



STi
SUPERINTENDÊNCIA DE
TECNOLOGIA DA INFORMAÇÃO



Universidade Federal Fluminense

Plano de Acessibilidade

1.0.0 2019



Serviço Público Federal
Ministério da Educação
Universidade Federal Fluminense
Superintendência de Tecnologia da Informação

Plano de Acessibilidade

Autor: Guilherme da Silva Alves Gonçalves

Data da última modificação: 19 de Fevereiro de 2019.



Sumário

[1. Introdução](#)

[1.1. Links relacionados](#)

[2. O desenvolvimento de websites e sistemas acessíveis](#)

[2.1. Padrões Web](#)

[2.2. Diretrizes e recomendações de acessibilidade](#)

[2.3. Avaliações e testes de acessibilidade](#)

[2.4. Manutenção da acessibilidade](#)

[3. Recomendações a serem seguidas](#)

[3.1. Marcação \(HTML\)](#)

[3.1.1. Respeitar os padrões](#)

[3.1.2. Organizar o código HTML de forma lógica e semântica](#)

[3.1.3. Utilizar corretamente os níveis de cabeçalho](#)

[3.1.4. Ordenar de forma lógica e intuitiva a leitura e tabulação](#)

[3.1.5. Fornecer âncoras para ir direto a um bloco de conteúdo](#)

[3.1.6. Não utilizar tabelas para diagramação](#)

[3.1.7. Separar links adjacentes](#)

[3.1.8. Dividir as áreas de informação](#)

[3.1.9. Não abrir novas abas ou janelas sem a solicitação do usuário](#)

[3.2. Comportamento \(Document Object Model – DOM\)](#)

[3.2.1. Disponibilizar todas as funções da página via teclado](#)

[3.2.2. Não criar páginas com atualização automática periódica](#)

[3.2.3. Não utilizar redirecionamento automático de páginas](#)

[3.2.4. Não incluir situações com intermitência de tela](#)

[3.2.5. Assegurar o controle do usuário sobre as alterações temporais do conteúdo](#)

[3.3. Conteúdo / Informação](#)

[3.3.1. Identificar o idioma principal da página](#)

[3.3.2. Informar mudança de idioma no conteúdo](#)

[3.3.3. Informar mudança de idioma no conteúdo](#)

[3.3.4. Informar o usuário sobre sua localização na página](#)

[3.3.5. Descrever links clara e sucintamente](#)

[3.3.6. Fornecer alternativa em texto para as imagens do website](#)



3.3.7. Disponibilizar documentos em formatos acessíveis

3.3.8. Em tabelas, utilizar títulos e resumos de forma apropriada

3.3.9. Associar células de dados às células de cabeçalho

3.3.10. Garantir a leitura e compreensão das informações

3.3.11. Disponibilizar uma explicação para siglas, abreviaturas e palavras incomuns

3.4. Apresentação / Design

3.4.1. Oferecer contraste mínimo entre plano de fundo e primeiro plano

3.4.2. Não utilizar apenas cor ou outras características sensoriais para diferenciar elementos

3.4.3. Permitir redimensionamento sem perda de funcionalidade

3.4.4. Possibilitar que o elemento com foco seja visualmente evidente

3.5. Formulários

3.5.1. Associar etiquetas aos seus campos

3.5.2. Estabelecer uma ordem lógica de navegação

3.5.3. Não provocar automaticamente alteração no contexto

3.5.4. Fornecer instruções para entrada de dados

3.5.6. Agrupar campos de formulário

4. Elementos padronizados de acessibilidade digital

5. Práticas desaconselhadas



1. Introdução

Na última década, a expansão prodigiosa da Internet vem revolucionando as formas de comunicação, de acesso à informação e de realização de negócios em todo o mundo ([eMAG, 2014](#)). Toda essa expansão, principalmente com a [Web 2.0](#), promoveu a possibilidade da participação das pessoas na produção e organização das informações. Mesmo quando o conteúdo não é gerado pelos usuários, este pode ser enriquecido através de comentários, avaliação, ou personalização.

Neste contexto, podemos destacar um grande problema enfrentado hoje em dia: a inacessibilidade à informação e ao conteúdo em websites da internet. Esse, afeta consideravelmente pessoas com deficiências visuais e físicas ao devido acesso à informação.

Com o objetivo de padronizar e auxiliar a produção de conteúdos, foram criados padrões de acessibilidade (ex: [WCAG](#)¹, [eMAG](#)², [WAI-ARIA](#)³) que fornecem diretrizes e recomendações a serem seguidas, tornando os conteúdos mais acessíveis a todas as pessoas.

Este documento visa apresentar de uma forma mais simplificada as principais recomendações para o desenvolvimento de *websites* acessíveis, sendo estes compatíveis com as diretrizes e recomendações fornecidas pela WCAG, eMAG e pela WAI-ARIA. Além disso, considerando também alguns aspectos adotados pelos sites governamentais brasileiros, como teclas de acesso rápido e contraste.

1.1. Links relacionados

1. <http://portal.mec.gov.br/acessibilidade>
2. <https://www.governoeletronico.gov.br>
3. <https://www.w3.org/TR/wai-aria/>
4. <https://www.w3.org/WAI/intro/wcag>

¹ Web Content Accessibility Guidelines.

² Modelo de Acessibilidade em Governo Eletrônico.

³ Accessible Rich Internet Applications.



2. O desenvolvimento de websites e sistemas acessíveis

A acessibilidade à *Web* refere-se a garantir acesso facilitado a qualquer pessoa, independente das condições físicas, dos meios técnicos ou dispositivos utilizados. No entanto, ela depende de vários fatores, tanto de desenvolvimento quanto de interação com o conteúdo (e-MAG, 2014). O desenvolvimento de *websites* e sistemas *Web* acessíveis é realizado em três etapas:

1. Seguir os padrões *Web*;
2. Seguir as diretrizes e recomendações de acessibilidade;
3. Realizar avaliações automáticas e manuais, e testes de acessibilidade com usuários deficientes.

2.1. Padrões *Web*

Para se criar um ambiente online efetivamente acessível é necessário, primeiramente, que o código esteja dentro dos [padrões *Web* internacionais definidos pelo W3C](#).

Os padrões de desenvolvimento *Web* do W3C, ou *Web Standards*, são um conjunto de recomendações que visam padronizar o conteúdo *Web*, possibilitando melhores práticas no desenvolvimento de páginas da internet. Uma página desenvolvida de acordo com os padrões *Web* deve estar em conformidade com as normas HTML, XML, XHTML e CSS, seguindo as regras de formatação sintática. Além disso, é muito importante que o código seja semanticamente correto, ou seja, que cada elemento seja utilizado de acordo com um significado apropriado, valor e propósito.

A conformidade com os padrões *Web* permite que qualquer sistema de acesso à informação interprete a mesma adequadamente e da mesma forma, seja por meio de navegadores, leitores de tela, dispositivos móveis (celulares, tablets, etc.) ou agentes de *software* (mecanismos de busca ou ferramentas de captura de conteúdo). Páginas que não possuem um código de acordo com os padrões do W3C apresentam comportamento imprevisível, e na maioria das vezes impedem ou pelo menos dificultam o acesso.



Neste sentido, quando *websites* e sistemas Web implementam corretamente os padrões *Web*, alguns benefícios podem ser obtidos:

1. Simplificação do desenvolvimento e a manutenção.
2. Organização do código.
3. Alinhamento da equipe de desenvolvimento ao padrão, no qual todos seguem as boas práticas e entendem o que foi feito, independente da autoria.
4. Qualidade do código.
5. Compatibilidade com navegadores futuros e outros dispositivos.
6. Carregamento e apresentação de página mais ágil.
7. Melhora na acessibilidade.
8. Facilidade de evoluir e adaptar.

Para conhecer as boas práticas em desenvolvimento de *websites* de acordo com os padrões veja a [Cartilha de Codificação dos Padrões Web em Governo Eletrônico \(ePWG\)](#), onde são listadas uma série de boas práticas.

2.2. Diretrizes e recomendações de acessibilidade

As diretrizes ou recomendações de acessibilidade explicam como tornar o conteúdo

Web acessível a todas as pessoas, destinando-se aos criadores de conteúdo *Web* (autores de páginas e criadores de *websites*) e aos programadores de ferramentas para

criação de conteúdo. A principal documentação nessa área é a WCAG (*Web Content Accessibility Guidelines*) atualmente em sua versão 2.0, desenvolvida pelo consórcio W3C a partir da criação do WAI (*Web Accessibility Initiative*), contendo as recomendações de acessibilidade para conteúdo *Web*.

O WAI ainda desenvolveu especificações para aplicações web (*Web Rica*), chamado WAI-ARIA (Accessible Rich Internet Applications), que busca resolver muitos dos problemas da camada de comportamento (DOM), sendo parte já implementada por alguns navegadores.

Por fim, o eMAG é o documento que norteia o desenvolvimento de *websites* e portais acessíveis no âmbito do governo federal. O eMAG traz muitos conceitos e recomendações aplicadas no WCAG e os organizam em categorias específicas.



2.3. Avaliações e testes de acessibilidade

Durante e após a construção do ambiente online de acordo com os padrões Web e as diretrizes de acessibilidade, é necessário testá-lo para garantir sua acessibilidade. Neste sentido, podemos dividir as avaliações de acessibilidade em três etapas sequenciais:

1. **Avaliações automáticas:** Avaliações de acessibilidade que podem ser feitas de maneira automática por meio de *plugins* de verificação e de *websites* de validação. É recomendado se utilizar os *plugins* durante o desenvolvimento e utilizar a verificação completa do ambiente online nos *websites* após seu desenvolvimento.
 - a. [Google Accessibility Developer Tools](#) (Plugin para navegador Chrome)
 - b. [Accessibility Engine \(axe\)](#) (Plugin para navegador Chrome e Firefox)
 - c. [Avaliador e Simulador de Acessibilidade em websites](#) (*website* - eMAG)
 - d. [DaSilva.org](#) (*website* - WCAG / eMAG)
 - e. [AccessMonitor](#) (*website* - WCAG)
 - f. [AChecker](#) (*website* - WCAG)
 - g. [Contrast Checker](#) (*website* - Validação de contraste de cores)

Outros *websites* e *plugins* podem ser encontrados no endereço <https://www.w3.org/WAI/ER/tools/>.

2. **Avaliações manuais:** Avaliações de acessibilidade manuais são importantes para se verificar problemas que podem não ter sido pegos na avaliação automática. Esta etapa de avaliação pode ser feita por meio de inspeções guiadas por *checklists*, com o intuito de verificar se os elementos atendem o mínimo de conformidade, e pelo uso de leitores de tela, a fim de validar se essas ferramentas estão lendo os conteúdos da forma esperada. Abaixo seguem alguns links de *checklists* e leitores de tela que podem ajudar na avaliação.
 - a. <https://www.cs.york.ac.uk/hci/webair/index.htm>
 - b. http://romeo.elsevier.com/accessibility_checklist/
 - c. <https://www.wuhcag.com/wcag-checklist/>
 - d. <https://www.governoeletronico.gov.br/documentos-e-arquivos/eM>



[AG-Checklist-acessibilidade-desenvolvedores.pdf](#)

- e. [NVDA \(Windows\)](#)
- f. [Orca \(Linux\)](#)
- g. [ChromeVox \(Google Chrome\)](#)

3. **Avaliações com usuários:** Por fim, outra etapa essencial da validação de uma página é a realização de testes com usuários reais (pessoas com deficiência ou limitações técnicas). Um usuário real poderá dizer se um *website* está realmente acessível, compreensível e com boa usabilidade e não simplesmente tecnicamente acessível. Quanto maior e mais diversificado o número de usuários reais participando da avaliação de acessibilidade, mais eficaz e robusto será o resultado.

2.4. Manutenção da acessibilidade

Segundo o documento do eMAG, a promoção da acessibilidade é um processo contínuo, recomenda-se que testes sejam realizados, de forma pontual, a cada alteração de conteúdo e validações globais em espaços determinados de tempo. O intervalo depende de diversos fatores, mas é altamente recomendável que se valide o *website/sistema* todo quando for feita a atualização de funcionalidades ou mudança de *layout*.



3. Recomendações a serem seguidas

Os padrões de acessibilidade compreendem recomendações ou diretrizes que visam tornar o conteúdo Web acessível a todas as pessoas, inclusive às pessoas com deficiência, destinando-se aos autores de páginas, projetistas de websites e aos desenvolvedores de ferramentas para criação de conteúdo. A observação destes padrões também facilita o acesso ao conteúdo da Web, independente da ferramenta utilizada (navegadores Web para computadores de mesa, laptops, telefones celulares, ou navegador por voz) e de certas limitações de ordem técnicas, como, por exemplo, uma conexão lenta, a falta de recursos de mídia, etc (e-MAG, 2014).

O documento de referência do eMAG divide as suas recomendações em seis seções relacionadas às necessidades de implementação:

1. Marcação (HTML)
2. Comportamento no DOM (Document Object Model)
3. Conteúdo / Informação
4. Apresentação / Design
5. Multimídia
6. Formulário

Entretanto, para este documento, desconsideraremos a seção de Multimídia, pois não trabalhamos com recursos sonoros e de vídeo em nossos sistemas. Caso precise das recomendações para essa seção verifique o [capítulo 3.5 do documento da eMAG](#).

Vale a pena destacar que todas as recomendações apresentadas no documento eMAG possuem uma correspondência nas diretrizes definidas no documento da WCAG

(<https://www.w3.org/Translations/WCAG20-pt-br/#guidelines>).

Por fim, em cada seção das recomendações também serão feitas adições de recomendações da WAI-ARIA, se baseando no documento de boas práticas da própria (<https://www.w3.org/TR/wai-aria-practices/>).



3.1. Marcação (HTML)

3.1.1. Respeitar os padrões

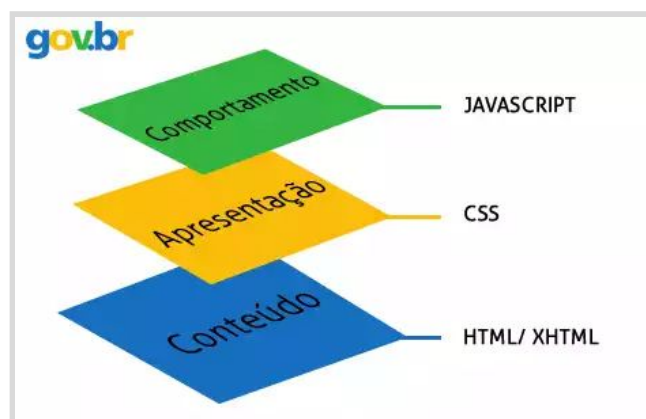
Ver WCAG 2.0 Critérios de Sucesso 4.1.1 e 4.1.2 (Princípio 4: Robusto)

Os Padrões *Web* são recomendações do W3C (World Wide Web Consortium), as quais são destinadas a orientar os desenvolvedores para o uso de boas práticas que tornam a web acessível para todos, permitindo assim que os desenvolvedores criem

experiências ricas, alimentadas por um vasto armazenamento de dados, os quais estão disponíveis para qualquer dispositivo e compatíveis com atuais e futuros agentes de usuário (ex: navegadores).

Outro ponto importante no respeito aos Padrões *Web* é a separação de camadas. As

camadas lógicas deverão ser separadas, de acordo com o objetivo para o qual elas foram desenvolvidas. Assim, para a camada de conteúdo devem ser utilizadas as linguagens de marcação, como HTML e XHTML. Para a camada de apresentação visual do conteúdo, utilizam-se as folhas de estilo CSS em qualquer uma de suas versões. Já para a camada que modifica o comportamento dos elementos, são utilizadas linguagens javascript e modelos de objeto (dom).



Fonte:

<http://emag.governoeletronico.gov.br/imagens/exemplos/camadas.png>



3.1.2. Organizar o código HTML de forma lógica e semântica

Ver WCAG 2.0 Critério de Sucesso 1.3.1 (Princípio 2: Adaptável)

O código HTML deve ser organizado de forma lógica e semântica, ou seja, apresentando os elementos em uma ordem compreensível e correspondendo ao conteúdo desejado. Cada elemento HTML deve ser utilizado para o fim que ele foi criado, levando em conta as [estruturas fornecidas pelo HTML5](#), o seu [valor semântico](#) e também a [aplicação de roles, estados e propriedades via WAI-ARIA](#).

3.1.3. Utilizar corretamente os níveis de cabeçalho

Ver WCAG 2.0 Critérios de Sucesso 1.3.1, 2.4.6 e 2.4.10 (Princípio 1 e 2: Perceptível e Operável)

Os níveis de cabeçalho (elementos HTML H1 a H6) devem ser utilizados de forma hierárquica, pois organizam a ordem de importância e subordinação dos conteúdos, facilitando a leitura e compreensão. Além disso, muitos leitores de tela utilizam a hierarquia de cabeçalhos como uma forma de navegação na página, pulando de um para outro, agilizando, assim, a navegação.

Ordem de subordinação: H1 > H2 > H3 > H4 > H5 > H6

3.1.4. Ordenar de forma lógica e intuitiva a leitura e tabulação

Ver WCAG 2.0 Critérios de Sucesso 1.3.2 e 2.4.3 (Princípio 1 e 2: Perceptível e Operável)

Deve-se criar o código HTML com uma sequência lógica de leitura para percorrer links, controles de formulários e objetos. Essa sequência é determinada pela ordem que se encontra no código HTML. Essa pode favorecer que a navegação venha após o conteúdo, para facilitar a navegação pelos usuários deficientes, ou fazer uso do atributo “tabindex” para direcioná-los dentro do *website*/sistema.



3.1.5. Fornecer âncoras para ir direto a um bloco de conteúdo

Ver WCAG 2.0 Critério de Sucesso 2.4.1 (Princípio 2: Operável)

Devem ser fornecidas âncoras, disponíveis na barra de acessibilidade ou barra superior do *website*, que apontem para links relevantes presentes na mesma página. Assim, é possível ir ao bloco de conteúdo desejado. Os links devem ser colocados em lugares estratégicos da página, como no início e fim do conteúdo e início e fim do menu. É importante ressaltar que o primeiro link da página deve ser o de ir para o conteúdo.



Para facilitar a utilização das âncoras, podem ser disponibilizados atalhos por teclado, utilizando o atributo `accesskey` nos links relevantes. Não pode haver repetição do mesmo `accesskey` em uma página.

```
▼ <ul id="accessibility">
  ▼ <li>
    ... ▶ <a accesskey="1" href="#acontent" id="link-conteudo">...</a> == $0
    </li>
  ▼ <li>
    ▶ <a accesskey="2" href="#anavigation" id="link-navegacao">...</a>
    </li>
  ▼ <li>
    ▶ <a accesskey="3" href="#SearchableText" id="link-buscar">...</a>
    </li>
  ▼ <li class="last-item">
    ▶ <a accesskey="4" href="#afooter" id="link-rodape">...</a>
    </li>
</ul>
```



3.1.6. Não utilizar tabelas para diagramação

Ver WCAG 2.0 Critério de Sucesso 1.3.1 (Princípio 1: Perceptível)

As tabelas devem ser utilizadas apenas para dados tabulares e não para efeitos de

disposição dos elementos na página. Para este fim, utilize as folhas de estilo e desenvolva *websites* [tableless](#).

3.1.7. Separar links adjacentes

Ver WCAG 2.0 Critério de Sucesso 1.3.1 (Princípio 1: Perceptível)

Links adjacentes devem ser separados por mais do que simples espaços, para que não fiquem confusos, em especial para usuários que utilizam leitor de tela. Para isso, é recomendado o uso de listas, onde cada elemento dentro da lista é um link. As listas podem ser estilizadas visualmente com CSS para que os itens sejam mostrados da maneira desejada, como um ao lado do outro (*display inline*).

Se estes links adjacentes formam um menu de navegação ou outra forma de navegação para os usuários é importante fazer uso de:

1. Elemento HTML <NAV> para o menu principal do *website*/sistema, equivalente ao se utilizar a `role="navigation"` do WAI-ARIA.
2. Atributo `role="menu"` e `role="menu-item"` para listas de links genéricas e seus respectivos itens filhos.
3. Atributo `role="menubar"` e `role="menu-item"` para listas de links horizontais e seus respectivos itens filhos.

3.1.8. Dividir as áreas de informação

Ver WCAG 2.0 Critério de Sucesso 3.2.3 (Princípio 3: Compreensível)

Áreas de informação devem ser divididas em grupos fáceis de gerenciar. As divisões

mais comuns são “topo”, “conteúdo”, “menu” e “rodapé”. Nas páginas internas deve-se manter uma mesma divisão para que o usuário se familiarize mais



rapidamente com a estrutura do *website*.

É importante fazer uso da semântica fornecida pelo HTML5 na divisão dos grupos de conteúdo dentro do *website* (HEADER, NAV, SECTION, ARTICLE, ASIDE e FOOTER), como também a semântica complementar fornecida pelo WAI-ARIA (BANNER, COMPLEMENTARY, CONTENTINFO, MAIN, NAVIGATION, SEARCH).

Uma explicação mais detalhada sobre esse tópico pode ser encontrada na recomendação 1.8 do manual do eMAG.

3.1.9. Não abrir novas abas ou janelas sem a solicitação do usuário

Ver WCAG 2.0 Critério de Sucesso 3.2.5 (Princípio 3: Compreensível)

A decisão de utilizar-se de novas instâncias – por exemplo abas ou janelas - para acesso a páginas e serviços ou qualquer informação deve ser de escolha do usuário. Por exemplo, deve-se evitar usar o atributo `target="_blank"` em links, sem que o usuário seja avisado.

Caso seja necessário abrir uma nova aba ou janela, informe no próprio link ao usuário que isso ocorrerá.

```
<a href="pagina_externa.html" target="_blank">Ir para página  
externa (abre nova janela)</a>
```

3.2. Comportamento (Document Object Model – DOM)

3.2.1. Disponibilizar todas as funções da página via teclado

Ver WCAG 2.0 Critérios de Sucesso 2.1.1, 2.1.2 (Princípio 2: Operável)

Devemos nos certificar que todos os comportamentos que são engatilhados pelo mouse também sejam pelo teclado. OBS: Alguns manipuladores de eventos são dispositivos-independentes, ou seja, se aplicam a qualquer dispositivo (mouse, teclado ou outro), como é o caso de: `onFocus`, `onBlur`, `onSelect`, `onChange`, e `onClick` (quando o `onClick` for utilizado em um



link ou elemento de formulário). Veja um exemplo abaixo de um botão que engatilha o evento de clique, podendo ser chamado pelo mouse ou pelo teclado.

```
<button id="btn" type="button" class="btn btn-primary">Clique em mim!</button>
```

```
document.getElementById('btn')  
.addEventListener('click', function(){  
  alert('Fui clicado!');  
});
```

3.2.2. Não criar páginas com atualização automática periódica

Ver WCAG 2.0 Critério de Sucesso 3.2.5 (Técnicas SVR1 e H76) (Princípio: Compreensível)

A atualização automática periódica – muito utilizada por canais de notícias - é comumente realizada através do uso do atributo `http-equiv` com conteúdo “refresh” da elemento meta no HEAD do documento (ex: `<meta http-equiv="refresh" content="30" />`, onde a página será recarregada a cada 30 segundos). O problema dessa atualização é que ela tira do usuário sua autonomia em relação à escolha (semelhante a abertura de novas instâncias em navegadores) e podem confundir e desorientar os usuários, especialmente usuários que utilizam leitores de tela.

3.2.3. Não utilizar redirecionamento automático de páginas

Ver WCAG 2.0 Critério de Sucesso 3.2.5 (Técnicas SVR1 e H76) (Princípio: Compreensível)

Não devem ser utilizadas marcações para redirecionar a uma nova página, como o uso do atributo `http-equiv` com conteúdo “refresh” do elemento META. Ao invés disso, deve-se configurar o servidor para que o redirecionamento seja transparente para o usuário.



3.2.4. Não incluir situações com intermitência de tela

Ver WCAG 2.0 Critério de Sucesso 2.3.1 (Princípio: Operável)

Não devem ser utilizados efeitos visuais piscantes, intermitentes ou cintilantes. Em pessoas com epilepsia fotosensitiva, o cintilar ou piscar pode desencadear um ataque epilético. A exigência dessa diretriz aplica-se também para propaganda de terceiros inserida na página.



Exemplo de intermitência de imagem

Fonte: <http://emag.governoeletronico.gov.br/imagens/intermitencia.png>

3.2.5. Assegurar o controle do usuário sobre as alterações temporais do conteúdo

Ver WCAG 2.0 Critério de Sucesso 2.2.2 (Princípio: Operável)

Conteúdos como *slideshows*, que “se movem”, rolagens, movimentações em geral ou animações não devem ser disparadas automaticamente sem o controle do usuário, mesmo em propagandas na página. Ao usuário deve ser repassado o controle sobre essas movimentações (quer seja por escolha de preferência de visualização da página, quer por outro método qualquer acessível a usuário com deficiência). Além disso, o usuário deve ser capaz de parar e reiniciar conteúdos que se movem, sem exceção.



Exemplo de *slideshow*, com mecanismos de navegação entre os slides.

Fonte: <https://getbootstrap.com/docs/4.0/components/carousel/>

3.3. Conteúdo / Informação

3.3.1. Identificar o idioma principal da página

Ver WCAG 2.0 Critério de Sucesso 3.1.1 (Princípio: Compreensível)

Deve-se identificar o principal idioma utilizado nos documentos. A identificação é feita por meio do atributo `lang` do HTML e, para documentos XHTML, é utilizado o `xml:lang`. Ele deve ser declarado em todas as páginas, pois além de auxiliar na acessibilidade do conteúdo, também permite melhor indexação pelos motores de busca.

```
<html lang="pt-BR">
```

3.3.2. Informar mudança de idioma no conteúdo

Ver WCAG 2.0 Critério de Sucesso 3.1.2 (Princípio: Compreensível)

Se algum elemento de uma página possuir conteúdo em um idioma diferente do principal, este deverá estar identificado pelo atributo **lang**. Essa recomendação não se aplica para nomes próprios ou termos técnicos que sejam compreendidos no contexto.

```
<p lang="de">
```

```
Da dachte der Herr daran, ihn aus dem Futter zu schaffen,  
aber der Esel merkte, daß kein guter Wind wehte, lief fort
```



und machte sich auf den Weg nach Bremen: dort, meinte er, könnte er ja Stadtmusikant werden.

</p>

3.3.3. Informar mudança de idioma no conteúdo

Ver WCAG 2.0 Critério de Sucesso 2.4.2 (Princípio: Operável)

O título da página deve ser descritivo e informativo, devendo representar o conteúdo principal da página, já que essa informação será a primeira lida pelo leitor de tela, quando o usuário acessar a página. O título é informado pelo elemento <title> e deve preferencialmente seguir a estrutura recomendada pelo [ePWG](#), que é [assunto principal da página] – [nome do sítio ou sistema] sem palavras extras.

Exemplos:

<title> Publicações - Ministério da Educação </title>
<title> Contato - Projeto Acessibilidade Virtual </title>

3.3.4. Informar o usuário sobre sua localização na página

Ver WCAG 2.0 Critério de Sucesso 2.4.8 (Princípio: Operável)

Deverá ser fornecido um mecanismo que permita ao usuário orientar-se dentro de um conjunto de páginas, permitindo que ele saiba onde está no momento. Assim, poderá ser utilizado o recurso de **breadcrumbs**, que são links navegáveis em forma de lista hierárquica os quais permitem que o usuário saiba qual o caminho percorrido até chegar à página em que se encontra no momento.

Exemplo: Um usuário navegando por um site de uma universidade encontra-se na seção de editais, que está dentro do menu “Ensino”. Acima do conteúdo, é disponibilizada a seguinte *Breadcrumb*:

Você está em: [Página inicial](#) > [Ensino](#) > Editais



```
<nav aria-label="Breadcrumb">
  <p id="breadcrumb-label">Você está em:</p>
  <ol class="breadcrumb" aria-labelledby="breadcrumb-label">
    <li class="breadcrumb-item"><a href="#">Página
inicial</a></li>
    <li class="breadcrumb-item"><a href="#">Ensino</a></li>
    <li class="breadcrumb-item active">Editais</li>
  </ol>
</nav>
```

OBS¹: No breadcrumb, todas as páginas do caminho, com exceção da qual está o usuário (posição atual), deverão estar implementadas como links e contidas dentro de uma lista.

OBS²: Você pode colocar uma tecla de acesso rápido (*accesskey*) para o *breadcrumb*, para o usuário poder com facilidade saber aonde ele está e qual o caminho que ele percorreu.

3.3.5. Descrever links clara e sucintamente

Ver WCAG 2.0 Critérios de Sucesso 2.4.4, 2.4.9 (Princípio: Operável)

Deve-se identificar claramente o destino de cada link, informando, inclusive, se o link remete a outro sítio. Além disso, é preciso que o texto do link faça sentido mesmo quando isolado do contexto da página.

Exemplo **incorreto**, mas comum :

[Clique aqui](#) para saber mais a respeito de acessibilidade.

Exemplo **correto**:

[Saiba mais a respeito de acessibilidade.](#)

Quando for um link para um download de arquivo, podemos fazer a seguinte forma:

[Manual sobre recomendações para descrição de imagens acessíveis \(.odt 150Kb\)](#)



3.3.6. Fornecer alternativa em texto para as imagens do website

Ver WCAG 2.0 Critério de Sucesso 1.1.1 (Técnica G95) (Princípio: Perceptível)

Leitura complementar: [O uso correto do texto alternativo \(documento pdf - 239 KB\)](#)

Deve ser fornecida uma descrição para as imagens da página, utilizando-se, para tanto o atributo **alt**. Entretanto ela deve ser breve e não deve ultrapassar dos **100 caracteres** ([Fonte WCAG 2.0](#)). Para descrever uma imagem com mais detalhes, pode-se fazer uso do atributo **aria-describedby**, tal como no exemplo abaixo.



Exemplo com HTML 5:

```
<figure role="group">  
    
  <figcaption id="caption" class="sr-only">  
    Figura 1: Imagem da árvore que deu  
    origem ao nome do nosso país: Pau-brasil  
  </figcaption>
```

```
</figure>
```

O atributo **aria-describedby** é sempre lido após o texto contido no atributo **alt**. Neste caso, o ideal é que não haja redundância em ambos os atributos.

Foram incorporados ao HTML5 os elementos `<figure>` e `<figcaption>`, que são utilizados em conjunto com o objetivo de agrupar uma imagem `` com a sua legenda, a qual aparecerá visualmente. Podemos adicionar uma role de agrupamento ao elemento `<figure>` (`role="group"`) para informar isso ao leitor de tela.

Podemos esconder elementos ou textos que apenas são úteis para



usuários deficientes visuais. Para isso utilizamos a classe CSS (**sr-only**) no elemento `<figcaption>`.

```
/* Extraído da biblioteca Bootstrap */  
.sr-only {  
  position: absolute;  
  width: 1px;  
  height: 1px;  
  padding: 0;  
  margin: -1px;  
  overflow: hidden;  
  clip: rect(0,0,0,0);  
  border: 0;  
}
```

Para imagens decorativas, aquelas que fazem parte da estilização do *website*, podemos colocar um atributo **alt** nulo (`alt=""`) e o leitor de tela irá ignorá-las.

3.3.7. Disponibilizar documentos em formatos acessíveis

Sem critérios de sucesso correspondentes no WCAG 2.0

Os documentos devem ser disponibilizados preferencialmente em HTML. Também podem ser utilizados arquivos para download no formato ODF, tomando-se os cuidados para que sejam acessíveis. Se um arquivo for disponibilizado em PDF, deverá ser fornecida uma alternativa em HTML ou ODF. Pode-se, também, utilizar a funcionalidade de verificação e correção de acessibilidade nos documentos das ferramentas do pacote Microsoft Office 2010 ou mais atual.

3.3.8. Em tabelas, utilizar títulos e resumos de forma apropriada

Ver WCAG 2.0 Critério de Sucesso 1.3.1 (Técnicas H43 e H63) (Princípio: Perceptível)

O título da tabela deve ser definido pelo elemento `<caption>` e deve ser o primeiro elemento utilizado após a declaração do elemento `<table>`. Em



casos de **tabelas extensas**, deve ser fornecido um **resumo** de seus dados através do atributo `summary` que deve ser declarado no elemento `<table>`.

```
<table summary="Esta tabela exibe os copos de café consumidos por
dia para cada programador, durante o mês de Janeiro.">
  <caption>Copos de café por Programador</caption>
  ...
</table>
```

[Veja mais sobre títulos e resumos de tabela na W3C](#)

3.3.9. Associar células de dados às células de cabeçalho

Ver WCAG 2.0 Critério de Sucesso 1.3.1 (Técnicas H43 e H63) (Princípio: Perceptível)

Em tabelas de dados simples, o uso apropriado do elemento `<th>` para os cabeçalhos e do elemento `<td>` para as células de dados é essencial para torná-las acessíveis. Para incrementar a acessibilidade, deve-se utilizar os elementos `<thead>`, `<tbody>` e `<tfoot>`, para agrupar as linhas de cabeçalho, do corpo da tabela e do final, respectivamente, com exceção de quando a tabela possuir apenas o corpo, sem ter seções de cabeçalho e rodapé. O W3C sugere utilizar o `<tfoot>` antes do `<tbody>` dentro da definição `<table>` para que o agente de usuário possa renderizar o rodapé antes de receber todas (potencialmente numerosas) linha de dados.

```
<table>
  <caption>...</caption>
  <thead>
    ...
  </thead>

  <tfoot>
    ...
  </tfoot>

  <tbody>
    ...
  </tbody>
</table>
```



3.3.10. Garantir a leitura e compreensão das informações

Ver WCAG 2.0 Critério de Sucesso 3.1.5 (Princípio: Compreensível)

O texto de um *website* deve ser de fácil leitura e compreensão, não exigindo do usuário um nível de instrução mais avançado do que o ensino fundamental completo. Quando o texto exigir uma capacidade de leitura mais avançada, deve ser disponibilizado informações suplementares que expliquem ou ilustrem conteúdo principal. Outra alternativa é versão simplificada do conteúdo em texto.

Existem algumas técnicas que auxiliam na melhora da inteligibilidade de textos, como, por exemplo:

- Desenvolver apenas um tópico por parágrafo;
- Utilizar sentenças organizadas de modo simplificado para o propósito do conteúdo (sujeito, verbo e objeto, preferencialmente);
- Dividir sentenças longas em sentenças mais curtas;
- Evitar o uso de jargão, expressões regionais ou termos especializados que possam não ser claros para todos;
- Utilizar palavras comuns no lugar de outras pouco familiares;
- Utilizar listas de itens ao invés de uma longa série de palavras ou frases separadas por vírgulas;
- Fazer referências claras a pronomes e outras partes do documento;
- Utilizar, preferencialmente, a voz ativa.

3.3.11. Disponibilizar uma explicação para siglas, abreviaturas e palavras incomuns

Ver WCAG 2.0 Critérios de Sucesso 3.1.3 e 3.1.4 (Princípio: Compreensível)

Ao menos na primeira ocorrência de siglas, abreviaturas ou palavras incomuns (ambíguas, desconhecidas ou utilizadas de forma muito específica), deve ser disponibilizada sua explicação ou forma completa. Essa explicação pode estar expressa no próprio texto, pode estar presente em um glossário ou, então, através da utilização do elemento `abbr`, conforme exemplos a seguir.

Exemplo 1:

```
<p>A ONU (Organização das Nações Unidas) é um órgão internacional.</p>
```




Exemplo 2:

```
<p>Bem-vindo à <abbr title="World Wide Web" lang="en">WWW</abbr>!</p>
```

3.4. Apresentação / Design

3.4.1. Oferecer contraste mínimo entre plano de fundo e primeiro plano

Ver WCAG 2.0 Critério de Sucesso 1.4.3 (Princípio: Perceptível)

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

Não deverão ser utilizadas imagens atrás do texto (*background*), pois acabam por dificultar a leitura e desviar a atenção do usuário.

- O mínimo de contraste recomendado pelo eMAG é a taxa **4,5 : 1**.
- Para alto contraste, o mínimo recomendado é a taxa **7 : 1**.

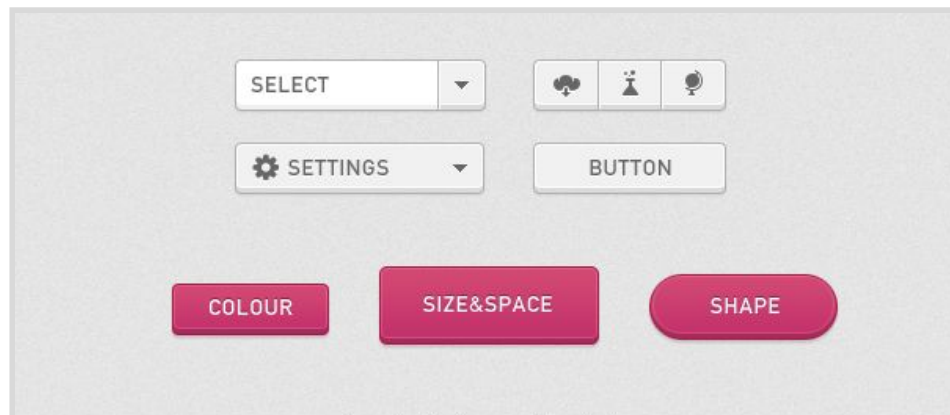
É possível verificar o contraste de cores por meio de ferramentas online como esta:

<https://webaim.org/resources/contrastchecker/>

3.4.2. Não utilizar apenas cor ou outras características sensoriais para diferenciar elementos

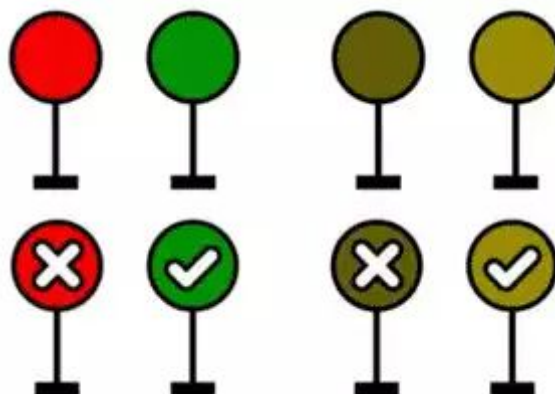
Ver WCAG 2.0 Critérios de Sucesso 1.3.3 e 1.4.1 (Princípio: Perceptível)

A cor ou outras características sensoriais, como forma, tamanho, localização visual, orientação, símbolos ou textos não devem ser utilizadas como o único meio para transmitir informações, indicar uma ação, pedir uma resposta ao usuário ou distinguir um elemento visual.



Fonte:

<https://webdesign.tutsplus.com/pt/articles/principles-for-successful-button-design--webdesign-6094>



Fonte: <http://emag.governoeletronico.gov.br/imagens/figura20.png>

A cor não deve ser o único elemento de diferenciação. A esquerda sinais com a visão normal, à direita os mesmo sinais vistos por quem tem daltonismo.

3.4.3. Permitir redimensionamento sem perda de funcionalidade

Ver WCAG 2.0 Critério de Sucesso 1.4.4 (Princípio: Perceptível)

A página deve continuar legível e funcional mesmo quando redimensionada para até 200%. Assim, é preciso garantir que, quando a página for redimensionada, não ocorram sobreposições nem o aparecimento de uma barra horizontal.



A página também deve continuar legível e funcional em outros tamanhos de tela, verificando sempre a responsividade do *layout*.



Fonte: <http://emag.governoeletronico.gov.br/imagens/figura24.png>

3.4.4. Possibilitar que o elemento com foco seja visualmente evidente

Ver WCAG 2.0 Critério de Sucesso 2.4.7 (Princípio: Operável)

A área que recebe o foco pelo teclado deve ser claramente marcada, devendo a área de seleção ser passível de ser clicada.

Por padrão, links e elementos de formulário já apresentam a borda destacada ao receberem o foco do teclado. Essa borda pode ser modificada via CSS para melhorar o destaque, mas não deverá ser removida. Recomenda-se que a espessura mínima da borda seja de 2px.

```
a:focus, a:hover {  
    border: 2px solid #00F;  
}
```

A pseudo-classe `:focus` é utilizada para definir o estilo de qualquer elemento HTML que receber o foco do teclado, como links e elementos de formulário. A pseudo-classe `:hover` é utilizada para definir o estilo de um elemento quando passa-se o mouse sobre ele.



3.5. Formulários

3.5.1. Associar etiquetas aos seus campos

Ver WCAG 2.0 Critério de Sucesso 1.3.1 (Técnica H44) (Princípio: Perceptível)

As etiquetas de texto `<label>` devem estar associadas aos seus campos (elementos `<input>`, `<select>` e `<textarea>`, à exceção do elemento `<button>`) correspondentes no formulário, através dos atributos `for` do `<label>` e `id` do `<input>`, os quais deverão ter o mesmo valor.

```
<label for="nome">Nome: </label>
<input type="text" name="nome" id="nome">
```

Quando não for possível ou prático de se adicionar um elemento `<label>` no cenário que estiver codificando, você poderá fazer uso do atributo `aria-label` para escrever o rótulo para o elemento.

```
<button aria-label="Fechar janela de diálogo"
onclick="dialog.close()">X</button>
```

3.5.2. Estabelecer uma ordem lógica de navegação

Ver WCAG 2.0 Critério de Sucesso 2.4.3 (Princípio: Operável)

Os elementos do formulário devem ser distribuídos corretamente através do código HTML, criando, assim, uma sequência lógica de navegação. Assim, os formulários devem primeiro ser codificados considerando a ordem lógica de navegação para depois serem organizados visualmente via CSS.

3.5.3. Não provocar automaticamente alteração no contexto

Ver WCAG 2.0 Critério de Sucesso 3.2.2 (Princípio: Compreensível)

Quando um elemento de formulário receber o foco, não deve ser iniciada uma mudança automática na página que confunda ou desorienta o usuário. Assim, as mudanças devem ocorrer através do acionamento de um botão.



3.5.4. Fornecer instruções para entrada de dados

Ver WCAG 2.0 Critério de Sucesso 3.3.2 (Princípio: Compreensível)

Para conteúdo que exigir entrada de dados por parte do usuário, devem ser fornecidas quando necessário, instruções de preenchimento juntamente com as etiquetas (elemento `<label>`).

Recomenda-se que a entrada de dados seja facilitada, como a exclusão de caracteres especiais em campos numéricos (Ex: número de documentos como CPF, datas, moeda), e a simplificação de campos.

Exemplo para entrada de datas:

```
<label for="data">Data (dd-mm-aaaa)</label>  
<input type="text" id="data" name="data"/>
```

Exemplo para entrada obrigatória de números:

```
<label for="cpf">CPF (somente números)  
  
</label>  
<input type="text" id="cpf" name="cpf" required>
```

Colocamos um elemento imagem no exemplo acima para indicar visualmente que o elemento é obrigatório e além disso informamos um `alt` para que a informação também esteja disponível para os leitores de tela. O atributo `required` no campo de entrada também nos ajuda nesse quesito da obrigatoriedade.

3.5.5. Identificar e descrever erros de entrada de dados e confirmar o envio das informações

Ver WCAG 2.0 Critério de Sucesso 3.3.1 (Princípio: Compreensível)

Quando um erro de entrada de dados for automaticamente detectado, o item que apresenta erro deve ser identificado e descrito ao usuário por texto.



Exemplo:

Avisos - dados inválidos

- Usuário inválido, mínimo de 4 caracteres!
- Nome completo inválido, o campo não pode estar em branco!
- Endereço inválido, mínimo de 4 caracteres!
- CEP inválido, somente é permitido a entrada de números!

Dados cadastrais

- Usuário
Mínimo 5 caracteres / sem espaços

Nome completo

Endereço

- Logradouro
- CEP
Somente números

Expandir demais campos

Cadastrar Limpar Cancelar

Fonte: <http://emag.governoeletronico.gov.br/imagens/figura39.png>

Uma forma de deixarmos os erros evidentes para os usuários deficientes visuais é colocar em foco a lista de erros reportados, logo após o envio do formulário. Outra forma que pode ser feita, caso a validação seja dinâmica, é focar a mensagem de erro do elemento atual, logo após a inserção do dado no campo. Tecnicamente isso ativará um evento **onBlur** no Javascript e o elemento que contém a mensagem de erro terá foco.

3.5.6. Agrupar campos de formulário

Ver WCAG 2.0 Critério de Sucesso 1.3.1 (Técnicas H71 e H85) (Princípio: Perceptível)

É recomendado que os campos com informações relacionadas sejam



agrupadas utilizando o elemento `<fieldset>`, principalmente em formulários longos. O agrupamento deverá ser feito de maneira lógica, associando o elemento `<legend>` explicando claramente o propósito ou natureza dos agrupamentos.

Exemplo em formulários:

```
<form method="post" action="...">
  <fieldset>
    <legend>Dados Pessoais</legend>
    <label for="nome">O seu Nome: </label>
    <input type="text" name="nome" id="nome">
    ...
  </fieldset>

  <fieldset>
    <legend>Dados Profissionais</legend>
    <label for="profissao">Sua profissão:</label>
    <input type="text" id="profissao" name="profissao">
    ...
  </fieldset>

  <fieldset>
    <legend>Informações de Contato</legend>
    <label for="email">E-mail: </label>
    <input type="text" id="email" name="email">
    ...
  </fieldset>
</form>
```

Exemplo em campos de seleção (elemento `<optgroup>` para agrupar as opções):

```
<label for="instituto">Qual a sua instituição?</label>
<select id="instituto" name="instituto">
  <optgroup label="Rio Grande do Sul">
    <option value="ifrs">IFRS</option>
    <option value="ifsul">IFSUL</option>
```



```
<option value="iffarroupilha">IFFarroupilha</option>
</optgroup>

<optgroup label="Santa Catarina">
  <option value="ifsc">IFSC</option>
  <option value="ifc">IFC</option>
</optgroup>

<optgroup label="Paraná">
  <option value="ifpr">IFPR</option>
</optgroup>
</select>
```




4. Elementos padronizados de acessibilidade digital

Os elementos padronizados de acessibilidade digital que devem estar presentes em todos os sítios do governo federal para facilitar o acesso ao cidadão, são:

- Teclas de atalho
- Possibilidade de visualizar o site em alto contraste
- Barra de acessibilidade
- Apresentação do mapa do website
- Página com a descrição dos recursos de acessibilidade

Esses elementos já fazem parte da identidade digital de governo para mais detalhes dessa iniciativa visite: <http://portalpadrao.gov.br/manuais> ou a [seção 4 do Documento da eMAG](#).



5. Práticas desaconselhadas

Estas práticas, apesar de comuns, configuram-se não só como empecilhos para o acesso de pessoas com deficiência, mas também, o acesso por dispositivos móveis. Portanto, devem ser evitadas durante o desenvolvimento. O uso de qualquer uma dessas práticas tem um impacto negativo significativo na experiência de uso do usuário.

- Uso de animações e aplicações FLASH
- Uso de CAPTCHAS tradicionais em formulários (Verificar CAPTCHAS acessíveis como o [reCAPTCHA V2](#))
- Tabelas para fins de diagramação
- Atualizações automáticas periódicas
- Elementos e atributos considerados depreciados pelo W3C. Exemplos: frame, applet, blink, marquee, basefont, center, dir, align, font, isindex, menu, etc.