

Capítulo

1

Introdução à Acessibilidade na Web: do Conceito à Prática

Renan Soares Germano, Maria Amelia Eliseo e Ismar Frango Silveira

Abstract

The constant presence of Internet use in our lives, when related to the high number of people with disabilities in society, makes accessibility an indispensable feature for contemporary digital systems. In addition to the social motivator that drives developers to make applications accessible, other factors can also be mentioned: financial - users with disabilities represent a consumer group that can generate profit through accessible digital systems that allow the consumption of products and services by all; pre-established requirements - there are cases where the implementation of a minimum level of accessibility is a mandatory requirement, such as in government applications. This chapter presents the current scenario of web accessibility and provides a series of resources to help create accessible web pages, improving the current scenario.

Resumo

A constante presença do uso da Internet em nossas vidas, quando relacionada com o alto número de pessoas com deficiência na sociedade, faz com que a acessibilidade seja uma característica indispensável para os sistemas digitais contemporâneos. Além do motivador social, que impele os desenvolvedores a tornarem as aplicações acessíveis, outros fatores também podem ser mencionados: financeiro - os usuários com deficiência representam um grupo consumidor que pode gerar lucro através de sistemas digitais acessíveis que permitem o consumo de produtos e serviços por todos; requisitos pré-estabelecidos - há casos em que a implementação de um nível mínimo de acessibilidade é um requisito obrigatório, como em aplicações governamentais. O presente capítulo apresenta o cenário atual da acessibilidade na web, e disponibiliza uma série de recursos para auxiliar na criação de páginas web acessíveis, melhorando o cenário atual.

1.1. Introdução

Estimativas mostram que a quantidade de pessoas vivendo com deficiência no mundo hoje é muito grande, chegando à casa dos bilhões. World Health Organization (2011) estima que cerca de 15% da população do mundo, ou 1 bilhão de pessoas, vive com algum tipo de deficiência. Enquanto isso, Return on Disability (2020) estima cerca de 1,85 bilhões de pessoas vivendo com algum tipo de deficiência, o que representa uma população maior do que a da China.

Esses números por si só já deixam claro a importância de suprir as necessidades desse grupo. Neste caso, é necessário prover meios alternativos e ferramentas auxiliares para que as pessoas com deficiência possam realizar, de maneira equivalente, as mesmas tarefas que qualquer outra pessoa. A partir disso, surge o conceito de acessibilidade.

Outro dado importante na atualidade é a quantidade de pessoas que utilizam a Internet no mundo. Como mostra a Tabela 1.1, pouco mais de 5,1 bilhões de pessoas possuem acesso à Internet hoje, o que representa cerca de 65% da população de todo o planeta. Relacionando esse fato aos dados apresentados no início da seção, fica claro a importância da acessibilidade na *web*: é a partir dela que é garantido para as pessoas com deficiência o direito de acesso às informações disponibilizadas na Internet.

Tabela 1.1. Quantidade estimada de usuários da Internet no mundo por região [Internet World Stats 2021].

Região	População	Usuário da Internet	Porcentagem da População
Ásia	4.327.333.821	2.762.187.516	63,8 %
Europa	835.817.920	736.995.638	88,2 %
África	1.373.486.514	594.008.009	43,2 %
América Latina / Caribe	659.743.522	498.437.116	75,6 %
América do Norte	370.322.393	347.916.627	93,9 %
Médio Oriente	265.587.661	198.850.130	74,9 %
Oceania / Austrália	43.473.756	30.385.571	69,9 %
Total	7.875.765.587	5.168.780.607	65,6 %

O cenário pandêmico atual e seus reflexos no aumento da utilização da Internet [We are Social 2021], e consequentes desafios advindos deste acontecimento no que se refere ao suporte à acessibilidade nas páginas dos *web sites*, também não poderia deixar de ser mencionado. Tal cenário mais uma vez reforça a importância da Internet e dos *web sites* nas vidas dos cidadãos contemporâneos, bem como da acessibilidade na *web*.

Ilustrando esses desafios, Fernández-Díaz et al. (2020) demonstram que há dificuldade de acesso a informações de saúde pública sobre a pandemia, divulgadas pela Organização Mundial da Saúde (OMS) em seu *web site*, para certos grupos de pessoas

com deficiência. Gleason et al. (2020) também apontam inacessibilidade das informações sobre a pandemia divulgadas pelo setor de saúde pública na *web*, apontando ainda que pessoas com deficiência também têm enfrentado dificuldades de acesso à produtos e serviços online, devido à falta de acessibilidade. Russ e Hamidi (2021) apontam a falta de conhecimento e conscientização por parte dos professores sobre a criação de conteúdos educacionais acessíveis, listando uma série de conselhos, baseados em literaturas anteriores, que devem ser seguidos para que o cenário presente melhore. E esses são apenas alguns poucos exemplos de estudos atuais que abordam o assunto.

Um dos recursos auxiliares, utilizados pelas pessoas com deficiência para utilizarem os *web sites* de maneira equivalente à uma pessoa sem deficiência, são as Tecnologias Assistivas (TAs). Elas podem ser programas instalados no computador do usuário ou um dispositivo físico que permite a interação entre o usuário e o computador. Alguns exemplos são listados em [W3C 2017c]

Apesar das TAs serem um importante recurso para os usuários com deficiência, pois melhoram a qualidade de vida deles uma vez que traz mais independência quando utilizando os *web sites*, somente elas não resolvem o problema. Para que elas funcionem adequadamente, as páginas devem ser implementadas pensando nessas tecnologias [W3C 2019c]. Por esse motivo, é responsabilidade principalmente dos desenvolvedores *web* promoverem o suporte à acessibilidade através da criação de páginas acessíveis.

Mesmo possuindo diversos recursos disponíveis para o desenvolvimento de *web sites* acessíveis, como as diretrizes internacionais de acessibilidade na *web* [W3C 2015a, W3C 2015b, W3C 2018b], especificações técnicas [W3C 2017a], validadores de código [W3C 2013, W3C 2019a] e avaliadores automáticos de acessibilidade [W3C 2016b], a literatura mostra que o nível de acessibilidade dos *web sites* ainda é baixo [Máñez-Carvajal et al. 2019, Martins et al. 2017, Gonçalves et al. 2021]. Mais do que isso, nem mesmo a existência de leis em alguns países contribui para a melhoria do cenário corrente [Barricelli et al. 2018, Carvajal et al. 2018, Kimmons 2017].

Dessa maneira, fica claro que o problema da falta de acessibilidade dos *web sites* não é um problema causado pela falta de tecnologia para sua implementação, mas sim pela falta de utilização de todos os recursos que existem hoje. Tal problema é causado pela falta de conscientização e motivação por parte dos responsáveis, e pela falta de conhecimentos técnicos para implementação [Cao et al. 2019, Inal et al. 2019, Ferati et al. 2020, Antonelli et al. 2018].

O artigo 9 da Convenção Sobre os Direitos das Pessoas com Deficiência, realizada pela Organização das Nações Unidas, em 2016, diz que cabe aos Estados Partes “(...) proporcionarem, a todos os atores envolvidos, formação em relação às questões de acessibilidade com as quais as pessoas com deficiência se confrontam (...)” [United Nations 2006]. Assim, o conteúdo deste capítulo contribui para formação de profissionais capacitados a lidarem com as questões de acessibilidade na *web*.

Neste contexto, o objetivo é conscientizar o leitor sobre a importância da acessibilidade, e o motivar para que leve a acessibilidade na *web* em consideração em seus projetos, incentivando-o a desenvolver *web sites* acessíveis. Para tanto, assume-se uma abordagem educacional, com a apresentação de fatos relevantes e demonstrações técnicas básicas e fundamentais para a criação de páginas *web* acessíveis. É importante destacar que não há o intuito de se esgotar todos os tópicos sobre acessibilidade na *web*, e que para isso mais tempo e outras atividades (práticas, inclusive) são necessárias.

O capítulo está organizado em quatro seções. Introdução (1) (esta seção) apresenta o cenário atual, a motivação e os objetivos. Referencial teórico (2) apresenta os trabalhos correlatos, além dos fatos e conceitos importantes sobre acessibilidade na *web*. A seção “Apresentação de páginas *web* com suporte à acessibilidade” (3) apresenta as demonstrações técnicas básicas e fundamentais para a criação de páginas *web* acessíveis. E por fim a conclusão (4) encerra com as considerações que os autores julgam importantes.

1.2. Referencial Teórico

1.2.1. Trabalhos correlatos

Na literatura pode-se encontrar relatos de experiência sobre a aplicação dos conceitos de acessibilidade em cursos de desenvolvimento *web* na graduação. Wang (2012) apresenta uma experiência realizada no curso de graduação em *Web Design* na disciplina “Multimídia e *Web Design*” e detalha tópicos e técnicas apropriados para o escopo do curso, bem como as estratégias de avaliação. Apesar dos tópicos apresentados terem se tornado antigos, mantém sua utilidade quando observados os seus conceitos de forma atualizada.

Nesta mesma linha, El-Glaly (2020) relata o processo de desenvolvimento de um curso de pós-graduação em acessibilidade, oferecido como disciplina opcional para estudantes de Engenharia de Software. E, a partir dos experimentos realizados, a autora conclui que a acessibilidade é um tópico importante que deve ser totalmente abordado na educação em Computação.

Em relação ao ensino de acessibilidade em cursos de graduação, Ferati e Vogel (2020) estudaram o conteúdo programático de catorze diferentes cursos relacionados ao desenvolvimento *web* de uma universidade sueca e, além disso, fizeram pesquisas e entrevistas com alunos e professores desta universidade, para avaliar como este tema era ensinado. Os resultados mostraram que a acessibilidade não é coberta nos cursos, embora tanto alunos quanto professores demonstraram ciência da importância do tema e concordam que deveria ser abordado nos cursos. Estes resultados também mostraram que, entre os desenvolvedores *web*, há pouca familiaridade com as diretrizes e políticas de acessibilidade, sendo um alerta para a necessidade do conhecimento deste tema entre estes profissionais, bem como sua inclusão nos cursos que tratam de desenvolvimento *web*.

Os achados de El-Glaly (2020) e Ferati e Vogel (2020) sugerem a necessidade de suporte ao tema acessibilidade, tanto para profissionais de desenvolvimento *web*

quanto para os educadores em acessibilidade. Assim, este capítulo busca transmitir e expandir alguns conceitos de acessibilidade, suas diretrizes e mostrar formas práticas de tratar o tema no desenvolvimento *web*. A literatura também mostra que é importante motivar os desenvolvedores *web* para que eles tenham interesse no tema, e que a motivação se dá principalmente ao trazer a questão para o mais próximo da vida real (deles e dos usuários das aplicações desenvolvidas por eles) e ao ensinar como introduzi-la tecnicamente nos *web sites*. Por isso, este capítulo traz conteúdos baseados nesses dois pilares.

1.2.2. Por que Desenvolver Web Sites Acessíveis?

Desenvolver *web sites* acessíveis é importante porque a acessibilidade é um direito humano. A Convenção Sobre os Direitos das Pessoas com Deficiência, em seu artigo 9 (Acessibilidade), enfatiza a importância de promover a acessibilidade nos meios digitais ao atestar que os Estados Partes devem assegurar às pessoas com deficiência o acesso, de maneira equivalente a qualquer outra pessoa, às informações, comunicações e outros serviços, incluindo em meios eletrônicos [United Nations 2006]. A acessibilidade como direito humano e fator indispensável para o desenvolvimento de uma sociedade igualitária e justa é amplamente discutida em [United Nations 2013a]. E, tal tópico também é abordado nos 17 objetivos de desenvolvimento sustentável para 2030 [United Nations 2013b].

Portanto, criar *web sites* acessíveis é importante porque as pessoas que precisam dos recursos de acessibilidade também podem ter acesso às informações, produtos, serviços e oportunidades disponibilizadas através deles. Dessa maneira, os desenvolvedores, *designers* e outros profissionais envolvidos e responsáveis no processo de criação desses *web sites* contribuem para o desenvolvimento de uma sociedade igualitária, na qual todos possuem as mesmas oportunidades, podem participar de maneira independente de todas as atividades da vida social e têm assegurado um direito fundamental humano.

Além de entender conceitualmente a acessibilidade na *web* e sua importância, também é importante ter em mente cenários reais nos quais ela se reflete. Alguns exemplos práticos que demonstram a importância da acessibilidade na *web* e da sua consequente contribuição para o desenvolvimento de uma sociedade justa e igualitária podem ser: redes sociais para comunicação, expressão e interação com outras pessoas; *web sites* de acesso à diferentes tipos de informações (noticiários, instruções de prevenção à saúde, como no caso da pandemia, etc.); *web sites* para agendamento de algum serviço (serviços públicos como emissão de segunda via de algum documento, por exemplo); *web sites* com recursos educacionais; e *web sites* com vagas de empregos ou sobre concursos públicos. Se qualquer um desses *web sites* forem inacessíveis, as pessoas que necessitam dos recursos de acessibilidade estarão em desvantagem com relação às pessoas que não necessitam desses recursos.

Outro fator que pode ser visto como um motivador para que a acessibilidade seja sempre levada em consideração quando criando *web sites* é que não somente as pessoas com deficiência se beneficiam dela. Os recursos de acessibilidade desenvolvidos ao

longo dos anos vêm se mostrando úteis para os usuários em geral, à exemplo das legendas, que além contribuir principalmente para o entendimento dos conteúdos multimídia por pessoas com deficiência auditiva, também podem ser úteis em um contexto no qual não é possível utilizar o áudio, ou mesmo quando se quer aprender outro idioma [W3C 2019c].

Além disso, o desenvolvimento de alguma deficiência ao longo da vida é uma condição humana, o que implica dizer que em algum momento todas as pessoas precisarão utilizar algum recurso de acessibilidade devido à alguma deficiência desenvolvida em decorrência do natural desgaste do corpo ao longo dos anos. Esse é um dos principais motivos apontados pela Organização Mundial da Saúde para a estimativa de que o número de pessoas com deficiência no mundo continua a crescer [World Health Organization 2011]. Portanto, caso o exercício de se colocar no lugar do usuário que precisa dos recursos de acessibilidade ainda não seja suficiente para motivar os desenvolvedores a se engajarem com a acessibilidade na *web*, talvez pensar que eles mesmos podem ser os futuros usuários com necessidades de acessibilidade de suas aplicações possa ajudar.

Em alguns casos, o desenvolvimento de *web sites* acessíveis é importante para que eles estejam de acordo com a lei vigente no país no qual for desenvolvido. Diversos países institucionalizaram a acessibilidade na *web* para os *web sites* desenvolvidos em seus territórios, sendo necessário garantir um nível mínimo de acessibilidade para que eles estejam de acordo com a lei vigente [W3C 2018a]. Para que esse nível mínimo de acessibilidade seja seguido é necessário estar de acordo com as diretrizes de acessibilidade disponibilizadas por cada lei (normalmente, as diretrizes de acessibilidade são baseadas nos padrões internacionais de acessibilidade, amplamente utilizados).

Por fim, o último motivo pelo qual os autores acreditam ser importante o desenvolvimento de *web sites* com acessibilidade é o financeiro. Sim, desenvolver *web sites* com acessibilidade pode gerar lucro. Um cenário muito simples é o de um *web site* que vende produtos e/ou serviços online: quanto maior for o número de usuários maior será a possibilidade de compras e consequente lucro. Portanto, é lógico afirmar que *web sites* acessíveis têm maior chances de gerarem lucros maiores do que *web sites* não acessíveis.

Como mencionado na introdução, é estimado que a quantidade de pessoas com deficiência no mundo é maior do que a população da China. Esse grupo, acrescido de seus amigos e familiares, podem chegar a cerca de 3,3 bilhões de pessoas, e juntos movimentam anualmente cerca 13 trilhões de dólares em rendimento disponível, deixando claro a importância do fator financeiro para a consideração da acessibilidade na *web* [Return on Disability 2020].

1.2.3. Conceitos Importantes

O primeiro conceito importante a se entender é a acessibilidade. Algumas definições são apresentadas a seguir.

1. Acessibilidade pode ser definida como um mecanismo social através do qual as pessoas com deficiência adquirem mais independência e qualidade de vida, podendo locomover-se livremente pelos meios físicos, expressar-se livremente e ter acesso à informação [United Nations 2013a].
2. Acessibilidade pode ser definida como o processo de adequação dos produtos, serviços, meios físicos e digitais para que possam ser utilizados por pessoas com diferentes tipos de deficiência [United Nations 2013a].
3. Acessibilidade poder ser definida como uma característica dos produtos, serviços, meios físicos e digitais que foram adaptados de modo que usuários com diferentes necessidades e preferências conseguem se utilizar deles [United Nations 2013a].

Fortemente conectado ao conceito de acessibilidade está o conceito de deficiência. A United Nations (2013a) define a deficiência como resultado da falta de acessibilidade. Ou seja, a deficiência não é vista como uma característica dos indivíduos, mas sim como o resultado da interação deles com elementos que impõem barreiras físicas e comportamentais, impedindo-os de participarem de maneira íntegra e efetiva em todas as atividades da sociedade.

Da definição anterior, destaca-se aqui as barreiras atitudinais. Estas resultam do comportamento humano com relação às pessoas com deficiência. Alguns exemplos podem ser o preconceito, a intolerância, criação de estereótipos e afins. Nesse sentido, pode-se considerar que a falta de conscientização (e até mesmo de interesse) sobre a importância e implementação da acessibilidade na *web* também se caracteriza como uma dessas barreiras atitudinais, contribuindo para o aparecimento da deficiência nos meios digitais.

A World Wide Web Consortium (W3C) é a organização internacional responsável pela criação e manutenção dos padrões da *web*. Seu objetivo é desenvolvê-la até que atinja o seu potencial máximo: torná-la acessível para todos [W3C 2021c]. Nesse sentido, a acessibilidade tem um importante papel ao eliminar as barreiras de utilização dos *web sites* para as pessoas com deficiência. Nesse contexto, o *Web Accessibility Initiative* (WAI) é um grupo da W3C focado na criação de tecnologias e diretrizes voltadas para o desenvolvimento da acessibilidade na *web* [W3C 2021d].

O objetivo da acessibilidade na *web* é tornar os *web sites* acessíveis para os usuários finais. Entretanto, também é importante garantir que outras tecnologias, envolvidas na criação e utilização dos *web sites*, sejam acessíveis. As outras duas principais tecnologias que devem ser acessíveis são as ferramentas de autoria (como editores de textos) para a codificação das páginas, e os agentes de usuário (navegadores e players de vídeo, por exemplo) para navegação na *web* [W3C 2019c].

Portanto, diretrizes de acessibilidade foram criadas para tornar as diferentes tecnologias da *web* e seus conteúdos acessíveis: *User Agent Accessibility Guidelines* (UAAG) [W3C 2015b], *Authoring Tool Accessibility Guidelines* (ATAG) [W3C 2015a] e a *Web Content Accessibility Guidelines* (WCAG) [W3C 2018b]. A UAAG deve ser utilizadas para criação de agentes de usuários acessíveis. A ATAG deve ser utilizada

para criação de ferramentas de autoria acessíveis. E a WCAG deve ser utilizada para criação de conteúdos acessíveis, como os textos, imagens, áudios e vídeos.

Enquanto a UAAG e a ATAG estão voltadas para a criação de aplicações cujos usuários finais são mais restritos, a WCAG está voltada para criação de conteúdos que serão acessados pelo público em geral. Por isso, a WCAG é o conjunto de diretrizes de acessibilidade mais difundido, sendo utilizada principalmente por desenvolvedores *web* e *web designers* (os criadores dos conteúdos dos *web sites*).

A primeira versão da WCAG (1.0) foi publicada em 1999 [W3C 1999]. A última versão publicada até a escrita deste capítulo é a 2.1 [W3C 2018b]. A versão 2.2 [W3C 2021e] será publicada em breve. O desenvolvimento da versão 3.0 [W3C 2021a] está em progresso, mas deve levar alguns anos até que seja concluída. As versões mais atuais não depreciam as anteriores, mas a W3C encoraja a utilização da versão mais atual, pois ela traz melhorias levantadas nas versões anteriores.

A WCAG é um documento não técnico (mas há recursos complementares que abordam detalhes de implementação). Ela está dividida em quatro princípios (perceptível, operável, entendível e robusto) e 13 diretrizes. Cada diretriz possui um conjunto de critérios de sucesso, cada um com um nível de prioridade - A, AA e AAA, do mais prioritário ao menos prioritário, respectivamente. Através dela é possível definir o nível de acessibilidade dos *web sites* em A, AA e AAA – os *web sites* recebem a categoria correspondente se todos os critérios de sucesso do respectivo nível, e dos níveis anteriores, forem atendidos [W3C 2021f].

1.3. Implementação de Páginas Web com Suporte à Acessibilidade

A construção das páginas de um *web site* se dá a partir de três linguagens principais: HTML, CSS e JavaScript. O HTML (*Hypertext Markup Language*) é utilizado para estruturar o conteúdo da página. O CSS (*Cascading Style Sheets*) é utilizado para formatar a aparência da página. E o JavaScript é utilizado para implementar elementos dinâmicos na página [W3C 2017a, W3C 2021b, ECMA 2021].

As próximas subseções apresentam dicas e exemplos práticos de implementações utilizando as linguagens para a construção das páginas *web* e outros recursos que contribuem para implementação do suporte à acessibilidade na fase de desenvolvimento. É importante destacar que os pontos apresentados não esgotam a lista de cuidados que devem ser tomados para que as páginas criadas sejam acessíveis, e foram escolhidos de acordo com a visão dos autores sobre a relevância de se falar um pouco mais detalhadamente sobre cada um deles. Os códigos das figuras das seções a seguir podem ser encontrados no link <https://github.com/rsg73626/html-and-css-accessibility-examples>.

1.3.1. O papel do HTML e do CSS no Suporte à Acessibilidade das Páginas Web

O HTML é composto por tags, marcações utilizadas para definir as diferentes seções de uma página. Existem diferentes tags para diferentes seções, de acordo com sua funcionalidade na página. É a partir das tags que é criado o Document Object Model (DOM) das páginas, uma estrutura em árvore contendo todos os conteúdos de uma

página. A partir dessa estrutura a árvore de acessibilidade é criada uma versão simplificada do DOM original. Ela é importante porque é a partir da árvore de acessibilidade que as TAs apresentam as informações das páginas para os usuários.

Em sua quinta versão, novas *tags* semânticas foram introduzidas ao HTML. A utilização delas auxilia a acessibilidade porque possuem semântica de maneira nativa, isto é, elas carregam em si mesmas o seu significado, que pode ser interpretado automaticamente pelas TAs através da árvore de acessibilidade das páginas. Isso torna possível transmitir para o usuário o conteúdo correto de cada informação, além de auxiliá-lo em como interagir com ela [W3C 2017a]. O layout apresentado na Figura 1.1 pode ser implementado por meio dos dois códigos HTML exibidos na Figura 1.2, um sem a utilização das *tags* semânticas do HTML 5 e outro com a utilização das *tags* semânticas do HTML5.

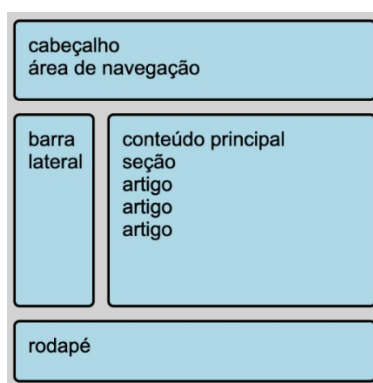


Figura 1.1. Exemplo de *layout* para uma página da web

<pre> <div class="header"> cabeçalho <div class="nav"> área de navegação </div> </div> <div class="aside"> barra lateral </div> <div class="main"> conteúdo principal <div class="section"> seção <div class="article"> artigo </div> <div class="article"> artigo </div> <div class="article"> artigo </div> </div> </div> <div class="footer"> rodapé </div> </pre>	<pre> <header> cabeçalho <nav> área de navegação </nav> </header> <aside> barra lateral </aside> <main> conteúdo principal <section> seção <article> artigo </article> <article> artigo </article> <article> artigo </article> </section> </main> <footer> rodapé </footer> </pre>
---	--

Figura 1.2. Implementações de *layout* para uma página da web. No quadro da esquerda, implementação sem uso das *tags* semânticas do HTML5. No quadro da direita, implementação com o uso das *tags* semânticas do HTML5.

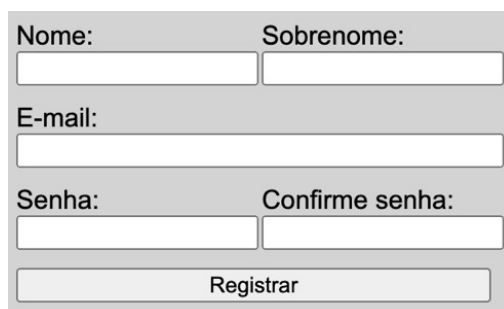
A *tag* *div*, apresentada na Figura 1.2, é uma marcação genérica e deve ser usada para agrupar outros elementos somente quando não houver outra *tag* semântica disponível para realizar o agrupamento [W3C 2017a]. As *tags* semânticas na segunda implementação da Figura 1.2 permitem que os usuários de TAs naveguem mais rapidamente entre as páginas através da identificação das seções (área de navegação e conteúdo principal, por exemplo). Além disso, pode-se dizer que o código da versão

com as *tags* semânticas ficou mais fácil de se entender e mais legível, pois não possui a configuração de classes para cada *tag*.

Quando criando as páginas é importante respeitar a responsabilidade de cada uma das linguagens utilizadas. Evitar a utilização de *tags* aninhadas para o posicionamento adequado dos elementos na tela de acordo com o *layout* definido, e deixar essa responsabilidade para o CSS, evita problemas de navegação quando utilizando o teclado ou TAs para a navegação. Isso ocorre porque a navegação com o teclado e as TAs não se baseia no conteúdo visual renderizado pelos navegadores nos monitores dos usuários, mas no conteúdo textual e estrutural dos documentos HTML.

As Figuras 1.4 e 1.5 mostram diferentes implementações para o formulário mostrando na Figura 1.3. Enquanto na primeira implementação (Figura 1.4) as *tags div* são utilizadas somente para definir uma estrutura em grade para ajudar a posicionar os elementos da página, a segunda implementação (Figura 1.5) separa as responsabilidades, estruturando os elementos no documento HTML com o uso das *tags* semânticas adequadas, e utilizando o recurso de *grid* do CSS para posicionar os elementos da maneira que foi definida no *layout* da Figura 1.3.

Além de contribuir para evitar problemas de navegação para os usuários que navegam pelas páginas *web* com o uso de teclado ou de alguma tecnologia assistiva, novamente também é possível dizer que o código HTML da Figura 1.5 ficou mais legível. Outro ponto positivo na separação das responsabilidades é que se caso seja necessário mudar o *layout* definido, somente o CSS precisará ser alterado.



Nome: Sobrenome:

E-mail:

Senha: Confirme senha:

Figura 1.3. Exemplo de formulário com *layout* com múltiplas linhas e colunas.

```
.grid { width: 300px; height: 200px; }
.row { width: 100%; height: 25%; }
.col { width: 50%; height: 100%; float: left; }
input { width: calc(100% - 10px); margin-right: 10px; }

<form method="POST" action="/register">
  <div class="grid">
    <div class="row">
      <div class="col">
        <label for="name"> Nome: </label> <input type="text" id="name"/>
      </div>
      <div class="col">
        <label for="sname"> Sobrenome: </label> <input type="text" id="sname"/>
      </div>
    </div>
    <div class="row">
      <label for="email"> E-mail: </label> <input type="email" id="email"/>
    </div>
    <div class="row">
      <div class="col">
        <label for="psw"> Senha: </label> <input type="password" id="psw"/>
      </div>
      <div class="col">
        <label for="cpsw"> Confirme senha: </label> <input type="password" id="cpsw"/>
      </div>
    </div>
    <div class="row">
      <input type="submit" value="Registrar"/>
    </div>
  </div>
</form>
```

Figura 1.4. Exemplo de implementação de formulário com *layout* com múltiplas linhas e colunas com o uso de *tags* aninhadas. No quadro superior, o código CSS, no quadro inferior, o código HTML.

Outro recurso de acessibilidade, específico para formulários, é a associação de uma tag label para cada uma das tags input. No código da Figura 1.4, isso é feito através da propriedade for da tag label (o valor desse atributo deve ser igual ao valor da propriedade id da tag input ao qual o label deve ser associado). No código da Figura 1.5, isso é feito envolvendo a tag input por uma tag label. Ambas as implementações permitem que, ao encontrar um input, as TAs transmitam adequadamente sua descrição para os usuários. Ao clicar no texto do label, o input correspondente é acionado para que o usuário insira os dados, o que também contribui para a acessibilidade, pois pode facilitar o preenchimento do formulário para usuários com dificuldades de utilização dos recursos de apontamento (mouse, por exemplo).

```

form {
  width: 300px; height: 200px;
  display: grid;
  grid-template-areas: 'name sname' 'email email' 'psw cpsw' 'submit submit';
}
input { width: 100%; }
#name { grid-area: name; padding: 5px; }
#sname { grid-area: sname; padding: 5px; }
#email { grid-area: email; padding: 5px; }
#psw { grid-area: psw; padding: 5px; }
#cpsw { grid-area: cpsw; padding: 5px; }
#submit { grid-area: submit; padding: 5px; }

```

```

<form method="POST" action="/register">
  <label id="name"> Nome: <input type="text"/> </label>
  <label id="sname"> Sobrenome: <input type="text"/> </label>
  <label id="email"> E-mail: <input type="email"/> </label>
  <label id="psw"> Senha: <input type="password"/> </label>
  <label id="cpsw"> Confirme senha: <input type="password"/> </label>
  <input type="submit" value="Registrar" id="submit"/>
</form>

```

Figura 1.5. Exemplo de implementação de formulário com *layout* com múltiplas linhas e colunas sem o uso de *tags* aninhadas. No quadro superior, o código CSS, no quadro inferior, o código HTML.

Por ser responsável pela formatação visual das páginas, o CSS possui uma grande influência na acessibilidade para deficientes visuais. É através dele que é possível controlar os tamanhos dos textos, botões e *link*, as cores, posicionamentos e contraste dos elementos. A seguir, dois fatores importantes sobre a implementação do código CSS e a sua influência na acessibilidade das páginas *web* são discutidos. O primeiro, um ponto de atenção que pode passar despercebido pelos desenvolvedores quando implementados os *layouts* com CSS. E o segundo, um recurso útil da linguagem para a implementação de diferentes esquemas de cores para a acessibilidade (importante para que os usuários possam utilizar os *web sites* com as cores cujo contraste é o mais adequado para elas, e para usuários com daltonismo também [W3C 2019b]).

Através do CSS é possível manipular a ordem com que os elementos são exibidos na tela para os usuários. No entanto, é importante que a ordem dos elementos na tela seja coerente com a ordem definida no código HTML, caso contrário, usuários que utilizam o teclado para navegar nas páginas, ou alguma tecnologia assistiva, podem ficar confusos quando navegando nos *web sites*. A ordem utilizada para se navegar por uma página através do teclado ou com uma tecnologia assistiva é a mesma definida no código HTML.

A Figura 1.6 apresenta a implementação de uma lista de itens com a visualização no navegador sendo realizada com os itens na mesma ordem definida no código HTML. As Figuras 1.7 e 1.8 mostram a mesma lista, mas agora com os itens em ordem inversa e aleatória, respectivamente, efeito criado a partir da utilização do CSS. Quando navegando pela página com o uso do teclado ou tecnologia assistiva (um leitor de tela, por exemplo), os usuários podem ficar confusos nos cenários das Figuras 1.7 e 1.8, pois a ordem com que eles serão apresentados para os usuários será diferente da ordem

exibida na tela (quando utilizando um leitor de tela, o elemento que está sendo lido para o usuário fica em evidência na interface; naturalmente, espera-se que o foco siga uma ordem lógica na tela [de cima para baixo, da esquerda para a direita, por exemplo]).

```

<h1>Plano de viagem</h1>
<ol>
  <li id="item1">Definir destino.</li>
  <li id="item2">Definir data.</li>
  <li id="item3">Pesquisar preços
    <ol>
      <li>Passagens.</li>
      <li>Estadia.</li>
      <li>Alimentação.</li>
      <li>Locomoção.</li>
    </ol>
  </li>
  <li id="item4">Comprar passagens.</li>
  <li id="item5">Fazer reservas.</li>
  <li id="item6">Arrumar as malas.</li>
</ol>

```

Plano de viagem

1. Definir destino.
2. Definir data.
3. Pesquisar preços
 1. Passagens.
 2. Estadia.
 3. Alimentação.
 4. Locomoção.
4. Comprar passagens.
5. Fazer reservas.
6. Arrumar as malas.

Figura 1.6. Implementação de lista com os itens sendo exibidos na mesma ordem definida no código HTML. No quadro da esquerda, código HTML. No quadro da direita, visualização da página no navegador.

A criação de variáveis em CSS permite a definição de valores que podem ser referenciados por toda a extensão do código CSS. Isso permite que a definição desses valores fique centralizada, facilitando a manutenção, pois uma vez que o valor precisa ser alterado, basta alterar em único lugar, e os outros lugares no qual a variável alterada é referenciada são automaticamente atualizados.

```

ol {
  display: flex;
  flex-direction: column-reverse;
}

```

Plano de viagem

6. Arrumar as malas.
5. Fazer reservas.
4. Comprar passagens.
3. Pesquisar preços
 4. Locomoção.
 3. Alimentação.
 2. Estadia.
 1. Passagens.
2. Definir data.
1. Definir destino.

Figura 1.7. Implementação de lista com os itens sendo exibidos em ordem inversa através do CSS. No quadro da esquerda, código CSS. No quadro da direita, visualização da página no navegador.

```

ol {
  display: grid;
  grid-template-areas: 'l2' 'l4' 'l1' 'l5' 'l3';
}
#item1 { grid-area: l1; }
#item2 { grid-area: l2; }
#item3 { grid-area: l3; }
#item4 { grid-area: l4; }
#item5 { grid-area: l5; }

```

Plano de viagem

2. Definir data.
4. Comprar passagens.
1. Definir destino.
5. Fazer reservas.
3. Pesquisar preços
 1. Passagens.
 2. Estadia.
 3. Alimentação.
 4. Locomoção.
6. Arrumar as malas.

Figura 1.8. Implementação de lista com os itens sendo exibidos em ordem aleatória através do CSS. No quadro da esquerda, código CSS. No quadro da direita, visualização da página no navegador.

Um recurso importante para a acessibilidade das páginas *web* é permitir que os usuários escolham as cores que melhores lhes atendem. Por exemplo, permitir que os usuários alterem entre diferentes combinações de cores para o plano de fundo e cores dos textos. A Figura 1.9 mostra a utilização das variáveis em CSS para implementar as cores de plano de fundo, textos de cabeçalho e textos normais. Enquanto isso, utilizando a mesma estrutura, mas somente mudando os valores das variáveis criadas, a Figura 1.10 mostra a alteração das cores. Esse é um exemplo de como esse recurso poderia ser utilizado em uma funcionalidade que dê opções de grupos de cores pré-determinados que possam ser mais adequadas para diferentes usuários, aumentando a acessibilidade das páginas *web*.

Para criar uma variável em CSS alguns passos são importantes. Primeiro, deve-se definir o escopo da variável, ou seja, em que parte da página a variável deve estar disponível. Isso é feito através de um seletor do CSS (seletores são um recurso da linguagem para selecionar os trechos de HTML que devem receber os estilos CSS que são codificados no bloco logo após a definição do seletor). Para definir uma variável CSS global, isto é, que pode ser referenciada a partir de qualquer lugar da página, basta realizar a sua definição a partir do seletor que indica uma *tag* global (podendo ser a *tag html* ou a *tag body*, por exemplo, porque elas são as duas *tags* que envolvem todas as outras *tags* em um documento HTML) (veja o quadro inferior esquerdo da Figura 1.9 e o quadro esquerdo da Figura 1.10).

Logo após criar o seletor do escopo no qual a variável será criada, é necessário usar dois traços (“--”), seguido do nome desejado para a variável, dois pontos (“:”), e o valor atribuído à variável. Os dois traços no início do nome são necessários para que seja possível diferenciar que se quer criar uma variável ao invés de se tentar usar uma propriedade do CSS. O valor pode ser qualquer valor de qualquer propriedade CSS. Para pegar o valor de uma variável no código CSS ao longo do documento de estilo, basta utilizar a palavra reservada “var”, seguido do nome da variável entre parêntesis (o nome deve conter os dois traços no início) (veja o quadro inferior esquerdo da Figura 1.9 e o quadro esquerdo da Figura 1.10).

Como dito anteriormente, as dicas de implementação desta seção não apresentam todos os recursos e cuidados que os desenvolvedores devem ter para que as páginas das aplicações *web* desenvolvidas por eles sejam acessíveis. Assim, existem outras fontes que podem ser consultadas para se aprender como levar em consideração a acessibilidade ao criar códigos HTML e CSS. Alguns deles são podem ser encontrados em [MDN 2021, WebAIM 2021a, W3C 2016a].

1.3.2. WAI-ARIA

O WAI-ARIA (2017a) (Web Accessibility Initiative – Accessible Rich Internet Applications) é uma especificação técnica que permite criar interfaces dinâmicas e elementos de interface personalizados acessíveis. Seu funcionamento se dá a partir do uso de propriedades e valores específicos nas tags do HTML (marcações ARIA). Essas

marcações fazem com que a árvore de acessibilidade, gerada a partir do DOM, seja modificada para conter informações relevantes sobre os conteúdos das páginas para a acessibilidade. As marcações permitem definir o papel, o estado e as propriedades dos elementos de interface. A partir da definição dessas características em um elemento de interface, quando uma tecnologia assistiva passar por ele, ela será capaz de transmitir para o usuário suas informações: para que serve, como interagir com ele e qual o seu estado.

The figure illustrates the definition of theme colors for a web page using variables in CSS. It is divided into three parts:

- HTML Code (top):** Shows a main container with an h1 heading 'Computador' and a paragraph describing a computer. A link is provided for more information: `wikipedia.org`.
- CSS Code (bottom left):** Shows CSS rules using variables:


```
body {
  --background: white;
  --h1-color: black;
  --text-color: black;
}
main { background-color: var(--background); }
h1 { color: var(--h1-color); }
p { color: var(--text-color); }
```
- Browser Visualization (bottom right):** Shows the rendered page with the heading 'Computador' in black, the paragraph in black, and the link in blue. The background is white.

Figura 1.9. Definição das cores do tema de uma página *web* com o uso de variáveis em CSS. Código HTML (quando superior). Código CSS (quadro inferior esquerdo). Visualização da página no navegador (quadro inferior direito).

The figure illustrates the alteration of theme colors for a web page using variables in CSS. It is divided into two parts:

- CSS Code (left):** Shows CSS rules using variables with different values:


```
body {
  --background: black;
  --h1-color: yellow;
  --text-color: white;
}
main { background-color: var(--background); }
h1 { color: var(--h1-color); }
p { color: var(--text-color); }
```
- Browser Visualization (right):** Shows the rendered page with the heading 'Computador' in yellow, the paragraph in white, and the link in blue. The background is black.

Figura 1.10. Alteração das cores do tema de uma página *web* com o uso de variáveis em CSS. Código CSS (quadro esquerdo). Visualização da página no navegador (quadro direito).

É importante dizer que com o uso do WAI-ARIA e das tags não semânticas do HTML é possível criar elementos personalizados, com sua própria semântica. No entanto, apesar de ser possível, por exemplo, a criação de um botão personalizado, é recomendado utilizar a tag padrão do HTML5 (`button`) para isso, pois ela já possui todas as implementações necessárias para que usuários de TAs sejam capazes de interagir com ele.

Por outro lado, o WAI-ARIA é um importante recurso para a implementação de elementos que não possuem uma *tag* nativa no HTML. Um exemplo seria a construção de um menu acessível. Com o WAI-ARIA é possível informar que determinada *tag* tem

um papel de menu, além de ser possível indicar quando uma opção que possui subopções, que podem ser expandidas e minimizadas, está com as subopções em visíveis ou não, promovendo assim acessibilidade ao dar suporte para usuários de leitores de tela.

Outro recurso comum nos *web sites* que o WAI-ARIA possibilita tornar acessível são os alertas personalizados. Tal elemento se caracteriza pela exibição de uma tela por cima do conteúdo atual, dando foco a algum conteúdo que deve ser apresentado para o usuário. Os alertas podem ser utilizados para diversas situações: transmitir alguma informação, como um aviso, alerta ou mensagem de erro; disponibilizar um meio para que o usuário insira um dado; permitir que o usuário realize alguma ação secundária; entre outros.

Mas, o fato de ele aparecer por cima do conteúdo atual pode causar problemas de acessibilidade se a implementação não for feita de maneira adequada. Para isso, o WAI-ARIA disponibiliza marcações para a criação de áreas de ativação. Essas áreas, uma vez que exibidas na tela, também são mencionadas pelas TAs.

A utilização do WAI-ARIA também está fortemente conectada à utilização de JavaScript nas páginas. Com o JavaScript é possível manipular os elementos das páginas através da árvore DOM, sendo possível mudar os atributos das *tags*, a aparência, adicionar *tags* e remover *tags* do DOM. Para que todo esse dinamismo também seja perceptível para usuários de TAs é necessário usar as marcações do WAI-ARIA de maneira adequada quando necessária, via JavaScript.

Para ilustrar a relação entre o JavaScript para tornar as páginas dinâmicas e o uso do WAI-ARIA, pode-se pensar no exemplo de um menu (Figura 1.11) com opções que ao serem clicadas exibem um submenu com mais opções. A ação de clicar em cada uma das opções pode ser capturada via JavaScript, e a partir da manipulação do DOM, exibir os respectivos submenus. Ao exibir um submenu, o código JavaScript também deve atualizar a propriedade da *tag* da opção que foi clicada para indicar para as TAs que o submenu daquela opção está aberto no momento.

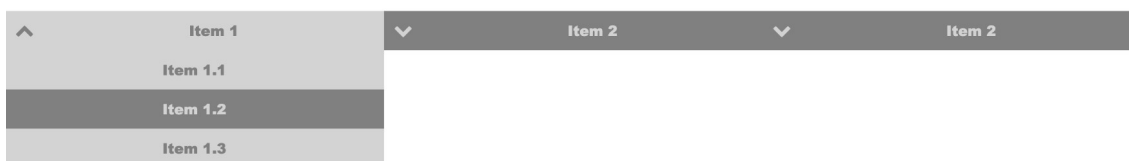


Figura 1.11. Exemplo de menu implementado com JavaScript e marcações do WAI-ARIA

1.3.3. Validadores de código, avaliadores de acessibilidade e outros recursos

Os validadores de código HTML e CSS são ferramentas que comparam um trecho de código escrito com as respectivas linguagens com as regras definidas na especificação de cada uma delas [W3C 2013, W3C 2019a]. Essas ferramentas ajudam a evitar que problemas de acessibilidade ocorram devido à escrita de algum código errado. Além disso, pela grande quantidade de palavras reservadas (termos que fazem parte das linguagens) envolvidos nas duas linguagens (HTML e CSS), é comum esquecer

algumas delas, ou escrever errado. Os validadores de código também podem ser úteis nesses casos.

Os avaliadores automáticos de acessibilidade são capazes de verificar se os *web sites* infringem alguma diretriz de acessibilidade. Tais ferramentas podem se basear em diferentes versões da WCAG, e as funcionalidades de cada uma delas variam. De modo geral, os avaliadores são capazes de identificar erros e pontos de atenção que necessitam ser verificados por uma pessoa para dizer com certeza se representam violação de alguma diretriz ou não.

Devido à complexidade das diretrizes de acessibilidade, diferentes avaliadores podem apresentar diferenças de interpretação das diretrizes em suas regras de avaliação, o que pode gerar resultados diferentes quando utilizando mais de uma ferramenta. Por exemplo, um erro que pode ser apresentado em uma ferramenta pode passar despercebido por outra. Mas, apesar dessas limitações, sua utilização ainda é válida e pode ajudar na melhoria do nível de acessibilidade das páginas avaliadas. É aconselhável utilizar mais de uma ferramenta para que elas se complementem (uma ferramenta pode cobrir um ponto falho da outra) [Abduganiev 2017].

Por causa de sua importância no processo de criação dos *web sites*, advinda da velocidade proporcionada na identificação de erros de acessibilidade, os avaliadores automáticos continuam a ser estudados pela comunidade científica, e novas ferramentas continuam a ser desenvolvidas [Abduganiev 2017]. Há ainda ferramentas desenvolvidas com recursos de inteligência artificial para que cada vez mais verificações possam ser feitas automaticamente [Alsaeedi 2020].

Existem diversos outros recursos auxiliares que podem ser utilizados pelos desenvolvedores *web* para a criação de *web sites* acessíveis. O principal deles são as documentações complementares das diretrizes de acessibilidade. Nelas, há dicas de como implementar os códigos para que cada um dos critérios de sucesso seja atendido [W3C 2016a]. Também existem recursos que simplificam as diretrizes de acessibilidade, como o *check list* criado pela WebAIM (2021d), e o WebAIR [Swallow et al. 2014]. Por fim, ferramentas para verificação do contraste das cores a serem utilizadas para os conteúdos das páginas e para os *links* também estão disponíveis e podem aumentar o nível de acessibilidade das páginas a serem criadas [WebAIM 2021b, WebAIM 2021c].

1.4. Conclusão

Este capítulo discutiu a importância e motivações para a elaboração de *web sites* acessíveis, além de mostrar algumas técnicas para a implementação de páginas *web* com suporte à acessibilidade. Os benefícios trazidos pela acessibilidade não se limitam apenas a eliminar barreiras para pessoas com deficiências, mas envolve os demais usuários promovendo uma melhor usabilidade, ou seja, tornando *web sites* mais eficazes, eficientes e conseqüentemente, mais fáceis de serem utilizados, inclusive para os usuários principiantes.

Há muitos desafios ainda em aberto para a adoção efetiva, por parte de desenvolvedores *web*, dos princípios de acessibilidade. Uma das barreiras é cultural: há que se criar, junto aos desenvolvedores, a consciência da importância de se considerar diretrizes de acessibilidade desde as fases mais iniciais do processo de desenvolvimento de *web sites*. Além disso, há problemas na formação desses desenvolvedores, do ponto de vista técnico, uma vez que não é sempre que conceitos relacionados à acessibilidade estão presentes nos currículos de cursos técnicos ou universitários. Nesse sentido, o presente capítulo pretendeu trazer uma contribuição neste sentido, fornecendo um material de referência que pode servir como uma primeira aproximação para os conceitos nele tratados. Espera-se que a sua leitura estimule o leitor a buscar aprofundamento neste tópico e que isso possa contribuir para a popularização das diretrizes de acessibilidade, resultando em um maior número de *web sites* acessíveis de qualidade.

Referências

- Abduganiev, S. G. (2017). Towards automated *web* accessibility evaluation: a comparative study. *Int. J. Inform. Technol. Comput. Sci*, 9(9), 18-44. DOI: <https://doi.org/10.5815/ijitcs.2017.09.03>
- Alsaeedi, A. (2020). Comparing *web* accessibility evaluation tools and evaluating the accessibility of *webpages*: proposed frameworks. *Information*, 11(1), 40. DOI: <https://doi.org/10.3390/info11010040>
- Antonelli, H. L., Rodrigues, S. S., Watanabe, W. M., & de Mattos Fortes, R. P. (2018, June). A survey on accessibility awareness of Brazilian *web* developers. In *Proceedings of the 8th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion* (pp. 71-79). DOI: <https://doi.org/10.1145/3218585.3218598>
- Barricelli, B. R., Sciarelli, P., Valtolina, S., & Rizzi, A. (2018). *Web* accessibility legislation in Italy: a survey 10 years after the Stanca Act. *Universal Access in the Information Society*, 17(1), 211-222. DOI: <http://dx.doi.org/10.1007/s10209-017-0526-z>
- Cao, S., & Loiacono, E. (2019, July). The state of the awareness of *web* accessibility guidelines of student *website* and app developers. In *International Conference on Human-Computer Interaction* (pp. 32-42). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-21902-4_3
- Carvajal, C. M., Piqueras, R. F., & Mérida, J. F. C. (2018). Evaluation of *Web* Accessibility of Higher Education Institutions in Chile. *International Education Studies*, 11(12), 140-148. DOI: <https://doi.org/10.5539/ies.v11n12p140>
- ECMA International. (2021). ECMA-262. Recuperado em 25 de agosto, 2021, de <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>.
- El-Glaly, Yasmine N. (2020). Teaching Accessibility to Software Engineering Students. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 121–127. DOI: <https://doi.org/10.1145/3328778.3366914>

- Ferati, M., & Vogel, B. (2020, March). Accessibility in *web* development courses: A case study. In *Informatics* (Vol. 7, No. 1, p. 8). Multidisciplinary Digital Publishing Institute. DOI: <https://doi.org/10.3390/informatics7010008>
- Fernández-Díaz, E., Iglesias-Sánchez, P. P., & Jambrino-Maldonado, C. (2020). Exploring WHO Communication during the COVID 19 Pandemic through the WHO Website Based on W3C Guidelines: Accessible for All?. *International journal of environmental research and public health*, 17(16), 5663. DOI: <https://doi.org/10.3390/ijerph17165663>
- Gleason, C., Valencia, S., Kirabo, L., Wu, J., Guo, A., Jeanne Carter, E., ... & Pavel, A. (2020, October). Disability and the COVID-19 Pandemic: Using Twitter to Understand Accessibility during Rapid Societal Transition. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 1-14). DOI: <https://doi.org/10.1145/3373625.3417023>
- Gonçalves, T. S., Simelio Solà, N., & Moreno Sardá, A. (2021). *Web* accessibility for citizens with reduced capacities on science portals: evaluation of MCTIC and Ciência Viva. *The Journal of International Communication*, 27(1), 106-125. DOI: <https://doi.org/10.1080/13216597.2021.1885463>
- Inal, Y., Rızvanoğlu, K., & Yesilada, Y. (2019). *Web* accessibility in Turkey: awareness, understanding and practices of user experience professionals. *Universal Access in the Information Society*, 18(2), 387-398. DOI: <https://doi.org/10.1007/s10209-017-0603-3>
- Internet World Stats. (2021). Internet World Stats: Usage and Population Statistics. Recuperado em 20 de agosto, 2021, de <https://www.Internetworldstats.com/stats.htm>.
- Kimmons, R. (2017). Open to all? Nationwide evaluation of high-priority *web* accessibility considerations among higher education *websites*. *Journal of Computing in Higher Education*, 29(3), 434-450. DOI: <https://doi.org/10.1007/s12528-017-9151-3>
- Máñez-Carvajal, C., Cervera-Mérida, J. F., & Fernández-Piqueras, R. (2019). *Web* accessibility evaluation of top-ranking university *Web* sites in Spain, Chile and Mexico. *Universal Access in the Information Society*, 1-6. DOI: <https://doi.org/10.1007/s10209-019-00702-w>
- Martins, J., Gonçalves, R., & Branco, F. (2017). A full scope *web* accessibility evaluation procedure proposal based on Iberian eHealth accessibility compliance. *Computers in Human Behavior*, 73, 676-684. DOI: <https://doi.org/10.1016/j.chb.2016.12.010>
- MDN. (2021). HTML: Boas práticas em acessibilidade. Recuperado em 25 de agosto, 2021, de <https://developer.mozilla.org/pt-BR/docs/Learn/Accessibility/HTML>.
- Return on Disability. (2020). 2020 Global Economics of Disability. Recuperado em 19 de agosto, 2021, de <https://www.rod-group.com/sites/default/files/2020%20Annual%20Report%20-%20The%20Global%20Economics%20of%20Disability.pdf>.
- Russ, S., & Hamidi, F. (2021, April). Online learning accessibility during the COVID-19 pandemic. In *Proceedings of the 18th International Web for All Conference* (pp. 1-7). DOI: <https://doi.org/10.1145/3430263.3452445>

- Swallow, D., Power, C., Petrie, H., Bramwell-Dicks, A., Buykx, L., Velasco, C. A., ... & Connor, J. O. (2014, July). Speaking the language of *web* developers: Evaluation of a *web* accessibility information resource (*WebAIR*). In International Conference on Computers for Handicapped Persons (pp. 348-355). Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-08596-8_54
- United Nations. (2006). Convention on the Rights of Persons with Disabilities (CRPD). Recuperado em 23 de agosto, 2021, de <https://www.un.org/development/desa/disabilities/convention-on-the-rights-of-persons-with-disabilities.html>.
- United Nations. (2013a). Accessibility and Development – Mainstreaming disability in the post-2015 development agenda. Recuperado em 23 de agosto, 2021, de https://www.un.org/disabilities/documents/accessibility_and_development.pdf.
- United nations. (2013b). Transforming our world: the 2030 agenda for sustainable development. Recuperado em 23 de agosto, 2021, de <https://sdgs.un.org/sites/default/files/publications/21252030%20Agenda%20for%20Sustainable%20Development%20web.pdf>.
- W3C. (1999). *Web* Accessibility Guidelines 1.0. Recuperado em 25 de agosto, 2021, de <https://www.w3.org/TR/WAI-WEBCONTENT/>.
- W3C. (2013). Markup Validation Service. Recuperado em 22 de agosto, 2021, de <https://validator.w3.org/>.
- W3C. (2015a). Authoring Tool Accessibility Guidelines (ATAG) 2.0. Recuperado em 22 de agosto, 2021, de <https://www.w3.org/TR/ATAG20/>.
- W3C. (2015b). User Agent Accessibility Guidelines (UAAG) 2.0. Recuperado em 22 de agosto, 2021, de <https://www.w3.org/TR/UAAG20/>.
- W3C. (2016a). Techniques for WCAG 2.0. Recuperado em 25 de agosto, 2021, de <https://www.w3.org/TR/WCAG20-TECHS/>.
- W3C. (2016b). *Web* Accessibility Evaluation Tools List. Recuperado em 22 de agosto, 2021, de <https://www.w3.org/WAI/ER/tools/>.
- W3C. (2017a). Accessible Rich Internet Applications (WAI-ARIA) 1.1. Recuperado em 22 de agosto, 2021, de <https://www.w3.org/TR/wai-aria-1.1/>.
- W3C. (2017b). HTML 5.2. Recuperado em 25 de agosto, 2021, de <https://www.w3.org/TR/html52/>.
- W3C. (2017c). Tools and techniques. Recuperado em 20 de agosto, 2021, de <https://www.w3.org/WAI/people-use-web/tools-techniques/>.
- W3C. (2018a). *Web* Accessibility Laws & Policies. Recuperado em 23 de agosto, 2021, de <https://www.w3.org/WAI/policies/>.
- W3C. (2018b). *Web* Content Accessibility Guidelines (WCAG) 2.1. Recuperado em 22 de agosto, 2021, de <https://www.w3.org/TR/WCAG21/>.
- W3C. (2019a). CSS Validation Service. Recuperado em 22 de agosto, 2021, de <https://jigsaw.w3.org/css-validator/>.

- W3C. (2019b). Colors with Good Contrast. Recuperado em 25 de agosto, 2021, de <https://www.w3.org/WAI/perspective-videos/contrast/>.
- W3C. (2019c). Introduction to *Web* Accessibility. Recuperado em 20 de agosto, 2021, de <https://www.w3.org/WAI/fundamentals/accessibility-intro/>.
- W3C. (2021a). W3C Accessibility Guidelines. Recuperado em 25 de agosto, 2021, de <https://www.w3.org/TR/wcag-3.0/>.
- W3C. (2021b). W3C Cascading Style Sheets home page. Recuperado em 25 de agosto, 2021, de <https://www.w3.org/Style/CSS/>.
- W3C. (2021c). W3C: Leading the *web* to its full potential. Recuperado em 25 de agosto, 2021, de <https://www.w3.org/>.
- W3C. (2021d). *Web* Accessibility Initiative WAI. Recuperado em 25 de agosto, 2021, de <https://www.w3.org/WAI/>.
- W3C. (2021e). *Web* Content Accessibility Guidelines (WCAG) 2.2. Recuperado em 25 de agosto, 2021, de <https://www.w3.org/TR/WCAG22/>.
- W3C. (2021f). *Web* Content Accessibility Guidelines (WCAG) Overview. Recuperado em 25 de agosto, 2021, de <https://www.w3.org/WAI/standards-guidelines/wcag/>.
- Wang, Ye Diana. (2012). A holistic and pragmatic approach to teaching *web* accessibility in an undergraduate web Design course. In Proceedings of the 13th annual conference on Information technology education (SIGITE '12). Association for Computing Machinery, New York, NY, USA, 55–60. DOI: <https://doi.org/10.1145/2380552.2380568>
- We are social. (2021). Digital 2021: global overview report. Recuperado em 20 de agosto, 2021, de <https://wearesocial.com/digital-2021>.
- WebAIM. (2021a). Accessible CSS. Recuperado em 25 de agosto, 2021, de <https://webaim.org/techniques/css/>.
- WebAIM. (2021b). Contrast Checker. Recuperado em 26 de agosto, 2021, de <https://webaim.org/resources/contrastchecker/>.
- WebAIM. (2021c). Link Contrast Checker. Recuperado em 26 de agosto, 2021, de <https://webaim.org/resources/linkcontrastchecker>.
- WebAIM. (2021d). *Web*AIM's WCAG 2 Checklist. Recuperado em 26 de agosto, 2021, de <https://webaim.org/standards/wcag/checklist>.
- World Health Organization. (2011). World report on disability 2011. World Health Organization.