

CENTRO PAULA SOUZA



FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Bacharel em Sistemas e Tecnologia da Informação

Samuel Henrique de Oliveira Guimarães

ACESSIBILIDADE DIGITAL:
Aplicações Web em JavaScript sob a perspectiva de usuários com limitações
visuais

Americana, SP

2013

FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Bacharel em Sistemas e Tecnologia da Informação

Samuel Henrique de Oliveira Guimarães

ACESSIBILIDADE DIGITAL:
**Aplicações Web em JavaScript sob a perspectiva de usuários com limitações
visuais**

Trabalho monográfico, desenvolvido em cumprimento à exigência curricular do Curso Superior de Bacharel em Análise de Sistemas e Tecnologia da Informação da Fatec Americana, sob orientação do Prof.^(o) Dr. Renato Kraide Soffner
Área de concentração: Interação Humano - Computador

Americana, S. P.

2013

**FICHA CATALOGRÁFICA elaborada pela
BIBLIOTECA – FATEC Americana – CEETPS**

G981a	<p>Guimarães, Samuel Henrique de Oliveira</p> <p>Acessibilidade digital: aplicações WEB em JavaScript sob a perspectiva de usuários com limitações visuais. / Samuel Henrique de Oliveira Guimarães. – Americana: 2013. 62f.</p> <p>Monografia (Graduação em Análise de Sistemas e Tecnologia da Informação). - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza.</p> <p>Orientador: Prof. Dr. Renato Kraide Soffner</p> <p>1. Interação homem-computador 2. WEB - rede de computadores I. Soffner, Renato Kraide II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana.</p> <p>CDU: 681.6 681.519</p>
-------	---

Bibliotecária responsável pela FC: Ana Valquiria Niaradi – CRB-8 região 6203

FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Bacharel em Sistemas e Tecnologia da Informação

Samuel Henrique de Oliveira Guimarães

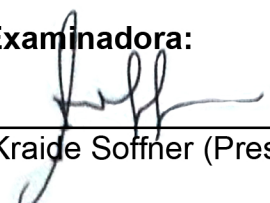
ACESSIBILIDADE DIGITAL:

**Aplicações Web em JavaScript sob a perspectiva de usuários com limitações
visuais**


Trabalho de conclusão de curso apresentado
à Faculdade de Tecnologia de Americana
como parte dos requisitos para obtenção do
título de Bacharel em Análise de Sistemas e
Tecnologia da Informação
Área de concentração: Interação Humano -
Computador

Americana, 05 de dezembro de 2013.

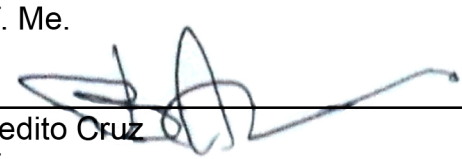
Banca Examinadora:



Renato Kraide Soffner (Presidente)
Prof. Dr.



Nivaldo Tadeu Marcusso
Prof. Me.



Benedito Cruz
Prof.

AGRADECIMENTOS

Agradeço a todos os meus amigos que me apoiaram incondicionalmente durante esta jornada e principalmente neste semestre que foi tão difícil.

Um agradecimento especial aos meus colegas e amigos de curso que contribuíram ao longo destes anos em todos os trabalhos e estudos que tivemos.

A meus irmãos por terem me apoiado e compreendido em um momento tão delicado com todas as adversidades que tivemos de passar.

Às demais pessoas, cuja colaboração direta ou indireta permitiu a realização deste trabalho.

RESUMO

Nas últimas décadas a *Internet* se consolidou como um dos mais importantes meios de comunicação na sociedade globalizada. O advento da rede mundial de computadores ajudou a promover a globalização, uma vez que diminuiu as barreiras físicas e se tornou um dos principais instrumentos facilitadores da comunicação. Entretanto, apesar da informação estar mais próxima e quase instantânea, quando se trata de pessoas portadoras de necessidades especiais, principalmente os deficientes visuais, o acesso a informação ainda é muitas vezes dificultoso. Devido este obstáculo, o estudo das práticas de acessibilidade e usabilidade na área de Interação Humano-Computador se tornou um tópico muito discutido e diversas tecnologias e ferramentas foram elaboradas e estudadas de forma a promover a acessibilidade digital e inclusão social através da *web*. Neste trabalho serão apresentadas as diretrizes, ferramentas, técnicas e tecnologias utilizadas para viabilizar a acessibilidade digital e aprimorar as interfaces de aplicações web modernas baseados em princípios de usabilidade e dos padrões *web*.

Palavras-chave: acessibilidade, usabilidade; padrões *web*, aplicações *web*

ABSTRACT

In the last decades, the Internet has established itself as one of the most important media in the globalized society. The advent of the World Wide Web has helped to promote globalization, since it decreased physical barriers and became a major communication enabler. However, despite the information being closer and almost instantaneous, when it comes to people with disabilities, especially the visually impaired, access to information is often difficult. Due this barrier, the study of accessibility and usability in Human-Computer Interaction has become a widely discussed topic and several technologies and tools were developed and studied in order to promote accessibility and digital inclusion through the web. In this paper, we will present the guidelines, tools, techniques and technologies used to provide accessibility and enhance interfaces of modern web applications based on principles of usability and web standards.

Keywords: *accessibility, usability; web standards, web applications*

SUMÁRIO

1. INTRODUÇÃO.....	8
2. ACESSIBILIDADE WEB	14
2.1. Contextualização da Deficiência Visual.....	14
2.2. Diretrizes e Contexto Histórico da Acessibilidade Web	19
2.2.1. Breve histórico do W3C.....	19
2.2.2. Estrutura Tecnológica do W3C	21
2.2.3. A Web Accessibility Initiative	21
2.2.4. Diretrizes de Conteúdo - WCAG	23
2.2.5. Tecnologias Assistivas.....	26
2.2.6. Tecnologias Assistivas Desenvolvidas no Brasil.....	28
3. APLICAÇÕES WEB E JAVASCRIPT	30
3.1. A Internet e Aplicações Client-Side.....	30
3.2. Contextualização das Tecnologias Envolvidas.....	32
3.2.1. HTML e Document Object Model	32
3.2.2. Motores JavaScript	37
3.2.3. RIAs em AJAX	38
3.2.4. Frameworks JavaScript	42
4. ARIA E RECOMENDAÇÕES PARA CONTEÚDOS DINÂMICOS.....	44

4.1. Accessible Rich Internet Applications	44
4.2. Landmark Roles	46
4.3. Live Regions e Aplicações em JavaScript.....	48
4.4. ARIA e Funcionamento com Leitores de Telas	50
5. CONSIDERAÇÕES FINAIS.....	51
REFERÊNCIAS.....	53
APÊNCIDE A – ESPECIFICAÇÕES WAI-ARIA SUPORTADAS PELO JAWS.....	60
ANEXO A – DIAGRAMA RDF DA ESPECIFICAÇÃO WAI-ARIA.....	61

1. INTRODUÇÃO

Acessibilidade significa não apenas permitir que pessoas portadoras de necessidades especiais (PNEs) participem de atividades que incluam o uso de produtos, serviços e informação, mas também a inclusão e extensão do uso destes por todas as parcelas presentes de uma determinada população (NBR 9050, 2004).

Com a revolução da rede mundial de computadores no século XX e a popularização da *Internet*, programas e informações puderam ser compartilhados de forma *online* e quase instantânea, quebrando paradigmas de distância e acesso a informação graças ao protocolo de comunicação HTTP (*Hypertext Transfer Protocol*). Este cenário estimulou e possibilitou que várias tarefas diárias importantes pudessem ser desempenhadas remotamente pela *web* através de programas que implementam sistemas de informação cuja camada de apresentação é escrita em HTML, de modo que a Interação Humano-Computador (IHC) seja assegurada pelos navegadores, tradicionalmente criados para a navegação na rede em sítios *web* estáticos.

[...] acessibilidade na *web* basicamente significa que pessoas com deficiências podem usar a *internet*. Mais especificamente, acessibilidade na *web* significa que pessoas com deficiências podem perceber, compreender, navegar e interagir com a *internet*. A acessibilidade na *web* abrange todas as inabilidades que afetam o acesso a ela, incluindo deficiências visuais, auditivas, físicas, fala, cognitivos e neurológicos. (TATCHER, J., 2006, p. 2)

De acordo com Tim Berners-Lee (1997), "O poder da *Web* está na sua universalidade. Acesso por todos, independentemente da deficiência é um aspecto essencial". Esta afirmação foi dada pelo diretor do W3C (*World Wide Web Consortium*) e inventor da *World Wide Web* durante a abertura do IPO (*International Program Office*) para o programa WAI (*Web Acessibility Initiative*), que visa desde sua fundação promover e alcançar funcionalidade total da *web* para pessoas com deficiências.

A WAI "desenvolve estratégias, diretrizes e recursos para ajudar a tornar a *Web* acessível a pessoas com deficiência" (WAI, 2011), enquanto o W3C desenvolve *standards* relacionados com a criação de conteúdos e tecnologias *web* (JACOBS, 2008).

Com a ajuda da informática e da *Internet* temos acesso a uma grande quantidade de serviços de utilidade pública, comerciais ou de entretenimento a qualquer instante. Também temos diversas tecnologias empregadas na *Internet* e diferentes formas de apresentar seu conteúdo, assim como diferentes sistemas operacionais e meios de acessar a própria *web*. Empresas de diferentes tamanhos e segmentos hoje adotam sistemas *online*, aplicações *web* em uma *Intranet* ou *Internet*, com a camada de apresentação comandada por intermédio de um navegador.

[...] Em relação às PNEs com limitação visual (pessoas com baixa-visão e cegos), a apropriação das tecnologias de informação e de comunicação (TICs) apresenta-se como uma forma de incluí-las digitalmente, comunicando-se e interagindo com os outros de forma que não se percebam as suas limitações. (ESTABEL; MORO; SANTAROSA, 2006, p. 94)

De acordo com Camargo Filho e Bica (2008, p. 1), “Pelos peculiaridades de sua deficiência severamente limitadora, os deficientes visuais [...] constituem um segmento digno de especial atenção por parte daqueles que lidam com acessibilidade digital”.

É evidente que os principais prejudicados quando falamos de acessibilidade digital sejam os indivíduos com limitações visuais, devido à importância dada a elementos gráficos e exigência de ferramentas de precisão para gestos ou navegação nas interfaces gráficas, como o *mouse* nos sistemas tradicionais e o toque dos dedos em aplicações voltadas telas sensíveis ao toque.

A falta de conhecimento de padrões de desenvolvimento relacionados à acessibilidade e usabilidade por parte dos desenvolvedores é uma das principais barreiras para criar uma *web* acessível. As principais diretrizes de acessibilidade são escritas pelos mesmos grupos que definem e mantêm as especificações das principais linguagens de programação da *web*, sendo algumas delas normas ISO/IEC. É preciso reconhecer o quanto tais diretrizes influenciam na qualidade da navegação dos PNEs.

Somente o desenvolvimento de aplicações *web* seguindo os padrões mantidos e definidos pelo W3C não é o suficiente para atender ao requisito da

acessibilidade digital. Desenvolvedores devem estar atentos às diretrizes de governos e outros órgãos e também dos fabricantes de softwares e tecnologias utilizados pelos PNEs a fim de garantir que suas aplicações *web* sejam de fato acessíveis.

A adoção dos padrões *web*, das diretrizes e recursos do WAI, dos estudos de usabilidade e entendimento das tecnologias envolvidas na acessibilidade digital garantem o funcionamento da “*web universal*” projetada por Tim Berners-Lee.

Com este trabalho serão apresentadas algumas das diretrizes e padrões desenvolvidos pelo W3C para a *web*, como os usuários PNEs acessam a *web* e quais suas dificuldades, quais os aspectos fundamentais para que aplicações *web* sejam consideradas *Rich Internet Applications* (RIA) e por último como torna-las acessíveis utilizando a especificação WAI-ARIA.

Esta pesquisa tem como objetivo principal expor as recomendações do W3C e do WAI em aplicações *web* e explanar como a acessibilidade é garantida neste tipo de aplicação. Será primeiramente apresentada uma contextualização a respeito das deficiências visuais. Em seguida, as diretrizes do WAI e as ferramentas utilizadas pelos usuários PNEs para navegar na *web*. No capítulo seguinte, o termo “aplicação *web*” será discutido, sendo expostas as teorias a respeito das características destas aplicações com um contexto histórico envolvendo as tecnologias que representam o tema. Por último serão apresentadas as recomendações do WAI para aplicações *web*, assim como a forma em que as ferramentas utilizadas por usuários PNEs se relacionam com estas especificações. Desta forma, serão elucidadas as recomendações para a adequação de aplicações *web* à usuários PNEs, contribuindo assim para que seja desenvolvida uma *Web* mais acessível.

Os conceitos desenvolvidos neste trabalho dizem respeito às tecnologias utilizadas para auxiliar a utilização de computadores por indivíduos portadores de necessidades especiais (ferramentas assistivas), às linguagens de programação (*HTML* e *JavaScript*), e as diretrizes e especificações do W3C, tendo em consideração suas características, seus escopos e suas limitações.

O presente trabalho tem ainda, com a finalidade de atingir o objetivo geral, os seguintes objetivos específicos listados a seguir:

- Pesquisa bibliográfica sobre acessibilidade, usabilidade, diretrizes e padrões de desenvolvimento de aplicações acessíveis;
- Identificar as ferramentas utilizadas por PNEs para fazer uso da *web*;
- Compreender do que se trata uma aplicação *web* e *quais recursos estão disponíveis para torná-las acessíveis*;

Esta pesquisa é relevante pois segundo Luisa (2006, p. 1) “a *Internet* e o computador alargaram os horizontes dos deficientes visuais, a quantidade de informações a que ele tem acesso através desses recursos é muito maior do que por meio de outros veículos”.

A acessibilidade digital pode ser vista como um fator importante na inclusão digital e social de um indivíduo pois as dificuldades em acesso à tecnologia podem prejudica-lo de tal forma que o impeçam de exercer sua plena cidadania:

[...] a exclusão digital aprofunda a exclusão socioeconômica. A inclusão digital deveria ser fruto de uma política pública com destinação orçamentária a fim de que ações promovam a inclusão e equiparação de oportunidades a todos os cidadãos. (ALMEIDA e DE PAULA, 2005, p. 59)

O estudo deste tema também é justificável pois no Brasil, a lei nº 7.853 de 24 de outubro de 1989, em seu Art. 1º estabelece normas gerais que asseguram o pleno exercício dos direitos individuais e sociais das pessoas portadoras de deficiências. No § 1º, são considerados os valores básicos de igualdade de tratamento e oportunidade, justiça social, do respeito à dignidade da pessoa humana, do bem-estar, e outros, indicados na Constituição.

A lei nº 8213 de 24 de julho de 1991, em seu Art. 93 estabelece as regras para preenchimento de cargos em empresas por pessoas portadoras de deficiências habilitadas na seguinte proporção:

- Até 200 empregados: 2%

- De 201 a 500 empregados: 3%
- De 501 a 1.000: 4%
- De 1001 em diante: 5%

Também fora firmado o decreto nº 5.296 de 2 de dezembro 2004, que em seu capítulo VI, dedica 14 artigos ao Acesso à Informação e à Comunicação, sobretudo das pessoas com deficiências visuais. Tal decreto torna obrigatória a acessibilidade digital nos portais e sítios *web* da administração pública e empresas prestadoras de serviços públicos na *Internet*, o que também inclui as aplicações *web*.

De acordo com os dados levantados no Censo 2000 (IBGE, 2002), 24 milhões de brasileiros apresentam algum tipo de deficiência, sendo destes 67% com alguma deficiência visual e 0,6% totalmente cegos. Também segundo a OMS (Organização Mundial de Saúde), cerca de 1% da população mundial apresenta algum grau de deficiência visual (OMS, 2013). Desta forma, o desenvolvimento desta pesquisa é importante a fim de delatar as principais dificuldades dos usuários PNEs com relação ao uso de aplicações *web* e firmar boas práticas para que estas aplicações entrem em conformidade com as diretrizes da W3C, WAI e também do Governo Federal, em razão principalmente dos esforços na inclusão destes indivíduos em igualdade de oportunidades e a exponente adoção de aplicações *web* nos ambientes corporativos, educacionais e públicos, pois, a acessibilidade digital desempenha um papel muito importante na inclusão social.

Com relação a metodologia da pesquisa, fora dividida em cinco etapas: (a) levantamento bibliográfico; (b) contextualização da acessibilidade; (c) delimitação do tema; (d) levantamento do contexto histórico e tecnológico; (e) especificações para aplicações acessíveis. Estas etapas são detalhadas a seguir:

- (a) Levantamento bibliográfico - Nesta etapa foram pesquisados livros, leis e normas para entender quais são os princípios de acessibilidade. Através desta pesquisa foi possível cruzar dados de diferentes organizações e ter um entendimento das leis e normas brasileiras que dizem respeito às pessoas com deficiência.

- (b) Contextualização da acessibilidade - Após a pesquisa inicial foi constatado que os usuários com deficiências visuais são os que mais precisam de soluções de acessibilidade na *web*. Foi necessário estudar quais os diferentes tipos de deficiências visuais e como os usuários PNEs utilizam a *web*.
- (c) Delimitação do tema - As diretrizes de acessibilidade propostas no passado dizem respeito ao conteúdo de sítios *web*, mas não a aplicações robustas e complexas como as que estão em uso desde a segunda metade da década de 2000. Nesta etapa o tema fora delimitado e o restante do trabalho se propõe a estudar apenas as aplicações *web* e as tecnologias que dizem respeito a elas.
- (d) Levantamento do contexto histórico e tecnológico - Nesta etapa fora estudado a história das tecnologias envolvidas nas aplicações *web*, como surgiram os termos e metodologias utilizados e como se relacionam com a acessibilidade e ferramentas utilizadas por PNEs.
- (e) Especificações para aplicações acessíveis - Na última etapa é apresentado o WAI-ARIA, conjunto de atributos que faz com que as ferramentas utilizadas por usuários PNEs interpretem componentes complexos e personalizados em aplicações *web*.

2. ACESSIBILIDADE WEB

2.1. Contextualização da Deficiência Visual

De todos os tipos de deficiência visual, a cegueira é a mais obstrutiva com relação a acessibilidade digital. De acordo com Conde (2012), a delimitação do grupamento dos deficientes visuais, cegos e portadores de visão subnormal, se dá por duas escalas oftalmológicas:

- Acuidade visual - Aquilo que se enxerga a determinada distância;
- Campo visual - Amplitude da área alcançada pela visão.

Conde (2012) ainda menciona que:

De um trabalho conjunto entre a *American Academy of Ophthalmology* e o Conselho Internacional de Oftalmologia, vieram extensas definições, conceitos e comentários a respeito, transcritos no Relatório Oficial do IV Congresso Brasileiro de Prevenção da Cegueira (vol-1, págs. 427/433, Belo Horizonte, 1980). Na oportunidade foi introduzido, ao lado de “cegueira”, o termo “visão subnormal” (*low vision*, em língua inglesa).

Amaurose, palavra de origem grega com o significado de “escurecer”, é o nome dado a perda completa da função visual, podendo ser unilateral ou bilateral. Não há percepção luminosa presente.

É considerado então cego ou de visão subnormal, aquele que apresenta desde ausência total de visão até alguma percepção luminosa que possa determinar formas a curtíssima distância. O termo deficiência visual não significa, então, total incapacidade para ver.

Queiroz (2012) distingue cegueira parcial da *amaurose*:

Falamos em “cegueira parcial” como aquela em que estão os indivíduos apenas capazes de CONTAR DEDOS a curta distância e os que só PERCEBEM VULTOS. O indivíduo é capaz de identificar também a direção de onde provém a luz. Mais próximos da cegueira total, mas ainda

considerados com cegueira parcial ou visão subnormal, estão os indivíduos que só têm PERCEPÇÃO e PROJEÇÃO LUMINOSAS. Nesse caso, há apenas a distinção entre claro e escuro.

Segundo Conde (2012) e Queiroz (2012), pedagogicamente é considerado cego aquele que mesmo com visão subnormal, necessita de instruções em Braille e o portador de visão subnormal, o indivíduo que lê tipos impressos ampliados ou com o auxílio de recursos ópticos.

Queiroz (2012) ainda explana a definição de Conde (2012) a respeito da definição de cegueira na medicina, da seguinte forma:

[...] uma pessoa é considerada cega se corresponde a um dos critérios seguintes: a visão corrigida do melhor dos seus olhos é de 20/200 ou menos, isto é, se ela pode ver a 20 pés (6 metros) o que uma pessoa de visão normal pode ver a 200 pés (60 metros), ou se o diâmetro mais largo do seu campo visual subentende um arco não maior de 20 graus, ainda que sua acuidade visual nesse estreito campo possa ser superior a 20/200. Esse campo visual restrito é muitas vezes chamado "visão em túnel" ou "em ponta de alfinete". Nesse contexto, caracteriza-se como indivíduo com visão subnormal aquele que possui acuidade visual de 6/60 e 18/60 (escala métrica) e/ou um campo visual entre 20 e 50°.

A NBR 9050 (ABNT, 2004), regulamentada pela Associação Brasileira de Normas Técnicas define as especificações para confecção de implantação de sinalizações táteis e sonoras para deficientes visuais.

Através do sistema braile, a informação pode ser passada através de textos, desde que o usuário PNE consiga fazer a leitura utilizando-se das mãos e dedos.

O sistema braile é muito eficiente no campo físico porém ineficaz nos espaços virtuais. Uma das principais evoluções da tecnologia nos últimos anos, as telas sensíveis ao toque ou *touchscreens* na língua inglesa, são um obstáculo para PNEs pois ainda não fornecem um *feedback* tátil em braile.

No decreto nº 5.296 (Brasil, 2004), o Art. 47 determina que “no prazo de até doze meses a contar da data de publicação deste Decreto, será obrigatória a acessibilidade nos portais e sítios eletrônicos da administração pública na rede mundial de computadores (*Internet*), para o uso das pessoas portadoras de deficiência visual, garantindo-lhes o pleno acesso às informações disponíveis”.

Nota-se a preocupação do Governo Federal em adaptar o conteúdo de sítios *web* da administração pública para deficientes visuais, o que evidencia que estes indivíduos podem e utilizam da *Internet* para desempenhar e estender sua cidadania.

Apesar da gravidade da cegueira e visão subnormal e da maioria das políticas e diretrizes adotadas visarem estes PNEs, o daltonismo e outras deficiências visuais também estão relacionados a acessibilidade digital:

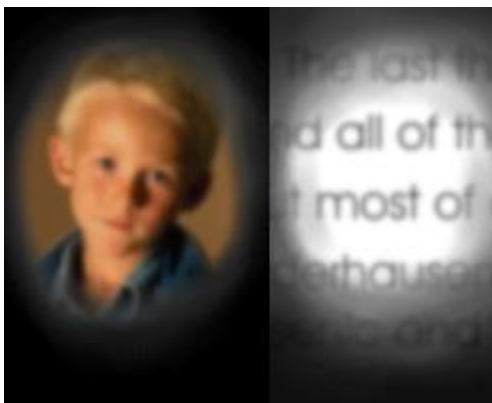
“Os problemas de acessibilidade mais sérios, dado o atual estado da *Web*, relacionam-se a usuários cegos e a usuários com outras deficiências visuais, posto que a maioria das páginas da *Web* é altamente visual. Por exemplo, é bastante comum ver combinações de cores de primeiro e segundo plano que tornam as páginas praticamente ilegíveis a usuários daltônicos.” (NIELSEN, 2000, p. 302)

Ainda que grande parte dos deficientes visuais não seja totalmente cega, obviamente estes indivíduos não são capazes de enxergar os elementos visuais dispostos na *web* da mesma maneira que os usuários não PNEs pois eles não têm uma visão útil suficiente para navegar na *web*. Um monitor de computador, uma tela sensível ao toque e uma ferramenta de precisão como mouse ou caneta não são muito úteis para deficientes visuais porque estes usuários não podem ver ou interagir com o que está em tela sem um *feedback* tátil ou auditivo.

Segundo o WebAIM (2013), a percepção visual das pessoas com visão subnormal é muito variável mas é definida como uma condição em que a vista de um indivíduo não pode ser completamente corrigida por óculos, desta forma interferindo nas atividades diárias, principalmente com a leitura. A visão subnormal é mais comum em indivíduos idosos, mas também pode ocorrer em indivíduos em qualquer idade

devido a mazelas definitivas ou temporárias como degeneração macular, glaucoma, diabetes ou catarata.

Figura 1 – Glaucoma



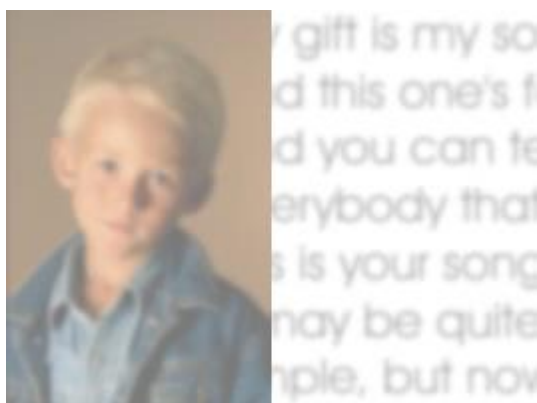
Fonte: WEBAIM, 2013

Figura 2 – Degeneração macular



Fonte: WEBAIM, 2013

Figura 3 – Catarata



Fonte: WEBAIM, 2013

Figura 4 – Retinopatia diabética



Fonte: WEBAIM, 2013

Conforme já citado, Nielsen (2000) afirma que devido à algumas combinações de cores, algumas páginas da web são completamente ilegíveis a usuários daltônicos.

As cores com indivíduos daltônicos tem dificuldade em diferenciar dependem do tipo de ausência da percepção de cor. Entretanto, as deficiências para as cores vermelho e verde são as mais comuns. Os diferentes tipos de daltonismo são separados por:

- Protanopia e protanomalia (Deficiência com vermelho) - Ausência da percepção de comprimentos de ondas longas (vermelho);

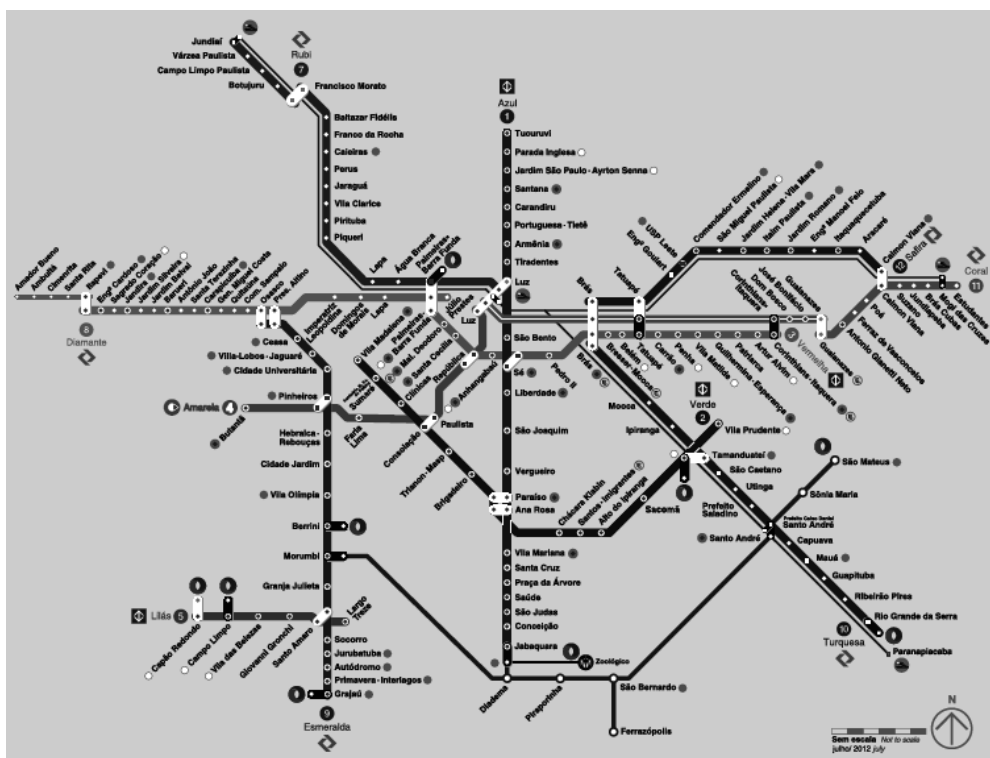
- Deuteranopia e deuteranomalia (Deficiência com a cor verde) - Ausência da percepção de comprimentos de ondas médias (verde);
- Tritanopia (Deficiência com a cor azul) - Ausência da percepção de comprimentos de ondas curtas (azul);
- Rod monocromia ou acromia (sem percepção de cores) - É a cegueira de todas as cores.

Figura 5 – Comparação dos tipos de daltonismo lado a lado. Da esquerda para a direita, visão normal, protanopia, deuteranopia, tritanopia e acromia



Fonte: WEBAIM, 2013

Figura 6 – Mapa do transporte metropolitano de São Paulo sob a perspectiva de um usuário com acromia



Fonte: Imagem manipulada digitalmente a partir do sítio web do Metrô de São Paulo

Devem ser considerados também outros indivíduos PNEs com deficiências não visuais ou até mesmo múltiplas deficiências que os impeçam de efetuar movimentos com teclado ou mouse e também a utilizarem ferramentas e tecnologias que os auxiliem na acessibilidade *web*.

2.2. Diretrizes e Contexto Histórico da Acessibilidade Web

2.2.1. Breve histórico do W3C

O Consórcio *World Wide Web* (W3C) fora fundado em 1994 por Tim Berners-Lee no Laboratório de Ciência da Computação do Instituto de Tecnologia de Massachusetts (MIT) em colaboração com a Organização Europeia para a Pesquisa Nuclear (CERN) e a Agência de Projetos de Pesquisa Avançada de Defesa (DARPA) com a finalidade de se tornar a principal organização que padroniza a *World Wide Web*, também inventada por Tim Berners-Lee em 1989 (W3C, 2008). Atualmente consiste de um consórcio que agrega empresas, órgãos governamentais e organizações independentes que contribuem com a criação de novos padrões para criação e a interpretação de conteúdo na *web*.

O W3C desenvolve os padrões e diretrizes para a *web* e desde sua fundação tem publicado seus artigos como “Recomendações do W3C” defendendo a “Interoperabilidade da *Web*”. Suas diretrizes são públicas e os padrões abertos, contribuindo com a comunidade de desenvolvedores e evitando a fragmentação da própria *web* (JACOBS, 2008).

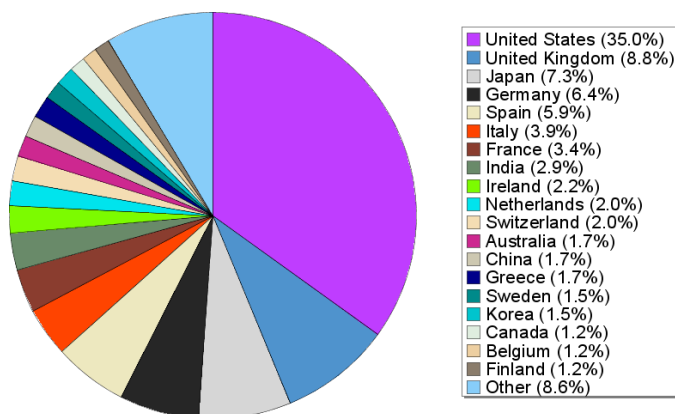
De acordo com Jacobs, o consórcio é financiado por uma combinação de recursos de seus filiados, doações para pesquisas e fontes de recursos públicos e privados, além de um programa para apoiadores. Possui escritórios em diversas regiões do mundo, incluindo no Brasil, promovendo as tecnologias do consórcio nos idiomas locais e incentivando a participação internacional em suas atividades através de eventos e fóruns de discussão.

As principais empresas multinacionais líderes do setor de tecnologia e comunicação são filiadas ao consórcio, segundo dados do W3C, sendo os três setores

econômicos com mais filiais as empresas de consultoria, os departamentos de pesquisa e desenvolvimento de universidades e as empresas de software. O Brasil tem como filiados o banco Caixa Econômica Federal, o Comitê Gestor da *Internet* no Brasil, o Departamento de Informática da PUC-RIO, o Serviço Federal de Processamento de Dados, o SENAC, a Universidade Federal de Alagoas, a Universidade Federal do Espírito Santo e a Universidade Federal de Pernambuco (W3C, 2013).

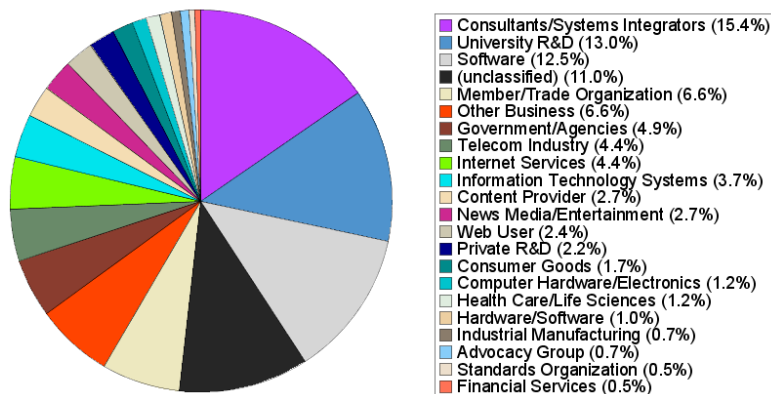
Segundo o W3C.br (2013), o escritório do W3C no Brasil está em funcionamento desde 2007, atuando em conjunto com o Comitê Gestor da *Internet* no Brasil propondo políticas e procedimentos relativos à regulamentação do uso da *Internet* e à recomendação de padrões técnicos e procedimentos operacionais para o desenvolvimento da *Internet* no Brasil. O W3C no Brasil também promove cursos, palestras e eventos em todo o território nacional, divulgando-os em seu *sítio web*.

Figura 7 - Distribuição dos filiados do W3C por país



Fonte: W3C, 2012

Figura 8 - Distribuição dos filiados do W3C por setor de negócios

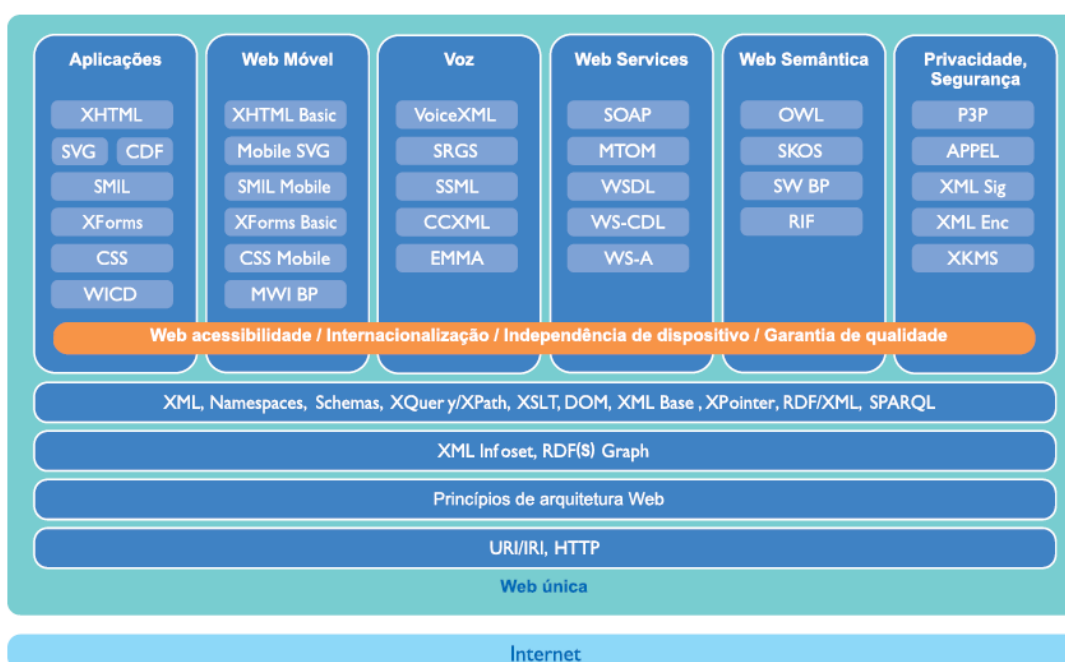


Fonte: W3C, 2012

2.2.2. Estrutura Tecnológica do W3C

O W3C apresenta um modelo em que define a arquitetura da *web* (também chamada de “*Web Única*”) em uma série de camadas no topo da arquitetura da *Internet*. A estrutura tecnológica do W3C se dá pela camada intermediária da *web*, com as áreas de interesse e as tecnologias desenvolvidas pelo consórcio.

Figura 9 – Estrutura tecnológica do W3C e arquitetura da Web



Fonte: W3C, 2012.

2.2.3. A Web Accessibility Initiative

De acordo com Henry e McGee ([s.d.]), a *Web* é fundamentalmente desenvolvida para funcionar para todas as pessoas, não importando o hardware, software, cultura, lugar ou habilidade física ou mental. Atingindo estes requisitos, a *web* se torna acessível para indivíduos com uma gama diversificada de habilidades em audição, movimento, visão e capacidade cognitiva.

Supostamente, o impacto da deficiência deveria ser drasticamente diferente na *Web* do que é sentido no mundo físico, uma vez que ela tem o poder de remover as barreiras de comunicação e interação que indivíduos portadores de deficiência encontram.

Com a missão de conduzir a *Web* a seu total potencial, o projeto de *Acessibilidade Web* como projeto do W3C foi concebido no final de 1996 como iniciativa de alguns membros do consórcio (DARDAILLER, 2009). Nesta data era evidente que o W3C precisava fazer mais do que apenas corrigir e guiar as tecnologias existentes, mas também expender muitos recursos em educação e ferramentas para provedores de conteúdo. Com isso em 1997 fora aberto o *WAI International Program Office* (IPO) para gerenciar novos fundos e as novas atividades do W3C.

Com o reconhecimento da ONU através da Convenção sobre os Direitos das Pessoas com Deficiência entrando em rigor em 2006 (ONU, 2008), o acesso a informação e tecnologias de comunicação, incluindo assim a *Web*, passaram a se tornar um direito humano básico.

É reconhecido também pelo W3C que acessibilidade provê a inclusão social não apenas a indivíduos PNEs, como também a idosos, pessoas em áreas rurais e países em desenvolvimento. Henry e Arch (2012) afirmam que acessibilidade é uma excelente estratégia para organizações pois contribui em diferentes aspectos de negócio:

- Fornecendo oportunidades iguais para pessoas com deficiência e também para usuários idosos, pessoas com baixo nível de alfabetização, pessoas não fluentes em determinados idiomas, usuários de dispositivos móveis, usuários de tecnologias antigas, com baixa largura de banda para *Internet* ou iniciantes na utilização da tecnologia;
- Tratamento da interoperabilidade e qualidade, redução de tempo de manutenção e carga de servidores, habilitando o conteúdo para diferentes tecnologias (atuais ou futuras) e configurações;
- Aumento da demanda e utilização de um sítio ou serviço *web*, contribuindo com a captação de recursos através de propaganda e ao mesmo tempo diminuindo custos em outras áreas. A acessibilidade também está relacionada a *Search Engine Optimization* (SEO);
- Fatores legais e políticos, acatando a leis, decretos, ordens e outros regulamentos de governos e outras organizações;

O WAI desenvolve as diretrizes para acessibilidade *web* que incluem “Conteúdo *web*” (sítios *web* e aplicações *web*), ferramentas de publicação (sistemas gerenciadores de conteúdo e *blogs*), navegadores e outros *user agents* (interpretadores), protocolos e outras especificações do W3C, incluindo o WAI-ARIA para *Rich Internet Applications* (RIA) acessíveis, que serão abordadas neste trabalho.

2.2.4. Diretrizes de Conteúdo - WCAG

Em 1999 o W3C publicou o primeiro documento com diretrizes para a acessibilidade na *web*, o WCAG 1.0 (WCAG, 2012). Tal documento fora desenvolvido em conjunto com indivíduos e organizações com o objetivo de definir um padrão internacional para garantir que o conteúdo na *web* fosse mais acessível a indivíduos PNEs. De acordo com o WCAG (2012), “conteúdo *web*” se refere a informação disponível em uma página ou aplicação, como por exemplo, textos, imagens, vídeos, sons ou, código e marcação que definem a estrutura e apresentação da *web*.

O WCAG 1.0 foi desenvolvido possuindo três níveis de prioridades, sendo:

- Prioridade 1: Estes requisitos precisam ser atendidos, caso contrário será impossível para um ou mais grupos de indivíduos acessar o conteúdo *web*. A aderência a este nível é descrita como “A”.
- Prioridade 2: Estes requisitos devem ser atendidos, caso contrário alguns grupos de indivíduos terão dificuldades em acessar o conteúdo *web*. A aderência a este nível é descrita como “AA” ou “A duplo”.
- Prioridade 3: Estes requisitos podem ser atendidos, de forma que facilitem o acesso a alguns grupos de indivíduos ao conteúdo *web*. A aderência a este nível é descrita como “AAA” ou “A triplo”.

Para o WCAG 1.0, foram propostas 14 diretrizes pelo W3C (1999):

- Recomendação 1 - Fornecer alternativas equivalentes ao conteúdo sonoro e visual: Proporcionar conteúdo que, ao ser apresentado ao usuário, transmita, em essência, as mesmas funções e finalidade que o conteúdo sonoro ou visual.

- Recomendação 2 - Não recorrer apenas à cor: Assegurar a percepção do texto e dos elementos gráficos quando vistos sem cores.
- Recomendação 3 - Utilizar corretamente marcações e folhas de estilo: Marcar os documentos com os elementos estruturais adequados. Controlar a apresentação por meio de folhas de estilo, em vez de elementos de apresentação e atributos.
- Recomendação 4 - Indicar claramente qual o idioma utilizado: Utilizar marcações que facilitem a pronúncia e a interpretação de abreviaturas ou texto em língua estrangeira.
- Recomendação 5 - Criar tabelas passíveis de transformação harmoniosa: Assegurar que as tabelas têm as marcações necessárias para poderem ser transformadas harmoniosamente por navegadores acessíveis e outros *user agents*.
- Recomendação 6 - Assegurar que as páginas dotadas de novas tecnologias sejam transformadas harmoniosamente: Assegurar que as páginas são acessíveis mesmo quando as tecnologias mais recentes não forem suportadas ou tenham sido desativadas.
- Recomendação 7 - Assegurar o controle do usuário sobre as alterações temporais do conteúdo: Assegurar a possibilidade de interrupção momentânea ou definitiva do movimento, intermitência, transcurso ou atualização automática de objetos ou páginas.
- Recomendação 8 - Assegurar a acessibilidade direta de interfaces do usuário integradas: Assegurar que a interface do usuário obedeça a princípios de design para a acessibilidade: acesso independente de dispositivos, operacionalidade pelo teclado, emissão automática de voz (verbalização).
- Recomendação 9 - Projetar páginas considerando a independência de dispositivos: Utilizar funções que permitam a ativação de elementos de página por meio de uma grande variedade de dispositivos de entrada de comandos.
- Recomendação 10 - Utilizar soluções de transição: Utilizar soluções de acessibilidade transitórias, para que as tecnologias de apoio e os navegadores mais antigos funcionem corretamente.

- Recomendação 11 - Utilizar tecnologias e recomendações do W3C: Utilizar tecnologias do W3C (de acordo com suas especificações) e seguir as recomendações de acessibilidade. Quando não for possível utilizar tecnologia W3C, ou quando tal utilização produzir materiais que não possam ser objeto de transformação harmoniosa, fornecer uma versão alternativa, acessível, do conteúdo.
- Recomendação 12 - Fornecer informações de contexto e orientações: Fornecer contexto e orientações para ajudar os usuários a compreenderem páginas ou elementos complexos.
- Recomendação 13 - Fornecer mecanismos de navegação claros: Fornecer mecanismos de navegação coerentes e sistematizados - informações de orientação, barras de navegação, mapa do sítio - para aumentar as probabilidades de uma pessoa encontrar o que procura em um dado sítio.
- Recomendação 14 - Assegurar a clareza e a simplicidade dos documentos: Assegurar a produção de documentos claros e simples, para que sejam mais fáceis de compreender.

Com o desenvolvimento tecnológico, as diretrizes do WCAG 1.0 começaram a se tornar obsoletas e a partir de 2001 o W3C iniciou o projeto da próxima versão do WCAG. Até 2006 o WCAG 2.0 era publicado como rascunho e desta vez a comunidade pode contribuir com a edição das novas diretrizes (CLARK, 2006). Em 11 de dezembro de 2008, o WCAG 2.0 foi oficialmente apresentado como uma recomendação do W3C, atualizando as diretrizes de sua versão anterior (WCAG, 2008).

O WCAG 2.0 (2008) categoriza suas recomendações em quatro princípios, cada uma contendo recomendações das prioridades A, AA ou AAA, devendo o conteúdo na *web* ser:

- Perceptível - A informação e os componentes da interface do usuário têm de ser apresentados aos usuários em formas que eles possam perceber.

- Operável - Os componentes de interface de usuário e a navegação têm de ser operáveis.
- Compreensível - A informação e a operação da interface de usuário têm de ser compreensíveis.
- Robusto - O conteúdo tem de ser robusto o suficiente para poder ser interpretado de forma concisa por diversos agentes do usuário, incluindo tecnologias assistivas.

Desde outubro de 2012, o WCAG 2.0 é reconhecido como um padrão ISO/IEC *International Standard* (ISO/IEC 40500:2012). Com a aprovação formal do WCAG 2.0 como norma ISO, alguns países que exigem que suas normas técnicas sejam ISO/IEC esperam reduzir a fragmentação da *web* e ampliar a implantação das diretrizes do W3C.

Durante o desenvolvimento do WCAG 2.0, um grupo de desenvolvedores liderados por Joe Clark (2006) propôs uma lista de correções e extensões para as recomendações do WCAG 1.0, fornecendo uma alternativa às especificações do WCAG 2.0. Esta lista de correções teve a colaboração de diversos editores e revisores, sendo que o anonimato do time de colaboradores fora preservado. Tal documento fora batizado de “WCAG Samurai” e é considerado uma “errata” às recomendações do WCAG 1.0. Esta errata não tem como objetivo substituir o WCAG 1.0, mas complementá-lo, não sendo possível seguir o WCAG Samurai em conjunto com o WCAG 2.0.

Clark (2006) faz duras críticas ao WCAG 2.0 a respeito do tamanho do documento e a dificuldade de entendimento dele, o que pode levar a diversas interpretações diferentes que não beneficiariam nem aos desenvolvedores e tão pouco aos usuários. Uma das recomendações mais radicais do WCAG Samurai por exemplo, é ignorar todas as diretrizes da Prioridade 3 ou AAA do WCAG.

2.2.5. Tecnologias Assistivas

“Tecnologia assistiva são recursos e serviços que visam facilitar o desenvolvimento de atividades diárias por pessoas com deficiência.

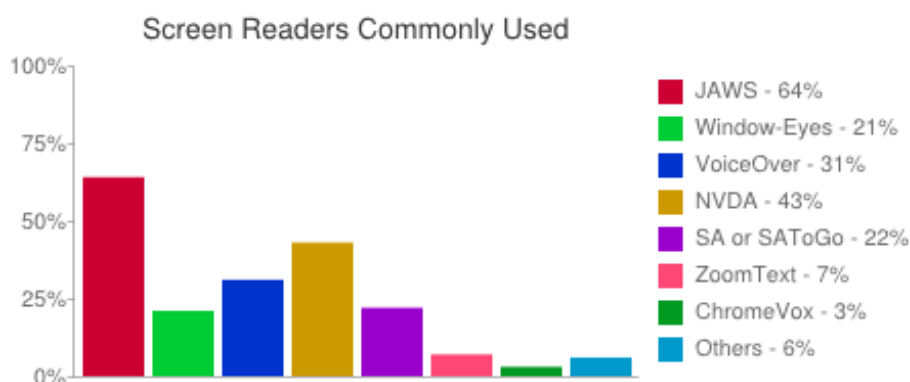
Procuram aumentar as capacidades funcionais e assim promover a independência e a autonomia de quem as utiliza”. (MELO, 2007, p. 94)

Dentre os recursos de tecnologias assistivas para deficientes visuais, existem *softwares* e *hardwares* voltados tanto para usuários cegos quanto com visão subnormal ou baixa visão. Com finalidade de abordar as tecnologias voltadas para a *web*, este trabalho será focado apenas em um subconjunto de tecnologias assistivas apresentadas como *softwares* leitores de telas.

Um software leitor de tela é um aplicativo que interage com o Sistema Operacional capturando as informações dispostas em forma de texto e apresentando-as em forma de uma resposta falada através de um sintetizador de voz ou imprimindo o texto em forma de braile, através de uma impressora específica (NVACCESS, 2013).

Atualmente existe uma quantidade considerável de softwares leitores de telas e motores sintetizadores de vozes. Segundo dados do WebAIM (2012), os *softwares* leitores de telas mais utilizados são, em quantidade de usuários entrevistados: JAWS, Window-Eyes, VoiceOver, NVDA, System Access, ZoomText e ChromeVox.

Figura 10 – Leitores de tela mais utilizados

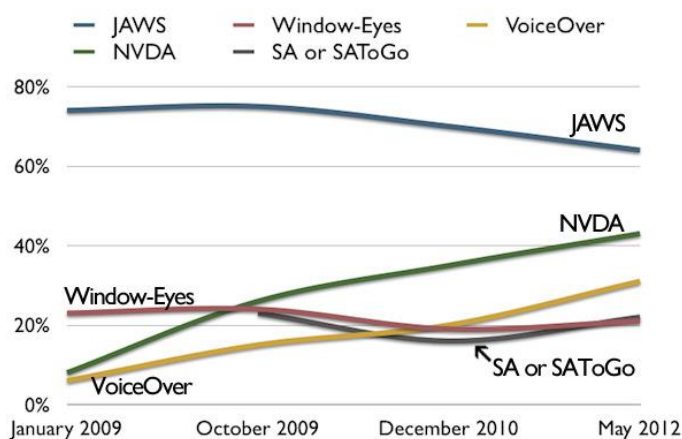


Fonte: WEBAIM, 2012

Ainda de acordo com o WebAIM, apesar da liderança do *software* JAWS, sua utilização tem caído enquanto ambos NVDA e VoiceOver tiveram um crescimento considerável nos últimos anos. Esta mudança pode ser justificada pelo tipo de licença de uso destes *softwares*, sendo o JAWS um *software shareware*, custando em sua versão mais completa \$1.095 em dólares estadunidenses (FREEDOM SCIENTIFIC,

2013), enquanto o NVDA tem licença GNU *General Public License*, é gratuito e *open-source* (NVACCESSS, 2013) e o VoiceOver empacotado e embutido com os sistemas operacionais OS X e iOS da Apple Inc.

Figura 11 – Mudanças no uso do leitor de tela ao longo do tempo.



Fonte: WEBAIM, 2012

2.2.6. Tecnologias Assistivas Desenvolvidas no Brasil

De acordo com Sonza (2008), os softwares Virtual Vision e Jaws são os leitores de tela mais utilizados no Brasil.

Apesar de não aparecer listado nas pesquisas do WebAIM, a liderança do Virtual Vision no mercado brasileiro é justificada pela forma de sua distribuição e seu histórico. Desenvolvido em 1997 pela empresa brasileira MicroPower, o Virtual Vision é considerado o único *software* de leitura de telas desenvolvido no Brasil capaz de funcionar sobre os aplicativos mais comuns no sistema operacional Windows (MICROPOWER VIRTUAL VISION, 2011). Em uma parceria com o banco Bradesco e a empresa Scopus, a MicroPower desenvolveu o produto e em 1998 fora lançado o “*Internet Banking para Deficientes Visuais*”.

A iniciativa garantiu que a aplicação de *Internet Banking* fosse premiada em 1999 na *ComputersWorld Smithsonian Awards* e foi responsável pelo surgimento de uma demanda de treinamento em informática para pessoas com deficiência visual

no Brasil. Desde o lançamento da aplicação, o banco Bradesco fornece a licença de uso do *software* para seus clientes deficientes visuais e em 2003, o Banco Real (atualmente pertencente ao grupo Santander) também passou a oferecer o *software* para seus clientes e funcionários.

Apesar de seu sucesso, Virtual Vision não foi o primeiro *software* leitor de tela desenvolvido no Brasil. Em 1993 o Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro (UFRJ) desenvolveu o DOSVOX, sendo que até 2002 o trabalho continuou em andamento com versões do *software* para o sistema operacional Windows, chamado de WINVOX (PROJETO DOSVOX, 2002).

Em 2005, o Ministério das Comunicações do Brasil (BRASIL, [s.d.]), em parceria com o CPqD (Centro de Pesquisa e Desenvolvimento em Telecomunicações) também desenvolveu um *software* leitor de telas também gratuito, com o objetivo de utilizá-lo em todos os seus programas de inclusão digital.

Além dos avanços tecnológicos que compreendem a arquitetura da *Web* proposta pelo W3C, navegadores e sistemas operacionais passaram por grandes atualizações nos últimos anos, possibilitando o desenvolvimento de aplicações com características similares a de *softwares* e aplicativos *desktop*. Infelizmente as alternativas nacionais parecem não ter acompanhado este ritmo, uma vez que com exceção do Virtual Vision, os outros *softwares* desenvolvidos no país não sofreram grandes atualizações nos últimos anos e portanto não suportam novas tecnologias em uso como HTML5 e AJAX.

3. APLICAÇÕES WEB E JAVASCRIPT

3.1. A Internet e Aplicações Client-Side

Nations (2012) define aplicações *web* como qualquer aplicação que utiliza um navegador como “cliente”, enquanto Rapoza (2009) descreve *Rich Internet Application* (RIA) como aplicações que tem características e funcionalidades de *softwares* tradicionais e que transferem todo o processamento da interface do *software* para o navegador porém mantendo a maior parte do processamento de dados num servidor de aplicação. O termo RIA fora introduzido pela primeira vez pela empresa Macromedia em 2002 quando a empresa apresentou a nova versão de seu *software* Macromedia Flash.

De acordo com Allaire (2002), a partir da segunda metade da década de 90, um crescimento explosivo da *Internet* e da *World Wide Web* levou a adoção generalizada de um novo modelo de conteúdo e aplicações usando computadores pessoais conectados à *Internet*. Este modelo prometeu reduzir o custo de desenvolvimento e entrega de aplicações em *desktops* para usuários finais, clientes e parceiros de negócios, e aumentar a gama de tipos de aplicações que poderiam ser desenvolvidas. A premissa deste modelo é centrada em um “cliente-magro” baseado em HTML e poderosos servidores de aplicação que componham e entreguem dinamicamente páginas estáticas com os dados processados para os navegadores.

Este modelo se provou bem sucedido até o começo dos anos 2000, mas sofria de desvantagens e limitações significativas, especialmente em torno da riqueza das interfaces destes aplicativos, da forma de distribuição de mídia e conteúdo, assim como em relação a sofisticação geral das soluções que poderiam ser construídas e ofertadas. Enquanto a *web* oferece conveniências significativas em termos de facilidade de implantação, as capacidades deste modelo de interação e programação para a *web* era sofrível para os usuários. Em muitos aspectos, a maior parte do desenvolvimento e implantação de aplicações *web* do final dos anos 1990 teve de se adaptar aos desafios impostos pela arquitetura inerente à *web* da época.

Para que uma aplicação seja considerada de fato uma RIA, segundo Allaire (2002), é necessário que esta aplicação atinja os seguintes requisitos:

- Tenha um *runtime* eficiente e de alta performance para execução do código, conteúdo e comunicações. O usuário de aplicações *web* está suscetível a enfrentar diversos desafios relacionados a performance, que incluem o processo de desenho dos elementos em HTML, o modelo de requisição-resposta dos servidores de aplicação, o tempo de espera de transmissão de dados e a falta de suporte a armazenamento de dados eficiente no *client-side*. De acordo com o W3C (2012), na especificação do HTML5 existem *Application Programming Interfaces* (APIs) para armazenamento local utilizando *cache* de aplicativo.
- Integre conteúdo, comunicação e interface em um único ambiente comum. A navegação em páginas estáticas na *web* era fragmentada entre a leitura de dados textuais em uma página e aplicativos de comunicação ou consumo de mídia em outra página. Uma solução de RIA deve fornecer uma profunda integração destes tipos de interação.
- Fornecer modelos de objetos poderosos e extensivos para interatividade. Em 2002 quando fora introduzido o conceito de RIA pela Macromedia, o suporte a aplicações *web* por intermédio de *JavaScript* e HTML ainda era muito limitado devido a deficiência do suporte a interatividade no *Document Object Model* (DOM) que uma aplicação rica requeria. Com a atualização dos motores *JavaScript* pelos navegadores mais recentes e a adoção do HTML5 (W3C, 2012) hoje é possível fornecer aplicações ricas sendo possível manipular o DOM de uma página HTML dinamicamente.
- Facilitar o desenvolvimento permitindo a reutilização de código e criação de componentes. De acordo com Allaire (2002), as aplicações ricas devem suportar o uso de componentes, permitindo assim que diferentes módulos das aplicações possam ser desenvolvidos por diferentes partes - terceiros, parceiros ou outros fornecedores - e sejam reutilizáveis ou personalizáveis, acelerando assim o desenvolvimento até mesmo para funcionalidades ou requisitos mais complexos.

- Permitir o uso de *web services* e *data services* de servidores de aplicação. A premissa das aplicações ricas é de que a camada de apresentação e a interface estejam separadas da lógica da aplicação hospedada na rede. Aplicações ricas devem prover um modelo para acesso a serviços remotos.
- Suportar ambientes *online* e *offline*. Aplicações *desktop* tem a vantagem de não requererem uma conexão com a *Internet* o tempo todo para funcionar enquanto aplicações *web* comuns requerem uma conexão persistente entre um cliente e um servidor. Uma aplicação rica deve ser hábil a funcionar *offline* ou em ambientes com condições de conexão muito variáveis como celulares e *tablets* por exemplo.
- Desenvolvimento em múltiplas plataformas e dispositivos. Seguindo a premissa de funcionar em ambientes *offline*, o desenvolvimento de aplicações ricas para múltiplas plataformas e aparelhos também é um fator importante a ser considerado.

As aplicações *web* se tornaram muito populares por conta da conveniência de usar o navegador como cliente. A facilidade de atualizar e manter estas aplicações sem a necessidade de distribuição ou instalação de *software* é um dos principais fatores responsáveis pela rápida adoção deste modelo, além do inerente suporte *cross-browser*.

3.2. Contextualização das Tecnologias Envolvidas

3.2.1. HTML e Document Object Model

HyperText Markup Language (Linguagem de Marcação de Hipertexto, HTML) é uma linguagem de marcação utilizada para construção e produção de páginas na *web*. Esta marcação, dada por etiquetas ou “*tags*”, cria uma estrutura para elementos textuais, figuras e formulários serem apresentados pelos navegadores (LONGMAN, 1998).

O HTML foi inventado por Tim Bernes-Lee na década de 1990. A estrutura do HTML é fortemente baseada em SGML (*Standard Generalized Markup Language*),

uma metalinguagem que fornece uma variedade de sintaxes para marcação que possam ser utilizadas por outras linguagens e aplicações e faz parte da norma ISO 8879:1986 (ISO, 1986).

Desde a primeira versão do HTML até hoje, cada versão reflete o estado da indústria de software e suas necessidades. Desde seu desenvolvimento foi pensado como uma linguagem acessível e "universal" mas, devido a regras sintáticas flexíveis demais e por questões de retro compatibilidade dos navegadores, muitas páginas na *web* não apresentam um código HTML válido, o que pode acarretar em problemas de acessibilidade. Entretanto, Clark (2006) afirma que um documento HTML válido não é automaticamente acessível.

Dos primeiros 5 anos de criação, o HTML passou por diferentes revisões e extensões. As primeiras quando era uma especificação CERN, depois pelo IETF (*Internet Engineering Task Force*) e a partir de 1995 através do W3C quando fora proposto o HTML 3.0. Em 1997 fora proposta uma extensão chamada de HTML 3.2 e no final do mesmo ano surgiu o HTML 4 (W3C, 2013).

Em 1998, o W3C decidiu parar o desenvolvimento do HTML e começou a desenvolver um equivalente baseado em XML (*eXtensible Markup Language*) batizado de XHTML, o que criou a especificação XHTML 1.0, formalizando-a em 2000. No mesmo ano, o W3C iniciou os projetos para a especificação do XHTML 2.0.

Em um *workshop* do W3C em 2004, a fundação Mozilla e a Opera Software apresentaram as primeiras propostas de evolução do HTML, que hoje é conhecido como HTML 5. A proposta inicial fora rejeitada pelo W3C, já que o consórcio preferiu continuar o desenvolvimento do XHTML.

Logo em seguida, Mozilla e Opera uniram forças com a Apple e iniciaram o trabalho de documentação e especificação do HTML5 sob nome de WHATWG (*Web Hypertext Application Technology Working Group*). Ainda de acordo com o W3C (2013), o WHATWG era baseado em vários princípios chave, como por exemplo:

- As tecnologias precisavam ser retro compatíveis;

- As especificações e implementações precisavam corresponder, mesmo que isto implicasse em alterar a especificação em vez da implementação;
- As especificações precisavam ser detalhadas o suficientes para que as implementações atingissem completa interoperabilidade sem a necessidade de engenharia reversa;

Apenas em 2006 o consórcio W3C demonstrou interesse em participar do desenvolvimento do HTML 5 e no começo de 2007 formou um grupo em conjunto com o WHATWG. O W3C ficou com o papel de registrar quais seriam os principais recursos do HTML 5, enquanto o WHATWG concentrou seus esforços no contínuo desenvolvimento dos padrões do HTML, mantendo a especificação e acrescentando novos recursos. Desde 2008, o W3C ficou encarregado de criar as recomendações para o HTML 5, publicando assim seu primeiro “*Working Draft*” (projeto de recomendação). Em dezembro de 2012, o W3C acrescentou o HTML 5 em sua lista de candidatos a recomendação (W3C, 2013).

Além do HTML, o W3C criou a especificação do *Document Object Model* ou Modelo de Objeto de Documentos (DOM) em 1998 (W3C, 2005). Segundo a Mozilla (2013), DOM é uma API independente de plataforma e linguagem que permite o acesso e atualização do conteúdo, estrutura e estilo de documentos *web* como HTML e XML dinamicamente através de linguagens como *JavaScript*. Anteriormente a especificação, cada navegador adotava seu próprio motor interpretador do DOM, causando severos problemas de interoperabilidade e conflitos *cross-browser*, fazendo com que o suporte a determinados navegadores fosse excluído ou limitado em algumas aplicações (MOZILLA DEVELOPER NETWORK, 2013).

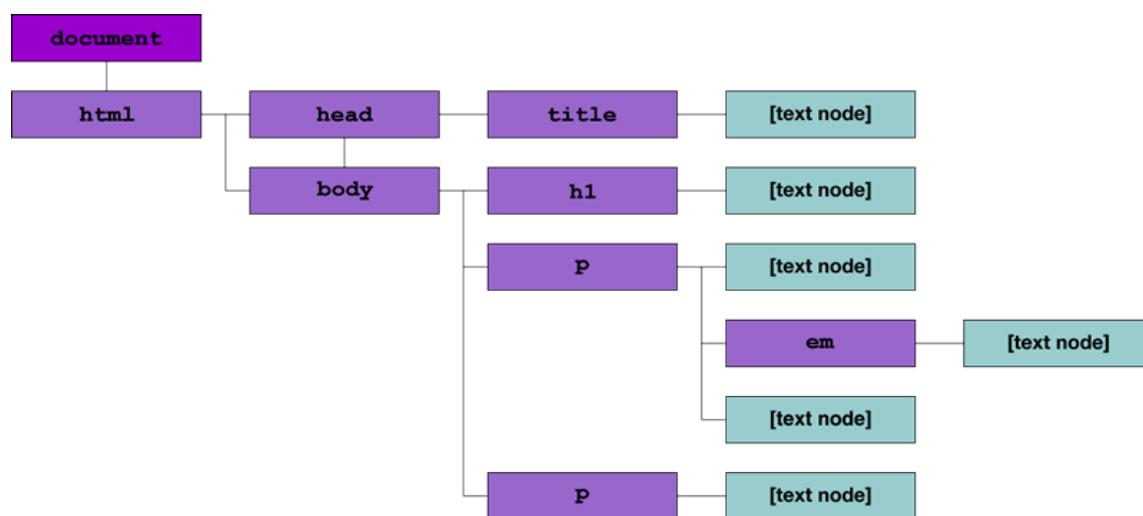
Todas as propriedades, métodos e eventos disponíveis para os desenvolvedores *web* para manipulação e criação de páginas *web* estão organizados em objetos. Estes objetos são acessíveis através de linguagens de script dos navegadores mais recentes.

O DOM é mais utilizado em conjunto com *JavaScript*, mas conforme já citado pelo W3C, fora desenvolvido para ser independente de qualquer linguagem de programação.

Quando uma página *web* é construída, são atribuídas etiquetas ou *tags* que recebem atributos e podem encapsular uma porção de dados ou textos. Esta página é interpretada pelo navegador e através de um estilo padrão, o documento é apresentado para o usuário. O desenvolvedor também pode atribuir uma classe ou identificador à esta *tag*, podendo assim manipula-la por alguma linguagem de programação ou modificar seu estilo através da linguagem CSS (*Cascading Style Sheet*), manipulando a formatação visual e sobrescrevendo o padrão do navegador (DEV.OPERA, 2009).

Basicamente, o DOM representa um documento *web* como uma árvore e cada elemento ou *tag* contido numa página é atribuído no DOM como um nó, com seus ramos apontando diretamente para os elementos que cada nó contém (seus elementos filhos) e o elemento que o contém (seu elemento pai).

Figura 12 – Representação de um documento HTML e sua árvore DOM.



Fonte: DEV.OPERA, 2009

O diagrama mostra a estrutura da árvore DOM de um documento HTML, mas não representa o que será exibido pelo navegador. Segundo Franklin (2011), o DOM é usado como base para uma outra árvore que os navegadores utilizam para a composição da tela, a *Render Tree* (Árvore de Renderização).

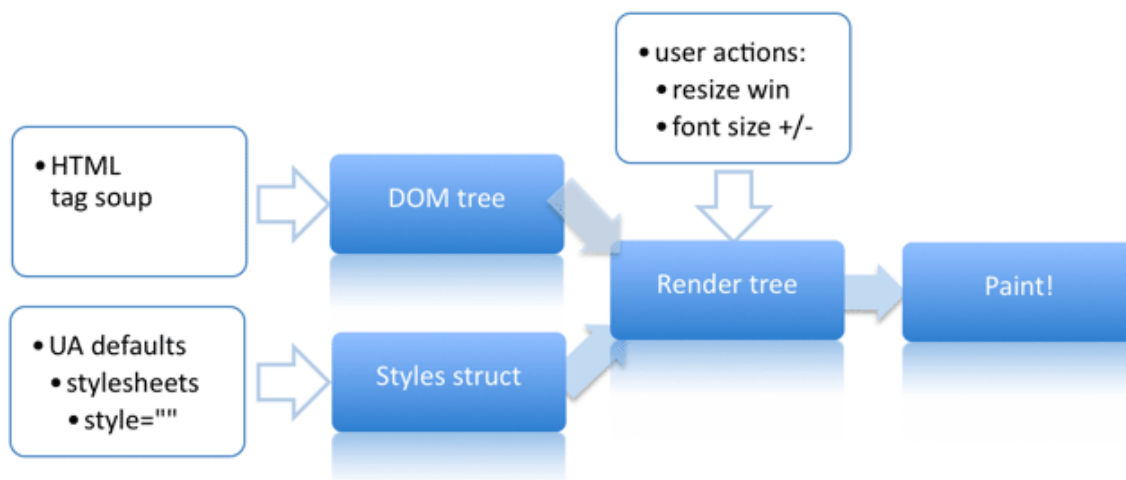
Hyatt (2007) explica que a classe base para todos os nós em uma árvore é chamado de Node.h e é subdividido em várias categorias, sendo as mais relevantes para composição do código no navegador os seguintes nós:

- Documento - A raiz da árvore é sempre o próprio documento. São conhecidas três classes de documento, Document, HTMLDocument e SVGDocument. O primeiro é utilizado por todos os documentos XML, o segundo por todos os documentos HTML e o terceiro para todos os documentos SVG;
- Elementos - Todas as *tags* ou etiquetas que aparecem em um documento HTML ou XML se tornam elementos. Na perspectiva do navegador, um elemento é um nó com uma *tag* que pode ser utilizada para criar subclasses específicas que possam ser processadas de acordo com a árvore de renderização;
- Texto - Texto puro que aparece entre elementos e que é transformado em nós de texto. São os caracteres que são escritos dentro das *tags*;

A árvore de renderização é a parte mais importante da composição de uma página. Além dos objetos na árvore DOM, também possui objetos como *scripts* e folhas de estilos. O processo da *Render Tree* é dado por três passos:

- *Attachment* - Após o navegador interpretar o DOM e criar seus nós, o método *attach* é chamado para iniciar a composição da tela. Primeiramente são adicionadas as folhas de estilo na árvore DOM e começa a estilização do documento;
- *RenderStyle* - Durante a execução do *attach* o *RenderStyle.h* é usado para guardar objetos de referência para cada uma das propriedades de CSS de um documento. O nó criado no DOM é procurado no documento CSS e caso seja encontrado, as propriedades da folha de estilo são aplicadas naquele elemento dentro da *Render Tree* até que este elemento seja destruído ou tenha algum valor alterado por *script*;
- *RenderObject* - Após a execução do método *RenderStyle*, é usado o *RenderObject* para posicionar um elemento dentro de um documento.

Figura 13 – Representação da Render Tree.



Fonte: FRANKLIN, A., 2011

Todos os navegadores tem em seu motor interpretador um conjunto de elementos HTML suportados. Se a *tag* presente no documento está também presente nesta lista, o processo de *attachment* começa assim que a árvore DOM é montada e em seguida são aplicados os estilos, dando continuidade à *Render Tree*. Quando algum *script* remove um elemento ou nó da árvore DOM, é necessário que o DOM reinterprete o código do documento e assim todo o processo acima descrito é refeito.

De acordo com Faulkner (2008), os grandes problemas com relação a manipulações no DOM e o funcionamento em conjunto por ferramentas assistivas, especialmente os leitores de tela, são que os usuários não tem acesso a estas alterações ou, quando tem, não estão cientes que uma mudança ocorreu em tela, já que isto não é anunciado pelos sintetizadores de voz.

3.2.2. Motores JavaScript

De acordo com a Mozilla Developer Network (2012), JavaScript é uma linguagem licenciada pela empresa Oracle e apesar do nome, é diferente da linguagem Java, também de propriedade da Oracle. Também conhecida como “JS”, é uma linguagem leve, interpretável, orientada a objetos e muito utilizada na *web* para programação de conteúdo dinâmico mas também está disponível em outros ambientes fora da *web*.

É um padrão *Ecma International*, sendo que todos os navegadores lançados a partir de 2012 suportam sua 5ª edição, o ECMAScript 5.1. Navegadores mais antigos suportam pelo menos a 3ª edição do padrão e uma 6ª revisão está em andamento.

Enquanto o HTML é utilizado para estruturar o conteúdo de uma página *web* e o CSS codifica os estilos e como a página deve ser apresentada, JavaScript é utilizado para criar efeitos ou *Rich Internet Applications* através da manipulação do DOM.

JavaScript também pode ser definido como um dialeto do ECMAScript, o que significa que algumas interpretações deste dialeto podem não ser diretamente compatíveis com outras. A tabela abaixo mostra algumas das principais implementações do ECMAScript no mercado:

Tabela 1 - Implementação do EcmaScript por navegador

Aplicação e Motor	Dialeto e versão	Edição do ECMAScript
Google Chrome / V8	JavaScript	ECMA-262, 5ª edição
Mozilla Firefox / SpiderMonkey	JavaScript 1.8.5	ECMA-262, 5ª edição
Safari / Nitro	JavaScript	ECMA-262, 5ª edição, revisão 1
Opera / Carakan e V8	JavaScript	ECMA-262, 5ª edição
Internet Explorer / Chakra	JavaScript	ECMA-262, 5ª edição

Fonte: Próprio autor

3.2.3. RIAs em AJAX

AJAX é o acrônimo da língua inglesa para *Asynchronous Javascript and XML*. Trata-se de uma metodologia que utiliza JavaScript e XML ou outros formatos de arquivos para troca de dados de maneira assíncrona dentro de uma página *web*, isto é, sem que o usuário precise fazer um carregamento total de uma página para outra para carregar conteúdo ou processar dados (ULLMAN; DYKES, 2007). O termo foi cunhado e difundido em 2005 por Jessé James Garret e foi rapidamente adotado por outros desenvolvedores e empresas.

Garret (2006) afirma que AJAX não é apenas uma tecnologia, mas várias tecnologias ou um conjunto de técnicas de programação que envolve:

- HTML e CSS para camada de apresentação;
- Apresentação dinâmica e interação através da manipulação do DOM;
- Intercâmbio e manipulação de dados utilizando XML, XSLT ou JSON;
- Recuperação de dados assíncronos através do objeto XMLHttpRequest;
- JavaScript para integrar todas estas tecnologias e fazê-las funcionar;

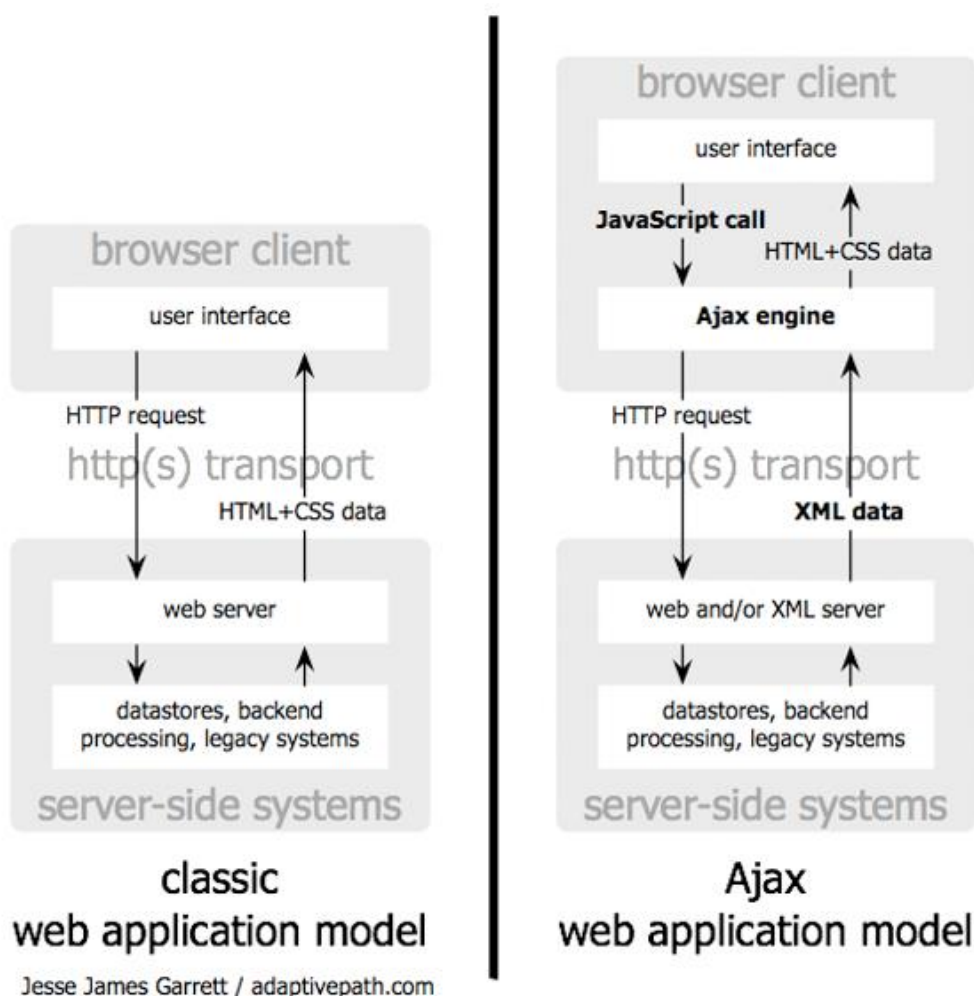
O modelo clássico de aplicação *web* é definido por Garret (2006) da seguinte forma: Ao interagir com algum elemento de uma dada interface, o usuário ativa um requisição através do protocolo HTTP para o servidor *web*. O servidor processa os dados e retorna como resposta uma página HTML para o cliente.

Este é o mesmo modelo que Allaire (2002) afirmara que se tornara bem sucedido até o começo da década de 2000 mas que perdera espaço para as *Rich Internet Applications*. Na data a afirmação de que as aplicações *web* necessitavam de um *runtime* para execução do código fazia sentido pois, a maioria das aplicações *web* desenvolvidas dependiam de um *plugin* além do navegador para seu funcionamento, tal como Java ou Flash, dada a dificuldade de se manter uma aplicação *web* estável e que funcionasse em todos os navegadores. As RIA e AJAX se diferem das aplicações *web* anteriores a estas abordagens na interação do usuário. Garret compara e explica:

[...] Essa abordagem faz muito sentido técnico, mas não faz tanto para uma boa experiência de usuário. Enquanto o servidor está fazendo a sua coisa, o que o usuário está fazendo? É isso mesmo, esperando. E a cada etapa em uma tarefa, o usuário espera mais um pouco.

Obviamente, se nós estivéssemos projetando a *Web* a partir do zero para aplicações, não faríamos os usuários esperarem. Uma vez que a interface está carregada, por que a interação com o usuário precisa pausar cada vez que a aplicação precisa de algo do servidor? Na verdade, por que o usuário precisa ver a aplicação se comunicar com o servidor em tudo? (GARRET, 2006)

Figura 14 – Comparação do modelo de aplicações web tradicionais do modelo de aplicações web em AJAX



Fonte: GARRET, 2006

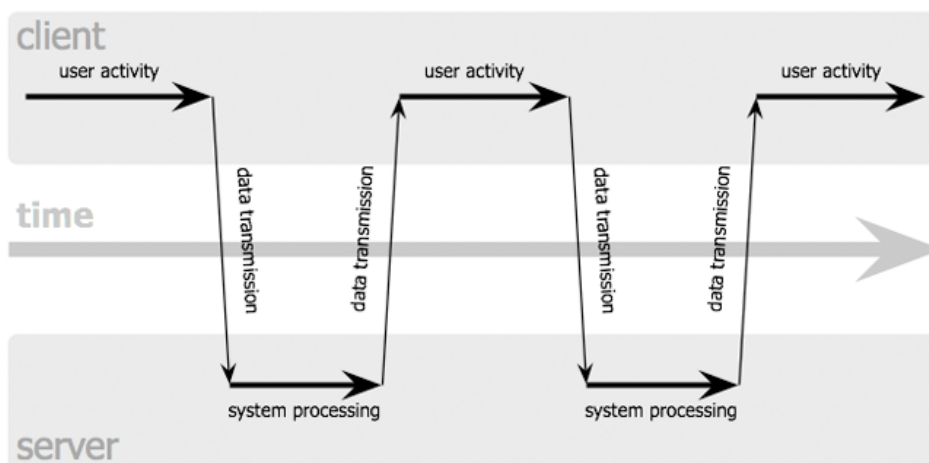
O AJAX é diferente das aplicações *web* tradicionais porque traz uma nova abordagem à interação com a *Web* por intermédio de um motor ou camada entre o usuário e o servidor. Ao carregar uma página *web* pela primeira vez, o navegador vai carregar um motor AJAX escrito em JavaScript que fica escondido do usuário.

Este motor pode ser responsável pela composição da tela que o usuário interage, o que não é obrigatório, mas é ele quem realiza a comunicação com o servidor no lugar do usuário. Este processo permite que o usuário interaja com a aplicação de forma assíncrona, independente da comunicação com o servidor. A metodologia pode ser utilizada não somente para uma aplicação *web* completa, mas também em pequenas partes dela que requeiram um carregamento assíncrono.

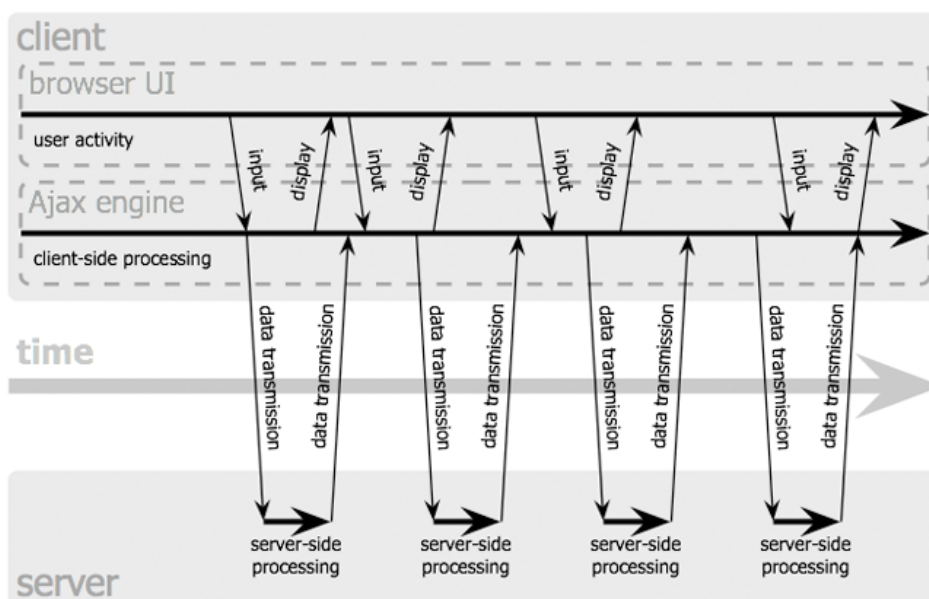
Na figura abaixo, Garret compara o fluxo de trabalho de um usuário em uma aplicação *web* tradicional e em uma aplicação *web* em AJAX:

Figura 15 – O padrão de interação síncrona de uma aplicação *web* tradicional (em cima), em comparação com o padrão assíncrono de uma aplicação Ajax (parte inferior).

classic web application model (synchronous)



Ajax web application model (asynchronous)



Jesse James Garrett / adaptivepath.com

Fonte: GARRET, 2006

As interações do usuário que normalmente seriam feitas através de uma requisição HTTP são trocadas por uma chamada AJAX através do objeto XMLHttpRequest (ULLMAN; DYKES, 2007). O JavaScript faz as requisições de maneira assíncrona sem obstruir o usuário, recebe os dados processados em formato

XML, JSON ou outros e manipula o DOM atualizando a interface com os dados recebidos em resposta do servidor.

Conforme fora citado anteriormente, a manipulação da árvore DOM implica em sérios problemas para usuários de ferramentas assistivas. No AJAX por conta do carregamento ser assíncrono, o usuário de um software leitor de telas não conseguiria saber qual área da página *web* está sendo atualizada e precisaria percorrer pelos elementos da página desde o começo até encontrar a informação desejada.

3.2.4. Frameworks JavaScript

Programação avançada com JavaScript pode ser muito difícil e consumir muito tempo, principalmente trabalhando com os complexos e diferentes motores de navegadores e realizando testes em diferentes cenários. Visando sanar estas dificuldades, vários programadores desenvolveram diferentes bibliotecas JavaScript para a comunidade. Estas bibliotecas também costumam ser chamadas de *frameworks* JavaScript.

Algumas bibliotecas tem funções ou propósitos bem específicos mas as mais usadas trazem métodos comuns ao JavaScript nativo para animações, manipulação do DOM e carregamento assíncrono com uma sintaxe mais fácil e já preparada para lidar com as dificuldades de se manter uma aplicação *cross-browser*.

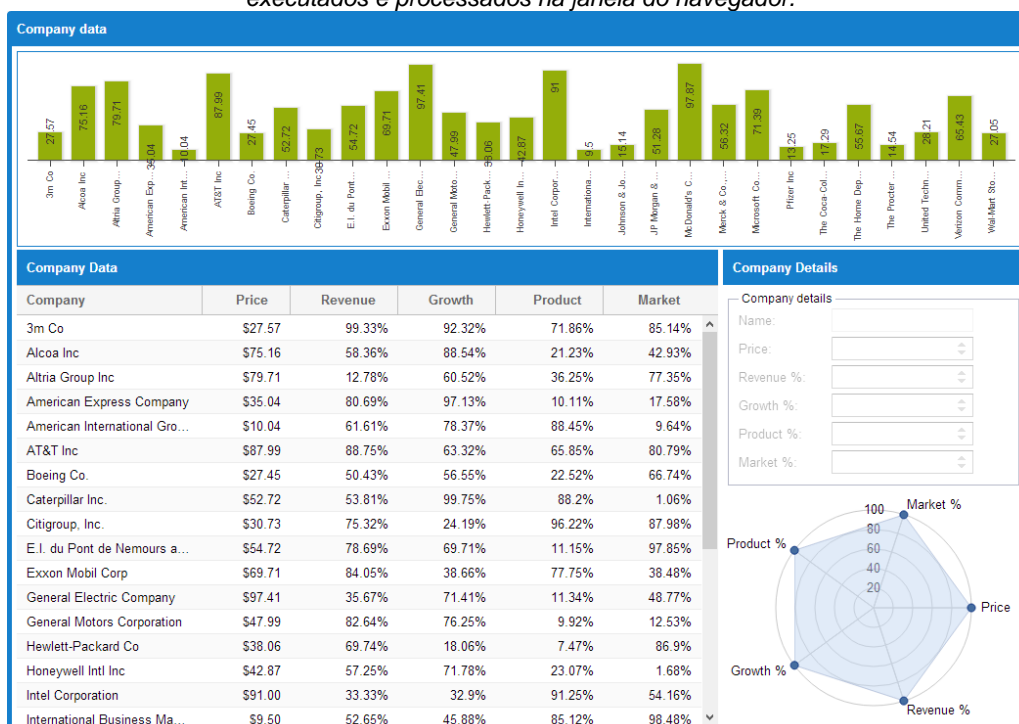
Segundo O W3Schools ([s.d.]), jQuery é a *framework* JavaScript mais popular na *web*. A biblioteca utiliza seletores CSS para acessar e manipular elementos no DOM, tendo uma sintaxe limpa e de fácil entendimento até mesmo para aqueles que não tem conhecimento avançado de programação JavaScript (JQUERY, [s.d.]).

Além de oferecer suporte a vários navegadores, também é extensível através de *plugins* desenvolvidos pelo próprio time de desenvolvimento da biblioteca ou por terceiros. A *framework* é utilizada pelas maiores empresas de tecnologia, serviços e entretenimento, tais como Google, Microsoft, IBM e Netflix (W3CSCHOOLS, [s.d.]). Seu código é aberto e a biblioteca é distribuída pela licença MIT.

Outro exemplo de bibliotecas JavaScript é o Sencha Ext JS. De acordo com seu desenvolvedor (SENCHA, [s. d.]), é a plataforma de desenvolvimento para aplicações *web* mais poderosa disponível no mercado, com centenas de componentes, várias APIs e documentação completa. Enquanto o jQuery oferece apenas métodos para controle de eventos, animações, manipulação de dados e objetos no DOM, o Ext JS traz componentes completos para desenvolvimento de aplicações, tais como gráficos, janelas modais, árvores de dados, tabelas com filtros e ordenação entre outros. Em contrapartida, sua licença é *open-source* apenas para desenvolvimento de aplicações dentro da licença GNU e também de código aberto. Para aplicações comerciais, o preço da biblioteca por desenvolvedor é de \$595,00 (SENCHA, [s. d.]).

Com relação a acessibilidade, o Sencha Ext JS segue as diretrizes do WAI para *Rich Internet Applications*, o WAI-ARIA, que será abordado no capítulo seguinte.

Figura 16 – Exemplo de aplicação web que utiliza a framework Sencha Ext JS. Os gráficos e tabelas são executados e processados na janela do navegador.



Fonte: SENCHA, s. d.

4. ARIA E RECOMENDAÇÕES PARA CONTEÚDOS DINÂMICOS

4.1. Accessible Rich Internet Applications

Conforme visto nos capítulos anteriores, uma página *web* comum é bem distinta de uma aplicação *web*. As aplicações ricas que antes precisavam de *plugins* para executá-las agora encontram no JavaScript e AJAX uma linguagem e bibliotecas confiáveis mudando a forma de interagir com o usuário utilizando de padrões abertos e tecnologias sólidas como HTML e CSS.

Esta mudança melhorou drasticamente a capacidade de resposta e usabilidade na *web*, porém sua técnica envolve muitos riscos e potenciais falhas à acessibilidade. Segundo a Mozilla Developer Network (2012), muitas bibliotecas JavaScript oferecem componentes que imitam comportamentos de interfaces bem conhecidas no *desktop*. Barras de rolagem, barras de menu, *sliders*, árvores de arquivos, abas e muitos outros componentes podem ser construídos com JavaScript, CSS e HTML.

Embora visualmente estes componentes sejam muito parecidos com componentes de aplicações *desktop*, eles não carregam uma informação semântica em sua estrutura que possa ser lida pelas ferramentas assistivas, além de modificar a árvore DOM de uma maneira que estas ferramentas não conseguem lidar. Partindo deste problema, o W3C criou a especificação WAI-ARIA, acrônimo para *Accessible Rich Internet Applications*.

Segundo o WebAIM (2013), os problemas com acessibilidade em *Rich Internet Applications* podem ser categorizados em:

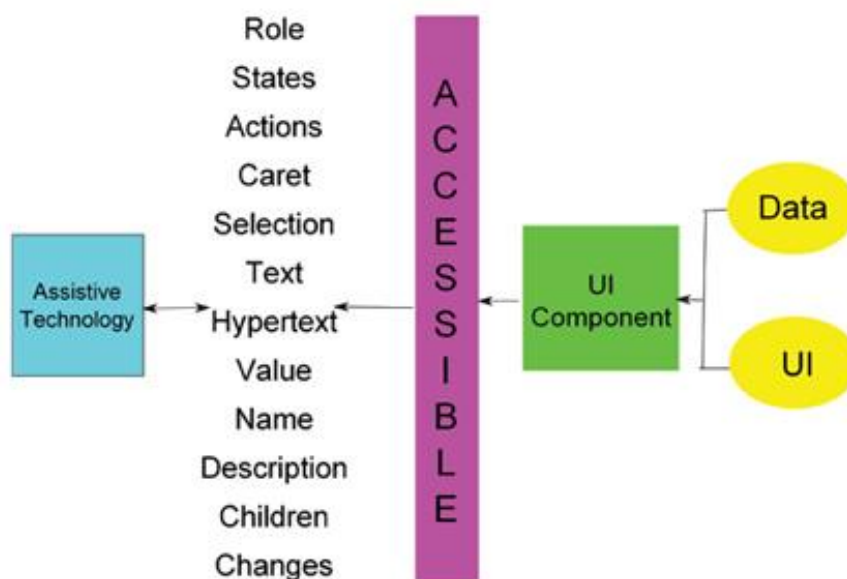
- Incapacidade de fornecer a estrutura semântica de áreas e funcionalidades da página (navegação, conteúdo principal, pesquisa, etc.);
- Inacessibilidade de conteúdo que é dinâmico e pode mudar dentro da página (atualizações de conteúdo por AJAX);

- Incapacidade de mudar o foco do teclado para elementos da página (por exemplo, definir o foco para uma mensagem de erro dentro da página ou para um campo de formulário);
- Dificuldade com teclado e acessibilidade do leitor de tela com componentes complexos e elementos de navegação (barras de navegação, árvores, listas, etc.);

Schwerdtfegger e Gibson ([s.d.]) definem o WAI-ARIA como uma *framework* para desenvolvimento de controles avançados de interface de usuário com HTML, AJAX, JavaScript e tecnologias relacionadas. Essa *framework* contém um conjunto de atributos que podem ser aplicados no HTML e mapeados pela API de acessibilidade dos navegadores.

Os autores afirmam que o WAI-ARIA atua como uma “ponte” entre as ferramentas assistivas e a aplicação para habilitar o acesso ao conteúdo rico com a informação semântica necessária para tornar o conteúdo acessível. Enquanto o WCAG traz as diretrizes para o conteúdo, o WAI-ARIA atribui significados às estruturas das aplicações *web*.

Figura 17 – A relação das ferramentas assistivas com o WAI-ARIA e as aplicações *web*.



Fonte: SCHWERDTFEGGER, R., GIBSON, B, s. d.

A especificação WAI-ARIA é subdividida em três tipos diferentes de atributos: *roles*, *states* e *properties*. *Roles* descrevem os componentes que não são

nativos ao HTML, como por exemplo, *sliders*, abas e janelas modais. *Properties* descrevem as características destes componentes, se eles são arrastáveis, precisam de outro elemento, tem algum pop-up associado a eles, etc. *States* descrevem o estado atual da interação com o elemento, informando a ferramenta assistiva se o elemento está ocupado, desabilitado, selecionado ou escondido (MOZILLA DEVELOPER NETWORK, 2012).

Os atributos ARIA foram projetados para serem automaticamente interpretados pelos navegadores e traduzidos para a API de acessibilidade nativa do sistema operacional. Quando uma aplicação *web* é marcada com ARIA em sua estrutura, as tecnologias assistivas são capazes de detectar, reconhecer e interagir com controles JavaScript da mesma forma que conseguiriam com equivalentes do *desktop*. Isto tem potencial para trazer uma experiência de usuário muito mais consistente do que era possível nas aplicações *web* da geração passada.

Tabela 2 – Suporte ao WAI-ARIA em ferramentas assistivas

Ferramenta Assistiva	Versão com suporte a ARIA	Versão com suporte a livre-regions
NVDA	2010.2	2011.1 apenas em Mozilla Firefox
Orca	Não suporta	Não suporta
VoiceOver	OS X 10.5 e iOS 4	OS X 10.7 e iOS 5
JAWS	8	10
Window Eyes	7	Não suporta
ZoomText	?	Não suporta

Fonte: MOZILLA DEVELOPER NETWORK, s. d.

Infelizmente nenhuma das ferramentas assistivas desenvolvidas no Brasil suporta o WAI-ARIA pois suas últimas atualizações foram lançadas antes de que a especificação ficasse pronta. Outras ferramentas assistivas ainda apresentam suporte superficial em forma de teste a ARIA (MOZILLA DEVELOPER NETWORK, 2012).

4.2. Landmark Roles

Nas especificações atuais do HTML e XHTML, não há mecanismos que identifiquem a função ou propósito de um determinado elemento em uma página ou aplicação de maneira que o computador consiga interpretar. Mesmo provendo uma estrutura correta do HTML seguindo as diretrizes do W3C, isso não necessariamente

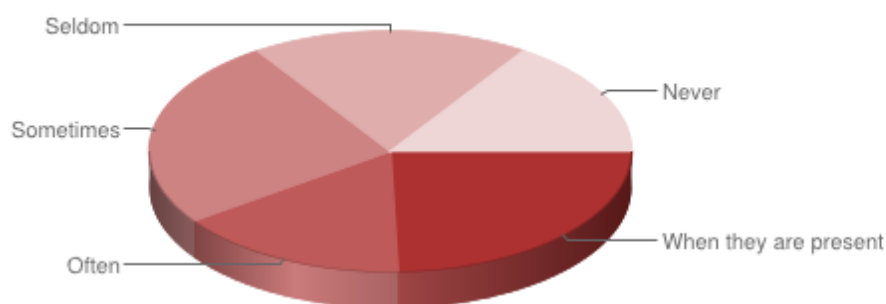
quer dizer que um *software* leitor de tela irá diferenciar uma área ou outra de uma página ou aplicação *web*.

O WAI-ARIA dá aos desenvolvedores o poder de determinar “papéis” para certas áreas ou componentes (WEBAIM, 2013). São chamados de *landmark roles* os atributos que ajudam a definir a estrutura de um documento. São eles:

- Article - Conteúdo que faça sentido no contexto do documento.
- Banner - Título, nome, logotipo do sistema.
- Complementary - Conteúdo de suporte para o conteúdo principal
- Contentinfo - Notas de rodapé, direitos autorais, links para declaração de privacidade, preferências, referências, etc.
- Main - O conteúdo principal ou central.
- Navigation - A área que contém a navegação.
- Search - Área que contém a busca ou pesquisa do sistema.

Segundo o WebAIM (2012), os usuários das ferramentas assistivas conhecem os *landmark roles* do WAI-ARIA, mas só os usam quando estão disponíveis:

Figura 18 – Uso de landmarks ARIA por usuários de leitores de telas
Usage of ARIA landmarks



Fonte: WEBAIM, 2012

Tabela 3 – Uso de *landmarks* ARIA por leitores de tela

Resposta	Nº de respondentes	% de respondentes
Quando presentes	408	24.6%
Frequentemente	262	15.8%
Algumas vezes	423	25.5%
De vez em quando	307	18.5%
Nunca	259	15.6%

Fonte: WEBAIM, 2012

4.3. Live Regions e Aplicações em JavaScript

Conforme já afirmado nos capítulos anteriores, leitores de tela e ferramentas assistivas tem dificuldades em “perceber” alterações no DOM de um documento. Faulkner (2008) explica que os leitores de tela fazem na verdade uma cópia do DOM da página *web* sendo exibida pelo navegador e a armazenada em um “*buffer* virtual”. É nessa cópia que o usuário de leitor de tela interage quando está navegando, acessando os elementos que demarcam as estruturas do documento através de atalhos no teclado do leitor de tela.

Este modo de navegação é conhecido como “*browser mode*” ou modo de navegação. Quando o usuário do leitor de tela está interagindo com campos de formulário, ele interage diretamente com o navegador e o *software* leitor de tela apenas lê os atributos destes campos e as teclas digitadas. Este modo é conhecido como “*forms mode*” ou modo de formulário.

Segundo Faulkner (2008), no modo de navegação, quando o usuário pode navegar através da tecla *TAB* ou listar os componentes de estrutura da página através de atalhos, uma das grandes limitações das versões do JAWS anteriores a 7.1 e a todas as versões do Windows Eyes é que a cópia do DOM é estática até que o usuário pressione um botão ou interaja com algum elemento da página.

Se ocorrer qualquer atualização assíncrona em algum componente da página, o leitor de tela não irá atualizar a lista de elementos gerada a partir do *buffer* virtual a não ser que o usuário atualize manualmente a cópia do DOM através de um atalho ou interaja com algum botão, link ou elemento com o evento *onclick* atribuído.

Visando sanar este problema, os desenvolvedores destes leitores de tela implementaram um comando que atualiza a cópia virtual do DOM em resposta a eventos JavaScript ao apertar as teclas *Enter* ou Barra de Espaço. Superficialmente esta parecia uma solução razoável mas que apresentava várias falhas, a começar por exigir que o usuário avise o leitor de tela que o *buffer* virtual precisa ser carregado e no caso de aplicações em AJAX, dado o tempo variável das chamadas a servidor,

eventualmente ocorreria uma desconexão entre o conteúdo que o usuário tem acesso e o conteúdo que é exibido pelo navegador.

Dado este problema, o WAI-ARIA trouxe em sua especificação as *live regions*. Com a marcação deste atributo, os desenvolvedores de aplicações *web* conseguem fazer com que componentes com carregamento assíncrono sejam acessíveis a leitores de tela e também permite ao desenvolvedor escolher se quer alertar o usuário, definir controles para a região, determinar a quantidade de conteúdo que será lida entre outras possibilidades.

Para definir uma *live region*, é necessário acrescentar a propriedade *aria-live* para o componente no HTML. Esta propriedade pode receber os atributos *off*, *polite* ou *assertive*. Estes valores determinam a ação do leitor de tela quando o elemento é atualizado:

- O valor *off* dirá para o leitor de tela não anunciar a atualização;
- Com o valor *polite*, o usuário será notificado sobre a mudança de conteúdo assim que ele finalizar sua tarefa atual. Esta notificação poderá vir na forma de um alerta sonoro ou uma indicação de áudio de atualização. O usuário então opta por pular para o conteúdo atualizado ou continuar onde está. Este valor é mais comum a notificações de status, progresso, atualização de valores ou mensagens de chat;
- O valor *assertive* alerta o usuário sobre mudanças de conteúdo imediatamente após ser atualizado. Este valor deve ser utilizado para atualizações importantes como por exemplo, mensagens de erro ou feedback para as ações do usuário;

O desenvolvedor também pode definir o conteúdo que deverá ser lido quando ocorrer uma atualização. Existem também *roles* especiais que definem certos tipos de conteúdo altamente dinâmicos, tais como alertas, logs e contadores. O alto nível de fidelidade com *live regions* permite uma grande flexibilidade tanto para desenvolvedores quanto para usuários finais (WEBAIM, 2013).

4.4. ARIA e Funcionamento com Leitores de Telas

Muitos desenvolvedores interpretam erroneamente que o ARIA modifica o comportamento ou a apresentação dos componentes para os usuários que não utilizam de ferramentas assistivas. A especificação fora desenvolvida para inserir uma semântica no código HTML que fosse útil às ferramentas assistivas. Acrescentar um *role* com valor de “checkbox” a uma *tag* “div” em uma página *web* não produzirá efeito nenhum ou transformará este elemento em uma caixa de seleção, mas o leitor de tela dirá ao usuário que ali se encontra um controle de caixa de seleção (FREEDOM SCIENTIFIC, [s.d.]). Cabe ao desenvolvedor atribuir um comportamento ao elemento que corresponda ao *role* descrito no componente.

Outra concepção errada, segundo publicação da empresa Freedom Scientific ([s.d.]), é que a interpretação marcação ARIA depende apenas da ferramenta assistiva. Os navegadores também desempenham um grande papel na acessibilidade. A maioria deles fornece uma API de acessibilidade e as ferramentas assistivas buscam nesta API a informação que é apresentada na tela. Isso significa que as ferramentas assistivas dependem totalmente do suporte dos navegadores à especificação ARIA para que estes controles funcionem.

Dos atributos ARIA listados no diagrama presente no “ANEXO A” deste trabalho (W3C, [s.d.]), são suportados pelo software leitor de telas JAWS em sua última versão todos atributos listados no quadro do “APÊNDICE A”:

5. CONSIDERAÇÕES FINAIS

Esta pesquisa buscou compilar questões relacionadas ao desenvolvimento de aplicações *web* e como os deficientes visuais utilizam a *Web*. Conforme visto nos capítulos anteriores, existem vários recursos livres, de padrões internacionais e abertos que desenvolvedores podem utilizar para garantir a acessibilidade e interação em *Rich Internet Applications*.

Com relação as diretrizes de conteúdo do WCAG, é importante que não só desenvolvedores procurem conhecê-las, mas também analistas, testadores e implementadores. Todas as áreas de estudo e vertentes que atuam no desenvolvimento da *Web* deveriam se preocupar com a acessibilidade, uma vez que os usuários portadores de necessidades especiais são muitos no Brasil e cada vez mais serviços que são oferecidos no mundo real também estão sendo ofertados *online*.

As recomendações do Governo Federal (BRASIL, 2004) podem não ser suficientes para portais e sistemas de administração pública, uma vez que não há qualquer menção ao uso de WAI-ARIA para componentes avançados nos programas de acessibilidade brasileiros. É também imprescindível o desenvolvimento de alternativas baratas e nacionais de ferramentas assistivas, uma vez que os *softwares* com licenças de uso comerciais não tem distribuição local e são caros, enquanto os *softwares* nacionais estão caminhando para obsolescência limitando o desenvolvimento de aplicações *web* que utilizem tecnologias modernas para uma experiência de usuário rica tanto para portadores de necessidades especiais quando para usuários comuns.

Faltam centros de treinamento, cursos, livros e palestras relacionados ao tema em português, o que afasta a comunidade acadêmica e também desenvolvedores do assunto.

Por último, apesar das análises do WebAIM (2012) abrangerem até mesmo usuários da América do Sul, faltam pesquisas relacionadas a acessibilidade na *web*

no Brasil, bem como uma análise mais desenvolvida sobre a *web* brasileira na perspectiva dos usuários portadores de necessidades especiais.

REFERÊNCIAS

ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos - apresentação**: NBR 9050. Rio de Janeiro, 2004. Disponível em: <http://www.pessoacomdeficiencia.gov.br/app/sites/default/files/arquivos/%5Bfield_generico_imagens-filefield-description%5D_24.pdf>. Acesso em: 30 jul. 2013.

ALLAIRE, J. **Macromedia Flash MX – A Next Generation rich client**. 2002. Disponível em: <<http://download.macromedia.com/pub/flash/whitepapers/richclient.pdf>>. Acesso em: 10 set. 2013.

ALMEIDA, L. B., DE PAULA, L. G. **O retrato da exclusão digital na sociedade brasileira**. Revista de gestão da tecnologia e sistemas de informação. São Paulo, ano 2005, v. 2, p.55-67. Disponível em: <<http://www.scielo.br/pdf/jistm/v2n1/05.pdf>>. Acesso em: 10 ago. 2013.

APPLE. **Apple – Acessibilidade – OS X – VoiceOver**. 2013. Disponível em <<http://www.apple.com/br/accessibility/osx/voiceover/>>. Acesso em: 11 ago. 2013.

BERNERS-LEE, T. **World Wide Web Consortium Launches International Program Office for Web Accessibility Initiative**. Disponível em: <<http://www.w3.org/Press/IPO-announce>>. Acesso em: 10 ago. 2013.

BRASIL. Lei n.º 7.853, de 24 de outubro de 1989. Dispõe sobre o apoio às pessoas portadoras de deficiência, sua integração social, sobre a Coordenadoria Nacional para Integração da Pessoa Portadora de Deficiência - Corde, institui a tutela jurisdicional de interesses coletivos ou difusos dessas pessoas, disciplina a atuação do Ministério Público, define crimes, e dá outras providências. **Diário Oficial da União**, Brasília, DF, 25 out. 1989.

BRASIL. Lei n.º 8.213, de 24 de julho de 1991. Dispõe sobre os Planos de Benefícios da Previdência Social e dá outras providências. **Diário Oficial da União**, Brasília, DF, 25 jul. 1991.

BRASIL. Decreto n.º 5.296, de 2 de dezembro de 2004. Regulamenta as Leis nos 10.048, de 8 de novembro de 2000, que dá prioridade de atendimento às pessoas que especifica, e 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas portadoras de deficiência ou com mobilidade reduzida, e dá outras providências. **Diário Oficial da União**, Brasília, DF, 3 dez. 2004.

BRASIL. Ministério das Comunicações. **Leitor de Telas** – Ministério das Comunicações. s.d. Disponível em <<http://www.mc.gov.br/acoes-e-programas/redes-digitais-da-cidadania/170-sem-categoria/22727-leitor-de-telas>>. Acesso em: 10 ago. 2013.

CAMARGO FILHO, S. F. M., BICA, F. **Acessibilidade digital para cegos: Um modelo de interface para utilização do mouse**. Porto Alegre, 2008. Disponível em: <http://sbie2008.virtual.ufc.br/CD_ROM_COMPLETO/sbie_artigos_completo/Acessibilidade%20digital%20para%20cegos%20-%20Um%20modelo%20de%20interface.pdf>. Acesso em: 10 ago. 2013.

CLARK, J. **WCAG Samurai**. 2006. Disponível em: <<http://www.wcagsamurai.org/>>. Acesso em: 10 ago. 2013.

CLARK, J. **To Hell with WCAG 2**. 2006. Disponível em: <<http://alistapart.com/article/tohellwithwcag2>>. Acesso em: 10 ago. 2013.

CHROMEVOX. **ChromeVox**. 2013. Disponível em <<http://www.chromevox.com/>>. Acesso em: 10 ago. 2013.

CONDE, A. J. M. **Instituto Benjamin Constant: Definindo a Cegueira e a Visão Subnormal**. Disponível em: <<http://www.ibr.gov.br/?itemid=94>>. Acesso em: 10 ago. 2013.

DARDAILLER, D. **WAI History: WAI early days**. 2009. Disponível em: <<http://www.w3.org/WAI/history>>. Acesso em: 27 jul. 2013.

DEV.OPERA. **Traversing the DOM**. 2009. Disponível em <<http://dev.opera.com/articles/view/traversing-the-dom/>>. Acesso em: 10 set. 2013.

ESTABEL, L. B.; MORO, E. L. da S.; SANTAROSA, L. M. C. **A inclusão social e digital de pessoas com limitação visual e o uso das tecnologias de informação e de comunicação na produção de páginas para a Internet**. 2006. Disponível em <<http://www.scielo.br/pdf/ci/v35n1/v35n1a10.pdf>>. Acesso em: 30 jul. 2013.

FAULKNER, S. **AJAX and Screen Readers** – Content Access Issues. 2008. Disponível em <<http://blog.paciellogroup.com/2008/02/ajax-and-screen-readers-content-access-issues/>>. Acesso em: 11 set. 2013.

FRANKLIN, A. **O que é DOM** – Document Object Model. 2011. Disponível em <<http://tableless.com.br/tenha-o-dom/>>. Acesso em: 10 ago. 2013.

FREEDOM SCIENTIFIC. **Products for low vision, blindness, and learning disabilities from Freedom Scientific**. 2013. Disponível em <<http://www.freedomscientific.com/>>. Acesso em: 10 ago. 2013.

FREEDOM SCIENTIFIC. **JAWS ARIA Support**. s. d. Disponível em <<http://www.freedomscientific.com/PDF/visionloss/manuals/DevDoc/JAWS-ARIA-Support.doc>>. Acesso em: 10 ago. 2013.

GARRET, J. J. **Ajax: A New Approach to Web Applications**. 2006. Disponível em: <<http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/>>. Acesso em: 23 out. 2013.

HENRY, S. L., ARCH, A. **Developing a Web Accessibility Business Case for Your Organization: Overview**. 2012. Disponível em: <<http://www.w3.org/WAI/bcase/Overview>>. Acesso em: 30 jul. 2013.

HENRY, S. L., MCGEE, L. **ACCESSIBILITY**. s.d. Disponível em: <<http://www.w3.org/standards/webdesign/accessibility>>. Acesso em: 30 jul. 2013.

HYATT, D. **WebCore Rendering I** – The Basics. 2007. Disponível em <<https://www.webkit.org/blog/114/webcore-rendering-i-the-basics/>>. Acesso em: 10 ago. 2013.

IBGE. Instituto Brasileiro de Geografia e Estatística. **Censo Demográfico - 2000 - Resultados da Amostra**. 2002. Disponível em: <<http://www.ibge.gov.br/home/>>

estatistica/populacao/censo2000/primeiros_resultados_amostra/brasil/pdf/tabela_1_1_3.pdf> Acesso em 27 ago. 2013.

ISO. International Organization for Standardization. **Information processing -- Text and office systems -- Standard Generalized Markup Language (SGML)**. ISO 8879:1986. Disponível em <http://www.iso.org/iso/catalogue_detail.htm?csnumber=16387>. Acesso em: 10 set. 2013.

JACOBS, Ian. **Sobre o Consórcio W3C**. 2008. Disponível em: <<http://w3c.br/sobre/#history-head>> Acesso em: 15 jul. 2013.

JQUERY. **jQuery**. s. d. Disponível em <<http://jquery.com/>>. Acesso em: 3 out. 2013.

LONGMAN. A. W. **Chapter 2**. A history of HTML. 1998. Disponível em <<http://www.w3.org/People/Raggett/book4/ch02.html>>. Acesso em: 10 set. 2013.

MICROPOWER VIRTUAL VISION. **Virtual Vision**. 2011. Disponível em < http://www.micropower.com.br/v4/tecnologia_virtualvision.html >. Acesso em: 10 ago. 2013.

MOZILLA DEVELOPER NETWORK. **Document Object Model (DOM)**. 2013. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/DOM>>. Acesso em: 10 set. 2013.

MOZILLA DEVELOPER NETWORK. **JavaScript**. 2012. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>>. Acesso em: 12 out. 2013.

MOZILLA DEVELOPER NETWORK. **JavaScript Technologies Overview**. 2012. Disponível em: < https://developer.mozilla.org/en-US/docs/Web/JavaScript/JavaScript_technologies_overview?redirectlocale=en-US&redirectslug=JavaScript%2FJavaScript_technologies_overview>. Acesso em: 12 out. 2013.

MOZILLA DEVELOPER NETWORK. **Web applications and ARIA FAQ**. 2012. Disponível em: <https://developer.mozilla.org/en-US/docs/Accessibility/ARIA/Web_applications_and_ARIA_FAQ>. Acesso em: 12 out. 2013.

NATIONS, Daniel. **Web Applications: What is a Web Application?** 2012. Disponível em: <http://webtrends.about.com/od/webapplications/a/web_application.htm>. Acesso em: 10 set. 2013.

NIELSEN, Jakob. **Projetando Websites**. Rio de Janeiro, Campus, 2000.

NVACCESS. **NV Access**. 2013 Disponível em: <<http://www.nvaccess.org/>>. Acesso em: 15 ago. 2013.

ORGANIZAÇÃO DAS NAÇÕES UNIDAS. **Convention on the Rights of Persons with Disabilities**. 2008. Disponível em: <<http://www.un.org/disabilities/default.asp?navid=12&pid=150>>. Acesso em: 30 ago. 2013.

ORGANIZAÇÃO MUNDIAL DE SAÚDE. **Visual impairment and blindness**. OMS. 2013. Disponível em: <<http://www.who.int/mediacentre/factsheets/fs282/en/>>. Acesso em: 12 out. 2013.

PROJETO DOSVOX. Projeto Dosvox. 2002. Disponível em: <<http://intervox.nce.ufrj.br/dosvox/>>. Acesso em: 10 ago. 2013.

PUPO, D. T., MELO, A. M., PÉREZ FERRÉS, S. **Acessibilidade: discurso e prática no cotidiano das bibliotecas**. São Paulo, SP: UNICAMP, 2008.

QUEIROZ, M. A. **Deficiência Visual: a cegueira e a baixa visão**. Disponível em <<http://www.bengalalegal.com/cegueira-e-baixa-visao>>. Acesso em: 10 ago. 2013.

RAPOZA, J. RIA War Is Brewing. 2008. Disponível em: <http://www.eweek.com/reviews/ria_war_is_brewing/>. Acesso em: 15 set. 2013.

SARTORETTO, M. L., BERSCH, R. **TECNOLOGIA ASSISTIVA**. O que é Tecnologia Assistiva. 2013. Disponível em <<http://www.assistiva.com.br/tassistiva.html>>. Acesso em: 20 ago. 2013.

SCHWERDTFEGGER, R., GIBSON, B. **An introduction to using WAI-ARIA to enable Web 2.0 accessibility**. s. d. Disponível em: <http://www-03.ibm.com/able/resources/wai_aria_intro.html>. Acesso em 24 out. 2013.

SENCHA. **JavaScript Framework for Building Rich Desktop Web Applications**. s. d. Disponível em: <<http://www.sencha.com/products/extjs/>>. Acesso em: 3 out. 2013

SONZA, A. P. **Ambientes virtuais acessíveis sob a perspectiva de usuários com limitação visual**. Porto Alegre. 2008. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/14661/000666392.pdf?sequence=1>>. Acesso em: 15 jul. 2013.

THATCHER, J. **Web accessibility: Web Standards and Regulatory Compliance**. Berkeley: Friends of Apress, 2006.

TONET, Luisa Hayder. **Pesquisa das ferramentas de acessibilidade computacional para deficientes visuais e as recomendações do W3C**. Guaíba. Ulbra. 2006. Disponível em <<http://guaiba.ulbra.br/seminario/eventos/2006/artigos/sistemas/161.pdf>>. Acesso em: 15 jul. 2013.

ULLMAN, C., DYKE, L. **What is Ajax?**. 2007. Disponível em: <<http://www.wrox.com/WileyCDA/Section/id-303217.html>>. Acesso em: 23 out. 2013.

W3C. **Current Members** - W3C. 2013. Disponível em: <<http://www.w3.org/Consortium/Member/List>>. Acesso em: 20 ago. 2013.

W3C. **HTML 5**. A vocabulary and associated APIs for HTML and XHTML. 2013. Disponível em <<http://www.w3.org/TR/html5/introduction.html>>. Acesso em: 7 set. 2013.

W3C. **W3C Brasil** – World Wide Web Consortium Escritório Brasil - Sobre. 2013. Disponível em: <<http://www.w3c.br/Sobre>>. Acesso em: 20 ago. 2013.

W3C. **WAI-ARIA Taxonomy**. s. d. Disponível em: <http://www.w3.org/TR/wai-aria/rdf_model>. Acesso em: 23 out. 2013.

W3C. **Web Content Accessibility Guidelines (WCAG) 2.0 aprovado como norma internacional ISO/IEC**. 2012. Disponível em: <<http://www.w3c.br/>

Noticias/WebContentAccessibilityGuidelineswcag20AprovadoComoNormalInternacio
nallsoiec>. Acesso em: 15 jul. 2013.

W3CSCHOOLS. **JavaScript Libraries**. JavaScript Frameworks (Libraries). 2013.
Disponível em <http://www.w3schools.com/js/js_libraries.asp>. Acesso em: 3 out.
2013.

WAI. **Web Accessibility Initiative (WAI)**. 2011. Disponível em <<http://www.w3.org/WAI/>>. Acesso em: 15 jul. 2013.

WCAG. **W3C Web Standard Defines Accessibility for Next Generation Web**.
2008. Disponível em <<http://www.w3.org/2008/12/wcag20-pressrelease.html>>.
Acesso em: 12 ago. 2013.

WCAG. **Web Content Accessibility Guidelines (WCAG) Overview**. 2012.
Disponível em <<http://www.w3.org/WAI/intro/wcag.php>>. Acesso em: 10 ago. 2013.

WEBAIM. **Accessibility of Rich Internet Applications**. 2013. Disponível em:
<<http://webaim.org/techniques/aria/>>. Acesso em 23 out. 2013.

WEBAIM. **Screen Reader User Survey #4**. 2012. Disponível em: <<http://webaim.org/projects/screenreadersurvey4>>. Acesso em: 12 ago. 2013.

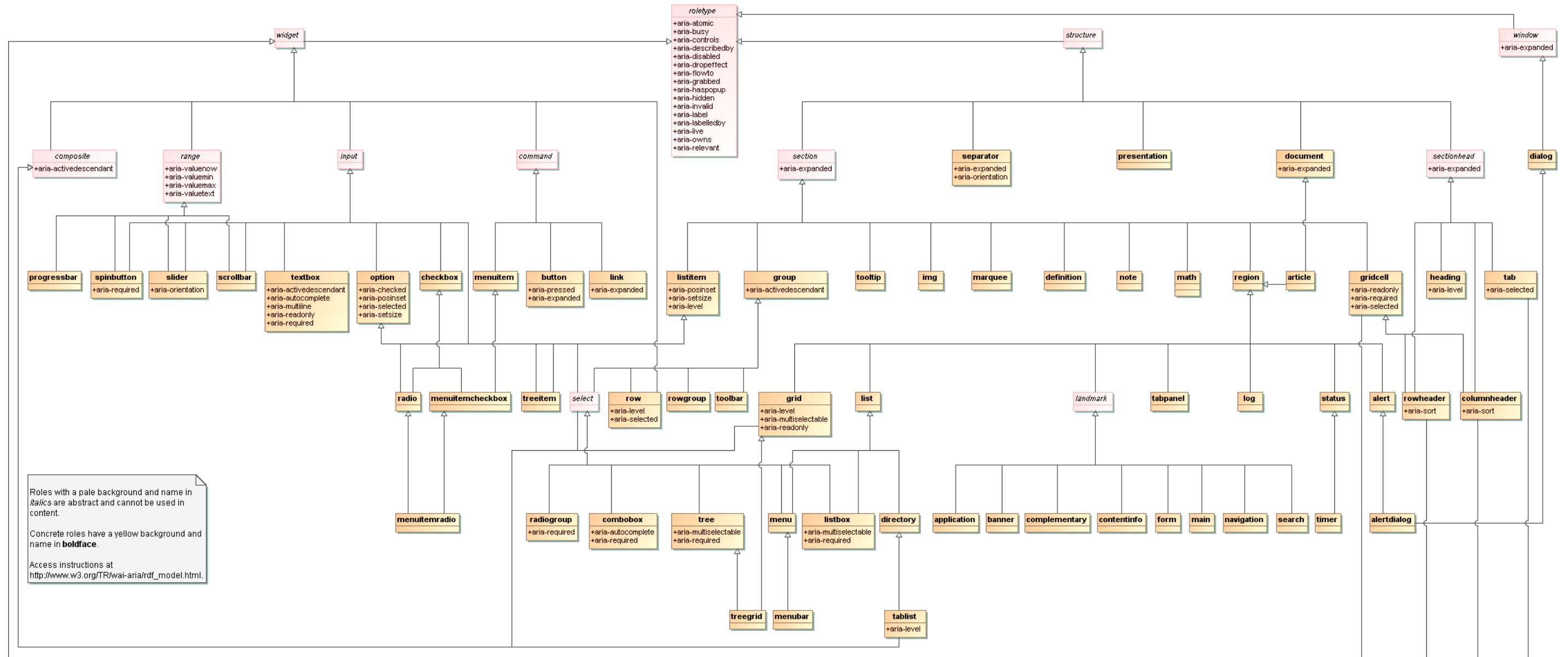
WEBAIM. **Web Accessibility in Mind**. s.d. Disponível em: <<http://webaim.org/articles/>>. Acesso em: 12 ago. 2013.

APÊNCIDE A – ESPECIFICAÇÕES WAI-ARIA SUPORTADAS PELO JAWS

Roles	Regions	States
alert	Role="application"	aria-activedescendant
alertdialog	Role="article"	aria-busy
button	Role="banner"	aria-checked
checkbox	Role="complementary"	aria-describedby
columnheader	Role="contentinfo"	aria-disabled
combobox	Role="document"	aria-expanded
dialog	Role="form"	aria-haspopup
document	Role="main"	aria-hidden
Grid	Role="navigation"	aria-invalid
gridcell	Role="region"	aria-label
group	Role="search"	aria-level
heading		aria-multiline
img		aria-orientation
link		aria-posinset
list		aria-pressed
listbox		aria-readonly
listitem		aria-required
menu		aria-selected
log		aria-setsize
menubar		aria-valuetext
menuitem		
menuitemcheckbox		
menuitemradio		
Note		
option		
presentation		
progressbar		
radio		
radiogroup		
row		
rowheader		
separator		
Scrollbar		
slider		
spinbutton		
tab		
tablist		
tabpanel		
textbox		
toolbar		
tooltip		
tree		
treegrid		
treeitem		

Fonte: FREEDOM SCIENTIFIC, s. d.

ANEXO A – Diagrama RDF da Especificação WAI-ARIA



Fonte: W3C, s. d.