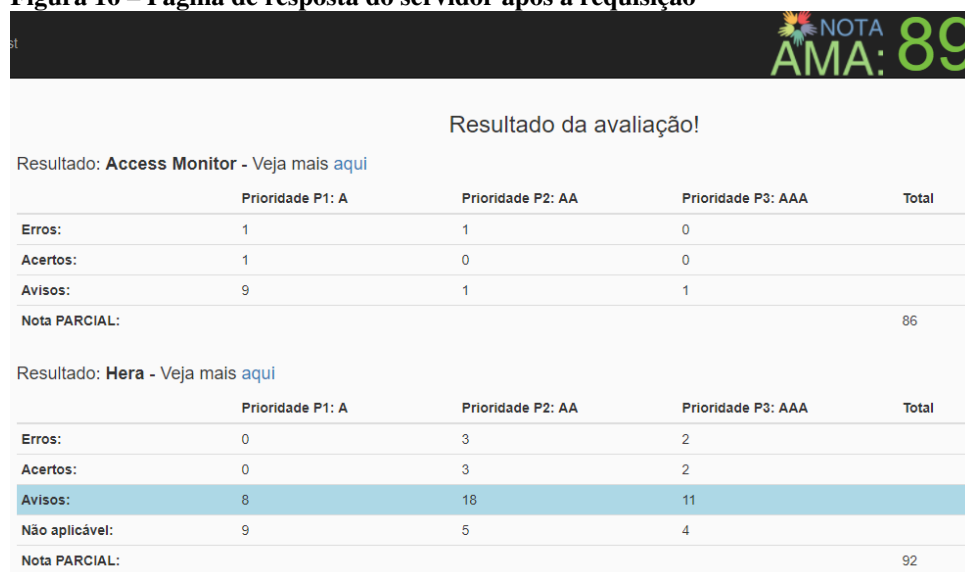


Fonte: *Print screen* da aplicação AMA

Como cada avaliador utiliza um sistema de pontuação ao final de uma avaliação, a ferramenta AMA também conta com um sistema próprio de pontuação ao final do resultado. Essa pontuação é uma média aritmética das notas dos avaliadores (soma de ambas as notas dos avaliadores dividido por dois), e então disponibilizada na página da ferramenta juntamente com o resultado, no canto superior direito da aplicação, com isso o desenvolvedor consegue uma

visão geral sobre o grau de acessibilidade que sua aplicação possui, como pode ser visto na Figura 16:

Figura 16 – Página de resposta do servidor após a requisição



Resultado da avaliação!				
Resultado: Access Monitor - Veja mais aqui				
	Prioridade P1: A	Prioridade P2: AA	Prioridade P3: AAA	Total
Erros:	1	1	0	
Acertos:	1	0	0	
Avisos:	9	1	1	
Nota PARCIAL:				86
Resultado: Hera - Veja mais aqui				
	Prioridade P1: A	Prioridade P2: AA	Prioridade P3: AAA	Total
Erros:	0	3	2	
Acertos:	0	3	2	
Avisos:	8	18	11	
Não aplicável:	9	5	4	
Nota PARCIAL:				92

Fonte: *Print screen* da aplicação AMA


5.1.2 Layout do Sistema

Dispondo de uma interface simples e amigável, a ferramenta desenvolvida ainda salva o retorno da avaliação individual de cada validador para que o usuário possa conferir o restante dos detalhes, conforme ilustra as Figuras 17 e 18 a seguir:


Figura 17 – Resultado da análise do validador *AccessMonitor*

pele *AccessMonitor*. O índice e um indicador que se destina ao uso exclusivo dos criadores do site Web. Todos os testes do *AccessMonitor* tem a sua fundamentação nas WCAG 2.0 do W3C.

Esta página não passa a bateria de testes do *AccessMonitor* de nível "A"



Nível	Testes realizados			
	Ok	Erros	Avisos	Total
A	1	1	9	11
AA	0	1	1	2
AAA	0	0	1	1



[\[versão linear\]](#)

II. Detalhe dos testes realizados

Elementos HTML encontrados na página	Erros		
	A	AA	AAA
<input type="checkbox"/> Texto alternativo em imagens			
<input type="checkbox"/> Marcação de cabeçalhos			
<input type="checkbox"/> Marcação de Links, menus e texto dos links			
<input type="checkbox"/> Links para contornar blocos de informação			
<input type="checkbox"/> Marcação de formulários			
<input type="checkbox"/> Standards W3C: (X)HTML + CSS	1		
<input type="checkbox"/> Elementos e atributos de apresentação/obsoletos		1	
<input type="checkbox"/> Metadados (título, navegação, redirecionamento, reinicialização)			
<input type="checkbox"/> Marcação do idioma principal da página			
<input type="checkbox"/> Contraste da informação			

Fonte: *Print screen* do site *AccessMonitor*

Figura 18 – Resultado da análise do validador *Hera*

http://moodle.pg.utfpr.edu.br/login/index.php [Rever novamente](#)

⚠ Tenha em linha de conta que, ao rever cada ponto, os scripts podem gerar conteúdos dinâmicos que merecem as mesmas considerações que os conteúdos estáticos.

 **Sumário**

- URL: http://moodle.pg.utfpr.edu.br/login/index.php
- Data/hora: 24/09/2017 - 8:07 GMT
- Total: **166 elementos**
- Análise automática: **15 segundos**
- Erros: **5 erros**
- A verificar manualmente: **37 pontos**
- Revisor: **(desconhecido)**
- Navegador: Sin identificar

Navegar por resultados

Utilize os links da tabela para rever manualmente cada um dos pontos ou comprovar os resultados obtidos na análise automática.

Estado dos pontos de verificação				
Prioridade	Verificar	Bem	Mal	N/A
 P1 HERA WCAG 1.0	8 ^p	--	--	9 ✓
 P2 HERA WCAG 1.0	18 ^p	3 ✓	3 ✗	5 ✓
 P3 HERA WCAG 1.0	11 ^p	2 ✓	2 ✗	4 ✓

Navegar por directrizes

Utilize os links para ver os pontos correspondentes a cada directriz de acessibilidade. Mostram-se todos os pontos, independentemente dos resultados obtidos na análise automática.

[Directriz 1](#) [Directriz 2](#) [Directriz 3](#) [Directriz 4](#) [Directriz 5](#) [Directriz 6](#) [Directriz 7](#) [Directriz 8](#) [Directriz 9](#) [Directriz 10](#) [Directriz 11](#) [Directriz 12](#) [Directriz 13](#) [Directriz 14](#)

Fonte: *Print screen do site Hera*

6 CONCLUSÃO

Neste capítulo são apresentadas algumas das percepções e considerações obtidas no desenvolvimento deste trabalho, bem como sugestões para trabalhos futuros.

6.1 CONSIDERAÇÕES FINAIS

O objetivo deste trabalho foi obter uma resposta para o problema “Por que pouco mais de 2% de todos os *sites* disponíveis na Internet apenas possuem o mínimo necessário para se dizer acessível?”. Por meio da fundamentação teórica, aplicação da pesquisa e desenvolvimento do sistema *Web*, cumpriu-se o propósito deste estudo ao estabelecer o objetivo central e também todos os objetivos específicos.

Na revisão de literatura deste trabalho de conclusão de curso, foi possível aprofundar temas que vem ganhando crescente importância no contexto das Tecnologias da Informação. Identificar os problemas atuais da sociedade no que diz respeito a acessibilidade e *Web*, reconhecer as limitações dessa parcela da população e entender as principais ferramentas que trabalham com acessibilidade foram fundamentais para alcançar os objetivos.

Foi possível, por meio de conhecimentos adquiridos tanto dentro quanto fora da UTFPR Câmpus Ponta Grossa, desenvolver uma ferramenta que possibilite um entendimento simplificado e padronizado sobre a disposição das informações tratadas pelos avaliadores em questão e, que incentive a verificação dos *sites* desenvolvidos para uma *Web* mais acessível.

6.2 TRABALHOS FUTUROS

Estudos futuros sobre acessibilidade *Web* e avaliadores de acessibilidade podem ser aprofundados para proporcionar melhorias na ferramenta já desenvolvida.

Dentre os temas passíveis de pesquisas futuras na UTFPR, destacam-se:

- Ampliar a quantidade de avaliadores na ferramenta para obter maior precisão no resultado final.
- Adicionar a opção de avaliar uma página *Web* por meio *upload* de arquivos físicos como uma alternativa sobre desenvolvimento local.
- Desenvolver opções de análise de outras formas de acessibilidade, como por exemplo, das folhas de estilo de uma página.

- Incluir uma forma de gerar um selo próprio sobre o resultado da ferramenta para incluir nos *sites*, como forma de propagar o incentivo e a preocupação quanto a acessibilidade *Web*.

Espera-se que os estudos e os resultados obtidos nesta pesquisa sejam compartilhados e enriquecidos por todos os interessados em aprofundar os estudos sobre acessibilidade, de modo a contribuir para o contínuo desenvolvimento da área.

REFERÊNCIAS

- ACCESSMONITOR. **Validador automático para as WCAG desenvolvido pela Unidade ACESSO da FCT**. 2013. Acessado em 10 de maio de 2017. Disponível em: <<http://www.acessibilidade.gov.pt/accessmonitor>>.
- BRASIL. **Lei n. 10.098, de 19 de dezembro de 2000**. 2000. Acessado em 18 de junho de 2016. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/L10098.htm>.
- _____. **DECRETO Nº 5.296 DE 2 DE DEZEMBRO DE 2004**. 2004. Acessado em 20 de maio de 2016. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2004/decreto/d5296.htm>.
- CHEERIO. **Fast, flexible & lean implementation of core jQuery designed specifically for the server**. 2016. Acessado em 20 de maio de 2016. Disponível em: <<https://www.npmjs.com/package/cheerio>>.
- CINTHYASAYS. **Free WCAG 2.0 and Section 508 Web Accessibility Scans**. 2003. Acessado em 20 de maio de 2016. Disponível em: <<http://www.cynthiasays.com>>.
- DASILVA. **O primeiro avaliador de Acessibilidade em Português para websites**. 2014. Acessado em 20 de maio de 2016. Disponível em: <<http://www.dasilva.org.br>>.
- EMAG. **Modelo de Acessibilidade em Governo Eletrônico**. 2014. Acessado em 17 de abril de 2016. Disponível em: <<http://emag.governoeletronico.gov.br>>.
- _____. **Acessibilidade na Web - Tecnologia Assistiva**. 2015. Acessado em 19 de junho de 2016. Disponível em: <<http://emag.governoeletronico.gov.br/cursodesenvolvedor/introducao/tecnologia-assistiva-leitores-de-tela.html>>.
- _____. **ASES - Avaliador e Simulador de Acessibilidade de Sítios**. 2016. Acessado em 17 de abril de 2016. Disponível em: <<http://www.governoeletronico.gov.br/acoes-e-projetos/e-MAG/ases-avaliador-e-simulador-de-acessibilidade-sitios>>.
- EXAMINATOR. **eXaminator Validador de Acessibilidade Web**. 2009. Acessado em 15 de maio de 2016. Disponível em: <<http://www.acessibilidade.gov.pt/webax/examinator.php>>.
- EXPRESS. **Fast, unopinionated, minimalist web framework for node**. 2017. Acessado em 16 de setembro de 2017. Disponível em: <<https://www.npmjs.com/package/express>>.
- HERA. **Reveno a Acessibilidade com Estilo**. 2005. Acessado em 27 de junho de 2005. Disponível em: <<http://www.sidar.org/hera/>>.
- IBGE. **Censo Demográfico. Vamos conhecer o Brasil, 2010**. 2010. Acessado em 17 de abril de 2016. Disponível em: <<http://7a12.ibge.gov.br/vamos-conhecer-o-brasil/nosso-povo/caracteristicas-da-populacao.html>>.
- LYNX. **Lynx Viewer**. 2010. Acessado em 20 de maio de 2017. Disponível em: <<http://www.delorie.com/web/lynxview.html>>.
- MACEDO, Hildebrando Rodrigues; CARVALHO, Alexandre Xavier Ywata de. **Aumento do acesso à internet em banda larga no Brasil e sua possível relação com o crescimento econômico: uma análise de dados em painel**. [S.l.], 2010.

MELO, A *et al.* Usabilidade, acessibilidade e inteligibilidade aplicadas em interfaces para analfabetos, idosos e pessoas com deficiência. **SIMPÓSIO BRASILEIRO SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS**, v. 8, 2009.

NEVILE, Liddy. Why is accessibility metadata proving difficult? In: **International Conference on Dublin Core and Metadata Applications**. [S.l.: s.n.], 2002. p. 237–241.

NODEBR. **O que é Node.js?** 2016. Acessado em 16 de setembro de 2017. Disponível em: <<http://nodebr.com/o-que-e-node-js>>.

NODEJS. **About Node.js®**. 2017. Acessado em 16 de setembro de 2017. Disponível em: <<https://nodejs.org/en/about>>.

OWASP. **Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet**. 2017. Acessado em 10 de maio de 2017. Disponível em: <https://www.owasp.org/index.php/Cross-Site_Request_Forgery_%28CSRF%29_Prevention_Cheat_Sheet>.

QUEIROZ, Marco Antonio de. **Métodos e Validadores de Acessibilidade Web**. 2008. Acessado em 20 de maio de 2016. Disponível em: <<http://www.acessibilidadelegal.com/13-validacao.php>>.

REQUEST. **Request is designed to be the simplest way possible to make http calls**. 2017. Acessado em 16 de setembro de 2017. Disponível em: <<https://www.npmjs.com/package/request>>.

TAW. **Supervise de forma continua la calidad de sus sitios web**. 1999. Acessado em 20 de maio de 2016. Disponível em: <<http://www.tawdis.net/servicios/monitorizacion/index.html?lang=es>>.

TECHOPEDIA. **Definition - What does Web Scraping mean?** 2017. Acessado em 18 de maio de 2017. Disponível em: <<https://www.techopedia.com/definition/5212/web-scraping>>.

TIOBE. **TIOBE Index for June of 2016**. 2016. Acessado em 16 de junho de 2016. Disponível em: <<https://www.tiobe.com/tiobe-index>>.

TREINAWEB. **O que é front-end e back-end?** 2017. Acessado em 10 de setembro de 2017. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-front-end-e-back-end>>.

VISIONAWARE. **All About Braille: Six Dots, Four Perspectives**. 2015. Acessado em 10 de setembro de 2017. Disponível em: <<https://www.visionaware.org/blog/visually-impaired-now-what/all-about-braille-six-dots-four-perspectives/12>>.

W3C. **Web Content Accessibility Guidelines 1.0**. 1999. Acessado em 17 de abril de 2016. Disponível em: <<https://www.w3.org/TR/WAI-WEBCONTENT>>.

_____. **Selecting Web Accessibility Evaluation Tools**. 2005. Acessado em 20 de maio de 2016. Disponível em: <<https://www.w3.org/WAI/eval/selectingtools>>.

_____. **Web Accessibility Evaluation Tools List**. 2016. Acessado em 20 de maio de 2016. Disponível em: <<https://www.w3.org/WAI/ER/tools>>.

_____. **An alt Decision Tree**. 2017. Acessado em 11 de setembro de 2017. Disponível em: <<https://www.w3.org/WAI/tutorials/images/decision-tree>>.

W3CBRASIL. **CARTILHA ACESSIBILIDADE NA WEB**. 2013. Acessado em 10 de setembro de 2017. Disponível em: <<http://www.w3c.br/pub/Materiais/PublicacoesW3C/cartilha-w3cbr-acessibilidade-web-fasciculo-I.html>>.

W3SCHOOLS. **The HTML DOM (Document Object Model)**. 2017. Acessado em 10 de setembro de 2017. Disponível em: <https://www.w3schools.com/js/js_htmlDOM.asp>.

WAVE. **Web Accessibility Evaluation Tool**. 2001. Acessado em 20 de maio de 2016. Disponível em: <<http://wave.webaim.org/about>>.