



Pós-Graduação em Ciência da Computação

BRUCE FÁBIAN REIS ALBUQUERQUE

**Avaliação de conformidade de sites governamentais à cartilha de
codificação dos Padrões Web em Governo Eletrônico - e-PWG**



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

Recife
2017

BRUCE FÁBIAN REIS ALBUQUERQUE

**Avaliação de conformidade de sites governamentais à cartilha de
codificação dos Padrões Web em Governo Eletrônico - e-PWG**

Dissertação apresentada ao Programa de Pós-Graduação do Centro de Informática da Universidade Federal de Pernambuco CIN-UFPE, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Leopoldo Motta Teixeira

Recife
2017

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

A345a Albuquerque, Bruce Fábian Reis
Avaliação de conformidade de sites governamentais a cartilha de
codificação dos padrões web em governo eletrônico - e-PWG / Bruce Fábian
Reis Albuquerque. – 2018.
85 f.: il., fig., tab.

Orientador: Leopoldo Motta Teixeira.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn,
Ciência da Computação, Recife, 2018.
Inclui referências, apêndices e anexos.

1. Ciência da computação. 2. Avaliação de conformidade. I. Teixeira,
Leopoldo Motta (orientador). II. Título.

004 CDD (23. ed.) UFPE- MEI 2018-052

Bruce Fábian Reis Albuquerque

**Avaliação de conformidade de sites governamentais a cartilha de
codificação dos Padrões Web em Governo Eletrônico - e-PWG**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre Profissional em 04 de janeiro de 2018.

Aprovado em: 04/01/2018.

BANCA EXAMINADORA

Prof. Kiev Santos da Gama
Centro de Informática / UFPE

Prof. Vanilson André de Arruda Burégio
Universidade Federal Rural de Pernambuco

Prof. Leopoldo Motta Teixeira
Centro de Informática / UFPE
(Orientador)

Dedido este trabalho a Deus, autor da minha fé. A Ele toda honra e glória.

Agradecimentos

Agradeço a todos que participaram dessa jornada.

Aos familiares, esposa, pais, irmã, primos e tios pela preocupação, força, disponibilidade em ajudar e orações.

A todos os amigos, que nos momentos de angústia se mostraram irmãos, especialmente aos que participaram junto comigo dessa trajetória, seja nas viagens, nas conversas, nos trabalhos em equipe, no estímulo, no cuidado, na companhia e simplesmente por estarem disponíveis quando precisei.

Ao meu orientador, Leopoldo, por toda ajuda, pelas dicas, profissionalismo, disponibilidade e paciência.

Ao Instituto Federal de Rondônia - IFRO, por proporcionar esta oportunidade e pela disponibilidade de recursos durante toda esta trajetória, sem a qual, não seria possível concluí-la.

A Deus, pelo sustento e por me conceder tamanha graça da qual não mereço.

Quando eu for grande, quero ser pequeno.

Resumo

A administração pública tem a obrigação ofertar serviços com excelência que possibilitem a melhor experiência possível ao usuário, devendo inclusive respeitar as características de cada região. Embasados nesse compromisso foram elaboradas recomendações publicadas em cartilhas e guias de padronizações para diversos fins, que visam orientar o desenvolvimento de aplicações e sites. Tendo em vista que o governo federal não possui nenhuma ferramenta que dê suporte automatizado aos desenvolvedores fazendo a checagem da lista de verificação proposta na cartilha de codificação durante o processo de desenvolvimento de sítios web, o presente trabalho teve como objetivo a criação de um software que otimize o desenvolvimento, verificando a adequação aos padrões fornecidos pelo governo em seu programa de governo eletrônico utilizando como referência as orientações contidas na cartilha de codificação dos Padrões Web em Governo Eletrônico, o e-PWG. Como forma de validação da ferramenta foram analisados 27 sítios de universidades federais sediadas em todas as unidades federativas do Brasil, o que constatou que há pontos de inconformidades com a cartilha de codificação em todos os sítios avaliados. A solução proposta pode contribuir para a melhoria da codificação de páginas web dos serviços governamentais e servir como referência para a melhoria/atualização da cartilha de codificação proposta atualmente.

Palavras-chave: e-PWG. Cartilha de codificação. Diretrizes.

Abstract

The public administration has the obligation to offer services with excellence that allow the best possible experience to the user, and must also respect the characteristics of each region. Based on this commitment, recommendations have been published in booklets and standardization guides that aim to guide the development of applications and sites. Given that the federal government does not have any tools that provide automated support to developers by checking the proposed checklist in the ePWG coding booklet during the web site development process, the present work aimed at creating a software that optimizes development by verifying compliance with the standards provided by the government in its e-government program using as reference the guidelines contained in the ePWG coding primer. As a form of validation of the tool, we analyzed 27 sites of federal universities based in all the federal units of Brazil, which found that there are points of nonconformity with the coding booklet in all the evaluated sites. The proposed solution can contribute to the improvement of the codification of web pages of the governmental services and serve as reference for the improvement / update of the currently proposed coding book.

Keywords: e-PWG. Coding booklet. Guidelines.

Lista de figuras

Figura 1 – e-GOV Brasil - linha do tempo	20
Figura 2 – e-GOV Brasil - linha do tempo (continuação)	21
Figura 3 – Diagrama de pacotes	26
Figura 4 – Camadas	27
Figura 5 – Fluxo do processo de conexão	28
Figura 6 – Declaração do charset em meta tags	31
Figura 7 – Estrutura básica de um documento HTML	32
Figura 8 – Regra CSS	35
Figura 9 – Referenciando link interno	39
Figura 10 – Resposta do serviço Google PageSpeed	41
Figura 11 – URL amigável	42
Figura 12 – Esquema http URL	42
Figura 13 – Declaração <i>font-size</i>	45
Figura 14 – Declaração <i>font</i>	45
Figura 15 – URL com o caractere “?”	57
Figura 16 – URL amigável	57
Figura 17 – Resultado da análise	60
Figura 18 – Análise do Doctype	61
Figura 19 – Análise do <i>Charset</i>	62
Figura 20 – Análise de Links quebrados	62
Figura 21 – Análise de URLs amigáveis	63
Figura 22 – Análise de acesso sem WWW	64
Figura 23 – Análise do JavaScript	64
Figura 24 – Análise das declarações de fontes	65
Figura 25 – Análise do Contraste	65

Lista de tabelas

Tabela 1 – Hypertext Transfer Protocol Status Code Registry	81
Tabela 2 – Perguntas da lista de verificação	85

Lista de abreviaturas e siglas

API	Application Programming Interface
ARPA	Advanced Research and Projects Agency
CERN	European Council for Nuclear Research
CMS	Content Management System
CSS	Cascading Style Sheets
DNS	Domain Name System
DOM	Document Object Model
e-GOV	Governo Eletrônico
e-MAG	Modelo de Acessibilidade em Governo Eletrônico
e-PWG	Padrões Web em Governo Eletrônico
HTML	HyperText Markup Language
HTTP	Protocolo de Transferência de Hipertexto
IDE	Integrated Development Environment
IETF	Internet Engineering Task Force
MIME	Multipurpose Internet Mail Extensions
MIT	Massachusetts Institute of Technology
PNG	Portable Network Graphics
RFC	Request for Comments
SSL	Secure Socket Layer
URI	Uniform Resource Identifier
UTF-8	8-bit Unicode Transformation Format
WAI	Web Accessibility Initiative
WCAG	Web Content Accessibility Guidelines
WHATWG	Web Hypertext Application Technology Working Group
WWW	World Wide Web
XHTML	eXtensible Hypertext Markup Language

Sumário

1	INTRODUÇÃO	15
1.1	Motivação e problema	16
1.2	Objetivos	17
1.2.1	Objetivo Geral	17
1.2.2	Objetivos Específicos	17
1.3	Organização do Trabalho	17
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	O Governo Eletrônico Brasileiro	19
2.1.1	e-MAG	21
2.1.2	e-PWG	22
2.1.3	A Cartilha de Codificação	22
2.2	<i>World Wide Web Consortium</i>	23
2.3	Validadores	24
3	SOLUÇÃO	26
3.1	A Ferramenta	26
3.1.1	Conexão	27
3.1.2	Implementação das diretrizes	28
3.1.3	Classes acessórias	28
3.1.4	Execução da ferramenta	29
3.1.5	Bibliotecas	29
3.2	Lista de Verificação	29
3.2.1	A página usa Doctype correto?	29
3.2.2	A página usa o <i>character set</i> de codificação de caracteres correto?	30
3.2.3	A página usa codificação válida?	32
3.2.4	As folhas de estilo CSS usadas pela página são válidas?	34
3.2.5	Há declarações de classes ou ID's desnecessárias?	37
3.2.6	Há <i>links</i> quebrados?	38
3.2.7	Qual é a performance velocidade de carregamento/tamanho da página?	40
3.2.8	A página utiliza URLs amigáveis?	41
3.2.9	As URLs funcionam sem o "www"?	42
3.2.10	Há erros de JavaScript?	43
3.2.11	Há quebra do desenho quando o usuário aumenta o tamanho da fonte?	44
3.2.12	O contraste das cores da página é suficiente?	45
3.2.13	Itens não implementados	47
3.2.13.1	<u>O código é bem estruturado?</u>	47

3.2.13.2	<u>A página funciona com o JavaScript desabilitado?</u>	48
3.2.13.3	<u>O conteúdo é acessível com as folhas de estilo desabilitadas?</u>	48
3.2.13.4	<u>A página é consistente em navegadores texto?</u>	48
3.2.13.5	<u>O conteúdo é legível quando impresso? Não há desperdício de papel ou tinta?</u>	48
3.2.13.6	<u>A página é bem visualizada em dispositivos portáteis?</u>	48
3.2.13.7	<u>A página é bem visualizada em diferentes resoluções de navegador e tela?</u>	49
4	AVALIAÇÃO	50
4.1	Metodologia	50
4.2	Cenários de teste	51
4.2.1	Doctype	52
4.2.2	Charset	52
4.2.3	Código HTML e CSS	52
4.2.4	Classes e IDs desnecessários	55
4.2.5	Links quebrados	56
4.2.6	URLs amigáveis	56
4.2.7	Código JavaScript	57
4.2.8	Fontes em unidades relativas	58
4.2.9	Contraste	58
4.3	Análise de sítios governamentais	59
4.3.1	Resultado <i>Doctype</i>	61
4.3.2	Resultado <i>Charset</i>	61
4.3.3	Links Quebrados	62
4.3.4	URL amigável	63
4.3.5	Acessa sem WWW	63
4.3.6	JavaScript	64
4.3.7	Declaração de fontes	65
4.3.8	Contraste	65
4.3.9	Resultado HTML, CSS, Classes e IDs e Performance	66
4.3.10	Considerações	66
5	CONCLUSÃO	68
5.1	Trabalhos relacionados	68
5.2	Considerações finais	69
5.3	Trabalhos futuros	69
	Referências	71

APÊNDICES	76
APÊNDICE A – Sítios avaliados	77
APÊNDICE B – Repositório	79
ANEXOS	80
ANEXO A – Valores de resposta do protocolo HTTP	81
ANEXO B – Lista de verificação da cartilha de codificação . . .	85

1 INTRODUÇÃO

A utilização de tecnologia pela sociedade vem crescendo exponencialmente nos últimos anos, revelando a facilidade que a *internet* provê aos seus consumidores para a prestação das mais variadas categorias de serviços. Em pesquisa realizada pelo Cetic.br¹ no período entre novembro de 2015 e junho de 2016, para o levantamento de indicadores sobre a proporção de indivíduos que já acessaram a internet, chegou-se ao número de 66% sobre o total da população, os que responderam afirmativamente.

Essa massiva utilização da internet como meio de comunicação, revela a necessidade de os serviços estarem sempre disponíveis em qualquer lugar, hora e por qualquer dispositivo, o que segundo Urdiales et al. (2004) levou diversos governos a adotarem medidas para oferecerem serviços e compartilharem informações pela web. Segundo DINIZ (2009), temas como desempenho, eficácia, transparência, mecanismos de controle, prestação de contas, entre outros, são relacionados ao processo de modernização da gestão pública, e por conseguinte requerem o uso de tecnologias para alavancar a eficiência da administração pública.

Dessa forma, o Governo Brasileiro lança, em 2008, os Padrões Brasil e-GOV que surgem como um conjunto de recomendações de boas práticas agrupadas em formato de cartilhas, com o objetivo de aprimorar a comunicação e o fornecimento de informações e serviços prestados por meios eletrônicos pelos órgãos do Governo Federal. A Cartilha de codificação foi a primeira a ser lançada (BRASIL, 2015a). Esta cartilha possui uma lista de verificações que os desenvolvedores devem seguir para validar o teor do conteúdo desenvolvido.

Nesse contexto de desenvolvimento de aplicações para internet, são utilizados padrões definidos internacionalmente para que o conteúdo escrito possa ser lido e acessado pelas mais diversas tecnologias que implementam esses padrões, tais como navegadores, leitores de tela, dispositivos móveis entre outros. O W3C² é o responsável atualmente pela definição desses padrões, desenvolvendo especificações técnicas e orientações para toda a comunidade de desenvolvedores que também são adotados pelo governo brasileiro no desenvolvimento de suas ferramentas web (BRASIL, 2010a).

O presente trabalho visa contribuir com uma análise de adequação de sítios governamentais para constatação de conformidade com os itens da lista de verificação da cartilha de codificação, através de uma ferramenta de validação automática que

¹ Com a missão de monitorar a adoção das tecnologias de informação e comunicação (TIC) – em particular, o acesso e uso de computador, Internet e dispositivos móveis – foi criado em 2005 o Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (Cetic.br). O Cetic.br é um departamento do Núcleo de Informação e Coordenação do Ponto BR (Nic.br), que implementa as decisões e projetos do Comitê Gestor da Internet do Brasil (Cgi.br).

² Disponível em : <<https://www.w3.org/>>. Acesso em nov. 2017.

contempla os itens fornecidos.

1.1 Motivação e problema

É responsabilidade da administração pública ofertar serviços com excelência que possibilitem a melhor experiência possível ao usuário, devendo inclusive respeitar as características de cada região. O governo possui recomendações definidas por meio de cartilhas, guias e padronizações que visam orientar o desenvolvimento de aplicações e sites (BRASIL, 2011).

Segundo os Padrões Web em Governo Eletrônico — e-PWG³, os órgãos do Governo Federal tem por objetivo aprimorar a comunicação e o fornecimento de informações e serviços prestados por meios eletrônicos, adotando tecnologias que permitam o aumento da eficácia, da eficiência e da transparência governamental.

A cartilha de codificação do e-PWG define orientações para desenvolvimentos de sítios do governo federal que são baseadas em boas práticas para codificação. Estas práticas tem o propósito de tornar os sites identificáveis, portáteis, relevantes, acessíveis e efetivos à população. Nesta cartilha são apresentadas 21 perguntas como um recurso para avaliar se as recomendações estão sendo seguidas durante o desenvolvimento ou melhoria dos sítios.

Embora a cartilha possua tais perguntas o governo federal não possui nenhuma ferramenta que dê suporte automatizado aos desenvolvedores fazendo a checagem da lista de verificação proposta na cartilha de codificação do e-PWG durante o processo de desenvolvimento de sítios web. Mesmo que existam algumas ferramentas de validação, descritas no Capítulo 2, e que contemplem alguns itens da lista, essas ferramentas tem seu foco voltado para a validação de acessibilidade e não contemplam todos os questionamentos da lista de verificação, que conforme a cartilha tem “o objetivo de aprimorar a comunicação, o fornecimento de informações e serviços prestados por meios eletrônicos pelos órgãos do Governo Federal” (BRASIL, 2010a).

O não atendimento às recomendações é percebido ao se navegar em sites onde há utilização excessiva, e por vezes desnecessária de recursos, como imagens e arquivos de scripts, onde o carregamento da página se torna demorado, principalmente para os usuários que utilizam redes 3G⁴ em dispositivos móveis e a não utilização de URLs amigáveis, que dificulta determinar a localização dentro do site acessado.

De acordo com o programa de governo eletrônico brasileiro, que tem como característica democratizar o acesso à informação, utilizando-se de tecnologias a fim

³ Disponível em: <<https://www.governoeletronico.gov.br/eixos-de-atuacao/governo/epwg-padroes-web-e-m-governo-eletronico>>. Acesso em dez. 2017.

⁴ Tecnologia para transmissão de dados e voz, fornecida por operadoras de telefonia.

de promover e ampliar a prestação de serviços públicos à população sendo eficiente e efetivo em suas ações, a seguinte questão de pesquisa foi desenvolvida:

Como verificar de forma automatizada a aderência dos sites governamentais às diretrizes fornecidas pelo programa de governo eletrônico em sua cartilha de codificação?

Deste modo pretende-se apresentar uma forma de melhoria no processo de desenvolvimento através de uma ferramenta para que sítios desenvolvidos no âmbito do governo federal possam ser avaliados durante seu desenvolvimento e estejam de acordo com o proposto pelas orientações da cartilha de codificação.

1.2 Objetivos

Esta sessão delinea sobre os objetivos geral e específicos para o andamento da pesquisa.

1.2.1 Objetivo Geral

Contribuir com a criação de uma ferramenta de suporte ao desenvolvimento de aplicações web no governo federal. A ferramenta visa analisar e verificar se sítios governamentais foram desenvolvidos atendendo as recomendações propostas pelo governo federal brasileiro na cartilha de codificação contida em seu programa de governo eletrônico (e-PWG).

1.2.2 Objetivos Específicos

Para concretização do objetivo geral os seguintes itens serão seguidos:

- Realizar um levantamento bibliográfico referente às tecnologias e conceitos elencados na cartilha de codificação do e-PWG.
- Desenvolver uma aplicação que será utilizada como instrumento para a validação de regras contidas na cartilha de codificação.
- Avaliar a precisão da ferramenta em um *benchmark* desenvolvido com base nas perguntas da cartilha.
- Analisar páginas oficiais do governo utilizando a ferramenta, para identificar sua adequabilidade aos padrões.

1.3 Organização do Trabalho

O restante do trabalho está dividido nos seguintes capítulos, a saber:

Capítulo 2: Fundamentação teórica — aborda uma revisão de literatura sobre temas ligados ao programa de governo eletrônico brasileiro; o World Wide Web Consortium; validadores de código e outras questões relativas ao tema.

Capítulo 3: Solução — descreve o processo de desenvolvimento da ferramenta baseado nas diretrizes da cartilha de codificação.

Capítulo 4: Avaliação — apresenta a avaliação da ferramenta por testes rodados localmente e a análise de portais de universidades federais brasileiras.

Capítulo 5: Conclusão — com apresentação dos resultados encontrados e debatidos durante a execução do trabalho e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo apresentaremos um arcabouço teórico e conceitual sobre principais conceitos que estão relacionados com a pesquisa.

2.1 O Governo Eletrônico Brasileiro

Segundo Rover et al. (2010, p. 3) o governo eletrônico tem como objetivo transformar a relação entre os governos, cidadãos e empresas, sobretudo em termos da agilidade e da transparência dos processos. Os autores explicam que o e-gov é um processo que apresenta riscos e que se não bem implementado pode desperdiçar recursos e falhar na prestação de serviços úteis a população.

O governo eletrônico pode ser definido com a utilização da tecnologia da informação e comunicação para promover maior eficiência e maior efetividade governamental, facilitando o acesso aos serviços públicos, permitindo ao grande público o acesso à informação, e tornando o governo mais responsável para com o cidadão, conforme diz Santos (2002, p. 6) fundamentado no *Pacific Council on International Policy* (2002).

O Programa de Governo Eletrônico Brasileiro é focado em três idéias principais, que são a participação cidadã, o gerenciamento do estado e a integração com parceiros. Esses pontos visam promover a democracia, dinamização dos serviços públicos e melhorar a eficiência da administração (BRASIL, 2015b).

O e-Gov no Brasil iniciou-se com o decreto presidencial do dia 3 de abril de 2000 que, em seu artigo primeiro, criava um grupo de trabalho interministerial com a finalidade de “[...] examinar e propor políticas, diretrizes e normas relacionadas com as novas formas eletrônicas de interação” (BRASIL, 2000).

Em 2002, após dois anos do início do programa, o Comitê Executivo do Governo Eletrônico - CEGE, publica o documento “2 Anos de Governo Eletrônico - Balanço de Realizações e Desafios Futuros”¹, e então define diretrizes gerais norteadas pelas seguintes linhas a seguir, conforme apresentado no histórico do Governo Eletrônico Brasileiro (BRASIL, 2015b):

- Promoção da cidadania como prioridade;
- Inclusão digital de forma indissociável do governo eletrônico;
- Software livre como estratégia de implementação;
- Gestão do conhecimento como gestão estratégica de articulação;

¹ Disponível em: <https://www.governoeletronico.gov.br/documentos-e-arquivos/E15_90balanco_2anos_egov.pdf>. Acesso em dez. 2017.

- Racionalização de recursos;
- Arcabouço integrado de políticas, sistemas, padrões e normas; e
- Integração das ações de governo eletrônico com outros níveis de governo e outros poderes.

Com o passar dos anos, outras políticas e modelos para a implementação do Programa de Governo Eletrônico foram sendo incorporadas, conforme apresentado nas Figuras 1 e 2, em especial o lançamento do e-MAG em 2004, recomendando a acessibilidade nos sítios da administração pública e o e-PWG em 2008, com ênfase em cartilhas de recomendações.

Figura 1 – e-GOV Brasil - linha do tempo



Fonte: Portal do Governo Eletrônico Brasileiro.

Figura 2 – e-GOV Brasil - linha do tempo (continuação)



Fonte: Portal do Governo Eletrônico Brasileiro.

2.1.1 e-MAG

O Modelo de Acessibilidade em Governo Eletrônico (e-MAG) tem o compromisso de ser o norteador no desenvolvimento e adaptação de conteúdos digitais do governo federal, garantindo o acesso a todos (BRASIL, 2014).

Foi lançado em 2005, e é a normativa oficial em acessibilidade nos portais e sítios eletrônicos da administração pública brasileira. Visa garantir o pleno acesso aos conteúdos disponíveis por pessoas com necessidades especiais. Este modelo é coerente com as necessidades brasileiras e está em conformidade com os padrões internacionais. Foi desenvolvido baseado em 14 normas de outros países acerca de acessibilidade digital. Dentre estas normas estão a *Section 508* do governo dos Estados Unidos, os padrões CLF do Canadá, as diretrizes irlandesas de acessibilidade e documentos de outros países como Portugal e Espanha (BRASIL, 2014).

O e-MAG especifica que para a criação de conteúdo acessível em um ambiente online é necessário que o código esteja de acordo com as recomendações internacionais do W3C, e recomenda a consulta à cartilha de codificação dos Padrões Web em Governo Eletrônico, como referencial de boas práticas.

2.1.2 e-PWG

A definição de padrões e métodos é um fator crítico para a implantação e operação do governo eletrônico e depende dessas definições para o sucesso das ações (BRASIL, 2014).

Nesse contexto surgem os Padrões Web em Governo Eletrônico- e-PWG, que auxiliam a administração pública a ofertar serviços à população objetivando a melhor experiência possível. A utilização dos padrões traz benefícios como a garantia do nível de qualidade e são definidos como:

[...] recomendações de boas práticas agrupadas em formato de cartilhas com o objetivo de aprimorar a comunicação e o fornecimento de informações e serviços prestados por meios eletrônicos pelos órgãos do Governo Federal (BRASIL, 2016).

Estes padrões estão distribuídos em 5 cartilhas, a seguir:

- Cartilha de Codificação;
- Guia de Administração;
- Cartilha de Usabilidade;
- Cartilha de Redação Web (webwriting);e
- Cartilha de Desenho e Arquitetura de Conteúdo.

A Cartilha de Codificação foi a primeira a ser lançada e dispõe de diretrizes para a criação de documentos HTML, CSS entre outras técnicas para se criar uma página adequada no contexto da administração pública. O Guia de Administração oferece subsídios para a manutenção e administração de sítios do governo. A cartilha de usabilidade também trabalha no contexto de administração e manutenção e contém as definições de usabilidades e testes que devem ser aplicados em sítios governamentais. A Cartilha de Redação Web, é o guia para elaboração de informação clara e estruturada no meio digital. Por fim, a Cartilha de Desenho e Arquitetura de Conteúdo, que segundo Oliveira e Eler (2015);Marcelo Medeiros Eler (2015) , visa padronizar os sítios dos órgão públicos, alinhando informações que otimizam a comunicação com o cidadão.

2.1.3 A Cartilha de Codificação

A Cartilha de codificação foi a primeira das cinco recomendações do e-PWG, e é o documento chave abordado nesta pesquisa. Esta cartilha contempla boas práticas em codificação para orientar os desenvolvedores a tornarem os sites da administração

pública mais efetivos e relevantes. São apresentados padrões web, prescritos pelo W3C, testes e mecanismos para comprovar a adequabilidade das páginas desenvolvidas.

A codificação, fator que motivou a criação deste documento, é descrito como:

[...] um conjunto de arquivos com função de conteúdo, apresentação e comportamento. É o esqueleto, a estrutura que dá suporte aos aspectos da página relacionados a apresentação, organização, navegação e usabilidade (BRASIL, 2010a).

Este documento também destaca a importância de se seguir os padrões web e de boas práticas de implementação, pois dessa forma, podem garantir seu desenvolvimento estruturado e evolutivo. Dessa forma, os sítios que seguem essas recomendações são mais ágeis e versáteis e facilitam a disponibilização da informação em diversos dispositivos (BRASIL, 2010a).

Em seu conteúdo, encontram-se orientações de como trabalhar com arquivos de apresentação (CSS) e de comportamento (JavaScript), elementos de marcação HTML, utilização da codificação de caracteres e Doctype apropriados, performance, URLs entre outras coisas. Em seu Capítulo 6, é apresentada uma lista de verificação, disponível na tabela 2 presente no anexo B, como um recurso de avaliação para as diretrizes propostas, nas fases de desenvolvimento, manutenção e evolução do sítio ou portal.

2.2 *World Wide Web Consortium*

O *World Wide Web Consortium*, ou somente W3C, como é conhecido, é um consórcio internacional que padroniza a Web onde participam neste processo, instituições filiadas, uma equipe de trabalho em tempo integral e a comunidade. A sua missão é “[...] conduzir a World Wide Web para que atinja todo seu potencial, desenvolvendo protocolos e diretrizes que garantam seu crescimento de longo prazo” (WORLD WIDE WEB CONSORTIUM, 2011).

O W3C foi formado em 1994, por Tim Berners-Lee no MIT², com o apoio do CERN³, ARPA⁴ e a Comissão Europeia. A visão do W3C era padronizar os protocolos e tecnologias usados para construir a web, de modo que o conteúdo estaria disponível para uma população tão ampla quanto possível. Após sua criação o W3C publicou diversas especificações entre elas o HTML 4.01, o formato de imagens PNG as Folhas de Estilo em Cascata (CSS) em todas as suas versões (WORLD WIDE WEB CONSORTIUM, 2014).

² Instituto de Tecnologia de Massachusetts

³ Conselho Europeu para Investigação Nuclear

⁴ Agência de Pesquisas em Projetos Avançados

Atuante nas áreas de Internacionalização, e-GOV, mobile Web, entre outros uma de suas iniciativas mais importantes é o WAI⁵, criado em 1999, voltado para a criação de padrões para acessibilidade na web que de acordo com Bach et al. (2009) as Diretrizes de Acessibilidade de Conteúdo na Web - WCAG, elaboradas por membros do W3C/WAI são a principal referência no assunto.

No Brasil, o W3C se faz presente por meio de um escritório, atuando em de grupos de trabalho como o “GT Dados Abertos Governo e Sociedade” que segundo Diniz e Guimarães (2013) existe desde 2011 e reúne membros do governo e representantes da sociedade civil.

2.3 Validadores

Abordando este tema a cartilha de codificação diz que:

Validação é a verificação que se faz para saber se um documento obedece às regras de linguagem em que foi escrito. Pode-se comparar a validação com a busca por erros gramaticais e de concordância em um texto escrito (BRASIL, 2010a).

Para Conforto e Santarosa (2002, p. 98) a validação pode ser feita de forma manual ou automática, sendo para que para esta, pode-se utilizar ferramentas ou serviços de análise e compatibilidade. As autoras citam que as vantagens de uma validação automática são a possibilidade de analisar muitos aspectos simultaneamente, o funcionamento rápido e a possibilidade de uma qualificação global da página.

Concernente à validação dos aspectos relativos à cartilha de codificação, o W3C possui ferramentas online⁶ como os validadores de código HTML e CSS e o *Link Checker*, ferramenta para checagem de *links* quebrados. A utilização desses programas é recomendada pelo governo em sua lista de recursos para acessibilidade⁷, e na cartilha de codificação do e-PWG.

Quanto aos validadores, Oliveira e Eler (2015) destaca o DaSilva, como o primeiro validador de acessibilidade em língua portuguesa, embasado nos princípios de acessibilidade recomendados pelo W3C e e-MAG. Os autores também citam o *eScanner*, lançado em 2013 como extensão do navegador *Google Chrome*, embasado na versão 3.0 do e-MAG.

Outra importante iniciativa brasileira é o Avaliador e Simulador de Acessibilidade em Sítios - ASES⁸, resultado da parceria entre o Departamento de Governo Eletrônico

⁵ Iniciativa de Acessibilidade na Web.

⁶ Disponível em: <<https://www.w3.org/developers/tools/>>. Acesso em dez. 2017.

⁷ Disponível em: <<https://www.governoeletronico.gov.br/eixos-de-atuacao/governo/acessibilidade/recurso-s-de-acessibilidade>>. Acesso em dez. 2017.

⁸ Disponível em: <<http://asesweb.governoeletronico.gov.br/ases/>>. Acesso em dez. 2017.

(DGE) e o Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS). Por ser um validador de acessibilidade criado no âmbito do governo federal, implementa as diretrizes do e-MAG e tem por objetivo auxiliar a construção de sítios que sejam acessíveis a qualquer pessoa, independente do seu tipo de deficiência e dispositivo de navegação (BRASIL, 201-).

O fato das ferramentas existentes não contemplarem toda a lista de verificação, validando somente pontos isolados, ou focarem na validação de acessibilidade, gera uma lacuna que a proposta deste trabalho visa preencher, onde uma única ferramenta realize a validações dos itens presentes na cartilha de codificação.

Este capítulo apresentou um arcabouço teórico, oferecendo assim, informações necessárias para o entendimento do contexto em que a pesquisa está alinhada. O próximo capítulo apresentará informações detalhadas da implementação da solução proposta para o andamento da pesquisa.

3 SOLUÇÃO

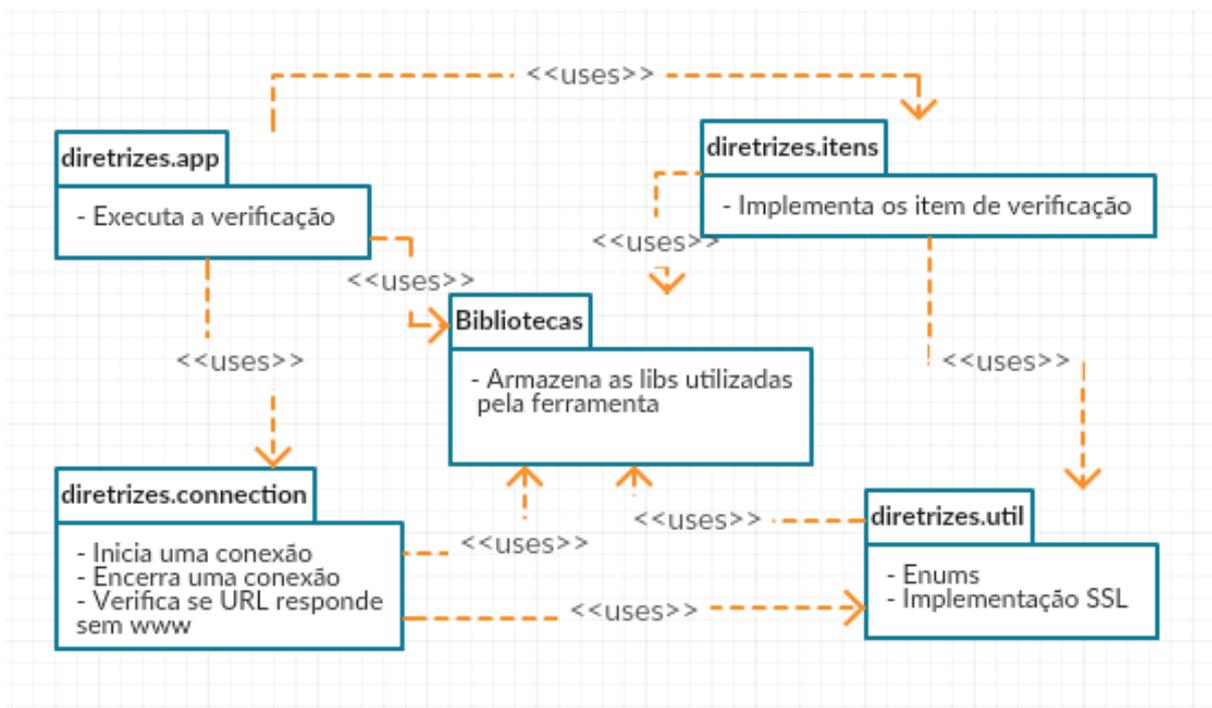
Este capítulo apresenta nossa abordagem para solução do problema, baseado na implementação automatizada das diretrizes presentes na lista de verificação da cartilha de codificação do e-PWG. Para isso, as diretrizes serão elencadas uma a uma, com suas características e a lógica como foram implementadas.

3.1 A Ferramenta

Para a realização da verificação automática foi desenvolvida uma ferramenta utilizando a linguagem Java¹ em sua versão 8, e a tecnologia *Maven*² para gerenciamento das bibliotecas utilizadas no projeto.

Os pacotes da ferramenta estão organizados conforme ilustra a Figura 3:

Figura 3 – Diagrama de pacotes



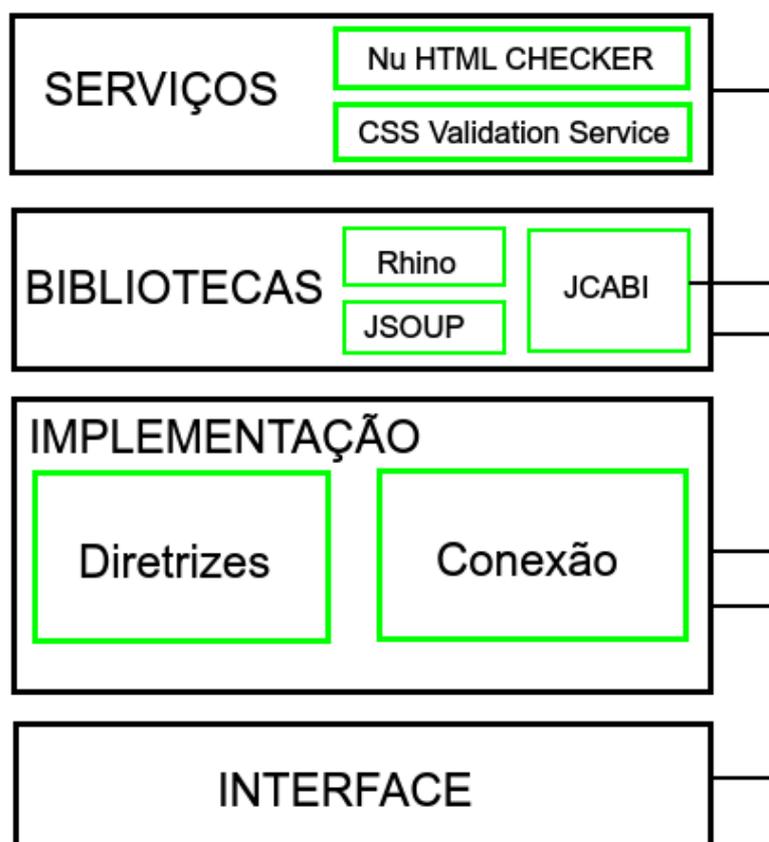
Fonte: Própria (2017).

A ferramenta utiliza serviços de outras fontes para realizar a validação dos itens da lista, as camadas da aplicação podem ser vistas na 4:

¹ Disponível em: https://www.java.com/pt_BR/

² Disponível em: <https://maven.apache.org/>

Figura 4 – Camadas



Fonte: Própria (2018).

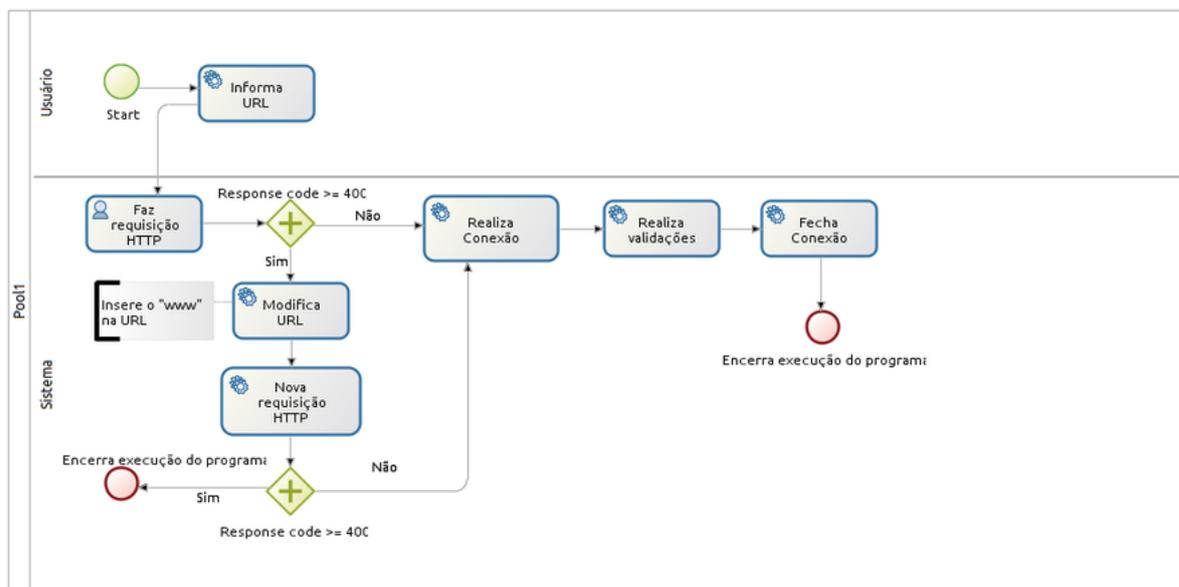
A execução da ferramenta é realizada pela IDE de desenvolvimento utilizada para codificação, onde a camada de *Interface* consiste na classe que instancia todas as validações e recebe um parâmetro de *URL* que serve como base para início da varificação.

3.1.1 Conexão

Esta camada é responsável pela conexão com as páginas que são analisadas. processo de conexão também é utilizado para realizar a validação do item da lista de verificação descrito na seção 3.2.9 que será explicado no decorrer deste capítulo.

A conexão é realizada informando a URL do sítio que se quer validar. O programa então, faz uma requisição HTTP para esse endereço e captura o *Response Code*. Se o código de resposta estiver fora do escopo 2xx: Sucesso e 3xx: Redirecionamento a conexão não é realizada, conforme mostra a Figura 5.

Figura 5 – Fluxo do processo de conexão



Fonte: Própria (2017).

Segundo a Internet Assigned Numbers Authority (2017), o escopo 2xx, significa que a ação foi recebida, entendida e aceita com sucesso, e o escopo 3xx, diz que outras ações devem ser tomadas para completar o pedido. A tabela 1 localizada no Anexo A apresenta a lista completa com os códigos de resposta HTTP.

3.1.2 Implementação das diretrizes

Esta camada consiste na agregação de todas as classes que fazem a validação da lista de verificação e que serão analisadas neste capítulo. Cada item da lista que foi implementado possui uma classe própria, sendo ao todo 12 itens. Os demais itens que não foram implementados estão descritos na Seção 3.2.13.

Para a codificação da verificação dos itens da lista, são utilizadas bibliotecas que se utilizam de serviços de validações externos como o serviço *Nu Html Checker* do W3C que valida código HTML, descrito no capítulo.

Como descrito, existem ferramentas próprias para alguns itens da lista, porém, a ferramenta desenvolvida condensa todas essas validações no mesmo *software*, auxiliando o desenvolvedor ao concentrar em um único ponto todas as requisições de validações.

3.1.3 Classes acessórias

A ferramenta possui um pacote onde se encontram classes acessórias que são utilizadas durante a implementação das validações. Basicamente há dois contextos, as

classes para tratamento das requisições em sítios que estão utilizando a tecnologia SSL sobre o protocolo HTTP e uma classe do tipo `enum`³ onde estão listados os *Doctypes* para validação do item de verificação contido na seção 3.2.1.

3.1.4 Execução da ferramenta

A validação dos itens da cartilha se dá através de uma classe específica que instancia todas as diretrizes implementadas. Nesta classe a URL referente ao sítio a ser validado é informada, uma conexão é aberta e os itens da lista são checados sequencialmente por ordem de instanciação.

3.1.5 Bibliotecas

Várias bibliotecas são utilizadas para a implementação da ferramenta. Estão presentes as bibliotecas que dão base a utilização da linguagem Java e algumas mais específicas que dão suporte a implementação das diretrizes. As principais bibliotecas são descritas a seguir:

- *Jsoup*⁴: biblioteca Java para trabalhar com o HTML. Através dela é possível extrair e manipular dados, usando os melhores métodos DOM, CSS e *Jquery*.
- *Jcabi W3C*⁵: biblioteca que contém adaptadores Java de validadores *online* do W3C. Foi utilizada para validação do código HTML e CSS.
- *Rhino*⁶: é uma implementação de código aberto do JavaScript escrita em Java e foi utilizada para validação do código JavaScript.

3.2 Lista de Verificação

A lista de verificação é um recurso para avaliar se as recomendações estão sendo seguidas no desenvolvimento, evolução ou manutenção do sítio ou portal (BRASIL, 2010a).

3.2.1 A página usa Doctype correto?

Conforme descreve KELLY (2002) o Doctype é necessário para que o documento cumpra os padrões *HTML*. Segundo EIS e FERREIRA (2012) o Doctype não é

³ <https://docs.oracle.com/javase/tutorial/java/javaOO/enum.html>

⁴ Disponível em: <https://jsoup.org/>

⁵ Disponível em: <http://w3c.jcabi.com/index.html>

⁶ Disponível em: <https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Rhino>

uma tag do HTML e sim uma instrução que possibilita ao navegador obter informações sobre qual versão do código a marcação foi escrita.

Sobre a necessidade de se declarar o Doctype, o W3C diz o seguinte:

Os DOCTYPEs são necessários por motivos legados. Quando omitido, os navegadores tendem a usar um modo de renderização diferente que é incompatível com algumas especificações. Incluir o DOCTYPE em um documento garante que o navegador faça uma tentativa de melhor esforço para seguir as especificações relevantes (WORLD WIDE WEB CONSORTIUM, 2017c).

A cartilha enfatiza que a recomendação do W3C é que se dê sempre preferência ao *Doctype Strict*, declaração esta que é utilizada para validação de documentos XHTML 4.01. Entretanto, atualmente o próprio W3C em sua lista de declaração de *Doctypes*⁷ diz que “a declaração de Doctype mais simples e confiável para usar é a definida em HTML5: <!DOCTYPE html>” (WORLD WIDE WEB CONSORTIUM, 2016b), sendo assim, assumiu-se que este Doctype será utilizado como parâmetro para a checagem deste item da lista de verificação.

Para avaliar se o Doctype declarado na página é o correto, a ferramenta, após a requisição de conexão bem sucedida, varre todos os nós do documento HTML, por meio da biblioteca *Jsoup*, e verifica se ocorre a incidência algum nó do tipo *DocumentType*, ou seja, se a declaração foi feita.

Caso o documento não possua declaração de *Doctype*, o programa retorna *FALSE*. Caso o documento possua a declaração, o *software* compara se esta é a mesma recomendada pelo W3C, retornando um resultado positivo apenas se essa afirmativa for verdadeira.

3.2.2 A página usa o *character set* de codificação de caracteres correto?

O IETF⁸, instituição que produz e mantém documentos de especificações técnicas de padrões que são utilizados em toda a internet em sua RFC 2277⁹, que trata sobre a política de conjunto de caracteres e idiomas define o termo *charset* como:

[...]um conjunto de regras para o mapeamento de uma sequência de octetos em uma sequência de caracteres, tal como a combinação de um conjunto de caracteres codificados e um esquema de codificação de caracteres (INTERNET ENGINEERING TASK FORCE, 1998).

Conforme exemplifica Park, Lim e Lim (2014), o *charset* é método de codificação para armazenar conjuntos de caracteres e sinais no computador.

⁷ Disponível em: <https://www.w3.org/QA/2002/04/valid-dtd-list.html>

⁸ The Internet Engineering Task Force.

⁹ Disponível em: <https://www.ietf.org/rfc/rfc2277.txt>

Desse modo a leitura de um documento é feita de acordo com o esquema de codificação de caracteres informado, fato este que comprova a necessidade da declaração do *charset* para a legibilidade apropriada evitando erros de interpretação.

Segundo o World Wide Web Consortium (2017a), o esquema de codificação de caracteres — *encoding*, apropriado é o UTF-8, chamado de *character set* universal.

O UTF-8 é um padrão industrial projetado para gerenciar e expressar caracteres utilizados no mundo todo podendo exibir sítios em qualquer idioma no mesmo padrão. Dessa forma um arquivo em Coreano pode ser exibido sem nenhum problema em um computador que não utilize o este idioma conforme explica Park, Lim e Lim (2014).

Para validação deste item da lista de verificação é utilizada a biblioteca *Jsoup*.

Após a conexão com a URL informada é feito o *parsing* do conteúdo HTML, que a partir deste momento se torna manipulável através da instanciação da classe *Document* da biblioteca, possibilitando a extração de dados e informações da página.

Todos os nós da página são varridos em busca de incidências da tag *<meta>*, onde é declarado o *charset* da página conforme ilustrado na figura 6¹⁰. A cartilha menciona que é possível a utilização do pseudo-atributo *<?xml version="1.0" encoding="utf-8" ?>*, porém esta verificação não foi contemplada pela ferramenta.

Figura 6 – Declaração do charset em meta tags

```
<!-- HTML 4 -->
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<!-- HTML5 -->
<meta charset="utf-8"/>
```

Fonte: Site Tableless.

O software então verifica se ocorre a declaração dos atributos *charset*, para HTML 5, ou *content*, para HTML de versões inferiores. Se esses atributos não forem declarados o programa retorna *FALSE*, invalidando este item de verificação.

Caso os atributos sejam declarados o seu conteúdo é extraído e comparado com o charset adequado, o UTF-8. Sendo a comparação verdadeira assume-se que o item está de acordo com a recomendação do *W3C* e da cartilha, retornando *TRUE*, e caso não seja o programa retorna *FALSE*.

Para a declaração do *content*, utilizado em versões inferiores ao HTML 5, devido a característica desse atributo em poder receber mais de um parâmetro como visto na Figura 6, que as palavras-chave *"text/html"*, indicador do tipo de conteúdo da página e *"charset=utf-8"* são utilizadas, todo o conteúdo é extraído e as palavras-chave são

¹⁰ Disponível em: <https://tableless.com.br/html-encode-utf-8/>. Acesso em dez. 2017.

separadas pela incidência do caractere “;” onde após esse procedimento, é verificado se alguma palavra-chave é igual a *string* “*charset=utf-8*”, que corresponde a declaração correta do *charset* dentro do atributo *content*.

Procedimento parecido é utilizado para a validação do atributo *charset*, utilizado no HTML 5, porém de forma simplificada, visto que este atributo recebe somente a palavra-chave referente ao *encoding*. Então é feita a comparação se a palavra-chave informada é igual a *string* “UTF-8”.

3.2.3 A página usa codificação válida?

A linguagem referenciada pela cartilha e utilizada para codificação de documentos é o HTML - Linguagem de Marcação de Hipertexto. O HTML foi criado por Tim Berners-Lee, criador do W3C, organização internacional que regulamenta padrões web. O HTML atualmente está na sua versão 5.

Segundo o World Wide Web Consortium (2017b), documentos HTML consistem de uma árvore de elementos e texto. Cada elemento é denotado no código com uma tag de início e uma final, sendo que em certos casos, essas tags podem ser omitidas e implícitas por outras tags. Um documento HTML básico pode ser observado na Figura 7¹¹ :

Figura 7 – Estrutura básica de um documento HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Sample page</title>
  </head>
  <body>
    <h1>Sample page</h1>
    <p>This is a <a href="demo.html">simple</a> sample.</p>
    <!-- this is a comment -->
  </body>
</html>
```

Print da página de recomendações do W3C

O HTML, então, é utilizado para definir a estrutura de um documento web de forma semântica utilizando dos elementos disponíveis na linguagem. HTML 5 é

¹¹ Disponível em: <<https://www.w3.org/TR/html51/introduction.html#a-quick-introduction-to-html>>. Acesso em dez. 2017.

a atual versão, e define a quinta revisão principal da linguagem central da *World Wide Web* (WORLD WIDE WEB CONSORTIUM, 2007a). Nesta versão foram implementados novos elementos, a possibilidade de utilização de APIs, melhoria em formulários, suporte nativo para gráficos vetoriais, melhorias em vídeos e áudio sem a necessidade de *plugins* entre outros.

Park, Lim e Lim (2014) explica que esta versão é uma especificação de padrão industrial que foi inicialmente desenvolvida por fornecedores de navegadores web. Esse grupo se chamou de WHATWG¹² e contava com membros da Opera, Mozilla e Apple conforme especifica LAWSON e SHARP (2011).

Em 2006, o W3C indicou interesse em participar do desenvolvimento do HTML 5, e em 2007 foi formado um grupo de trabalho para trabalhar com o WHATWG na especificação. As empresas Apple, Mozilla e a Opera permitiram ao W3C publicar a especificação sob os direitos autorais do W3C, mantendo uma versão com a licença menos restritiva no site WHATWG (WORLD WIDE WEB CONSORTIUM, 2017b). Após alguns anos trabalhando juntos, em 2012, devido aos pontos objetivos diferentes entre os grupos, o W3C cria uma nova equipe para cuidar do HTML 5 e preparar um rascunho de trabalho para a próxima especificação.

Nesse contexto, a validação deste item é baseada na linguagem de marcação definida acima, onde se verifica a incidência de erros na linguagem conforme descrito na cartilha de codificação:

Validação é a verificação que se faz para saber se um documento obedece às regras de linguagem em que foi escrito. Pode-se comparar a validação com a busca por erros gramaticais e de concordância em um texto escrito (BRASIL, 2010a).

A forma de validação implementada é simples e utiliza a biblioteca *Jcabi W3C*. A validação ocorre após a conexão com a página onde é realizado o *parse* do documento que é convertido em uma *string*. Após isso a *string* contendo todo o conteúdo HTML é submetida como parâmetro ao método *validate()* da classe *ValidationResponse* que retorna *TRUE* caso não ocorra problema na validação, e *FALSE* caso contenha algum erro no código.

A biblioteca *Jcabi W3C*, utiliza o serviço de validação de marcação do próprio W3C para verificar a adequação do código. Quando o programa é executado a biblioteca acessa de forma transparente o serviço *Nu Html Checker*¹³ onde a resposta contém todas as informações fornecidas pelo servidor W3C.

¹² Grupo de trabalho que desenvolve e mantém HTML e tecnologias vinculadas a ele.

¹³ Disponível em: <<https://validator.w3.org/nu/>>. Acesso em dez. 2017.

O *Nu Html Checker* é o *backend* de outros serviços de validação como o checker.html5.org, html5.validator.nu e validator.w3.org/nu e seu código fonte está disponível em sua página no *GitHub*¹⁴ sob a licença MIT¹⁵.

Este serviço implementa os requisitos de conformidade de documentos nas especificações HTML utilizando como parâmetro as seguintes seções da especificação:

- Justificativa para erros de nível de sintaxe; e
- Justificativa para restrições sobre modelos de conteúdo e sobre os valores dos atributos.

Nestas seções, resumidamente, haverá indicativos de erros para marcações onde podem ocorrer potenciais erros de segurança, acessibilidade, usabilidade, interoperabilidade ou porque podem resultar em um desempenho fraco ou podem fazer com que seus scripts falhem de maneiras difíceis de solucionar, conforme descreve o desenvolvedor da ferramenta.

3.2.4 As folhas de estilo CSS usadas pela página são válidas?

Conforme citado no item anterior, a validação é uma forma de verificar se um documento está de acordo com as regras da linguagem em que foi escrito.

O alvo deste item de validação são as folhas de estilo em cascata, comumente chamadas apenas de CSS, abreviação do termo em inglês *Cascading Style Sheet*, onde sua primeira versão foi lançada em 1996 como recomendação oficial do W3C.

O CSS pode ser definido da seguinte forma:

CSS é a linguagem para descrever a apresentação de páginas da Web, incluindo cores, layout e fontes. Permite adaptar a apresentação a diferentes tipos de dispositivos, como telas grandes, telas pequenas ou impressoras. O CSS é independente do HTML e pode ser usado com qualquer linguagem de marcação baseada em XML(WORLD WIDE WEB CONSORTIUM, 2016a).

Conforme descrito na cartilha de codificação do e-PWG (BRASIL, 2010a), através das folhas de estilo em cascata é possível controlar todo o *layout* do documento sendo parte da camada de apresentação que é a parte visual do conteúdo.

Buckley (2015), diz que todos os navegadores modernos implementam o CSS para renderização de documentos HTML, porém, navegadores de diferentes empresas

¹⁴ Disponível em: <<https://github.com/validator/validator>>. Acesso em dez. de 2017.

¹⁵ Licença permissiva criada pelo Instituto de Tecnologia de Massachusetts comumente utilizada em softwares livres.

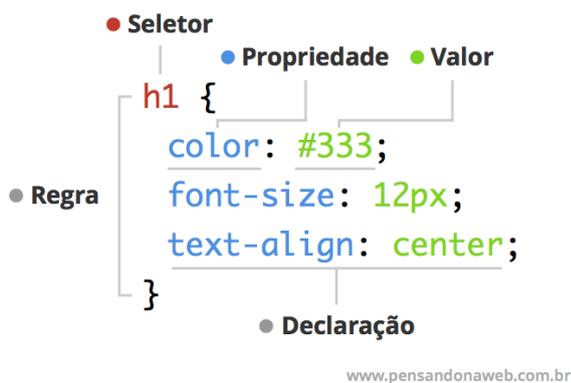
utilizam mecanismos de *layouts* - *layout engines*, para realizar a renderização do estilo, e cita as seguintes *engines* como as mais populares atualmente:

- Gecko (empregado pelo Firefox da Mozilla);
- Trident (empregado pelo Internet Explorer da Microsoft);
- KHTML (empregado pelo Konqueror do KDE);
- WebKit (empregado pelo Safari da Apple);
- Presto (empregado pelo Opera da Opera Software);
- Blink (empregado pelo Chrome do Google).

Os diferentes mecanismos de renderização podem, ao final, apresentar resultados minimamente diferentes uns dos outros, porém conforme orientação da cartilha de codificação devemos considerar que isso é algo inerente ao meio. O importante é que a página ou aplicação funcione (BRASIL, 2010a).

O CSS consiste na declaração de regras que tem por finalidade atingir itens codificados no documento HTML, podendo em uma única regra estilizar todos os elementos de um determinado tipo de uma vez, bem como estilizar itens específicos através de ids e classes. Conforme descreve (SILVA, 2015), a regra CSS é a unidade básica de uma folha de estilo e é composta de duas partes básicas, o seletor e a declaração conforme mostra a Figura 8¹⁶.

Figura 8 – Regra CSS



Fonte: site pensandonaweb.com.br

Descrição dos componentes da regra CSS:

¹⁶ Disponível em: <<http://pensandonaweb.com.br/aprendendo-html-a-linguagem-da-web/>>. Acesso em dez. de 2017.

- Seletor — elementos da marcação HTML que se quer modificar;
- Declaração — Composta de propriedade e valor, é a área que recebe os parâmetros de estilização;
- Propriedade — composto da característica a ser estilizada;
- Valor — é a quantificação ou qualificação da propriedade;

A cartilha de codificação diz que há três formas de se chamar ou anexar uma folha de estilo no documento HTML, porém apenas a folha externa é indicada pelos padrões e boas práticas. As outras duas formas, *inline* e interna devem ser evitadas pois misturam a camada de estrutura, definida pela marcação HTML com a camada de apresentação. A utilização da folha externa garante maior flexibilidade devido ao fato de qualquer necessidade de alteração, o arquivo HTML não precisará ser modificado. Além disso, o arquivo externo pode ser compartilhado por diversos documentos.

A declaração externa de arquivo CSS é feita através da tag `<link>` conforme descrito no código 3.1:

Código 3.1 – Declaração para arquivo CSS externo

```
<link rel="stylesheet" href="estilo.css" type="text/css" />
```

Segundo descrito nas recomendações do W3C, o atributo *rel*¹⁷ informa o tipo do *link* e se não foi declarado, não possui palavras-chave, ou se nenhuma das palavras-chave usadas é permitida, o elemento não cria nenhum *link*. A palavra chave *stylesheet* indica ao navegador que o *link* é um recurso externo. O *href*¹⁸ define o destino do *link* e não deve estar vazio e o atributo *type*¹⁹ é meramente consultivo, sua finalidade é indicar ao cliente o MIME type²⁰ do recurso vinculado.

O processo de validação deste item consistiu em capturar o CSS utilizado pelo HTML e submetê-lo ao método de validação da biblioteca *Jcabi W3C*, assim como realizado na seção 3.2.3.

Como descrito acima há apenas uma maneira indicada pra se utilizar o CSS, que é através de um *link* para um arquivo externo, portanto nas páginas que tem a incidência de estilos *inline* ou internos, considerou-se que o código não está seguindo as orientações da cartilha, logo é considerada não apta.

¹⁷ Disponível em: <<https://www.w3.org/TR/html5/document-metadata.html#attr-link-rel>>. Acesso em dez. de 2017.

¹⁸ Disponível em: <<https://www.w3.org/TR/html5/document-metadata.html#attr-link-href>>. Acesso em dez. de 2017.

¹⁹ Disponível em: <<https://www.w3.org/TR/html5/document-metadata.html#attr-link-type>>. Acesso em dez. de 2017.

²⁰ Mecanismo que informa ao cliente qual o tipo do documento.

A captação do CSS para validação se deu através da biblioteca *Jsoup*, de forma que o seguinte processo foi seguido:

- Varredura de todos os elementos do tipo `<link>` no documento HTML;
- Verifica-se se o elemento `<link>` possui o atributo `“rel”` com a palavra-chave `“stylesheet”` declarada;
- Caso a premissa anterior seja verdadeira é obtida a localização do arquivo presente no atributo `“href”`;
- Realiza-se uma tentativa de conexão com esse `link`; se foi possível o acesso ao arquivo, o seu conteúdo é capturado e submetido ao método de validação da biblioteca *Jcabi W3C*, que retorna os valores `TRUE`, caso a validação esteja correta e `FALSE` caso não.

A biblioteca *Jcabi W3C*, de forma transparente, utiliza o *CSS Validation Service*²¹, software livre criado pelo W3C para auxiliar *web designers* e desenvolvedores web na verificação de folhas de estilo em cascata (WORLD WIDE WEB CONSORTIUM, 2007b). O serviço de validação utiliza como referência as especificações atuais de CSS mantidas pelo W3C.

3.2.5 Há declarações de classes ou ID's desnecessárias?

As classes e os id's são atributos utilizados na marcação para identificação de elementos ou grupos de elementos. Conforme especificação do W3C os id's deverão ser exclusivos de um único elemento; de forma contrária, as classes podem ser declaradas em diversos elementos na marcação HTML formando um agrupamento que possibilita a detecção desses elementos por seletores CSS, métodos como o `getElementsByClassName()` no DOM e outros recursos desse tipo.

A cartilha de codificação não é muito específica quanto ao atendimento deste item, porém, o seguinte texto é informado: “Utilize os atributos `“class”` e `“id”` com parcimônia[. . .]” (BRASIL, 2010a).

Dessa forma, para avaliação deste item da lista foi utilizada a seguinte abordagem:

- Verificação de classes e id's declarados, porém, sem referências nos códigos CSS e JavaScript;

²¹ Disponível em: [<https://jigsaw.w3.org/css-validator/>](https://jigsaw.w3.org/css-validator/). Acesso em dez. de 2017.

- Verificação de classes e id's que estão nulos, ou seja, os atributos “*class*” e “*id*” estão presentes em algum elemento na marcação, porém, não foram atribuídas as palavras-chave, tornando a declaração dos atributos sem utilidade.

A validação inicia varrendo o código HTML e verificando se há incidência dos atributos “*class*” ou “*id*” em todos os elementos declarados. Caso algum elemento possua uma declaração nula a validação retorna *FALSE*, indicando que o código não está apto para este item de verificação.

Nos elementos que possuem id's ou classes declaradas corretamente o programa captura as palavras-chave informadas e as armazena em uma lista. Após esse procedimento o sistema captura o conteúdo das folhas de estilo e códigos *JavaScript* utilizados pelo documento HTML e analisa se há a incidência das palavras-chave que foram armazenadas anteriormente na lista. Caso alguma palavra-chave não tenha correspondência o programa retorna *FALSE*. Se todas as palavras-chave estiverem sendo utilizadas, o programa retorna *TRUE*, considerando apto este item de validação.

3.2.6 Há *links* quebrados?

De acordo com Xavier (2013) o *link* é ideia motriz do hipertexto. É ele que dá origem, visibilidade e viabilidade ao hipertexto em rede.

Fragoso (2009) exemplifica dizendo que os *hiperlinks* são conexões automatizadas que dão acesso a outro módulo de informação, sendo grandemente flexíveis podendo ser utilizados de diversas formas com objetivos e motivações distintas.

Como “unidades de conexão em hipertexto” (WORLD WIDE WEB CONSORTIUM, 2003), fazem referências de um documento para outro (*link* externo) ou referenciam locais de um mesmo documento (*link* interno). Por meio do *link* é possível acessar os mais diversos recursos disponíveis em formato digital, como páginas web, documentos dos mais variados formatos, imagens, áudios, vídeos entre outros.

O termo “*link quebrado*” pode ser entendido quando a premissa básica de um *link* não é atendida, ou seja, quando aponta para um recurso que não está disponível. Kolb, Ruckenbauer e Esterbauer (2001) exemplificam que a origem de um *link* quebrado pode ocorrer quando são apagados dados referenciados de um servidor que hospeda recursos, dados referenciados por outras instâncias.

Partindo desses princípios a verificação deste item é feita realizando requisições para os *links* informados no documento HTML. O sistema verifica todas as âncoras, que neste contexto são os *links* que apontam para o mesmo documento, referenciadas pelo elemento `<a>`, e demais URLs informadas nos atributos “*src*” e “*href*” nos elementos que possuem esses atributos declarados.

A definição de uma âncora ocorre quando qualquer elemento da árvore possui atributos do tipo “*id*” ou “*name*” declarados e não nulos, e estes são referenciados pelo atributo “*href*” de uma tag `<a>` tendo como escopo de referência outra seção do mesmo documento.

O W3C define o atributo *name* da seguinte forma:

Este atributo nomeia a âncora atual para que seja o destino de outro link. O valor desse atributo deve ser um nome de âncora exclusivo. O escopo deste nome é o documento atual. (WORLD WIDE WEB CONSORTIUM, 1999b).

Quanto a utilização dos *id*'s para formação de âncoras a especificação do World Wide Web Consortium (1999a) informa que “o atributo *id* pode ser usado para criar uma âncora na tag de início de qualquer elemento (incluindo o elemento A)”.

A verificação das âncoras começa com uma varredura em todos os elementos marcados no documento HTML que possuem declarados os atributos *id* ou *name*. As palavras-chave declaradas nesses atributos são armazenadas em uma lista para posterior comparação.

O próximo passo é o mapeamento do conteúdo existente no atributo *href* presente na tag `<a>`, onde o *link* declarado começa com uma cerquilha, conhecida pelo caractere “#”, seguido por uma palavra-chave como mostra a figura 9.

Figura 9 – Referenciando link interno

```
<a href="#section-example-1" title="Acesse o exemplo 1">  
  Seção de exemplo 1  
</a>
```

Fonte: Própria (2017).

O conteúdo declarado no atributo *href* então é comparado com as palavras-chave que foram inseridas na lista, caso o link contenha uma palavra-chave que não encontre correspondência é retornado *FALSE*, indicando, dessa forma, que neste item a recomendação não está sendo atendida.

Prosseguindo, o programa segue validando o conteúdo inserido nos atributos *href* e *src*. O conteúdo inserido no atributo *src* especifica a localização de um recurso externo sendo comumente utilizado para referenciar imagens nas tags `` e também para apontar para *scripts* que manipulam o comportamento da página, através da tag `<script>`. O *href* é usado para encontrar recursos na web, sendo em grande parte declarado nas tags `<a>` definindo um *link* entre a tag atual e o endereço de destino e pela tag `<link>` contendo a localização de arquivos CSS.

Para cada URL inserida nos atributos acima mencionados é realizada uma conexão para obtenção do *Response Code* da requisição, da mesma forma utilizada na Seção 3.1.1. Caso o código de resposta de alguma requisição esteja fora do escopo aceitável, 2xx: Sucesso e 3xx: Redirecionamento, não realizando a conexão, fica constatado que o *link* não cumpre com sua finalidade, estando quebrado, logo este item de verificação não é atendido.

Caso todas as conexões às URLs informadas sejam realizadas assume-se então que o este item da lista está de acordo com o recomendado.

3.2.7 Qual é a performance velocidade de carregamento/tamanho da página?

Os serviços eletrônicos disponibilizados pelo Governo Brasileiro permitem estreitar a relação com os cidadãos e estão embasados em três aspectos fundamentais, sendo eles a universalização dos serviços, governo ao alcance de todos e infraestrutura avançada conforme explica Oliveira e Eler (2015).

Nesse contexto, para o desenvolvimento desses serviços há uma gama de recursos que estão disponíveis para chamar a atenção do usuário como vídeos, imagens e áudios. No entanto, estes devem ser utilizados com moderação, pois “a sobrecarga de recursos diminui a velocidade de exibição da página, o que degrada a experiência do usuário (ROCHA; PIMENTEL, 2013, p.10).

Uma das diretrizes da cartilha de usabilidade do e-PWG é o respeito a velocidade de conexão do público-alvo, pois grande parte dos cidadãos ainda enfrentam problemas de velocidade na conexão e mesmo em instituições educacionais, por motivos internos, a banda larga pode se tornar lenta (BRASIL, 2010b). Essa latência na transferência de dados, faz com que a disponibilização do conteúdo seja demorada comprometendo a utilização do sistema.

Dessa forma ao abordar este tema a cartilha de codificação orienta que as páginas devem ser leves, onde “[. . .] devem ter preferencialmente até 50Kb, somados código, conteúdo e imagens. Recomenda-se que as páginas não ultrapassem 70Kb de tamanho” (BRASIL, 2010a).

Para a checagem deste item da lista é utilizado a API do serviço *Google PageSpeed Insights*²² que segundo (CARVALHO, 2015, p. 54), foi desenvolvido com intuito de ajudar programadores na criação de sítios mais rápidos e adaptáveis a dispositivos móveis.

Ao utilizar esse serviço, que requer apenas uma URL como parâmetro, é retornado uma lista de sugestões para tornar essa página mais rápida e outras informações

²² Disponível em: <<https://developers.google.com/speed/pagespeed/insights/>>. Acesso em dez. 2017.

como os tamanhos dos arquivos CSS e JavaScript, das imagens que estão sendo exibidas e do HTML como pode ser observado na figura .

Figura 10 – Resposta do serviço Google PageSpeed

```
{
  "kind": "pagespeedonline#result",
  "id": "https://www.ufpe.br/",
  "responseCode": 200,
  "title": "Universidade Federal de Pernambuco - UFPE",
  "ruleGroups": {
    "SPEED": {
      "score": 67
    }
  },
  "pageStats": {
    "numberResources": 125,
    "numberHosts": 10,
    "totalRequestBytes": "26909",
    "numberStaticResources": 70,
    "htmlResponseBytes": "296345",
    "textResponseBytes": "94358",
    "cssResponseBytes": "694661",
    "imageResponseBytes": "1506654",
    "javascriptResponseBytes": "1796111",
    "otherResponseBytes": "45725",
    "numberJsResources": 35,
    "numberCssResources": 12
  }
},
```

Fonte: <<https://www.googleapis.com/pagespeedonline/v2/runPagespeed?url=http://www.ufpe.br>>. Acesso em dez. 2017.

O programa ao obter os dados deste serviço, faz a soma dos tamanhos dos arquivos HTML, JavaScript, CSS e Imagens, retornados em *bytes*, e converte-os para *kilobytes*, unidade de medida informada na cartilha, e verifica se o valor ultrapassa o recomendado. Caso o resultado esteja acima de 70Kb, valor recomendado pela cartilha, o item não é considerado apto.

3.2.8 A página utiliza URLs amigáveis?

URLs amigáveis, são definidas assim de acordo com Solaro e Yu (2007) pois são fáceis de ler e entender pelos seres humanos, são facilmente rastreadas por motores de busca, navegadores e outras aplicações web.

Os mesmos autores dizem que:

Uma URL amigável não inclui cadeias de consulta e, de preferência, não se refere diretamente a nenhum script de página do servidor ativo (ASP), script de interface de gateway comum (CGI) ou outro módulo executável (SOLARO; YU, 2007).

A orientação da cartilha de codificação para este caso é que cada página de um sítio deve ser identificada por uma URL única e compreensível salientando “que muitos mecanismos de busca não indexam páginas que contenham pontos de interrogação

(?) ou outro caractere (como um & ou =) na sua URL“ (BRASIL, 2010a), conforme apresentado na figura 11.

Entende-se então que uma URL amigável não deve possuir parâmetros de consulta, artifício amplamente utilizados por programadores ao enviar dados por formulários. Além disso, deve-se evitar informar a extensão do arquivo que está sendo acessado, geralmente associados a linguagem que está sendo utilizada para codificação, como o “.php” para o PHP ou o “.jsp” para o Java.

Figura 11 – URL amigável

Certo: <http://www.portal.gov.br/contato>

Errado: http://www.portal.gov.br/default.php?reg=2&p_secao=18

Fonte: Padrões Web em Governo Eletrônico e-PWG - Cartilha de Codificação

Dessa forma a checagem deste item é realizada verificando se na URL informada há a incidência dos caracteres, “?”, “&” e “=”. Caso algum destes caracteres esteja presente na URL assume-se que este item não está de acordo com o recomendado.

3.2.9 As URLs funcionam sem o “www”?

A World Wide Web, comumente conhecida como a própria internet, foi criada por Tim Berners-Lee em 1989, foi criada para encorajar físicos a compartilharem informações sob uma rede (BERNERS-LEE, 1994).

O seu acrônimo WWW desde então se faz presente nas URLs como forma de indicar que tal recurso está na internet e faz parte da grande teia de hipermídia que conecta o mundo todo através do protocolo de transferência de hipertexto. O acrônimo então é registrado nos servidores DNS como sendo parte da URL, antecedendo o nome do *host* que hospeda o recurso.

Entretanto, conforme a especificação do Protocolo de Transferência de Hipertexto — HTTP, pelo (INTERNET ENGINEERING TASK FORCE, 1999, p. 18) visualizada na figura 12, o acrônimo WWW não faz parte do esquema http URL, não sendo necessária sua utilização.

Figura 12 – Esquema http URL

```
http_URL = "http:" "://" host [ ":" port ] [ abs_path [ "?" query ] ]
```

Fonte: Internet Engineering Task Force

A cartilha de codificação aconselha então que a URL responda sem o WWW, e informa que muitos navegadores já desconsideram o trígama, podendo o usuário digitar apenas o nome do domínio (BRASIL, 2010a).

Para validação desta diretriz é necessário informar a URL que se quer verificar omitindo o WWW. O sistema então tenta realizar uma conexão com o endereço informado. Caso a conexão seja feita com sucesso assume-se que o item está de acordo com a recomendação.

Para as URLs que não respondem, o sistema insere o WWW no endereço e realiza uma nova conexão, para que os demais itens da lista possam ser validados, porém, para este item específico a recomendação já não é mais cumprida.

Esta validação é a primeira a ser realizada pelo programa, na conexão inicial do sistema descrita na seção 3.1.1.

3.2.10 Há erros de JavaScript?

O JavaScript é uma linguagem de programação interpretada baseada em objetos que em conjunto com elementos HTML pode ser usada para tornar as aplicações mais fáceis de usar e flexíveis conforme explica Stothard (2000).

FLANAGAN (2006, p. 1) diz que o JavaScript é mais comumente utilizado em navegadores de internet e que o propósito geral da linguagem é permitir que scripts possam interagir com o usuário, controlando o navegador e alterando o conteúdo do documento exibido nas janelas do *browser*. Essa utilização do Javascript é conhecida como *client-side*, onde o código é embutido nas páginas HTML e o processamento é feito localmente na máquina do usuário e não no servidor.

A cartilha de codificação insere o JavaScript na camada de processamento, camada esta que tem por objetivo melhorar experiência do usuário manipulando os elementos da camada de conteúdo que é definida pela marcação HTML (BRASIL, 2010a).

O mesmo documento informa que a forma correta de utilização do JavaScript deve ser sempre por arquivos externos e nunca utilizando codificação dentro do documento HTML, evitando a mistura das camadas de conteúdo e processamento. A forma correta de se chamar um arquivo JavaScript é apresentando no código 3.2.

Código 3.2 – Referenciando arquivo JavaScript

```
<script type="text/javascript" src="/js/script.js"></script>
```

A especificação que o JavaScript implementa é conhecida como *ECMAScript*,

padronizada pela *ECMA International*²³ na especificação ECMA-262²⁴.

Para este item, foi adotado o mesmo procedimento utilizado na validação das camadas de conteúdo (HTML) e apresentação (CSS) ou seja, a busca por erros no código que podem prejudicar o funcionamento da página.

O processo é iniciado capturando todo código JavaScript descrito no documento HTML através da tag `<script>`. Após isso é utilizada a biblioteca *Rhino*, que recebe o código por parâmetro em seu método `parse()`. O método analisa o código e verifica se há inconformidades, onde, caso ocorram, é constatado que há erros na escrita do código. Também é verificado se há incidência de código JavaScript dentro do documento HTML, inserido na tag `<script>`, caso ocorra, assume-se que a página não atende ao item desta seção.

A biblioteca *Rhino* é uma implementação *open source* de JavaScript escrita em Java, que possui todos os recursos da versão 1.7 do JavaScript. A biblioteca conta com um compilador que traduz o código JavaScript para Java, e também outros recursos como um *shell* para execução dos *scripts*.

3.2.11 Há quebra do desenho quando o usuário aumenta o tamanho da fonte?

Este item inicialmente foi pensado em ser validado apenas com a intervenção manual, pois, é necessário a ação do usuário para manipular o tamanho da fonte no navegador, porém, a cartilha de codificação orienta aos desenvolvedores utilizarem unidades de tamanho relativas:

Evite o uso de tamanhos de fontes em unidades absolutas px ou pt nas folhas para mídias digitais, dando preferência pelo uso de tamanhos relativos tipo em e %. Para o tamanho padrão de texto utilize 1em ou 100%, use tamanhos menores apenas para informações de copyright ou semelhante(BRASIL, 2010a).

As unidades de medida são recursos utilizados para se especificar o tamanho de determinados elementos da página. Diversas propriedades do CSS necessitam dessa definição para especificação de largura, margem, espaçamento interno, bordas e fontes.

SCHEIDT (2015) explica que no tipo de unidade de medida relativa o tamanho de um elemento é afetado pelo tamanho do outro, sendo que este tipo é redimensionado de melhor forma em diferentes tipos de mídias:

²³ Associação criada em 1961, dedicada a padronização de sistemas de informação e comunicação (PINHO, 2017, p. 1)

²⁴ Disponível em: <http://www.ecma-international.org/publications/standards/Ecma-262.htm>. Acesso em dez. 2017.

- *em*: Relativo ao tamanho da fonte base. Se o tamanho do *font-size* do *body*, for 15px e o tamanho do elemento for 2em, isso definirá o seu tamanho em 30px.
- %: Percentagem - define o tamanho do elemento em relação ao contêiner em que está contido. Um elemento descrito com largura de 100%, preencherá completamente a largura de seu contêiner, independente do tamanho da tela.

Dessa forma, visando atender a recomendação descrita na cartilha, onde é preferível utilizar medidas relativas nas fontes, o sistema realiza esta validação, primeiramente obtendo todo o código CSS relacionado ao HTML. Em seguida, são analisadas todas as propriedades CSS que estão modificando os valores dos tamanhos das fontes. Caso o valor de alguma propriedade esteja utilizando as unidade de medida absolutas “*pt*” ou “*px*”, este item da lista é considerado inválido.

Conforme especificação, a propriedade “*font-size*” indica a altura desejada de glifos da fonte (WORLD WIDE WEB CONSORTIUM, 2013). É possível, também, a utilização da forma abreviada, utilizando apenas a palavra “*font*,” porém desta forma é possível não somente informar o tamanho da fonte, mas também outras características como a família, o estilo, o peso entre outros.

As figuras 13 e 14 abaixo, mostram a devida utilização dessas propriedades para manipulação dos tamanhos.

Figura 13 – Declaração *font-size*

```
em { font-size: 150% }  
em { font-size: 1.5em }
```

Fonte: World Wide Web Consortium.

Figura 14 – Declaração *font*

```
p { font: condensed oblique 12pt "Helvetica Neue", serif; }
```

Fonte: World Wide Web Consortium.

3.2.12 O contraste das cores da página é suficiente?

Este item é pouco abordado pela cartilha, é dito apenas que “é importante verificar se o jogo de cores da página proporciona contraste suficiente para leitura do conteúdo” (BRASIL, 2010a).

Entretanto, o Modelo de Acessibilidade em Governo Eletrônico — e-MAG, que tem por finalidade ser o norteador para o desenvolvimento e adaptação de conteúdos

digitais do Governo Federal, trata sobre este assunto, desse modo foi utilizado como base para o entendimento da proposta.

O Modelo de Acessibilidade recomenda oferecer contraste mínimo entre plano de fundo e primeiro plano, explicando que:

As cores do plano de fundo e do primeiro plano deverão ser suficientemente contrastantes para que possam ser visualizadas, também, por pessoas com baixa visão, com cromodeficiências ou que utilizam monitores de vídeo monocromático (BRASIL, 2014).

Consoante ao e-MAG o W3C dispõe um documento²⁵ fornecendo técnicas para ferramentas de avaliação e reparo de acessibilidade. O ponto de checagem deste documento em seu item 2.2 orienta para que se certifique de que as combinações de cores de primeiro plano e de fundo proporcionem um contraste suficiente quando vistas por alguém com deficit de cor ou quando vistos em uma tela em preto e branco. Também é informado que para uma boa visibilidade a diferença de cor e brilho entre duas cores devem ser superiores ao intervalo definido, sendo calculados da seguinte forma:

Diferença de brilho:

$$((\text{Red value} \times 299) + (\text{Green value} \times 587) + (\text{Blue value} \times 114)) / 1000.$$

Diferença entre duas cores:

$$(\text{maximum}(\text{Red value 1}, \text{Red value 2}) - \text{minimum}(\text{Red value 1}, \text{Red value 2})) + (\text{maximum}(\text{Green value 1}, \text{Green value 2}) - \text{minimum}(\text{Green value 1}, \text{Green value 2})) + (\text{maximum}(\text{Blue value 1}, \text{Blue value 2}) - \text{minimum}(\text{Blue value 1}, \text{Blue value 2})).$$

O cálculo trabalha com os valores RGB²⁶ das cores, onde os maiores valores são subtraídos dos menores correspondentes e por fim é realizada a soma geral. O resultado da diferença entre o brilho de duas cores não pode ser inferior a 125 e o resultado da diferença de duas cores não pode ser inferior a 500, caso contrário a diferença entre as cores não é considerada adequada.

O cálculo acima foi implementado na ferramenta, que trabalha da seguinte forma:

- O CSS é capturado, considerando tanto codificação interna quanto externa.
- São capturados os valores das cores de fundo, referentes das seguintes propriedades: “background”, “background-color”, “bgcolor” e “background-image”;
- São capturados os valores das cores das fontes referentes a propriedade: “color”;

²⁵ Disponível em: <<https://www.w3.org/TR/AERT/#color-contrast>>. Acesso em dez. 2017.

²⁶ Abreviatura de um sistema de cores onde segue o padrão Vermelho(R), Verde(G) e Azul(B).

- O cálculo da diferença de cores é aplicado entre as cores de fundo e cores de fonte;
- Caso ocorra uma diferença inferior aos valores adequados para brilho ou cor a validação não é atendida.

O sistema realiza a comparação de cores, de forma que para cada cor de fundo encontrada todas as cores de fonte são comparadas a esta. Também é feita a conversão automática dos valores declarados de forma hexadecimal para RGB para que os valores possam ser aplicados a regra de cálculo.

3.2.13 Itens não implementados

A lista de verificação possui um total de 21 itens, porém, nem todos foram implementados, pois, foi considerada a necessidade de intervenção manual para validação dos demais. Para o teste destes itens é necessária a intervenção direta do usuário nas configurações do navegador ou a visualização em outros dispositivos e *softwares* para análise da página desenvolvida.

Dois itens não continham informações necessárias para que fosse possível o entendimento, e portanto não foram contemplados. A falta de esclarecimento quanto ao que seriam os “aspectos da apresentação” e o que são as “imagens da apresentação” mencionados nos itens abaixo foi o fator decisivo para a não implementação destes.

- O sítio usa CSS para todos os aspectos da apresentação?
- As imagens de apresentação estão incluídas nas CSS?

A seguir segue uma breve descrição dos demais itens.

3.2.13.1 O código é bem estruturado?

A cartilha de codificação já apresenta a solução em seu conteúdo ao indicar que a validação manual é uma forma de testar este item.

[...]Uma forma de se testar se essa camada foi estruturada corretamente é abrir o documento, sem estilos, no navegador e verificar se este é compreensível com uma estrutura seqüencial, que permita a leitura de forma natural (BRASIL, 2010a).

E comprova que uma validação automática não garante a qualidade do código:

As ferramentas de validação permitem verificar não-conformidades no código ou CSS e é útil para o diagnóstico de erros no código que podem estar interferindo em seu funcionamento. No entanto, a validação não garante a qualidade da marcação, nem previne o uso inadequado e o abuso de tags (BRASIL, 2010a).

3.2.13.2 A página funciona com o JavaScript desabilitado?

É necessária a intervenção manual do usuário, alterando as configurações do navegador.

3.2.13.3 O conteúdo é acessível com as folhas de estilo desabilitadas?

É necessária a intervenção manual do usuário, alterando as configurações do navegador.

3.2.13.4 A página é consistente em navegadores texto?

A cartilha explica que:

O navegador de texto exibe todo o conteúdo das páginas na Internet em formato texto, sem imagem, sons ou animações, e pode ser usado tanto pelas pessoas portadoras de deficiência visual como pelas que acessam a Internet com conexão discada (BRASIL, 2010a).

Para tanto é necessária a intervenção humana, verificando se o conteúdo é apresentado de forma consistente nos programas navegadores de texto.

3.2.13.5 O conteúdo é legível quando impresso? Não há desperdício de papel ou tinta?

Apesar da possibilidade de se escrever um estilo próprio para impressão, é difícil dar certeza se este código irá poupar ou não recursos.

Este código apenas será utilizado ao imprimir a página, mas caso o programador utilize tipos de fontes inapropriadas, não formate corretamente o tamanho do texto, não remova banners, menus e outros conteúdos desnecessários, o desperdício continuará ocorrendo mesmo com a utilização de folha de estilo própria pra impressão.

3.2.13.6 A página é bem visualizada em dispositivos portáteis?

O CSS por meio de suas *Media Queries*, mecanismo que permite a atribuição de tipos de estilos diferentes para mídias específicos (WORLD WIDE WEB CONSORTIUM, 2012), permite que sejam desenvolvidas folhas de estilos específicas para os mais

diversos dispositivos, incluindo os portáteis. Entretanto, a utilização de *Media Queries* no código CSS não garante que a página seja bem visualizada, devendo o programador realizar o teste manual acessando a página em dispositivos de diversas resoluções para verificar a adequação da página.

3.2.13.7 A página é bem visualizada em diferentes resoluções de navegador e tela?

Assim como descrito no item anterior, é possível criar estilos específicos para resoluções diferentes, porém, o teste manual é necessário para comprovação da efetividade da boa visualização.

Neste capítulo, a implementação da solução proposta na pesquisa foi descrita contemplando as tecnologias abordadas pelos itens da lista de verificação e a lógica de processamento para avaliação destes itens. O próximo capítulo, apresenta uma análise de sites governamentais realizada pela ferramenta desenvolvida, que tem o objetivo de verificar o nível de conformidade desses sites à cartilha de codificação.

4 AVALIAÇÃO

Este capítulo apresenta os resultados da avaliação de sítios de universidades federais brasileiras, utilizando a ferramenta descrita no Capítulo 3, onde é discutido item a item o nível de adequação às recomendações da cartilha de codificação. Também apresentamos neste capítulo, a metodologia de testes utilizada como parâmetro para confirmação da efetividade da ferramenta. Desse modo, a Seção 4.1, descreve brevemente a metodologia utilizada para avaliação da solução oferecida a Seção 4.2 apresenta o resultado da avaliação dos sítios governamentais.

4.1 Metodologia

Segundo WASLAWICK (2014), o método de pesquisa diz qual o caminho para se chegar ao objetivo proposto. Na busca por resultados satisfatórios, os seguintes métodos foram utilizados conforme menciona SILVA e MENEZES (2005):

Do ponto de vista da natureza a pesquisa é aplicada, pois objetiva gerar conhecimentos para aplicação prática e dirigidos à solução de problemas específicos. Envolve verdades e interesses locais. Esse ponto alinha-se ao proposto devido ao fato de solucionar o objetivo da pesquisa, no qual refere-se ao problema específico de não haver uma forma automatizada (prática) para suporte no desenvolvimento de sítios com base nas diretrizes oferecidas de forma condensada em um único software.

Do ponto de vista do procedimento técnico a pesquisa será experimental: quando se determina um objeto de estudo, selecionam-se as variáveis que seriam capazes de influenciá-lo, definem-se as formas de controle e de observação dos efeitos que a variável produz no objeto. São descritas as técnicas, materiais e mecanismos para a realização da pesquisa. Esse ponto alinha-se ao proposto devido aos cenários de testes desenvolvidos onde a solução proposta é examinada, e também a posterior análise dos dados coletados utilizando a ferramenta em um universo específico escolhido.

A abordagem quantitativa é utilizada, uma vez que garante que este procedimento garante a precisão dos resultados e evita distorções da análise e da interpretação (GIL, 2006).

Segundo Leopardi (2002), a abordagem quantitativa assegura a objetividade e credibilidade dos achados, onde é observada no resultado da análise dos sítios governamentais, que confere o nível de adequação aos itens recomendados.

4.2 Cenários de teste

Com o objetivo de que a ferramenta desenvolvida fosse considerada apta, testes foram realizados para que os itens implementados respondessem de forma satisfatória aos itens da lista de verificação.

Os testes consistiram na criação e modificações de páginas HTML, bem como arquivos CSS e JavaScript, onde erros propositais foram inseridos. Após isso, essas páginas foram submetidas ao programa que realizou a validação conforme descrito nas especificações do Capítulo 3.

Para que fosse possível a realização dos testes, foi montando ambiente local para hospedagem das páginas HTML, onde foi utilizado o Apache HTTP Server¹. Dessa forma, nas sub-seções deste capítulo descreveremos como os itens da lista de verificação foram validados. Entretanto, dois itens não puderam ser validados localmente devido suas peculiaridades, são eles:

- Qual é a performance velocidade de carregamento/tamanho da página?; e
- As URLs funcionam sem o “www”?.

Para o teste de performance o serviço *Google Page Speed*, utilizado para obter o tamanho dos arquivos, não funcionou para uma URL hospedada em um ambiente local, apesar disso, várias páginas disponíveis na web foram utilizadas para verificação deste item com o intuito de observar se alguma anormalidade na execução da verificação aconteceria, fato este que não ocorreu.

Para a verificação das URLs com o trígama WWW, devido ao ambiente local não dispor de um servidor DNS para registro e alterações de nomes de domínio, a verificação dessa funcionalidade também foi realizada em outros sítios.

Foi identificado, que alguns sites não puderam ser validados pois a conexão sempre era recusada, lançando uma exceção, porém os sites podem ser acessados normalmente em navegadores.

Analisando a causa da exceção *java.net.NoRouteToHostException*², descobriu-se que normalmente isso ocorre quando o *host* remoto não pode ser alcançado por causa de um *firewall* interativo, ou se um roteador intermediário estiver desativado. Todavia, na análise dos sítios descritos nesta pesquisa, não houve ocorrências deste tipo.

¹ Disponível em: <<https://httpd.apache.org/>>. Acesso em dez. 2017.

² Disponível em: <<https://docs.oracle.com/javase/7/docs/api/java/net/NoRouteToHostException.html>>. Acesso em dez. 2017.

4.2.1 Doctype

Para garantir que a verificação do Doctype esteja ocorrendo de forma correta, em uma página HTML foram realizados os seguintes procedimentos:

- Doctype não declarado;
- Doctype incorreto declarado, onde foi utilizado o Doctype HTML 4.01 Strict³;
- Doctype inexistente, declarado com um nome qualquer, assumindo o cenário de um possível erro de digitação do programador;
- Doctype declarado corretamente, conforme descrito em 3.2.1.

Em todos os cenários a ferramenta se comportou de forma ideal.

4.2.2 Charset

Este item tem o objetivo de garantir que o *charset* recomendado seja utilizado no desenvolvimento das páginas HTML. Desse modo, as seguintes situações foram abordadas e implementadas no documento de teste que foi avaliado pelo programa:

- Tag `<meta>` não declarada, não sendo possível, dessa forma, a declaração do *encode*.
- *Charset* declarados nulos nos atributos *“content”* ou *“charset”* na tag `<meta>`;
- *Charset* inexistente declarado, para assumir um possível erro de digitação do programador;
- *Charset* incorreto declarado, onde foi utilizado o esquema de codificação ISO-8859-1;
- *Charset* declarado corretamente.

4.2.3 Código HTML e CSS

Para estes itens, erros de escrita no documento HTML e no código CSS foram propositalmente inseridos, esperando que a ferramenta não valide em ambos os casos. As seguintes situações foram contempladas:

- Título da página não informado entre na tag `<title>`;

³ Declaração completa: `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`

- Tags declaradas incorretamente conforme mostra o código 4.1:

Código 4.1 – tags não fechadas

```
<p>
  tag span não <span fechada </span>
</p>

<div class="abcd"
  A div acima não está fechada
</div>

`, elemento de bloco, foi declarado dentro de um `<span>` elemento de linha conforme o código 4.2:

#### Código 4.2 – Elemento de bloco

```

 <h1>cabeçalho errado</h1>

```

- Textos soltos dentro da tabela sem estarem inseridos em uma tag apropriada.
- Elementos fora de contexto, como um `<tr>`, indicador de uma linha de uma tabela, inserido dentro de um `<ul>`, elemento que cria uma lista não ordenada, como mostra o código 4.3:

#### Código 4.3 – Elemento fora de contexto

```

 <tr>texto</tr>

```

- Comentário declarado incorretamente, conforme o código 4.4:

#### Código 4.4 – Comentário incorreto

```
<!-- declaração incorreta, faltou um "-" no início da
 marcação -->
```

- Atributos declarados incorretamente, como o `href` e o `rel`, mostrados no código 4.5:

#### Código 4.5 – Declaração de atributos

```

 Clique aqui!

```

```
<link href="css/css.css" re="stylesheet" />
```

- Tag `<img>` sem o atributo “*alt*” declarado, conforme código 4.6;

#### Código 4.6 – Imagem sem atributo alt

```

```

- Estilização do documento interna, dentro da tag `<style>` e *inline*, utilizando o atributo *style* um elemento qualquer, como mostra os exemplos no código 4.7:

#### Código 4.7 – Estilização interna e inline.

```
<html>
//estilização interna
 <head>
 <style>
 body{
 //propriedade: valor;
 }
 </style>
 </head>
</html>

<div style="background-color: #000; padding: 20px;">
 Div com estilização inline
</div>
```

- Utilização de propriedades CSS inexistentes, como mostra o código 4.8:

#### Código 4.8 – Propriedades CSS inexistentes

```
div{
 cor: #F00;
 largura: 81em;
 margem: 2px 3px 2px 3px;
}
```

- Declaração de valores não válidos e/ou não reconhecidos na regra CSS, conforme o código 4.9:

#### Código 4.9 – Valores css não válidos

```
div{
 width: ab50px;
 background-color: #0099KK;
}
```

- Textos soltos no código CSS, ou seja, não estão dentro de comentários ou não fazem parte da regra CSS;
- Propriedades não separadas com o caractere ponto e vírgula ( ; ), exemplificado no código 4.10 :

**Código 4.10 – Propriedades CSS não separadas com ";"**

```
body{
 font-family: sans-serif
 font-size: 1em
 color: #999
 padding: 10px
}
```

O serviço de validação de marcação do W3C dispõe de uma lista de erros<sup>4</sup> comentados que auxiliam o programador a identificar os demais erros que podem ocorrer na marcação HTML. É mister informar que devido a grande possibilidade de erros que podem ocorrer para a invalidação do HTML, os testes realizados, foram assim feitos por serem considerados erros comuns que ocorrem durante o desenvolvimento. Erros de digitação, falta de delimitadores entre regras ou a ausência de elementos na marcação por exemplo, são itens considerados pelo serviço de validação indispensáveis.

#### 4.2.4 Classes e IDs desnecessários

Conforme descrito no Capítulo 3, classes e ids devem ser utilizados com cautela. Atualmente é grande a disponibilidade de *frameworks* CSS no mercado, e com seu uso baseado na utilização de classes já estilizadas, o documento HTML é saturado de declaração de atributos que propiciam uma utilização desnecessária favorecendo a quebra da recomendação.

Desse modo, como forma de validar a ferramenta conforme a lógica descrita na Seção 4.2.4 tais procedimentos foram adotados:

- Elemento com atributo “*class*” declarado porém sem palavra-chave definida;
- Elemento com atributo “*id*” declrado porém sem palavra-chave definida;
- Elemento com atributo “*class*” declarado, porém sem incidência da palavra-chave nos arquivos CSS e JS;
- Elemento com atributo “*id*” declarado, porém sem incidência da palavra-chave nos arquivos CSS e JS;

<sup>4</sup> Disponível em: <<https://validator.w3.org/docs/errors.html>>. Acesso em dez. 2017.

#### 4.2.5 Links quebrados

Esta validação tem o objetivo de eliminar a incidência de links quebrados, que muitas vezes ocorrem não por erro do programador mas pela indisponibilidade de recursos que foram removidos, tiveram seus endereços alterados ou por erro de usuários ao informarem links de forma errada através de CMS. Essa validação propicia que as ligações entre recursos sejam sempre válidas, e que no contexto de governo eletrônico, as informações oferecidas aos usuários estejam sempre disponíveis.

Como forma de testar se esta implementação está obedecendo a lógica descrita na Seção 3.2.6 as circunstâncias a seguir foram implementadas:

- Link para arquivo JavaScript inexistente, declarado no atributo “*src*” da tag `<script>`;
- Link para arquivo CSS inexistente, declarado no atributo “*href*” tag `<link>`;
- Âncora declarada no atributo “*href*” da tag `<a>` inexistente;
- Link para uma página de internet inexistente, declarado no atributo “*href*” da tag `<a>`;
- Referência a uma imagem inexistente, declarada no atributo “*src*” da tag `<img>`;
- Declarações nulas nos atributos “*href*” e “*src*”

#### 4.2.6 URLs amigáveis

A URL amigável como recurso de auxílio ao usuário, facilitando a identificação e compreensão da página acessada e validada pela ferramenta analisando o endereço informado e verificando a conformidade com a lógica descrita na Seção 3.2.8. Para isso o seguinte teste foi realizado:

- Foram criados arquivos HTML contendo em seus nomes os caracteres que não são recomendados.

Visando simular um ambiente real, o teste também foi realizado numa instância local do CMS Wordpress<sup>5</sup>. Esse gerenciador de conteúdo contempla um ambiente completo para disponibilização de sites e blogs. Foi escolhido por possuir a funcionalidade de alteração de *links* permanentes, que foram modificados, e submetidos a ferramenta para análise da validação conforme as Figuras 15 e 16.

<sup>5</sup> Disponível em: <<https://br.wordpress.org/>>. Acesso em dez. 2017.

Figura 15 – URL com o caractere “?”



Fonte: Própria (2017).

Figura 16 – URL amigável



Fonte: Própria (2017).

#### 4.2.7 Código JavaScript

Assim como na Seção 4.2.3, para a validação deste item, erros foram propositalmente inseridos em códigos JavaScript, esperando que não ocorra a validação. Os testes foram baseados em erros de sintaxe que estão descritos a seguir, e visualizados no código 4.11:

- Sinal “=” esperado;
- Sinais de parênteses ausentes;
- Sinais de colchetes ausentes;
- Caracteres ilegais em nomes de variáveis;
- Utilização da instrução “return”, de forma inválida, fora de uma função;
- Utilização do “try”, sem posterior declaração do “catch” ou “finally”, para tratamento de exceções.

Código 4.11 – Ausência de parênteses e chaves

```
@var1 @var2 = 1;

function minhaFuncao {
 //função sem o "()"
}

function minhaFuncao(){ //sem o colchete de fechamento

if(cont <> 1 { // if sem o ")"

var #@! = 2; //caracteres inválidos

var a = 1;
```

```
var b = 2;
if(a < b){
 return true; //return fora do escopo de uma função.
}

try {
 //código aqui
}
var a = "texto";
//ausência de catch ou finally após declaração do try.
```

#### 4.2.8 Fontes em unidades relativas

Esta validação verifica se há incidência de unidades de medidas em *pixels* (px) ou pontos (pt), para atribuição de valores de tamanho às fontes utilizadas. Dois testes foram realizados para esta validação, conforme apresentado no código 4.12:

- Declarações de tamanhos de fontes utilizando “*pt*” e “*px*” dentro utilizando a propriedade “*font-size*”;
- Declarações de tamanhos de fontes utilizando “*pt*” e “*px*” dentro utilizando a propriedade “*font*”.

#### Código 4.12 – Declarações de fontes

```
body{
 font: italic bold 16px Arial, Helvetica, Sans-serif;
}
.p1{
 font: italic bold 16pt Arial, Helvetica, Sans-serif;
}
.p2{
 font-size: 16px;
}
a{
 font-size: 16pt;
}
```

#### 4.2.9 Contraste

O contraste verifica a diferença entre duas cores. Conforme implementado na ferramenta, foram capturados os valores das cores de fundo, e os valores das cores de fontes. Os testes para este item consistiram em declarações de diversos valores nas propriedades acima citadas com a finalidade de comparar se o código está adequado ou não ao item 3.2.12.

Os valores para cores de fundo utilizados foram os seguintes:

- Branco: código hexadecimal #FFFFFF, e código RGB (255, 255, 255);
- Preto: código hexadecimal #000000, e código RGB (0, 0, 0).

Os valores das cores de fontes foram os seguintes:

- Uma variação de cinza claro: código hexadecimal #F5F5F5, e código RGB (245, 245, 245);
- Amarelo: código hexadecimal #FFFF00, e código RGB (255, 255, 0).

O resultado apresentado para a diferença de brilho entre duas cores foi :

- Branco com cinza claro: 30;
- Branco com amarelo: 255;
- Preto com cinza claro: 735;
- Preto com amarelo: 510;

O resultado apresentado para a diferença das cores foi:

- Branco com cinza claro: 10;
- Branco com amarelo: 30;
- Preto com cinza claro: 245;
- Preto com amarelo: 255;

Os valores 30, para a diferença de brilho e 10 e 30 para a diferença de cores, são inferiores ao recomendado, conforme descrito na seção 3.2.12, dessa forma, para este cenário, o resultado de não conformidade com a regra foi retornado pela ferramenta, que se mostrou coerente com os parâmetros estipulados na fórmula de cálculo para contraste.

### 4.3 Análise de sítios governamentais

Após a ferramenta ser submetida a testes para comprovar sua efetividade, foram escolhidos 26 sítios de universidades federais brasileiras, sendo um por estado, conforme descritos na tabela ?? presente no apêndice A. A avaliação foi feita entre os dias 13 e 17 de novembro de 2017. A página avaliada foi a *home page*, ou seja, a página inicial de cada sítio, exceto para avaliação das URLs amigáveis, onde foram escolhidas

páginas de notícias das referidas instituições por serem mais complexas e possuírem informações textuais em maior volume do que a URL presente na página inicial.

A Figura 17 apresenta o resultado final coletado através da ferramenta, onde *TRUE*, representado pela letra “T”, significa que o item foi validado corretamente e *FALSE* representando pela letra “F” que a página não está atendendo a recomendação. Os itens da lista de verificação na tabela são apresentados conforme ordem declarada na cartilha de codificação.

É observado que em nenhum item houve 100% de aprovação. É interessante informar que nos itens de validação de código HTML, de código CSS, IDs e Classes desnecessários e Performance, todos os sites avaliados foram reprovados. A descrição mais detalhada é vista nas seções deste capítulo.

**Figura 17 – Resultado da análise**

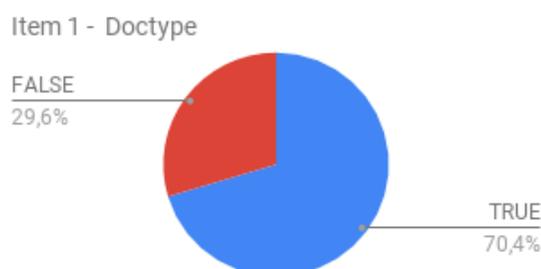
Estado	Itens de verificação											
	#1	#2	#3	#4	#5	#7	#8	#9	#10	#11	#16	#17
Acre	T	T	F	F	F	F	F	T	T	F	F	F
Alagoas	T	T	F	F	F	F	F	T	F	F	F	F
Amapá	F	T	F	F	F	F	F	T	F	F	F	F
Amazonas	F	T	F	F	F	F	F	T	T	F	F	F
Bahia	T	T	F	F	F	F	F	T	T	F	T	F
Ceará	F	T	F	F	F	F	F	T	T	F	F	F
Distrito Federal	T	T	F	F	F	F	F	T	T	F	F	F
Espírito Santo	F	T	F	F	F	F	F	T	T	F	F	F
Goiás	T	T	F	F	F	F	F	T	F	F	F	F
Maranhão	F	T	F	F	F	F	F	F	T	F	F	F
Mato Grosso	T	T	F	F	F	F	F	T	T	F	F	F
Mato Grosso do Sul	T	T	F	F	F	F	F	T	F	F	F	F
Minas Gerais	T	T	F	F	F	F	F	T	T	F	F	F
Pará	T	T	F	F	F	F	F	T	T	F	F	F
Paraíba	T	T	F	F	F	F	F	T	T	F	F	T
Pernambuco	T	T	F	F	F	F	F	T	F	F	F	F
Paraná	T	T	F	F	F	F	F	T	T	F	F	F
Piauí	T	T	F	F	F	F	F	T	T	F	F	F
Rio de Janeiro	F	T	F	F	F	F	F	T	T	F	T	F
Rio Grande do Norte	T	T	F	F	F	F	F	T	T	F	F	F
Rio Grande do Sul	F	T	F	F	F	F	F	T	T	F	T	F
Rondônia	F	F	F	F	F	F	F	F	T	F	F	F
Roraima	T	T	F	F	F	F	F	F	T	F	F	F
Santa Catarina	T	T	F	F	F	F	F	T	T	F	F	F
São Paulo	T	T	F	F	F	F	F	T	T	F	F	F
Sergipe	T	T	F	F	F	F	F	T	T	T	F	F
Tocantins	T	T	F	F	F	F	F	T	T	F	F	F

Fonte: Própria (2017).

### 4.3.1 Resultado *Doctype*

A análise do *Doctype* constatou que 70,4% dos sítios avaliados estão em conformidade com a cartilha como ilustrado no gráfico da Figura 18.

**Figura 18 – Análise do Doctype**



Fonte: Própria (2017).

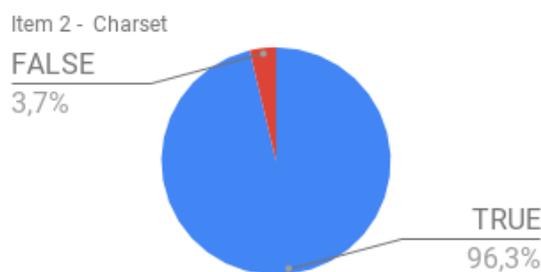
Visitando alguns dos sítios que não estão em conformidade, apesar de não constar a informação de quando foram criados, é notável pelo visual, que foram criados há alguns anos, e que provavelmente o padrão HTML 5, não havia sido difundido totalmente. Devido a este fato, é interessante ressaltar também, que muitas IDEs<sup>6</sup>, softwares utilizados pelos programadores para criação de páginas web, para agilizar o desenvolvimento, criam automaticamente a estrutura básica das páginas, de forma que o Doctype é inserido sem a intervenção do desenvolvedor.

Devido a possível época que essas páginas foram desenvolvidas, e a utilização de IDEs para a codificação das páginas, o doctype referente ao HTML 4.01, encontrado nas páginas que não estão em conformidade, foram inseridos nos documentos e por falta de atualização estes Doctypes permanecem até hoje.

### 4.3.2 Resultado *Charset*

A análise dos *charsets* utilizados nas páginas avaliadas constatou que grande parte dos sítios estão atendendo a recomendação com o total de adequação de 96,3%.

<sup>6</sup> Ambiente de Desenvolvimento Integrado

Figura 19 – Análise do *Charset*

Fonte: Própria (2017).

O único sítio avaliado que não está atendendo a recomendação é o da Fundação Universidade Federal de Rondônia - UNIR, onde o *charset* utilizado é o ISO-8859-1.

#### 4.3.3 Links Quebrados

A análise de links quebrados, constatou que apenas 5 universidades estão atendendo a recomendação. O percentual de desacordo é de 81,5% conforme ilustra o gráfico da Figura 20:

Figura 20 – Análise de Links quebrados



Fonte: Própria (2017).

Os erros mais recorrentes foram de links que retornaram o status da requisição nos valores 404, página não encontrada e erros na faixa 5xx, que são relacionados ao servidor. Abaixo seguem alguns exemplos:

- <https://www.ufpe.br/fale-conosco> — status: 404;
- <http://www.cce.ufpr.br/portal/> — status: 504;
- <http://www.icsa.ufpa.br/> — status: 500;
- <http://proexti.ufam.edu.br/politica-de-extensao> — status: 404;
- <http://www.proplan.ufes.br/materia.asp?codMater=129> — status: 404

- [http://portais.ufma.br/PortalUfma/paginas/pagina\\_interna.jsf?pagina=325&tipo=6](http://portais.ufma.br/PortalUfma/paginas/pagina_interna.jsf?pagina=325&tipo=6)  
— status: 500

#### 4.3.4 URL amigável

A análise de URLs amigáveis, foi realizada através de páginas de notícias encontradas nos sites das universidades. Utilizar a *home page*, não retrataria a realidade, devido ao fato de que a página inicial de um site, comumente é apenas o endereço registrado no servidor DNS como <https://www.unir.br/> ou <https://www.ufpe.br/>.

Desse modo, após a apuração, descobriu-se que 88,9% dos sites estão adequados a este item.

**Figura 21 – Análise de URLs amigáveis**



Fonte: Própria (2017).

Apenas 3 universidades não estão atendendo a esta recomendação: UNIR, UFMA e UFRR.

Algumas URLs que foram verificadas são apresentadas abaixo:

- <https://www.unir.br/index.php?pag=noticias&id=24324>;
- <http://ufrj.br/ultimas-noticias?start=90>;
- <http://portais.ufma.br/PortalUfma/paginas/noticias/noticia.jsf?id=50892>;
- <https://ufrj.br/noticia/2017/11/09/ufrr-recebe-exposicoes-e-apresentacoes-no-ii-estfic>;
- [https://www.ufpe.br/agencia/noticias/-/asset\\_publisher/VQX2pzmP0mP4/content/mpf-promove-seminario-sobre-direitos-indigenas-no-nordeste/40615](https://www.ufpe.br/agencia/noticias/-/asset_publisher/VQX2pzmP0mP4/content/mpf-promove-seminario-sobre-direitos-indigenas-no-nordeste/40615).

#### 4.3.5 Acessa sem WWW

Quanto às páginas de aprovação neste item que acessam sem o trígama WWW na URL, 81,5% atendem a recomendação.

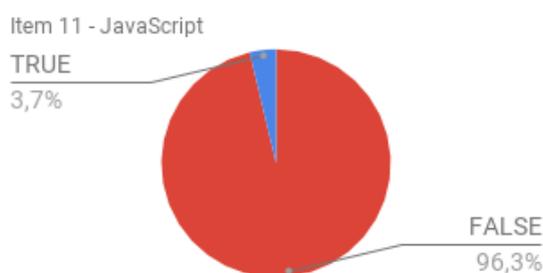
**Figura 22 – Análise de acesso sem WWW**

Fonte: Própria (2017).

Cinco universidades não estão de acordo, sendo elas: UFAL, UNIFAP, UFG, UFMS e UFPE. É recomendado que sejam feitas alterações nos servidores da infraestrutura interna que hospedam esses sítios para resolução do problema.

#### 4.3.6 JavaScript

A avaliação do JavaScript constatou que apenas a Universidade Federal de Sergipe - UFS, atende ao recomendado. O percentual de desacordo é de 96,3%.

**Figura 23 – Análise do JavaScript**

Fonte: Própria (2017).

Alguns sites apresentaram erros no código, sendo a causa principal a ausência do caractere “;” em algum trecho do código. Porém, o que definiu o alto percentual de desacordo com a recomendação é a utilização do código JavaScript dentro do documento HTML, onde a cartilha preconiza que a utilização desse tipo de script seja sempre em um arquivo externo. Um dos motivos é a utilização de código JavaScript, para configuração de plugins, e ajustes em elementos da página utilizando a biblioteca *Jquery*<sup>7</sup>.

<sup>7</sup> Disponível em: <<https://jquery.com/>>. Acesso em dez. 2017.

#### 4.3.7 Declaração de fontes

A análise constatou que somente 3 sites estão em acordo com o recomendado, sendo assim, o restante equivalente a 88,9% não obedece as definições.

**Figura 24 – Análise das declarações de fontes**



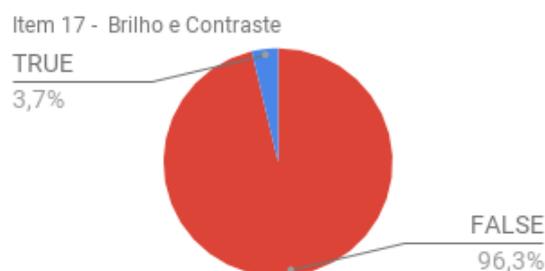
Fonte: Própria (2017).

A utilização do pixel como unidade de medida para qualquer propriedade da regra CSS ainda é muito presente. Devido sua fácil utilização, e por não ser variável é considerada como a medida mágica das CSS, sendo frequentemente uma boa medida para se usar especialmente se o estilo requer alinhamento do texto com imagens (WORLD WIDE WEB CONSORTIUM, 2010). Dessa forma é possível entender o porquê do grande percentual de utilização do pixel como unidade de medida de fontes.

#### 4.3.8 Contraste

A análise do contraste que consiste em verificar a diferença entre duas cores e a diferença do brilho dessas cores, mostrou que apenas uma universidade, A UFPB, está atendendo a recomendação, sendo 96,3% o índice de descordo.

**Figura 25 – Análise do Contraste**



Fonte: Própria (2017).

Uma possível explicação para esse alto nível de desacordo é a declaração de várias cores para fontes. O fato do script comparar para cada cor de fundo declarada

todas as cores de fontes presentes, também aumenta a possibilidade de ocorrer um valor de contraste não recomendado.

#### 4.3.9 Resultado HTML, CSS, Classes e IDs e Performance

Para todos esses quatro itens não houve nenhum sítio avaliado que estivesse completamente de acordo com as regras de validação.

Erros de marcação HTML recorrentes ocorreram nas declarações de atributos, como a falta da utilização do “*alt*” dentro do elemento de imagem, `<img>`, e ids duplicados. Quanto ao código CSS, valores não reconhecidos, valores não permitidos, atributos inexistentes e estilização *inline* foram maioria no não atendimento as recomendações. Na validação de classes e ids todos os sítios possuem declarações desnecessárias desses atributos. Por fim, no item de performance os valores obtidos foram bem acima do recomendado. O sítio com o menor valor avaliado, foi o da Universidade Federal do Espírito Santo — UFES, com 1820Kb, porém muito alto em comparação ao valor recomendado.

#### 4.3.10 Considerações

Após o desenvolvimento da ferramenta e da análise de sítios, alguns pontos foram percebidos quanto ao atendimento das recomendações e também das especificações contidas na cartilha. Desse modo, esta pesquisa visa sugerir, também, algumas modificações para melhoria da cartilha de codificação.

A última versão da cartilha codificação data de julho de 2010. Há referências a padrões que não são mais recomendados e sugere-se que sejam alterados, como o Doctype, onde a atual versão recomendada não é citada.

Há itens pouco desenvolvidos, onde a falta de informação ocasionou o não atendimento a dois itens da lista mencionada na Seção 3.2.13. Dessa forma, recomenda-se que esses itens sejam mais trabalhados.

Hoje em dia, com o avanço da tecnologia CSS, o conceito de sites responsivos é indispensável para o atendimento do item 20 da lista de verificação que questiona se a página é bem visualizada em dispositivos portáteis.

Segundo Rogatnev (2015, p. 2) a abordagem responsiva no desenvolvimento de sites poupa tempo e dinheiro, pois ao invés de se criar estilos para cada dispositivo ou navegador é criado apenas um com manifestações diferentes do mesmo design. Dessa forma os sites não devem ser criados apenas para serem flexíveis, mas também adaptáveis a qualquer dispositivo. Devido a isso, uma abordagem explícita do conceito de responsividade para criação de sites é objeto indispensável para ser abordado na

cartilha.

O último ponto que foi considerado passível de alteração foi percebido após a verificação de performance das páginas. A cartilha recomenda páginas com peso de no máximo 70kb. Porém, nenhum site sequer chegou perto deste número.

O governo federal em 2013, inaugurou o projeto de Identidade Digital de Governo que visa padronizar todos os sítio web ligados a ele (OLIVEIRA; ELER, 2015). O projeto foi desenvolvido por meio de um sistema gerenciador de conteúdo e está disponibilizado no portal da Secretaria Especial de Comunicação Social - SECOM<sup>8</sup>, vinculada a Presidência da República.

Nesse contexto, durante a análise dos sítios, percebeu-se que as universidades UFAC, UFMA, UFPA, UFPB, UFPI, UFRR, UFSC e UFT estão utilizando o template padrão. Então, foi feita uma análise ao site modelo do Portal Padrão<sup>9</sup>, e o resultado obtido foi de 12718Kb, valor bem acima do recomendado. Devido ao fato do projeto de Identidade Digital do Governo abranger toda a esfera federal, há a possibilidade de que um dia, todos os sites governamentais utilizem o portal padrão, o que inviabilizará o atendimento deste item da cartilha conforme avaliação feita ao modelo. Dessa forma sugere-se que o valor do tamanho das páginas, somado os recursos utilizados, seja analisado, de acordo com o cenário atual e contemple um valor que seja possível de ser alcançado.

<sup>8</sup> Disponível em: <<http://www.secom.gov.br/atuacao/comunicacao-digital/identidade-digital-1/documentacao-por-cms>>. Acesso em dez. 2017.

<sup>9</sup> Disponível em: <<http://www.portalpadrao.gov.br/>>. Acesso em dez. 2017.

## 5 CONCLUSÃO

Este capítulo apresenta os trabalhos relacionados, considerações finais e trabalhos futuros.

### 5.1 Trabalhos relacionados

Vilella (2003), quando a estruturação de portais governamentais era bastante recente, realizou uma avaliação de portais estaduais brasileiros visando a prestação de serviços públicos na web focando em três dimensões, conteúdo, usabilidade e funcionalidade.

Carvalho (2013), em sua pesquisa, realizou a avaliação de acessibilidade, codificação e acessibilidade de portais de universidades públicas brasileiras. Utilizou os validadores de HTML e CSS, do W3C para checagem de código, o ASES para acessibilidade e a lista de checagem de usabilidade da *User Effect*<sup>1</sup>. Esta pesquisa constatou que nenhuma instituição se adequava 100% as recomendações das 3 dimensões avaliadas. Um ponto de destaque é a avaliação do tempo de carregamento das páginas, presente na avaliação de usabilidade, onde o tamanho aceitável do sítio para um carregamento rápido é de 100Kb, em que todas as universidades reprovaram.

Santos et al. (2013) realizaram um estudo para avaliação dos sites corporativos dos Tribunais de Contas brasileiros com o objetivo de constatar em que estágio de evolução encontra-se cada um deles. Para isso, utilizaram as cartilhas de codificação, usabilidade, guia de administração e redação web do e-PWG, como pressupostos de boas práticas. Nesta avaliação foi aplicado um questionário que incorporou os principais construtos e diretrizes de avaliação traduzidos das cartilhas referentes aos padrões web.

Oliveira e Eler (2015), em sua pesquisa apresenta o cenário atual da acessibilidade em sítios web ligados ao Governo federal. Ao todo, 39 sítios referentes aos ministérios do governo federal pela ferramenta DaSilva. Foram realizadas validações de código HTML e CSS e também avaliação de acessibilidade aos padrões WCAG e e-MAG. Como resultado foram apresentados números preocupantes com relação à acessibilidade digital, e a constatação de que sítios governamentais importantes não estão acessíveis a todos os cidadãos.

Com base nos trabalhos acima citados, é constatado que nenhuma análise feita por estas pesquisas contempla toda a cartilha de codificação. O Foco principal dessas análises é a verificação de usabilidade de sites governamentais, onde, por vezes, são utilizadas vários serviços de validações. O diferencial da solução proposta nesta pes-

<sup>1</sup> Empresa de consultoria especializada em usabilidade.

quisa é contemplar os itens de verificação da cartilha de codificação, condensando-os em uma única ferramenta, facilitando a análise e auxiliando o desenvolvedor a encontrar pontos de inconformidade antes do portal ser disponibilizado para a população.

## 5.2 Considerações finais

O governo tem a responsabilidade de ofertar serviços de excelência da melhor maneira possível aos cidadãos. Com a informatização de processos e serviços, é necessário que os meios para cumprir com esta competência, estejam adequados para atender toda a população. Algumas vezes, esses ambientes são disponibilizados sem seguir recomendações que foram feitas com o intuito de melhorar o serviço prestado.

A pesquisa foi desenvolvida em duas etapas, onde buscou, propor uma solução automatizada para responder perguntas da lista de verificação da cartilha de codificação do e-PWG, possibilitando fazer a checagem de conformidade às recomendações em um único software. E também, fazer uma análise em 27 sítios de universidades brasileiras, para verificar o nível de atendimento a essas recomendações.

Através da análise dos sítios das universidades federais, apresentada no Capítulo 4 pode-se identificar que mesmo após 7 anos da publicação da cartilha de codificação, não ocorreu nenhum caso que atenda a todos os itens checados no universo analisado. É importante lembrar que houve itens que necessitam de validação manual, sendo que possivelmente, uma análise completa de todos os itens da lista poderia aumentar o percentual de não adequação.

Dentre as contribuições, estão presentes algumas sugestões de atualização da cartilha, visto o avanço de algumas tecnologias e o surgimento de conceitos e técnicas que surgiram nos últimos anos e contribuem para o atendimento das recomendações, logo, melhorando a disponibilização de serviços ao usuário.

Dessa forma, este trabalho resulta em uma solução útil aos profissionais envolvidos no processo de desenvolvimento de software das instituições ligadas ao governo federal brasileiro.

## 5.3 Trabalhos futuros

Como proposta de trabalhos futuros desta pesquisa, sugere-se a melhoria da ferramenta apresentada, nos seguintes pontos:

- Disponibilização de *interface web*, facilitando assim o acesso à validação;
- Melhorar a apresentação das mensagens de retorno das validações;

- Utilizar a estratégia de linhas de produtos de *softwares*, possibilitando ao usuário customizar a ferramenta com as opções disponíveis. E também, oportunizando a implementação de outras recomendações relacionadas a este tema.

E quanto aos sítios da administração pública, pequisas a respeito das tecnologias e das formas utilizadas nos itens da lista de verificação, com o objetivo de possibilitar o aumento do percentual de adequabilidade dos sites.

## Referências

- BACH, C. F. et al. Diretrizes de acessibilidade: uma abordagem comparativa entre WCAG e e-MAG. In: SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO, V., 2009. [S.l.], 2009. p. 73 – 74.
- BERNERS-LEE, T. World Wide Web. *Computers in Physics*, v. 8, n. 3, p. 298 – 299, 1994.
- BRASIL. Decreto não numerado de 3 de Abril de 2000. Institui Grupo de Trabalho Interministerial para examinar e propor políticas, diretrizes e normas relacionadas com as novas formas eletrônicas de interação. Brasília, abril 2000.
- BRASIL. Ases. 201–. Disponível em: <<https://softwarepublico.gov.br/social/ases>>. Acesso em: Dezembro de 2017.
- BRASIL. *Padrões Web em Governo Eletrônico e-PWG - Cartilha de Codificação*. 2010. Disponível em: <<http://epwg.governoeletronico.gov.br/cartilha-codificacao>>. Acesso em: Janeiro de 2017.
- BRASIL. *Padrões Web em Governo Eletrônico e-PWG - Cartilha de Usabilidade*. 2010. Disponível em: <<http://epwg.governoeletronico.gov.br/cartilha-usabilidade%23s2>>. Acesso em: Dezembro de 2017.
- BRASIL. *Modelo de Acessibilidade em Governo Eletrônico v.3.0*. 2011. Disponível em: <<https://www.governoeletronico.gov.br/documentos-e-arquivos/e-MAG%20V3.pdf>>. Acesso em: Janeiro de 2017.
- BRASIL. *eMAG - Modelo de Acessibilidade em Governo Eletrônico*. 2014. Disponível em: <<http://emag.governoeletronico.gov.br/>>. Acesso em: Dezembro de 2017.
- BRASIL. *eMAG - Modelo de Acessibilidade em Governo Eletrônico. Apresentação / Design*. 2014. Disponível em: <<http://emag.governoeletronico.gov.br/%23s3.4>>. Acesso em: Dezembro de 2017.
- BRASIL. *Histórico do Programa de Governo Eletrônico Brasileiro*. 2015. Disponível em: <<https://www.governoeletronico.gov.br/sobre-o-programa/historico>>. Acesso em: Janeiro de 2017.
- BRASIL. *Histórico do Programa de Governo Eletrônico Brasileiro*. 2015. Disponível em: <<https://www.governoeletronico.gov.br/sobre-o-programa/historico>>. Acesso em: Dezembro de 2017.
- BRASIL. *ePWG - Padrões Web em Governo Eletrônico*. 2016. Disponível em: <<https://www.governoeletronico.gov.br/eixos-de-atuacao/governo/epwg-padroes-web-em-governo-eletronico>>. Acesso em: Dezembro de 2017.
- BUCKLEY, S. *Attribute Grammars: An Executable Specification for CSS Layout*. 2015. Dissertação (Mestrado) — Macquarie University, Sydney - Australia.
- CARVALHO, C. V. de. *Programa de Governo Eletrônico Brasileiro: Avaliação da Acessibilidade, Codificação e Usabilidade dos Portais das Universidades Federais*. 2013. 44 p. Dissertação (Mestrado Profissional em Gestão de Políticas Públicas e Segurança Social) — UNIVERSIDADE FEDERAL DO RECÔNCAVO DA BAHIA.

- CARVALHO, R. N. M. de. *Usabilidade, acessibilidade e qualidade da web da administração pública portuguesa*. 2015. 140 p. Dissertação (Mestrado em Engenharia Informática) — Instituto Universitário de Lisboa. Disponível em: <<http://hdl.handle.net/10071/10911>>. Acesso em: Dezembro de 2017.
- CONFORTO, D.; SANTAROSA, L. M. C. Acessibilidade à Web: Internet para todos. *Informática na educação: teoria & prática*, Porto Alegre, v. 5, n. 2, p. 87 – 102, Novembro 2002.
- DINIZ, E. H. O governo eletrônico no Brasil: perspectiva histórica a partir de um modelo estruturado de análise. *Revista de Administração Pública*, v. 43, n. 1, p. 23 – 48, 2009.
- DINIZ, V.; GUIMARÃES, C. B. dos S. Os desafios para um país digitalmente aberto. In: VI CONGRESSO CONSAD DE GESTÃO PÚBLICA, 2013, Brasília. Brasília, 2013.
- EIS, D.; FERREIRA, E. HTML - A alma do Client-Side. In: \_\_\_\_\_. *HTML5 e CSS3 com farinha e pimenta*. São Paulo: Tableless.com.br, 2012. cap. 3, p. 21 –. ISBN 9781105096358.
- FLANAGAN, D. *JavaScript: the definitive guide*. 5. ed. [S.l.]: O'Reilly Media, Inc., 2006.
- FRAGOSO, S. Padrões de conectividade e links na web: uma proposta teórico-metodológica para avançar a Webometria e a Análise de Hiperlinks. In: XXXII CONGRESSO BRASILEIRO DE CIÊNCIAS DA COMUNICAÇÃO, 2009, Curitiba. Curitiba, 2009. Acesso em: Dezembro de 2017.
- GIL, A. C. *Metodologia do ensino superior*. 3. ed. São Paulo: Atlas, 2006.
- INTERNET ASSIGNED NUMBERS AUTHORITY. *Hypertext Transfer Protocol (HTTP) Status Code Registry*. 2017. Disponível em: <<https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml>>. Acesso em: Novembro de 2017.
- INTERNET ENGINEERING TASK FORCE. *Policy on Character Sets and Languages*. 1998. Disponível em: <<https://www.ietf.org/rfc/rfc2277.txt>>. Acesso em: Novembro de 2017.
- INTERNET ENGINEERING TASK FORCE. *RFC 2616. Hypertext Transfer Protocol – HTTP/1.1*. 1999. Disponível em: <<https://www.ietf.org/rfc/rfc2616.txt>>. Acesso em: Dezembro de 2017.
- KELLY, B. Web Watch: An accessibility analysis of UK university entry points. *Ariadne*, n. 33, 2002.
- KOLB, A.; RUCKENBAUER, H.; ESTERBAUER, R. *Ciberética : responsabilidade em um mundo interligado pela rede digital*. 1. ed. São Paulo: Loyola, 2001. Edição Brasileira.
- LAWSON, B.; SHARP, R. *Introducing html5*. 2. ed. [S.l.]: New Riders, 2011.
- LEOPARDI, M. T. *Metodologia da pesquisa na saúde*. Santa Maria: Pallotti, 2002.
- OLIVEIRA, A. D. A.; ELER, M. M. Acessibilidade em governo eletrônico: um estudo sobre a aplicação de padrões Web em sítios gov. br. In: XI SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO, 2015. [S.l.], 2015. p. 683 – 690.

PARK, E.; LIM, Y.; LIM, H. A Study of Web Accessibility of Websites Built in HTML5 - Focusing on the Top 100 Most Visited Websites. *International Journal of Multimedia and Ubiquitous Engineering*, v. 9, n. 4, p. 247 – 256, 2014.

PINHO, D. M. de. *ECMAScript 6: entre de cabeça no futuro do JavaScript*. 1. ed. [S.l.]: Casa do Código, 2017.

ROCHA, Z.; PIMENTEL, M. *TÉCNICAS PARA MELHORIA DE PERFORMANCE EM APLICAÇÕES WEB NO LADO DO CLIENTE*. 2013. 41 p. Monografia (SISTEMAS DE INFORMAÇÃO) — UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO.

ROGATNEV, N. *Responsive Web Design*. 2015. 67 p. Monografia (Information Technology) — MAMK - University of Applied Sciences.

ROVER, A. J. et al. Avaliação de portais e sítios governamentais no Brasil. Aires José Rover; Fernando Galindo. (Org.). *O Governo Eletrônico e suas múltiplas facetas*. Zaragoza: Lefis Series, v. 10, p. 11 – 38, 2010.

SANTOS, P. M. et al. Ranking dos tribunais de contas brasileiros: uma avaliação a partir dos padrões web em governo eletrônico. *Rev. Adm. Pública [online]*, v. 47, n. 3, p. 721 – 744, 2013. ISSN 0034-7612.

SANTOS, R. de Jesus Leone dos. GOVERNO ELETRÔNICO: O QUE SE DEVE E O QUE NÃO SE DEVE FAZER. In: XVI CONCURSO DE ENSAYOS Y MONOGRAFÍAS DEL CLAD SOBRE REFORMA DEL ESTADO Y MODERNIZACIÓN DE LA ADMINISTRACIÓN PÚBLICA “GOBIERNO ELECTRÓNICO”, 2002, Caracas. Caracas, 2002.

SCHEIDT, F. A. *Fundamentos de CSS: criando design para sistemas web*. [S.l.]: Outbox Livros Digitais, 2015.

SILVA, E. L. da; MENEZES, E. M. *Metodologia da pesquisa e elaboração de dissertação*. 4. ed. Florianópolis: UFSC, 2005.

SILVA, M. S. Regra CSS. In: \_\_\_\_\_. *Fundamentos de HTML5 e CSS3*. 1. ed. São Paulo: Novatec, 2015. cap. 4, p. 70 – 70. ISBN 978-85-7522-438-0.

SOLARO, J. A.; YU, J. *Friendly URLs*. 2007. U.S. Patent n. 7,293,012, 6 nov. 2007. Acesso em: Dezembro de 2017.

STOTHARD, P. The sequence manipulation suite: JavaScript programs for analyzing and formatting protein and DNA sequences. *BioTechniques*, v. 28, n. 6, 2000.

URDIALES, C. et al. eGovernment and Identity Management: a Signature Coding Method for PIN generation. In: EULAT'04 WORKSHOP ON EGOVERNMENT AND EDEMOCRACY, 2004, Santiago de Chile. Santiago de Chile., 2004.

VILELLA, R. M. *Conteúdo, usabilidade e funcionalidade: três dimensões para avaliação de portais estaduais de Governo Eletrônico na Web*. 2003. 263 p. Dissertação (Ciência da Informação) — Universidade Federal de Minas Gerais.

WASLAWICK, R. S. *Metodologia de Pesquisa para ciência da computação*. 2. ed. Rio de Janeiro: Elsevier, 2014.

- WORLD WIDE WEB CONSORTIUM. *anchors with id*. 1999. Disponível em: <<https://www.w3.org/TR/html4/struct/links.html%23anchors-with-id>>. Acesso em: Dezembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *Links*. 1999. Disponível em: <<https://www.w3.org/TR/html4/struct/links.html%23adef-name-A>>. Acesso em: Dezembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *W3C Glossary and Dictionary*. 2003. Disponível em: <<https://www.w3.org/2003/glossary/>>. Acesso em: Dezembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *HTML Design Principles*. 2007. Disponível em: <<https://www.w3.org/TR/html-design-principles/>>. Acesso em: Dezembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *Serviço de validação CSS*. 2007. Disponível em: <<https://jigsaw.w3.org/css-validator/about.html>>. Acesso em: Dezembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *Folhas de Estilo Web Dicas & truques CSS*. 2010. Disponível em: <[https://www.w3.org/Style/Examples/007/units.pt\\_BR.html](https://www.w3.org/Style/Examples/007/units.pt_BR.html)>. Acesso em: Dezembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *Sobre o W3C*. 2011. Disponível em: <<http://www.w3c.br/Sobre/>>. Acesso em: Dezembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *Media Queries*. 2012. Disponível em: <<https://www.w3.org/TR/css3-mediaqueries/>>. Acesso em: Dezembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *CSS Fonts Module Level 3*. 2013. Disponível em: <<https://www.w3.org/TR/2013/CR-css-fonts-3-20131003/>>. Acesso em: Dezembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *The history of the Web*. 2014. Disponível em: <[https://www.w3.org/wiki/The\\_history\\_of\\_the\\_Web](https://www.w3.org/wiki/The_history_of_the_Web)>. Acesso em: Dezembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *HTML & CSS - What is CSS?* 2016. Disponível em: <<https://www.w3.org/standards/webdesign/htmlcss>>. Acesso em: Dezembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *Recommended Doctype Declarations to use in your Web Document*. atualizado em 2016, 2016. Disponível em: <<https://www.w3.org/QA/2002/04/valid-dtd-list.html>>. Acesso em: Novembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *Encoding*. 2017. Disponível em: <<https://www.w3.org/TR/encoding>>. Acesso em: Novembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *HTML5 - A quick introduction to HTML*. 2017. Disponível em: <<https://www.w3.org/TR/html51/introduction.html%23a-quick-introduction-to-html>>. Acesso em: Dezembro de 2017.
- WORLD WIDE WEB CONSORTIUM. *The DOCTYPE*. 2017. Disponível em: <<https://www.w3.org/TR/html52/syntax.html%23doctype>>. Acesso em: Dezembro de 2017.

---

XAVIER, A. C. dos S. Navegação, links e referência no hipertexto. In: \_\_\_\_\_. *A era do hipertexto: linguagem e tecnologia*. 2. ed. Recife: Pipa Comunicação, 2013. cap. 5, p. 268 – 311. ISBN 978-85-66530-26-1.

## **APÊNDICES**

## APÊNDICE A – Sítios avaliados

	Universidade	Sigla	Endereço Avaliados
AC	Universidade Federal do Acre	UFAC	<a href="http://ufac.br/">http://ufac.br/</a>
AL	Universidade Federal de Alagoas	UFAL	<a href="http://ufal.edu.br/">http://ufal.edu.br/</a>
AP	Universidade Federal do Amapá	UNIFAP	<a href="http://unifap.br/public/">http://unifap.br/public/</a>
AM	Universidade Federal do Amazonas	UFAM	<a href="http://ufam.edu.br/">http://ufam.edu.br/</a>
BA	Universidade Federal da Bahia	UFBA	<a href="https://ufba.br/">https://ufba.br/</a>
CE	Universidade Federal do Ceará	UFC	<a href="http://ufc.br/">http://ufc.br/</a>
DF	Universidade de Brasília	UNB	<a href="http://unb.br/">http://unb.br/</a>
ES	Universidade Federal do Espírito Santo	UFES	<a href="http://ufes.br/">http://ufes.br/</a>
GO	Universidade Federal de Goiás	UFG	<a href="https://ufg.br/">https://ufg.br/</a>
MA	Universidade Federal do Maranhão	UFMA	<a href="http://portais.ufma.br/PortalUfma/index.jsf">http://portais.ufma.br/PortalUfma/index.jsf</a>
MT	Universidade Federal do Mato Grosso	UFMT	<a href="http://ufmt.br/ufmt/site/">http://ufmt.br/ufmt/site/</a>
MS	Universidade Federal do Mato Grosso do SUL	UFMS	<a href="https://ufms.br/">https://ufms.br/</a>
MG	Universidade Federal de Minas Gerais	UFMG	<a href="https://ufmg.br/">https://ufmg.br/</a>

	Universidade	Sigla	Endereço Avaliados
PA	Universidade Federal do Pará	UFPA	<a href="https://portal.ufpa.br/">https://portal.ufpa.br/</a>
PB	Universidade Federal da Paraíba	UFPB	<a href="http://ufpb.br/">http://ufpb.br/</a>
PR	Universidade Federal de Pernambuco	UFPE	<a href="http://ufpe.br">http://ufpe.br</a>
PE	Universidade Federal do Paraná	UFPR	<a href="http://ufpr.br/portalufpr/">http://ufpr.br/portalufpr/</a>
PI	Universidade Federal do Piauí	UFPI	<a href="http://ufpi.br/">http://ufpi.br/</a>
RJ	Universidade Federal do Rio de Janeiro	UFRJ	<a href="https://ufrj.br/">https://ufrj.br/</a>
RN	Universidade Federal do Rio Grande do Norte	UFRN	<a href="http://ufrn.br/">http://ufrn.br/</a>
RS	Universidade Federal do Rio Grande do Sul	UFRGS	<a href="http://ufrgs.br/ufrgs/inicial">http://ufrgs.br/ufrgs/inicial</a>
RO	Universidade Federal de Rondônia	UNIR	<a href="https://unir.br">https://unir.br</a>
RR	Universidade Federal de Roraima	UFRR	<a href="http://ufrr.br/">http://ufrr.br/</a>
SC	Universidade Federal de Santa Catarina	UFSC	<a href="http://ufsc.br/">http://ufsc.br/</a>
SP	Universidade Federal de São Paulo	UNIFESP	<a href="http://unifesp.br/">http://unifesp.br/</a>
SE	Universidade Federal de Sergipe	UFS	<a href="http://ufs.br/">http://ufs.br/</a>
TO	Universidade Federal do Tocantins	UFT	<a href="http://ww2.uft.edu.br/">http://ww2.uft.edu.br/</a>

## **APÊNDICE B – Repositório**

O acesso aos fontes do programa encontra-se em: <https://bitbucket.org/brucealbuquerque/diretrizesmaven>.

## **ANEXOS**

## ANEXO A – Valores de resposta do protocolo HTTP

Tabela 1 – Hypertext Transfer Protocol Status Code Registry

Value	Description
100	Continue
101	Switching Protocols
102	Processing
103	Early Hints
104 – 199	Unassigned
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
207	Multi-Status
208	Already Reported
209 – 225	Unassigned
226	IM Used
227 – 299	Unassigned
300	Multiple Choices
301	Moved Permanently

---

---

Value	Description
302	Found
303	See Other
304	Not Modified
305	Use Proxy
306	(Unused)
307	Temporary Redirect
308	Permanent Redirect
309 – 399	Unassigned
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Payload Too Large
414	URI Too Long

---

---

---

---

Value	Description
415	Unsupported Media Type
416	Range Not Satisfiable
417	Expectation Failed
418 – 420	Unassigned
421	Misdirected Request
422	Unprocessable Entity
423	Locked
424	Failed Dependency
425	Unassigned
426	Upgrade Required
427	Unassigned
428	Precondition Required
429	Too Many Requests
430	Unassigned
431	Request Header Fields Too Large
432 – 450	Unassigned
451	Unavailable For Legal Reasons
452 – 499	Unassigned
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout

---

---

---

---

Value	Description
505	HTTP Version Not Supported
506	Variant Also Negotiates
507	Insufficient Storage
508	Loop Detected
509	Unassigned
510	Not Extended
511	Network Authentication Required
512 – 599	Unassigned

---

---

Fonte: Internet Assigned Numbers Authority (IANA)

## ANEXO B – Lista de verificação da cartilha de codificação

**Tabela 2 – Perguntas da lista de verificação**

1	A página usa Doctype correto?
2	A página usa o character set de codificação de caracteres correto?
3	A página usa codificação válida?
4	A(s) folha(s) de estilo CSS usada pela página é (são) válida(s)?
5	Há declarações de classes ou ID's desnecessárias?
6	O código é bem estruturado?
7	Há links quebrados?
8	Qual é a performance velocidade de carregamento/tamanho da página?
9	A página utiliza URL's amigáveis?
10	As URLs funcionam sem o "www"?
11	Há erros de javascript?
12	A página funciona com o javascript desabilitado?
13	O conteúdo é acessível com as folhas de estilo desabilitadas?
14	O sítio usa CSS para todos os aspectos da apresentação?
15	As imagens de apresentação estão incluídas nas CSS?
16	Há quebra do desenho quando o usuário aumenta o tamanho da fonte?
17	O contraste das cores da página é suficiente?
18	A página é consistente em navegadores texto?
19	O conteúdo é legível quando impresso? Não há desperdício de papel ou tinta?
20	A página é bem visualizada em dispositivos portáteis?
21	A página é bem visualizada em diferentes resoluções de navegador e tela?

Fonte: Padrões Web em Governo Eletrônico e-PWG - Cartilha de Codificação