

eMAG - Modelo de Acessibilidade em Governo Eletrônico

[Programa de Governo Eletrônico Brasileiro](#)

Versão 3.1 - Abril 2014

[Download do Modelo em PDF \(2,2 MBytes\)](#)

Esta obra está licenciada por uma licença [Creative Commons - Atribuição 4.0 Internacional \(CC BY 4.0\)](#).

Agradecimentos

O Departamento de Governo Eletrônico agradece à: Bruna Poletto Salton, Jucélia Poletto Almeida, Agebson Rocha Façanha, André Luiz Rezende, Andréa Poletto Sonza, Ângela Guimarães, Felipe Zap, Gleison Samuel do Nascimento, Jorge Fiore de Oliveira Junior, Júlia Marques Carvalho da Silva, Marco Antônio de Queiroz (in memoriam), Marcus Vinícius Bennett Ferreira, Maurício Covolan Rosito, Reinaldo Ferraz, Renato Busatto, Ricardo Moro, Ueslei Paterno e Woquiton Fernandes, que tornaram a versão 3.0 do eMAG possível. Por sua contribuição na versão 3.1 agradecemos à Anderson Luís Porto Costa, Everaldo Carniel, Lael Nervis, Jason Pilotti e Lívio Siqueira Lima.

O DGE também agradece a:

André Pimenta Freire, Carol Scarton, Diego Roger Ramos Freitas, Edercio Marques Bento, Jorge Fernandes, Leonelo Dell Anhol Almeida, Maria Cecília Calani Baranauskas, Thiago Prado de Campos, Vagner Figueredo de Santana.

Pelas contribuições, via consulta pública, com sugestões, esclarecimentos e correções para o presente documento.

Equipe do DGE – Departamento de Governo Eletrônico

Ana Elisa Dourado Salina Gabriel – Analista em Tecnologia da Informação

César Gonçalves do Bomfim – Analista em Tecnologia da Informação

Edson Santos Martins Leite - Analista em Tecnologia da Informação

Fernanda Hoffmann Lobato – Coordenadora Geral

Higo Gomes Mesquita - Analista em Tecnologia da Informação

[Voltar ao sumário](#)

1. Introdução

Uma das principais atribuições do Governo Federal é promover a inclusão social, com distribuição de renda e diminuição das desigualdades. Entre as diversas iniciativas que visam atingir esse objetivo, o governo investe no uso adequado e coordenado da tecnologia porque compreende a inclusão digital como caminho para a inclusão social.

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. Mas a que se deve este fenômeno? Basicamente, deve-se ao seu potencial para atingir instantaneamente um grande número de pessoas, independentemente de localização geográfica e de contexto sócio-cultural.

Neste contexto, a inacessibilidade de sítios eletrônicos exclui uma parcela significativa da população brasileira do acesso às informações veiculadas na internet. O governo brasileiro - tendo em vista suas atribuições - não pode aceitar tal situação na entrega de informações e serviços sob sua responsabilidade.

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

As recomendações do eMAG permitem que a implementação da acessibilidade digital seja conduzida de forma padronizada, de fácil implementação, coerente com as necessidades brasileiras e em conformidade com os padrões internacionais. É importante ressaltar que o eMAG trata de uma versão especializada do documento internacional WCAG (*Web Content Accessibility Guidelines: Recomendações de Acessibilidade para Conteúdo Web*) voltado para o governo brasileiro, porém o eMAG não exclui qualquer boa prática de acessibilidade do WCAG.

[Voltar ao sumário](#)

1.1 O acesso de pessoas com deficiência

O computador e a Internet representam um enorme passo para a inclusão de pessoas com deficiência, promovendo autonomia e independência. Mas como pessoas com deficiência utilizam o computador?

No que se refere a acesso ao computador, as quatro principais situações vivenciadas por usuários com deficiência são:

Sumário

[Agradecimentos](#)

[1. Introdução](#)

[1.1 O acesso de pessoas com deficiência](#)

[1.2 Sobre as versões do eMAG](#)

[1.3 Alterações na versão 3.1](#)

[1.4 Legislação](#)

[2. O processo para desenvolver um sítio acessível](#)

[2.1 Primeiro passo: Padrões Web](#)

[2.2 Segundo passo: Recomendações de Acessibilidade](#)

[2.3 Terceiro passo: Avaliação de Acessibilidade](#)

[2.4 Manutenção da acessibilidade](#)

[2.5 Recursos, ferramentas e leitura complementar](#)

[3. Recomendações de Acessibilidade](#)

[3.1 Marcação](#)

[3.2 Comportamento \(DOM\)](#)

[3.3 Conteúdo/Informação](#)

[3.4 Apresentação/Design](#)

[3.5 Multimídia](#)

[3.6 Formulário](#)

[4. Elementos padronizados de acessibilidade digital no Governo Federal](#)

[5. Práticas desaconselhadas](#)

[6. Glossário](#)

[7. Anexo 1 – Tabela de Contraste de Cores](#)



Acessível
em Libras

- Acesso ao computador sem mouse: no caso de pessoas com deficiência visual, dificuldade de controle dos movimentos, paralisia ou amputação de um membro superior;
- Acesso ao computador sem teclado: no caso de pessoas com amputações, grandes limitações de movimentos ou falta de força nos membros superiores;
- Acesso ao computador sem monitor: no caso de pessoas com cegueira;
- Acesso ao computador sem áudio: no caso de pessoas com deficiência auditiva.

Muitas vezes, a deficiência não é severa o suficiente a ponto de tornar-se uma barreira à utilização do computador. Entretanto, na maioria das páginas da Web, as pessoas cegas ou com baixa visão, pessoas com deficiência auditiva, com dificuldade em utilizar o mouse, por exemplo, encontram barreiras de acessibilidade que dificultam ou impossibilitam o acesso aos seus conteúdos.

Muitas pessoas também apresentam outras limitações relacionadas à memória, resolução de problemas, atenção, compreensão verbal, leitura e linguística, compreensão matemática e compreensão visual. Uma pessoa com dislexia, por exemplo, pode apresentar dificuldade de leitura de uma página devido a um desenho inadequado. Por isso, um sítio desenvolvido considerando a acessibilidade deve englobar diferentes níveis de escolaridade, faixa etária e pouca experiência na utilização do computador, bem como ser compatível com as diversas tecnologias utilizadas para acessar uma página da Web.

Um dos aliados das pessoas com deficiência para o uso do computador são os recursos de tecnologia assistiva, que auxiliam na realização de tarefas antes muito difíceis ou impossíveis de realizar, promovendo, desta maneira, a autonomia, independência, qualidade de vida e inclusão social de pessoas com deficiência.

Existe atualmente uma enorme gama de recursos de tecnologia assistiva, desde artefatos simples até objetos ou softwares mais sofisticados e específicos, de acordo com a necessidade de cada pessoa. Uma pessoa com limitado movimento das mãos, por exemplo, pode utilizar um teclado adaptado que contém teclas maiores ou um mouse especial para operar o computador. Já as pessoas com baixa visão podem recorrer a recursos como ampliadores de tela, enquanto usuários cegos podem utilizar softwares leitores de tela para fazer uso do computador.

Apesar de sua enorme importância na promoção da acessibilidade às pessoas com deficiência, os recursos de tecnologia assistiva, por si só, não garantem o acesso ao conteúdo de uma página da Web. Para tal, é necessário que a página tenha sido desenvolvida de acordo com os padrões Web (Web Standards) e as recomendações de acessibilidade, os quais serão abordados ao longo deste documento.

[Voltar ao sumário](#)

1.2 Sobre as versões do eMAG

Para a elaboração da versão 2.0 do eMAG foi realizado um estudo das regras de acessibilidade através de um método comparativo entre as normas adotadas por diversos países, como a Section 508 do governo dos Estados Unidos, os padrões CLF do Canadá, as diretrizes irlandesas de acessibilidade e documentos de outros países, entre eles Portugal e Espanha. Também foi realizada uma análise detalhada das regras e pontos de verificação do órgão internacional WAI/W3C, presentes na WCAG 1.0. O governo brasileiro, comprometido com a inclusão, buscou, através da elaboração do Modelo de acessibilidade do governo eletrônico, facilitar o acesso para todas as pessoas às informações e serviços disponibilizados nos sítios e portais do governo.

O eMAG foi desenvolvido em 2004 baseado no estudo de 14 normas existentes em outros países acerca de acessibilidade digital. Dentre as normas analisadas estavam a Section 508 do governo dos Estados Unidos, os padrões CLF do Canadá, as diretrizes irlandesas de acessibilidade e documentos de outros países como Portugal e Espanha. Também foi realizada uma análise detalhada das regras e pontos de verificação do órgão internacional WAI/W3C, presentes na WCAG 1.0.

Assim, a primeira versão do eMAG, a 1.4, elaborada pelo Departamento de Governo Eletrônico em parceria com a ONG Acessibilidade Brasil, foi disponibilizada para consulta pública em 18 de janeiro de 2005, e a versão 2.0, com as alterações propostas, foi disponibilizada em 14 de dezembro do mesmo ano.

Em 2007, a Portaria nº 3, de 7 de maio, institucionalizou o eMAG no âmbito do sistema de Administração dos Recursos de Informação e Informática – SISPI, tornando sua observância obrigatória nos sítios e portais do governo brasileiro.

As versões 1.4 e 2.0 eram divididas em dois documentos:

- A visão do cidadão, voltada a todos os cidadãos brasileiros e gestores, apresentava o modelo de acessibilidade de forma simples e tinha cerca de 16 páginas.
- A cartilha técnica, voltada a desenvolvedores de sítios, apresentava detalhadamente a proposta de implementação das recomendações de acessibilidade em sítios do governo, tinha 44 páginas com 57 recomendações de boas práticas e era voltada a área técnica. As recomendações estavam divididas em três níveis de prioridades de acordo com o WCAG 1.0.

A divisão do eMAG em dois documentos apresentou alguns inconvenientes durante o processo de disseminação do Modelo, como a dificuldade das pessoas entenderem as áreas da Visão do Cidadão e seu relacionamento com a aplicação efetiva da acessibilidade. O aprendizado durante os seis anos da versão 2.0



Acessível
em Libras

do eMAG e o lançamento da versão 2.0 do WCAG em 2008 marcaram o caminho para a revisão do Modelo.

A revisão do modelo 2.0, que resultou na versão 3.0, foi desenvolvida através da parceria entre o Departamento de Governo Eletrônico e o Projeto de Acessibilidade Virtual da RENAPI (Rede de Pesquisa e Inovação em Tecnologias Digitais).

A elaboração da versão 3.0 foi embasada na versão anterior do eMAG, apoiando-se na WCAG 2.0, lançada em dezembro de 2008, e considerando as novas pesquisas na área de acessibilidade à Web, sendo a minuta enviada para consulta de 30 especialistas nas diversas áreas da acessibilidade e tipos de deficiência. Apesar de utilizar a WCAG como referência, e estar alinhado a esta, o eMAG 3.0 foi desenvolvido e pensado para as necessidades locais, visando atender as prioridades brasileiras.

Seguindo a diretriz do programa de Governo Eletrônico de promover a Cidadania, o documento-proposta passou por Consulta Pública no período de novembro de 2010 a janeiro de 2011, recebendo contribuições tanto pelo sistema de Consulta Pública do Portal do Programa, quanto por e-mail. Em 21 de setembro de 2011, a versão 3.0 do eMAG foi lançada oficialmente no evento Acessibilidade Digital – Um Direito de Todos.

A versão 3.0 do eMAG unificou em apenas um documento o Modelo, retirando a separação entre visão técnica e visão do cidadão. Outra decisão foi o abandono dos níveis de prioridade A, AA e AAA, visto que o padrão é voltado as páginas do Governo, não sendo permitido exceções com relação ao cumprimento das recomendações. Além disso, no eMAG 3.0 foi incluída a seção chamada “Padronização de acessibilidade nas páginas do governo federal”, com o intuito de padronizar elementos de acessibilidade que devem existir em todos os sítios e portais do governo.

[Voltar ao sumário](#)

1.3 Alterações na versão 3.1

A **versão 3.1 do eMAG** apresenta melhorias no conteúdo do texto para torná-lo mais compreensível.

O subitem O processo para desenvolver um sítio acessível, ganhou um capítulo próprio.

O capítulo 4 teve sua terminologia alterada de “Padrões de acessibilidade digital no Governo Federal” para “Elementos padronizados de acessibilidade digital no Governo Federal”. A quantidade de itens também foi alterada, eram 7 e agora são 5, incluindo uma padronização para a primeira folha de contraste e removendo as funcionalidades aumentar e diminuir fonte da barra de acessibilidade, pois os navegadores de internet já possuem estas funcionalidades nativas e são conhecidas pela maioria dos usuários. Outros elementos foram removidos do capítulo como “Apresentação de formulário”, “Conteúdo alternativo para imagens” e “Apresentação de documentos”, pois estes já são contemplados pelas 45 recomendações deste documento, o que causava redundância.

As recomendações foram numeradas de acordo com as seções do eMAG, o que facilitará a manutenção das recomendações do modelo:

- Marcação
- Comportamento (DOM)
- Conteúdo/Informação
- Apresentação/Design
- Multimídia
- Formulário

Outras modificações nas recomendações:

- [Algumas recomendações tiveram seus enunciados refeitos com uma redação mais clara;](#)
- [A recomendação numero 5 foi movida para comportamento, sendo numerada como 2.1;](#)
- [A Recomendação 27 foi movida para sua sequencia lógica próxima sendo numerada como 3.2;](#)
- [A 31 mudou para marcação sendo numerada como 1.8.](#)

Foram inseridos novos exemplos, inclusive com o uso de HTML5 e WAI-ARIA para determinadas recomendações.

Por fim, os links de Recursos e Leitura complementar foram retirados para permitir alterações mais dinâmicas ao conteúdo, que pode ser consultado na página do eMAG no Portal do Programa de Governo Eletrônico:

<http://www.governoeletronico.gov.br/eixos-de-atuacao/governo/acessibilidade/emag-modelo-de-acessibilidade-em-governo-eletronico>

[Voltar ao sumário](#)



Acessível
em Libras

1.4 Legislação

Estão listados os principais documentos, que fazem parte da legislação que norteia o processo de promoção da acessibilidade e a implementação do eMAG:

- 1. Comitê Brasileiro (CB-40) da Associação Brasileira de Normas Técnicas (ABNT), que se dedica à normatização no campo de acessibilidade, atendendo aos preceitos de desenho universal. O Comitê possui diversas comissões, definindo normas de acessibilidade em todos os níveis, desde o espaço físico até o virtual;
- 2. Lei Nº 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas portadoras de deficiência ou com mobilidade reduzida, e dá outras providências.
- 3. Decreto número 5296, de 2 de dezembro de 2004, que regulamenta as leis nº 10.048, de 8 de novembro de 2000, que dá prioridade de atendimento às pessoas que especifica, e 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas com deficiência, e dá outras providências;
- 4. Portaria nº 3, de 7 de maio de 2007, que institucionalizou o eMAG no âmbito do sistema de Administração dos Recursos de Informação e Informática (SISP), tornando sua observância obrigatória nos sítios e portais do governo brasileiro.
- 5. Convenção Internacional sobre os Direitos das Pessoas com Deficiência (2007), de 30 de março de 2007, elaborada pela Nações Unidas, define, em seu artigo 9º, a obrigatoriedade de promoção do acesso de pessoas com deficiência a novos sistemas e tecnologias da informação e comunicação, inclusive à Internet.
- 6. Decreto nº 6949, de 25 de agosto de 2009, que promulga a Convenção Internacional sobre os Direitos das Pessoas com Deficiência, tornando-a um marco legal no Brasil.
- 7. Instrução Normativa MP/SLTI Nº04, de 12 de novembro de 2010, dispõe sobre o processo de contratação de Soluções de Tecnologia da Informação pelos órgãos integrantes do Sistema de Administração dos Recursos de Informação e Informática - SISP do Poder Executivo Federal.
- 8. Lei nº 12.527, de 18 de novembro de 2011 (Lei de Acesso à Informação), que no seu artigo 8º, parágrafo 3º, inciso VIII preconiza que: “Os sítios de que trata o § 2º deverão, na forma de regulamento, atender, entre outros, aos seguintes requisitos: (...) adotar as medidas necessárias para garantir a acessibilidade de conteúdo para pessoas com deficiência”.
- 9. Decreto de nº 7724, de 16 de maio de 2012, que regulamentou a lei de acesso a informação, indica no seu artigo 8º parágrafo 3º: “Os sítios na Internet dos órgãos e entidades deverão, em cumprimento às normas estabelecidas pelo Ministério do Planejamento, Orçamento e Gestão, (...)”.

[Voltar ao sumário](#)

2 . O processo para desenvolver um sítio acessível

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. O processo para desenvolver um sítio acessível é realizado em três passos:

- [1. Seguir os padrões Web;](#)
- [2. Seguir as diretrizes ou recomendações de acessibilidade;](#)
- [3. Realizar a avaliação de acessibilidade.](#)

[Voltar ao sumário](#)

2.1 Primeiro passo: 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 Web. 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.



Acessível
em Libras

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.

Para conhecer as boas práticas em desenvolvimento de sítios de acordo com os padrões veja a página do Escritório Brasileiro do W3C <http://w3c.br/>. A Cartilha de Codificação dos Padrões Web em Governo Eletrônico (ePWG), disponível em <http://www.governoeletronico.gov.br/eixos-de-atuacao/governo/epwg-padroes-web-em-governo-eletronico/cartilha-de-codificacao>, lista uma série de boas práticas.

[Voltar ao sumário](#)

2.2 Segundo passo: 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 sítios) e aos programadores de ferramentas para criação de conteúdo. A principal documentação nessa área é a WCAG (*Web Content Accessibility Guidelines* - http://www.w3.org/TR/#tr_Accessibility_All em inglês) 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), ainda boa parte em status de 'rascunho' chamado WAI-ARIA (*Accessible Rich Internet Applications* – http://w3.org/standards/techs/aria#w3c_all), 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 sítios e portais acessíveis no âmbito do governo federal.

[Voltar ao sumário](#)

2.3 Terceiro passo: Avaliação de Acessibilidade

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.

No caso dos padrões Web, há validadores automáticos. No que diz respeito às diretrizes de acessibilidade, é necessário realizar, inicialmente, uma validação automática, que é realizada através de softwares ou serviços online que ajudam a determinar se um sítio respeitou ou não as recomendações de acessibilidade, gerando um relatório de erros. Uma das ferramentas que podem ser utilizadas é o ASES, avaliador e simulador de acessibilidade em sítios.

É preciso salientar que, apesar de tornarem a avaliação de acessibilidade mais rápida e menos trabalhosa, os validadores automáticos por si só não determinam se um sítio está ou não acessível. Para uma avaliação efetiva, é necessária uma posterior validação manual.

A validação manual é necessária porque nem todos os problemas de acessibilidade em um sítio são detectados mecanicamente pelos validadores. Para a validação manual, são utilizados checklists de validação humana.

Deve-se lembrar que após cada teste, os ajustes devidos devem ser feitos e novamente testados.

Assim, os passos sugeridos para a avaliação de acessibilidade em um sítio são os seguintes:

- 1. Validar os **códigos do conteúdo HTML e das folhas de estilo**;
- 2. Verificar o **fluxo de leitura da página**. A forma mais simples é inibir o CSS, imagens e scripts, lendo apenas o HTML da página. Boa parte dos navegadores possuem ferramentas ou extensões que permitem essa visualização. Outra opção é utilizar navegadores textuais, como o Lynx ou um leitor de tela. Para maiores detalhes, ver documento Descrição dos Leitores de Tela, disponível em: <http://www.governoeletronico.gov.br/eixos-de-atuacao/governo/acessibilidade/material-de-apoio>
- 3. Realizar a validação automática de acessibilidade utilizando o ASES e outros avaliadores automáticos sugeridos no Capítulo 4;
- 4. Realizar a validação manual. A validação manual é uma etapa essencial na avaliação de acessibilidade de um sítio, já que os validadores automáticos não são capazes de detectar todos os problemas de acessibilidade em um sítio, pois muitos aspectos requerem um julgamento humano. Por exemplo, validadores automáticos conseguem detectar se o atributo para descrever imagens foi utilizado em todas as imagens do sítio, mas somente uma pessoa poderá verificar se a descrição da imagem está adequada ao seu conteúdo. Para realizar uma validação manual efetiva, o desenvolvedor deverá ter conhecimento sobre as diferentes tecnologias, as barreiras de acessibilidade enfrentadas por pessoas com deficiência e as técnicas ou recomendações de acessibilidade. A validação manual deve ser feita preferencialmente com dispositivos de tecnologia assistiva como leitores de tela. Deve-se percorrer toda página apenas utilizando teclado, verificando comportamentos, atalhos, folhas alternativas de contraste, se os textos alternativos estão descritos de acordo com a imagem e seu contexto, entre outros. Listas com os itens a serem testados (checklists) na validação humana podem ser encontradas em <http://www.governoeletronico.gov.br/eixos-de-atuacao/governo/acessibilidade/material-de-apoio>.



Acessível
em Libras

- **5. Teste com usuários reais.** 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 sítio 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.

[Voltar ao sumário](#)

2.4 Manutenção da acessibilidade

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 sítio todo quando for feita a atualização do Sistema de Gestão de Conteúdo ou mudança de desenho.

Recomenda-se, também, que constem informações sobre a acessibilidade do sítio (ver Capítulo 4 - Elementos padronizados de acessibilidade digital no Governo Federal), incluindo formas de contato em caso de dificuldade de acesso .

[Voltar ao sumário](#)

2.5 Recursos, ferramentas e leitura complementar

Na página <http://www.governoeletronico.gov.br/eixos-de-atuacao/governo/acessibilidade/material-de-apoio> é possível encontrar ferramentas, aplicativos, cursos, tutoriais e links para outros sites que auxiliam a implementação da acessibilidade.

[Voltar ao sumário](#)

3. Recomendações de Acessibilidade

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 sítios 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.

Para facilitar a implementação das recomendações, no eMAG elas são separadas por seções de acordo com as necessidades de implementação:

- [Marcação](#)
- [Comportamento \(DOM\)](#)
- [Conteúdo/Informação](#)
- [Apresentação/Design](#)
- [Multimídia](#)
- [Formulário](#)

No eMAG, diferente da WCAG internacional, as recomendações de acessibilidade deste documento não estão divididas por níveis de prioridade e sim por área. Por se tratar de recomendações para páginas de governo, todas as recomendações necessárias para determinada situação devem ser seguidas. Assim, se a página é a área de contato, as recomendações de formulário (além das de marcação, conteúdo, etc) devem ser seguidas, se apresentar vídeo, atenção especial deve ser dada as recomendações de multimídia.

[Voltar ao sumário](#)

3.1 Marcação

Recomendação 1.1 – Respeitar os Padrões Web

[Ver WCAG 2.0 Critérios de Sucesso 4.1.1 e 4.1.2](#)

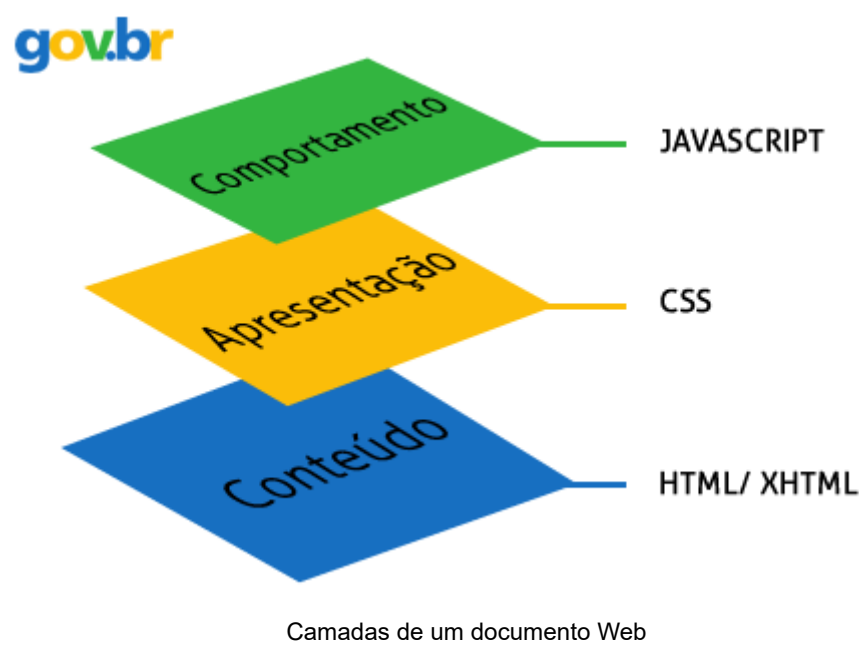
Os Padrões Web são recomendações do W3C (World Wide Web Consortium), as quais são destinadas a orientar os desenvolvedores para o uso de boas práticas que tornam a web acessível para todos, permitindo assim que os desenvolvedores criem experiências ricas, alimentadas por um vasto armazenamento de dados, os quais estão disponíveis para qualquer dispositivo e compatíveis com atuais e futuros agentes de usuário (ex: navegadores).

Quando tratamos de acessibilidade as principais recomendações são as de Web Design e Aplicações (<http://www.w3c.br/Padroes/WebDesignAplicacoes>), que referem-se aos padrões para o desenvolvimento de páginas Web, incluindo HTML5 CSS, SVG, Ajax, e outras tecnologias para Aplicações Web (WebApps), assim como o padrão internacional de acessibilidade WCAG, internacionalização e dispositivos móveis.



Acessível
em Libras

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).



Para mais detalhes a respeito dos padrões de desenvolvimento web, ver a Cartilha de Codificação Padrões Web e-GOV do padrão e-PWG, disponível em: <http://www.governoeletronico.gov.br/eixos-de-atuacao/governo/epwg-padroes-web-em-governo-eletronico/cartilha-de-codificacao>.

[Voltar ao sumário](#)

Recomendação 1.2 – Organizar o código HTML de forma lógica e semântica

[Ver WCAG 2.0 Critério de Sucesso 1.3.1](#)

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.

Assim, marcação semântica adequada deve ser utilizada para designar os cabeçalhos (h1, h2, h3), as listas (ul, ol, dl), texto enfatizado (strong), marcação de código (code), marcação de abreviaturas (abbr), marcação de citações longas (blockquote), etc. Dessa forma, as páginas poderão ser apresentadas e compreendidas sem recursos de estilização, tal como as folhas de estilo. Além disso, o código semanticamente correto é muito importante para usuários com deficiência visual, pois os leitores de tela descrevem primeiro o tipo de elemento e depois realizam a leitura do conteúdo que está dentro desse elemento.

Exemplo correto

<h1>Padrões Web</h1>

Menu 1

Menu 2

<h2>Web Semântica</h2>

<blockquote>

O poder da web está em sua

Exemplo Incorreto

</p>

</p>

<h1>Padrões Web</h1>

<p>Menu 1

<p>Menu 2

<h2>Web Semântica</h2>

<p>

O poder da web está em sua

universalidade.

Ser acessada por todos,

Exemplo correto em HTML5 (utilizando ARIA)

<header role='banner'>

<h1>Padrões Web</h1>

</header>

Menu 1

Menu 2

<h2>Web Semântica</h2>

[Ir ao exemplo navegável Recomendação 1.2](#)

[Voltar ao sumário](#)

Recomendação 1.3 – Utilizar corretamente os níveis de cabeçalho

[Ver WCAG 2.0 Critérios de Sucesso 1.3.1 e 2.4.10](#)

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. Conceitualmente, existem seis níveis de títulos, sendo o H1 o mais alto, ou seja, deverá corresponder ao conteúdo principal da página, assim é recomendável que toda página

tenha apenas um H1. Já os níveis do H2 ao H6 poderão ser utilizados mais de uma vez na página, mas sem excesso e com lógica textual, obedecendo uma hierarquia. Para compreender melhor os níveis de título pode-se tomar como exemplo um sítio de um livro, onde o nome do livro é o H1, os capítulos são H2, os subcapítulos são H3 e assim por diante.

Exemplo

```
<h1>Técnicas culinárias</h1>
<p>A seguir os segredos que facilitam
a vida na cozinha.</p>

<h2>Legumes, folhas e
vegetais</h2>

<h3>Baba do quiabo</h3>
<p>Para eliminar a baba do
quiabo, lave-o ainda inteiro, seque-o e coloque-o numa tigela com um
pouco de suco de limão, deixando repousar durante 15 minutos. Depois
```

[Ir ao exemplo navegável](#) **Recomendação 1.3**

Exemplo de marcação adequada de cabeçalhos

Portal Brasil - Ordem de cabeçalhos e outros elementos semânticos.



[Voltar ao sumário](#)

Recomendação 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](#)

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.

É recomendável disponibilizar o bloco do conteúdo principal da página antes do bloco de menu. Isso facilita o acesso por quem navega via teclado, permitindo um acesso mais ágil, sem a necessidade de navegar por todos os itens de menu antes de chegar ao conteúdo. Apesar de atalhos (como links e teclas) auxiliarem nesse sentido, alguns usuários não sabem e podem ser de difícil utilização para pessoas com deficiência motora. Algumas das formas não funcionam em interfaces mais simples, como o WebVox do DOSVOX. Lembrando que ao utilizar CSS, visualmente, os blocos de menu e conteúdo podem ser dispostos em qualquer local da página.

Exemplo: bloco de conteúdo antes do bloco de menu

```
<body>
  <div id="cabecalho">
    ...
  <div id="conteudo">
  <div id="menu">
    <div id="rodape">
  </body>
```

OBS: Em alguns casos, pode-se forçar a navegação via atributo tabindex. No entanto, se houver a necessidade de utilizar o tabindex, o mesmo deverá ser utilizado com a semântica correta e ser verificado manualmente se o fluxo fornecido é realmente o desejado. O uso do tabindex pode resultar em uma ordem e tabulação inconsistente.

[Voltar ao sumário](#)

Recomendação 1.5 – Fornecer âncoras para ir direto a um bloco de conteúdo

[Ver WCAG 2.0 Critério de Sucesso 2.4.1](#)

Devem ser fornecidas âncoras, disponíveis na barra de acessibilidade, 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 de 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. Para o governo federal são recomendados atalhos para o menu principal, para o conteúdo e para a caixa de pesquisa. Maiores detalhes a esse respeito podem ser encontrados no *capítulo 3 – Elementos de acessibilidade nas páginas do Governo Federal*.

Devem ser utilizados ambos os atributos name e id para que as âncoras funcionem em todos os navegadores e tecnologias assistivas, tanto textuais quanto gráficos, já que há os que suportam ambos os atributos e os que suportam apenas um deles.

Os links indicadores de início e fim de conteúdo e início e fim de menu podem estar ocultos na página utilizando folhas de estilo. No entanto, é necessário tomar o cuidado de ocultá-los de forma que eles continuem acessíveis aos leitores de tela. Os mecanismos para ocultar elementos e seus efeitos na acessibilidade são os seguintes:

CSS	Efeito na Tela	Efeito na Acessibilidade
visibility:hidden;	O elemento fica oculto, mas continua a ocupar o espaço que normalmente ocuparia	O conteúdo é ignorado pelos leitores de tela
display:none;	O elemento fica oculto e não ocupa espaço	O conteúdo é ignorado pelos leitores de tela
height: 0; width: 0; overflow: hidden;	O elemento fica oculto e não ocupa espaço	O conteúdo é ignorado pelos leitores de tela
text-indent: -999em;	O conteúdo é movido para “fora da tela”, não sendo mais visível, mas links podem ser focalizados de maneira imprevisível	Os leitores de tela acessam o conteúdo, mas somente texto e elementos inline
position: absolute; left: -999em;	O conteúdo é removido de sua posição, não ocupando espaço e é movido para “fora da tela”, ficando oculto	Os leitores de tela acessam o conteúdo

Atributos em CSS para ocultar elementos de acessibilidade

Tabela 1: Atributos em CSS para ocultar elementos de acessibilidade

Exemplo

Topo da Página (na barra de acessibilidade)

```
<ul id="atalhos">
<li><a href="#irconteudo"
accesskey="1">Ir Conteúdo[1]</a></li>
<li><a href="#irmenu" accesskey="2">Ir
para menu principal[2]</a></li>
<li><a href="#irbusca">Ir para busca [3]
</a></li>
</ul>
```

Conteúdo da Página

```
<div>
<a href="#" id="irconteudo"
class="oculto">Início do conteúdo</a>
<!-- Conteúdo -->
</div>
```

Menu Principal da Página

```
<div>
<a href="#" id="irmenu"
class="oculto">Início do menu</a>
<!--itens de menu -->
</div>
```

Formulário

```
<form action="#"method="post">
<fieldset>
<legend>Buscar</legend>
<label for="busca">Pesquise
aqui</label>
<input type="text" href="#"
id="irbusca" value="Pesquise aqui" />
<input type="submit" value="Buscar"
class="buscar" name="buscar" />
</fieldset>
```

[Ir ao exemplo navegável](#) [Recomendação 1.5](#)

[Voltar ao sumário](#)

Recomendação 1.6 – Não utilizar tabelas para diagramação

[Ver WCAG 2.0 Critério de Sucesso 1.3.1 \(Técnica H51\).](#)

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.

Exemplo correto – Utilização de tableless para diagramação

```
</head>
<body>
<div id="topo" role="banner">
<div>
<!--barra de acessibilidade-->
</div>
<h1>Exemplo sem tabelas!</h1>
</div>
<div id="menu" role="navigation">
```



Exemplo incorreto - Utilização de tabela para leiaute

```
</head>
  <body>
    <table border="1">
      <tr id="topo">
        <td colspan="2">
          <!--barra de acessibilidade-->
        </td>
        <td colspan="3">
          <h1>Exemplo com
```

[Voltar ao sumário](#)

Recomendação 1.7 – Separar links adjacentes

[Ver WCAG 2.0 Critério de Sucesso 1.3.1 \(Técnica H48\).](#)

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.

Caso os links estejam no meio de um parágrafo de conteúdo texto, pode-se utilizar vírgulas, parênteses, colchetes, pipe, entre outros, para fazer a separação.

Exemplo correto

```
</li>
  <ul id="menu">
    <li> <a href="home.html">Home</a>
    <li> <a href="pesquisa.html">Pesquisa</a></li>
    <li> <a href="mapasite.html">Mapa do Site</a></li>
  </ul>
  <!-- Conteudo do Site -->
```

Exemplo incorreto

```
<p id="menu">
  <a href="#menu">Pular o menu</a>
<br>
  <a href="home.html">Home</a><br>
  <a href="pesquisa.html">Pesquisa</a><br>
  <a href="mapasite.html">Mapa do Site</a>
</p>
```

Em HTML5 e ARIA

```
<div id="menu" role="menu">
  <div role="menuitem"> <a href="home.html">Home</a></div>
  <div role="menuitem"> <a href="pesquisa.html">Pesquisa</a></div>
  <div role="menuitem"> <a href="mapasite.html">Mapa do Site</a></div>
</div>
```

[Ir ao exemplo navegável Recomendação 1.7](#)

Existem três roles da especificação ARIA que podem ser utilizados para as listas: o role navigation, que indica o menu principal do site (esse role possui a mesma função que a elemento NAV do HTML5). O segundo e o terceiro roles são muito semelhantes, o role menu e o role menubar. Os dois possuem a função de indicar menus que não são o principal da página. A diferença entre os dois é que o role menu possui um sentindo mais geral enquanto o menubar é ideal para criações de menu na horizontal. Um exemplo de uso de menu pode ser a utilização de sub-menus dentro do menu principal. Os roles menu e menubar podem possuir filhos com role menuitem para indicar que se trata de um item daquele menu.

Exemplo:

```
</head>
  <body>
    <header id="topo"
      role="banner">
      <div>
        <!--barra de acessibilidade-->
      </div>
      <h1>Exemplo com tabela</h1>
    </header>
    <nav id="menu" role="navigation">
```

[Voltar ao sumário](#)

Recomendação 1.8 – Dividir as áreas de informação

[Ver WCAG 2.0 Critério de Sucesso 3.2.3 \(Técnica G61\).](#)

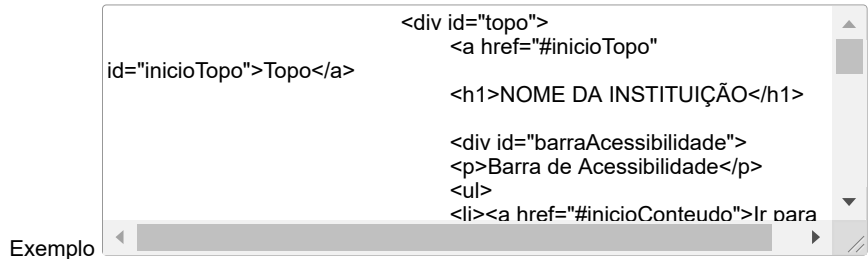
Áreas de informação devem ser divididas em grupos fáceis de gerenciar. As divisões mais comuns são “topo”, “conteúdo”, “menu” e “rodapé”. Nas páginas internas deve-se manter uma mesma divisão para que o usuário se familiarize mais rapidamente com a estrutura do sítio. É importante destacar, entretanto, que a página inicial pode ter uma divisão diferente das páginas internas, pois normalmente ela contém mais elementos. O exemplo a seguir mostra a divisão da página inicial de um sítio contendo os blocos “topo”, “menu”, “conteúdo” e “rodapé”, além da barra de acessibilidade contendo os atalhos.



Acessível
em Libras



Exemplo de divisão de blocos de conteúdo.



A divisão em blocos de conteúdo representa a base para a utilização dos atalhos, permitindo que o usuário tenha rápido acesso à área desejada. Para mais detalhes sobre a barra de acessibilidade e os atalhos, ver capítulo 4 – Elementos padronizados de acessibilidade digital no Governo Federal.

É importante que as diversas páginas de um sítio possuam um estilo de apresentação coerente e sistemático, mantendo-se um padrão de estrutura. Assim, elementos principais de navegação deverão ser mantidos na mesma posição em todas as páginas, com exceção da página inicial que, muitas vezes, apresenta uma estrutura diferenciada.

Exemplo

Um sítio possui um logotipo, um título, um formulário de pesquisa e uma barra de navegação. Esses elementos aparecem na mesma ordem relativa em cada página do sítio em que se repetem. Em uma das páginas, não há o formulário de pesquisa, mas o restante dos itens continua na mesma ordem.

Em HTML5 e ARIA

As elementos estruturais do HTML HEADER, NAV, SECTION, ARTICLE, ASIDE e FOOTER já são interpretadas pela maioria dos leitores de tela. No entanto, é importante utilizar os landmarks roles de ARIA associados a elas, para garantir que todos os recursos de tecnologia assistiva realizem a interpretação da informação. Apesar de a informação ficar duplicada, os leitores de tela que reconhecem o HTML5 darão preferência ao HTML e, dessa forma, não irão ler duas vezes a mesma informação.

Exemplos de elementos estruturais no HTML5:

- <header>
Contém informações introdutórias para uma página ou para uma seção dela.
- <nav>
É utilizada para seções da página que contenham links para outras páginas ou seções, como o menu principal, por exemplo.
- <section>
Representa uma seção genérica ou uma seção que contém uma aplicação. Funciona de maneira parecida com a elemento <div>, separando seções do documento.
- <article>
Representa uma seção da página independente, como uma postagem em um blog, um comentário em um fórum, entre outros.
- <aside>
Contém conteúdo relacionado à área principal do documento.
- <footer>
É utilizada para demarcar o rodapé da página ou de uma seção do conteúdo.



Exemplos de *landmark* roles em ARIA:

- Banner
Contém informações voltadas ao site como um todo e não somente à página.
Representa as informações normalmente contidas no topo das páginas, como o título e o logo do site e a barra de acessibilidade.
- Complementary
Conteúdo adicional ou complementar ao conteúdo principal.
- Contentinfo
Contém informações gerais do site como um todo, que normalmente aparecem no rodapé, como copyright, contato, etc.
- Main
Contém o conteúdo principal da página.
- Navigation
Contém os links para navegar pelas partes do site ou da página, como é o caso do menu.
- Search
Esta seção contém o formulário de busca do site.



Acessível
em Libras

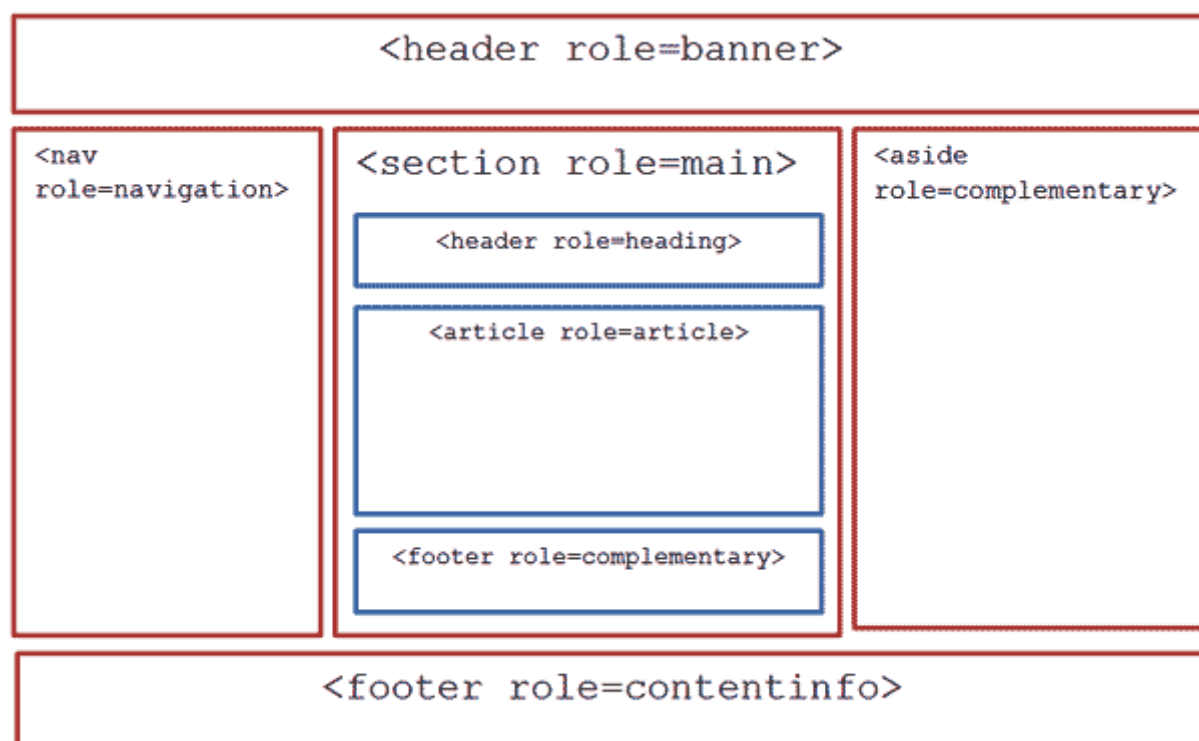
O elemento HEADER pode ser utilizado várias vezes em uma mesma página sem problemas de semântica, pois ele possui duas funções diferentes. A primeira é a de conter as informações do topo da página e, para isso, deverá ser utilizado o role banner. A segunda função é para indicar um cabeçalho de seção e pode ser utilizada várias vezes no documento. Neste caso, deve ser utilizado o role heading e, preferencialmente, deverá conter algum nível de título.

O conteúdo principal do site não possui um elemento específico no HTML5 como os demais, então a sua identificação é feita com o valor main no role.

O elemento FOOTER, assim como o HEADER, pode ser utilizado mais que uma vez na mesma página, pois ele pode ser utilizado para dois objetivos. O primeiro é para o rodapé geral da página que deve receber o valor contentinfo para o role. O segundo é para indicar rodapé em seções e, nesse caso, deve ser utilizado o role complementary.

O elemento SECTION do HTML5 é indicado para separar seções dentro do documento, podendo ser utilizado quantas vezes forem necessárias na mesma página. O ARIA role que contém o mesmo papel do elemento possui o valor section.

Abaixo, temos um exemplo de estrutura de uma página utilizando as elementos estruturais do HTML5 e ARIA roles. Além dos landmarks, também foram utilizados outros roles, como é o caso do heading (utilizado para identificar títulos) e do article (identifica conteúdo que faz sentido isolado, que é independente, como uma postagem em um blog).



Estrutura com HTML5 e ARIA.

Como podemos ver, o HTML5 ampliou consideravelmente a quantidade de elementos, os quais, aliados aos roles do WAI ARIA, vêm facilitar a aplicação de semântica em áreas estruturais. Porém, mesmo com a aplicação dessas novas ferramentas ainda devem ser fornecidas âncoras para as áreas relevantes da página

(ver Recomendação 6), pois a marcação semântica ainda não é suficiente para garantir que os leitores de tela ou outros agentes de usuário que possuem funcionalidades específicas para navegação pelo teclado, consigam listar esses novos elementos.

Toda divisão de conteúdo que possua um significado importante deve ser feita utilizando o elemento SECTION no lugar do elemento DIV, pois o DIV no HTML5 possui a função de apenas agrupar elementos que precisam ser agrupados, mas não possuem uma separação semântica dos demais elementos.

Exemplo de código utilizando HTML5 e ARIA roles para as diferentes áreas de informação, além de manter os atalhos para conteúdo, menu e busca:

```
<header role="banner">
<a href="#inicioTopo" id="inicioTopo">Topo</a>
  <h1>NOME DA INSTITUIÇÃO</h1>
  <div id="barraAcessibilidade">
    <p>Barra de Acessibilidade</p>
    <ul>
      <li>
<a href="#inicioConteudo">Ir para conteúdo [1]</a></li>
```

[Voltar ao sumário](#)

Recomendação 1.9 – Não abrir novas instâncias sem a solicitação do usuário

[Ver WCAG 2.0 Critério de Sucesso 3.2.5](#)

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. Assim, não devem ser utilizados:

- Pop-ups;
- A abertura de novas abas ou janelas;
- O uso do atributo target=“ _blank”;
- Mudanças no controle do foco do teclado;
- Entre outros elementos, que não tenham sido solicitadas pelo usuário.

É muito importante que os links abram na guia ou janela atual de navegação, pois os usuários com deficiência visual podem ter dificuldade em identificar que uma nova janela foi aberta. Além disso, estando em uma nova janela, não conseguirão retornar à página anterior utilizando a opção voltar do navegador. Quando for realmente necessária a abertura de um link em nova janela, é recomendado que tal ação seja informada ao usuário no próprio texto do link. Isso permite ao usuário decidir se quer ou não sair da janela ou aba em que se encontra e, caso decida acessar o link, ele saberá que se trata de uma nova aba ou janela.

```
<a href="universal.html"> Desenho Universal
(abre em nova janela) </a>
```

Exemplo

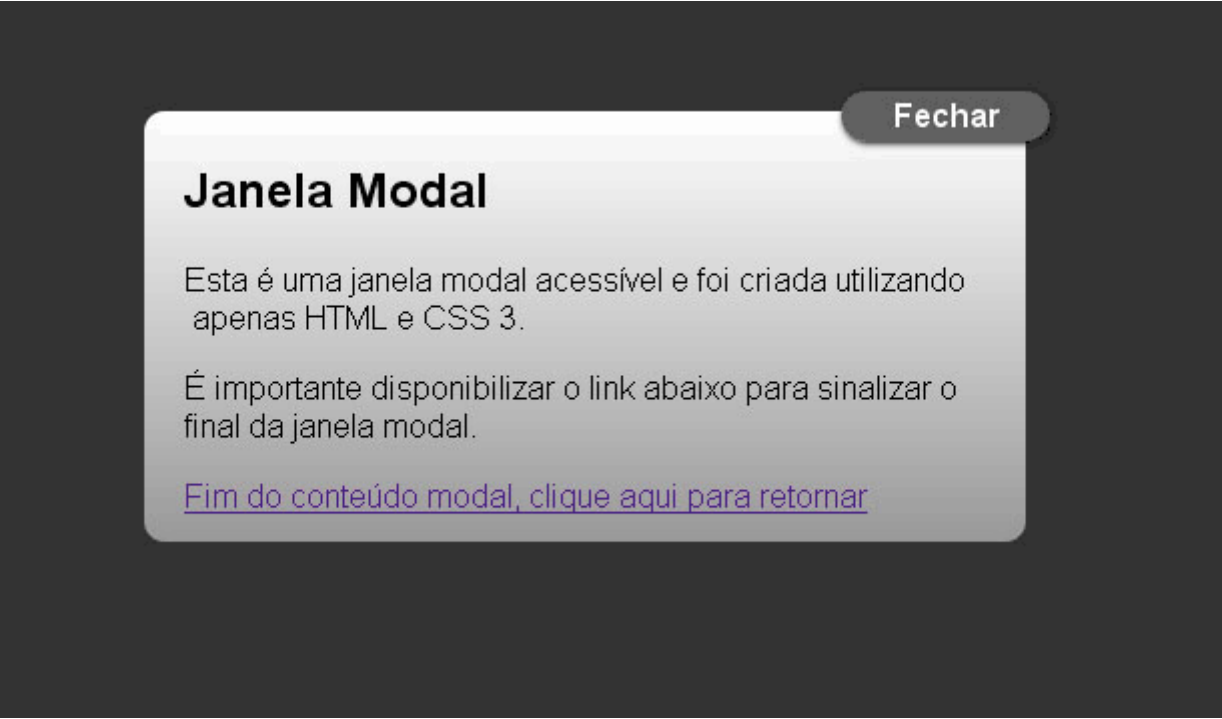
As janelas modais, como as lightbox, fazem com que o foco do teclado permaneça “atrás” da janela, ou seja, o usuário navegando pelo teclado não tem acesso ao conteúdo dessas janelas. Além disso, as janelas modais, em geral, não apresentam um bom nível de acessibilidade em dispositivos móveis. Assim, se houver real necessidade de utilizar esse tipo de elemento, é preciso garantir que o foco seja remetido para o início do conteúdo da janela modal, que o conteúdo dentro da mesma seja acessível e que seja possível retornar facilmente para o site navegando pelo teclado.

Exemplo

Um exemplo simples para criação de uma janela modal foi colocado abaixo contendo apenas HTML e CSS3.



Acessível
em Libras



Exemplo de janela modal

HTML

```
<div id="conteudo" role="main">

<a href="#openModal">Abrir Janela Modal</a>

<div id="openModal" class="modalDialog">
  <div>

    <a href="#close" class="close">Fechar</a>
    <h2>Janela Modal</h2>
    <p>
```

3.2 Comportamento (Document Object Model – DOM)

Recomendação 2.1 - Disponibilizar todas as funções da página via teclado

Ver WCAG 2.0 Critérios de Sucesso 2.1.1 e 2.1.2

Todas as funções da página desenvolvidas utilizando-se linguagens de script (javascript) devem ser programadas, primeiramente, para o uso com teclado. O foco não deverá estar bloqueado ou fixado em um elemento da página, para que o usuário possa mover-se pelo teclado por todos os elementos.

Algumas funções específicas do mouse possuem uma função lógica correspondente via teclado, conforme mostrado na tabela a seguir:

Evento do teclado	Evento correspondente do mouse
onkeydown	onmousedown
onkeyup	onmouseup
onkeypress	onclick*
onfocus*	onmouseover
onblur*	onmouseout

Eventos de teclado e seus correspondentes aos de mouse

Tabela 2: Eventos de teclado e seus correspondentes aos de mouse.

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).

Quando forem utilizados múltiplos manipuladores de eventos para uma ação, de maneira que ela funcione tanto pelo mouse como pelo teclado, é importante testar o resultado final em diferentes navegadores e utilizando diferentes recursos de tecnologia assistiva, para garantir que o evento seja, de fato, acessível.

Dê preferência por utilizar o *onclick*/*onkeypress* em vez de *onmousedown*/*onkeydown* e *onmouseup*/*onkeyup*, pois estes últimos fazem com que o evento seja disparado automaticamente através do teclado. Se houver real necessidade de utilização destes eventos, deverá ser feito um controle sobre qual tecla deverá ser acionada para que o evento ocorra, conforme o exemplo a seguir:

Exemplo HTML

```
<script src="js/exemplo.js">
type="text/javascript"</script>
<noscript>Seu navegador não tem
suporte a JavaScript ou está desativado!</noscript>
...
<p><a href="#"id="link">Portal Brasil</a>
</p>
```

Arquivo JavaScript (exemplo.js)

```
varx=document.getElementById("link")
x.onkeydown=function(e){
varpressedkey
if(typeofevent!=='undefined'){
pressedkey=window.event.keyCode
}else{//outros navegadores
pressedkey=e.keyCode//identifica
}
}
```

Existem funções do mouse que não possuem uma função correspondente via teclado, como é o caso de duplo clique (*dblclick*). Nesses casos, é necessário implementar a função de maneira alternativa, como, por exemplo, incluindo botões que executem, pelo teclado, a função de forma equivalente. O evento *onclick* já funciona pelo



teclado (tecla ENTER) na maioria dos navegadores. Os menus do tipo dropdown e outros que contiverem subitens devem ser acessíveis pelo teclado.

Em HTML5 e ARIA

Menus suspensos são ótimos para organizar sites muito grandes, porém esses menus podem acarretar problemas na acessibilidade, especialmente no que diz respeito ao acesso aos subitens. Esses menus, em muitos casos, são desenvolvidos de forma que funcionam apenas com eventos do mouse, o que se torna um grande problema de acessibilidade. Abaixo, é apresentado um exemplo que utiliza HTML5 e ARIA, além de implementação de javascript, pensando-se na acessibilidade nos subitens.

HTML 5

```
<nav role='navigation' aria-label='Menu principal' class='topnav' id='menu'><ul>
  <li>
    <a href='index.php'>Página inicial</a></li>
    <li>
    <a href='noticias.php'>Notícias</a></li>
    <li>
    <a href='eventos.php'>Eventos</a></li>
    <li>
    <a href='acessibilidade-
```

CSS

```
/* utilizado para o menu ficar na vertical */
#menu li{
display: inline-block;
}

/*utilizado para o link ficar no mesmo tamanho que a li */
#menu li a{
display: block;
```

Javascript:

O Javascript abaixo adiciona as funções do link mostrar/ocultar, porém deve ser utilizado junto com a biblioteca do jQuery (<http://jquery.com/>).

javascript para mostrar/ocultar submenu

```
$(document).ready(function(){
  $(".leitor-de-telas div, .eventos .lista, .dicas .myhide").hide();

  $(".expandir").click(function() {
    $(this).parent().next().slideToggle("slow");

    var text = $(this).text();

    if( text.search("Expandir") != -1 )
    { // -1 é quando não encontra a string
```

[Voltar ao sumário](#)

Recomendação 2.2 – Garantir que os objetos programáveis sejam acessíveis

[Ver WCAG 2.0 Critérios de Sucesso 2.1.1 e 2.1.2](#)

Deve-se garantir que scripts e conteúdos dinâmicos e outros elementos programáveis sejam acessíveis e que seja possível sua execução via navegação. Além de proporcionar o uso por teclado, estratégias devem ser adotadas para proporcionar o acesso a todos independente de seu dispositivo. A funcionalidade drag-an-drop pode ter como alternativa o uso dos atalhos de recortar e colar.

Exemplo correto

Página HTML

```
<a href="cadastro.html" id="cadastro">Cadastre-se agora!</a>
```

Página JavaScript (.js)

```
function pop() {
  alert("Você vai fazer um novo cadastro!");
}
var element = document.getElementById("cadastro");
element.onclick = pop;
```

Exemplo incorreto

Página HTML

```
<a href="javascript:pop()">Cadastre-se agora!</a>
```

Página JavaScript (.js)

```
function pop() {
  alert("Você vai fazer um novo cadastro!");
}
```



Acessível em Libras

Nesse caso, se o navegador não tiver suporte a scripts, o usuário ficará impossibilitado de acessar o link.

OBS: A função “alert” do javascript não gera um pop-up e sim uma mensagem que é lida por todos os leitores de tela.

Quando não for possível que o elemento programável seja diretamente acessível, deve ser fornecida uma alternativa em HTML para o conteúdo.

Quando o script for utilizado em uma página da Web, uma forma de fornecer uma alternativa para ele é através do elemento NOSCRIPT. Este elemento pode ser utilizado para mostrar conteúdos em navegadores que não suportam scripts ou que tenham o script desabilitado. No entanto, se o navegador tiver suporte a scripts e estes estiverem habilitados, o elemento NOSCRIPT será ignorado. Dessa forma, a utilização do elemento NOSCRIPT para um script inacessível não garante que o objeto seja acessível. Assim, a recomendação é que o próprio script seja desenvolvido tomando-se o cuidado para que ele seja acessível, e o elemento NOSCRIPT deve ser utilizado para abranger os casos em que scripts não são suportados.

[Voltar ao sumário](#)

Recomendação 2.3 - Não criar páginas com atualização automática periódica

[Ver WCAG 2.0 Critério de Sucesso 3.2.5 \(Técnicas SVR1 e H76\).](#)

A atualização automática periódica – muito utilizada por canais de notícias - é comumente realizada através do uso do o atributo http-equiv com conteúdo “refresh” da elemento meta no HEAD do documento (ex: <’meta http-equiv=“refresh” content=“30” />, onde a página seria 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.

Como exemplo de uma boa prática, em uma interface Web para e-mail (Webmail), um desenvolvedor pode fornecer um botão ou link para buscar novos e-mails recebidos em vez de atualizar automaticamente. Em páginas onde o limite de tempo é absolutamente necessário, o usuário deverá ser informado que a página é atualizada automaticamente.



O Webmail Expresso possui um botão de atualizar para carregamento de novas mensagens

[Voltar ao sumário](#)

Recomendação 2.4 – Não utilizar redirecionamento automático de páginas

[Ver WCAG 2.0 Critério de Sucesso 3.2.5 \(Técnicas SVR1 e H76\).](#)

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 - ver ePWG - Cartilha de Codificação em <http://epwg.governoeletronico.gov.br/cartilha-codificacao>

[Voltar ao sumário](#)

Recomendação 2.5 – Fornecer alternativa para modificar limite de tempo

[Ver WCAG 2.0 Critério de Sucesso 2.2.1](#)

Em uma página onde há limite de tempo para realizar uma tarefa deve haver a opção de desligar, ajustar ou prolongar esse limite. Essa recomendação não se aplica a eventos em que o limite de tempo é absolutamente necessário.

Deve-se lembrar que, em ambos os casos, o limite de tempo deve ser informado.

Exemplo

Na inserção de dados em um formulário para obtenção de um benefício ou consulta a processo, o cidadão deve ter tanto tempo quanto for necessário para o preenchimento de seus dados. O site deve possuir um mecanismo que permita ao cidadão aumentar o limite de tempo, disponibilizando por exemplo um botão para aumentar o tempo em 5 minutos.

[Voltar ao sumário](#)

Recomendação 2.6 – Não incluir situações com intermitência de tela



[Ver WCAG 2.0 Critério de Sucesso 2.3.1](#)

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: A imagem abaixo representa um elemento piscante, onde as intermitências ocorrem a cada 0,2 segundos.



Exemplo de intermitência de imagem

[Voltar ao sumário](#)

Recomendação 2.7 – Assegurar o controle do usuário sobre as alterações temporais do conteúdo

[Ver WCAG 2.0 Critério de Sucesso 2.2.2](#)

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.

É desejável que o usuário tenha também o controle da velocidade desses conteúdos.



Exemplo de slideshow, com mecanismos de parada e navegação entre os slides.

[Voltar ao sumário](#)

3.3 Conteúdo / Informação

Recomendação 3.1 – Identificar o idioma principal da página

[Ver WCAG 2.0 Critério de Sucesso 3.1.1](#)

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.

Exemplo



Acessível
em Libras

Em HTML 4.01

```
<html lang="pt-BR">
<head>
  <title>documento escrito em
português do Brasil</title>
  ...
</html>
```

Em XHTML 1.1

```
<html
xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt-BR">
<head>
  <title>documento escrito em
português do Brasil</title>
  ...
</html>
```

Em HTML5

```
<html lang="pt-br">
<head>
  <title>documento escrito em português do Brasil</title>
  ...
</html>
```

[Voltar ao sumário](#)

Recomendação 3.2 – Informar mudança de idioma no conteúdo

[Ver WCAG 2.0 Critério de Sucesso 3.1.2](#)

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.

Exemplo

XHTML

```
<p xml:lang="de">
  Futter zu schaffen,
  Wind wehte, lief fort
  Bremen: dort, meinte er,
  Da dachte der Herr daran, ihn aus dem
  aber der Esel merkte, daß kein guter
  und machte sich auf den Weg nach
  könnte er ja Stadtmusikant werden.
</p>
```

HTML

```
<p lang="de">
  Futter zu schaffen,
  wehte, lief fort
  Bremen: dort, meinte er,
  Da dachte der Herr daran, ihn aus dem
  aber der Esel merkte, daß kein guter Wind
  und machte sich auf den Weg nach
  könnte er ja Stadtmusikant werden.
</p>
```

[Voltar ao sumário](#)

Recomendação 3.3 – Oferecer um título descritivo e informativo à página

[Ver WCAG 2.0 Critério de Sucesso 2.4.2](#)

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, ou recursos estilísticos. Na página inicial do sistema ou portal, basta seguir a estrutura [nome do sítio ou sistema].

Exemplo 1

A página inicial do Projeto de Acessibilidade Virtual apresenta o seguinte título:

```
<title>
  Projeto Acessibilidade Virtual
</title>
```

Exemplo 2

A página secundária “Contato” do Projeto de Acessibilidade Virtual apresenta o seguinte título:

```
<title>
  Contato - Projeto Acessibilidade
Virtual
</title>
```

Exemplo 3



Acessível
em Libras

A página interna “Publicações” do sítio do Ministério da Educação apresenta o seguinte título:

Educação

<title>
Publicações – Ministério da
</title>

Para mais detalhes a respeito do título descritivo, ver ePWG - Cartilha de Codificação do ePWG (disponível em: <http://epwg.governoeletronico.gov.br/cartilha-codificacao>)

[Voltar ao sumário](#)

Recomendação 3.4 – Informar o usuário sobre sua localização na página

[Ver WCAG 2.0 Critério de Sucesso 2.4.8](#)

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 “migalha de pão” (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 sítio de uma universidade encontra-se na seção de editais, que está dentro do menu “Ensino”. Acima do conteúdo, é disponibilizada a seguinte Migalha de pão:

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

OBS: Na migalha de pão, 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.

[Voltar ao sumário](#)

Recomendação 3.5 – Descrever links clara e sucintamente

[Ver WCAG 2.0 Critérios de Sucesso 2.4.4 e 2.4.9](#)

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.

É preciso tomar cuidado para não utilizar a mesma descrição para dois ou mais links que apontem para destinos diferentes. Da mesma forma, links que remetem ao mesmo destino devem ter a mesma descrição.

Exemplo Incorreto

da Bahia e Triângulo

<h2>Educação Superior</h2>
<p>Tomam posse os reitores das federais

<p> Leia mais notícias sobre Educação Superior </p>

Exemplo Correto

da Bahia e Triângulo

<h2>Educação Superior</h2>
<p>Tomam posse os reitores das federais

<p> Leia mais notícias sobre Educação Superior </p>

Repare no exemplo incorreto acima que não há necessidade de utilizar o atributo title em links, já que esse atributo não é bem suportado por recursos de tecnologia assistiva, como leitores de tela, não tem utilidade para quem navega apenas pelo teclado e não tem bom suporte em dispositivos móveis, como celulares, entre outros problemas. Assim, se você quiser fornecer informações adicionais para um link, faça-o no próprio texto do link ou no contexto, e jamais repita o conteúdo do link no atributo title.

Exemplo Incorreto

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

Exemplo Correto

[Saiba mais a respeito de acessibilidade](#)

OBS: Não se deve fazer a utilização de links do tipo “clique aqui”, “leia mais”, “mais”, “saiba mais”, “veja mais”, “acesse a lista”, pois estas expressões não fazem sentido fora do contexto. Muitos usuários de leitores de tela navegam por links, tornando descrições como “Clique aqui”, “Veja mais” insuficientes para o usuário saber o destino do link, ou localizá-lo na página.

Em links de arquivos para download, é necessário informar a extensão e o tamanho do arquivo no próprio texto do link, conforme exemplo a seguir.



```
<a href="cartilha.odt">Cartilha de
Acessibilidade do W3C (.odt 150Kb)</a>
```

Exemplo

[Voltar ao sumário](#)

Recomendação 3.6 – Fornecer alternativa em texto para as imagens do sítio

Ver WCAG 2.0 Critério de Sucesso 1.1.1 (Técnica G95).

Deve ser fornecida uma descrição para as imagens da página, utilizando-se, para tanto o atributo alt.

Exemplo 1



Exemplo de descrição de imagem.

```

```

No código:

No caso de banners e outras imagens que reproduzam textos, o ideal é reproduzir o texto escrito.

Exemplo 2



Exemplo de descrição de banner

```
<a
href="http://www.dominiopublico.gov.br/">

</a>
```

No código:

Descrever qualquer imagem, em geral, é algo bastante subjetivo e a descrição deve ser adaptada ao contexto em que a imagem se encontra. Para mais detalhes de como escrever um texto alternativo, veja o tutorial O uso correto do texto alternativo na seção do eMAG no portal de Governo Eletrônico (<http://www.governoeletronico.gov.br/eixos-de-atuacao/governo/acessibilidade/material-de-apoio>).

Apesar de não haver um limite de caracteres para o atributo alt, ele é utilizado para descrições sintéticas, em poucas palavras ou em uma frase curta. Para imagens mais complexas que exigem uma descrição mais detalhada, como infográficos, por exemplo, deve-se fornecer, além do alt, a descrição no próprio contexto ou um link para a descrição longa logo após a imagem. Deve ficar claro para o usuário que esse link remete para a descrição longa da imagem, conforme o exemplo a seguir.

Exemplo 3

Acessível
em Libras



Exemplo de descrição de infográfico - A descrição textual está disponível em outra página

No código da página do infográfico:

```

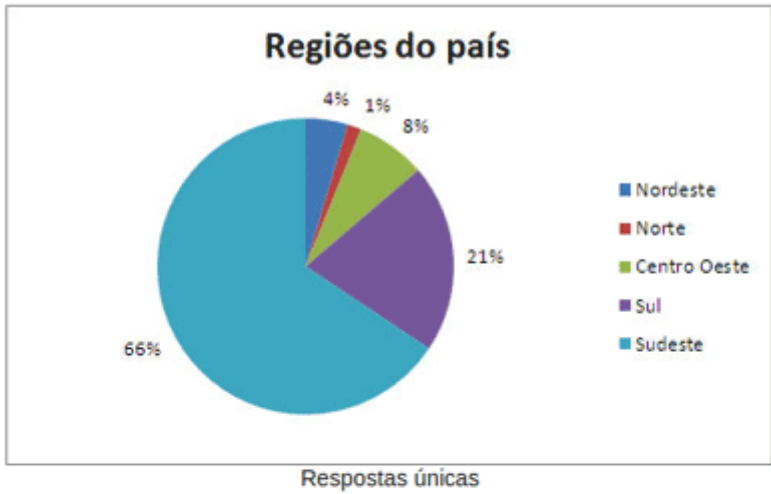
<p><a href="selos-e-certificacoes.html">Conteúdo texto do Infográfico</a> - disponível</p>
```

Código em selos-e-certificacoes.html

```
<h1>Selos e certificações conscientes</h1>
<p>Conheças os selos e certificações que atestam a qualidade e responsabilidade das empresas na produção</p>
<dl>
<dt>Orgânico Brasil</dt>
<dd>Selo do Sistema Brasileiro de Avaliação de Conformidade Orgânica (SISOrg) identifica e controla a produção nacional de alimentos orgânicos, caracterizada por processo
```

Lembrando que em gráficos simples, as informações podem estar contidas na própria página ajudando o entendimento, inclusive, de pessoas que tem dificuldade com visualizações de gráficos ou queiram utilizar as informações para outros fins, como pesquisas.





Percentual de respondentes por regiões do país

Nordeste	4%
Norte	1%
Centro Oeste	8%
Sul	21%
Sudeste	66%

Dados simples podem ser representados tanto graficamente quanto em forma de tabelas.

Em HTML5

Foram incorporados ao HTML5 os elementos `FIGURE` e `FIGCAPTION`, que são utilizados em conjunto com o objetivo de agrupar uma imagem `IMG` com a sua legenda, a qual aparecerá visualmente.

O elemento `FIGURE` não é usado apenas para marcação de imagens, ele define um bloco de conteúdo independente que normalmente contém linhas de códigos, imagens, gráficos, diagramas e vídeos que fazem referência ao conteúdo principal do documento. O elemento `FIGCAPTION` é opcional e exibe uma legenda para o elemento `FIGURE`. Esse novo elemento age como subtítulo opcional ou legenda para qualquer conteúdo contido dentro do elemento `FIGURE` e não substitui o atributo `alt` nas descrições das imagens. Pode-se ter várias imagens, vídeos, gráficos, dentro de um único elemento `FIGURE`. Caso haja um elemento `FIGCAPTION` este deve ser o primeiro ou o último filho do elemento `FIGURE` e só um `FIGCAPTION` irá representar a legenda.

```
<figure>
  
  <figcaption>
    Figura 1: Imagem da árvore que deu
    origem ao nome do nosso país: Pau-brasil
  </figcaption>
</figure>
```

Exemplo
Que seria traduzido visualmente na página da seguinte forma:



Uso dos elemento figure e figcaption

OBS: Imagens que não transmitem conteúdo, ou seja, imagens decorativas, devem ser inseridas por CSS. No exemplo a seguir, o separador pontilhado logo abaixo do título de nível 3 “Acessibilidade web” é uma imagem decorativa cuja função foi inserida através das folhas de estilo. Note que em alguns casos a imagem pode ser substituída por um código CSS.

Exemplo



Título em texto com imagem decorativa de separador pontilhado



HTML

<h3>Acessibilidade web</h3>

CSS

h3 {
color: #64860B;
font-size: 1.4em;
font-weight: lighter;
padding: 5px 0 10px;
height: 30px;
border-bottom: 1px dotted #666;
}

[Voltar ao sumário](#)

Recomendação 3.7 – Utilizar mapas de imagem de forma acessível

[Ver WCAG 2.0 Critério de Sucesso 1.1.1 \(Técnica H24\)](#)

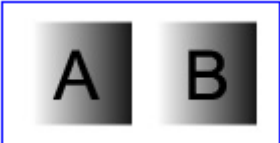
Um mapa de imagens é uma imagem dividida em áreas selecionáveis definidas por elemento AREA . Cada área é um link para outra página Web ou outra seção da página atual. É um recurso em desuso, mas pode ser útil na acessibilidade de infográficos, por exemplo.

Esse recurso não deve ser utilizado para menus ou seleção de regiões para serviços.

Existem dois tipos de mapas de imagem: mapas do lado do cliente.

Para mapas de imagem do lado do cliente, devem ser fornecidas descrições através do atributo alt para cada uma das zonas ativas, ou seja, para cada um dos links que receberá o foco.

Exemplo (mapa de imagem do lado do cliente)



Mapa de imagem exemplo 1

Código:

<map name="Map" id="Map">
 <area shape="rect" coords="8,10,63,59" href="a.html" alt="Link para a seção [A]">
 <area shape="rect" coords="77,9,126,61" href="b.html" alt="Link para a seção [B]">
</map>

Além dos mapas de imagem do lado do cliente, existem os do lado do servidor.

No entanto, é recomendada a utilização de mapas de imagem do lado do cliente, já que para mapas de imagem do lado do servidor não é possível fornecer um alt para cada uma das zonas ativas, somente para o mapa como um todo, não sendo possível, portanto, torná-lo acessível. No entanto, se for realmente necessária sua utilização, devem ser fornecidos links redundantes relativos a cada região ativa do mapa de imagem, conforme o exemplo a seguir, para que, desta forma, usuários com leitores de tela possam ter acesso ao seu conteúdo.

Exemplo (mapa de imagem do lado do servidor)



Mapa de imagem exemplo 2

Código:

<p>
 Área Verde -
 Área Amarela -
 Área Azul

[Voltar ao sumário](#)

Recomendação 3.8 – 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.

OBS: O ODF (Open Document Format) é um formato aberto de documento adotado pela e-PING (Arquitetura de Interoperabilidade em Governo Eletrônico) que pode ser implementado em qualquer sistema. O ODF engloba formatos como: ODT (Open Data Text) para documentos de texto, ODS (Open Data Sheet) para planilhas eletrônicas, ODP (Open Data Presentation) para apresentações de slides, entre outros.

Muitos softwares já utilizam esses formatos, como é o caso do OpenOffice, BrOffice, Google Docs, Abiword e StarOffice. O Microsoft Office 2010 também inclui suporte para ODF. Para versões anteriores do Microsoft Office, pode ser instalado um Add-in gratuito para suporte aos formatos ODF.

[Voltar ao sumário](#)

Recomendação 3.9 – Em tabelas, utilizar títulos e resumos de forma apropriada

[Ver WCAG 2.0 Critério de Sucesso 1.3.1 \(Técnicas H39 e H73\)](#)

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.

Exemplo

<table summary="Esta tabela exibe os copos de café consumidos por cada senador, o tipo de café (descafeinado ou normal), com açúcar ou sem açúcar.">

<caption>Copos de café por Senador</caption>

...

Para mais detalhes veja o tutorial *Tabelas Acessíveis* (<http://www.governoeletronico.gov.br/eixos-de-atuacao/governo/acoes/acessibilidade/material-de-apoio>), na seção do eMAG no Portal do Programa de Governo Eletrônico.

[Voltar ao sumário](#)

Recomendação 3.10 – Associar células de dados às células de cabeçalho

[Ver WCAG 2.0 Critério de Sucesso 1.3.1 \(Técnicas H43 e H63\)](#)

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.

Exemplo 1

<table>

<caption>Demonstrativo do Patrimônio</caption>

<thead>

<tr>

<th>Tipos</th>

<th>Valores (R\$)</th>

<th>Percentual</th>

</tr>

</thead>

Tipos	Valores (R\$)	Percentual
Recursos Financeiro	56.879,63	51,36%
Bens Móveis	25.691,23	23,20%
Bens Imóveis	28.169,36	25,44%
Total	110.740,22	100%

Demonstrativo do Patrimônio

Imagem da tabela descrita no código do exemplo 1

Para tabelas mais complexas, é necessário utilizar marcações para associar as células de dados com as células de cabeçalho. A maneira mais adequada de realizar esse procedimento é utilizar os atributos *id/headers* ou *scope/col*. No primeiro, pode-se associar qualquer célula de conteúdo a qualquer célula de cabeçalho, utilizando o mesmo valor para o atributo id e para o header. No segundo caso, a associação é automática, sendo mais utilizado em tabelas de associação direta, nas quais é dado o valor col para o atributo scope nos cabeçalhos. Nos exemplos a seguir, é possível verificar a utilização do *id/headers* e do *scope/col*.

Exemplo 2

<table>

<caption>Resultado do Concurso</caption>

<tr>

<th id="vaga">Vaga</th>

<th id="candidato">Nome do candidato</th>

<th id="basico">Prova de Conhecimento Básico</th>

<th id="especifico">Prova de Conhecimento Específico</th>

Vaga	Nome do candidato	Prova de Conhecimento Básico	Prova de Conhecimento Específico
Técnico Administrativo	Paulo da Silva	8	16
	Pedro Pontes	7	15
Técnico em Informática	João Pereira	9	17

Resultado do Concurso

Imagem da tabela descrita no código do exemplo 2

Exemplo 3

<table summary="Tabela de vendas de maçã e banana efetuadas no ano de 2010">
<caption>Vendas 2010</caption>
<thead>
<tr>
<th
scope="col">Mês</th>
<th scope="colgroup"
colspan="2">Rio Grande do Sul</th>

Mês	Rio Grande do Sul		Santa Catarina	
Janeiro	Maçã	Banana	Maçã	Banana
Feveiro	1000	1500	3000	1000
Março	2000	1500	3500	500

Vendas 2010

Imagem da tabela descrita no código do exemplo 3

Exemplo 4

<table summary="Tabela com resumo das despesas de transporte durante o mês de Agosto nas cidades de Porto Alegre e Florianópolis">
<caption>Relatório de despesa de viagem</caption>
<thead>
<tr>
<th>Data</th>
<th id="alimentacao"
axis="despesas">Alimentação</th>

Data	Alimentação	Hotel	Transporte
Porto Alegre			
25 de agosto de 2010	R\$ 37,74	R\$ 112,00	R\$ 45,00
26 de agosto de 2010	R\$ 27,28	R\$ 112,00	R\$ 45,00
Subtotal	R\$ 65,02	R\$ 224,00	R\$ 90,00
Florianópolis			
27 de agosto de 2010	R\$ 96,25	R\$ 109,00	R\$ 36,00
28 de agosto de 2010	R\$ 35,00	R\$ 109,00	R\$ 36,00
Subtotal	R\$ 131,25	R\$ 218,00	R\$ 72,00
Total	R\$ 196,27	R\$ 442,00	R\$ 162,00

Relatório de despesa de viagem

Exemplo da tabela do exemplo 4 com o uso do atributo axis

[Voltar ao sumário](#)

Recomendação 3.11 – Garantir a leitura e compreensão das informações

Ver WCAG 2.0 Critério de Sucesso 3.1.5

O texto de um sítio 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.

Para mais informações sobre como escrever textos para web, acesse ePWG Cartilha de Redação Web (<http://epwg.governoeletronico.gov.br/cartilha-redacao>) e o manual

[Voltar ao sumário](#)

Recomendação 3.12 – Disponibilizar uma explicação para siglas, abreviaturas e palavras incomuns

Ver WCAG 2.0 Critérios de Sucesso 3.1.3 e 3.1.4

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 – Explicação no próprio texto

```
<p>A ONU (Organização das Nações Unidas) é um órgão internacional criado em 1945.</p>
```

Exemplo 2 – Utilização de glossário

Um site sobre desenvolvimento Web fornece um link de menu para o Glossário, que contém explicações para termos como HTML, CSS, Tag, atributo, etc.

Exemplo 3 – Utilização do elementos abbr

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

[Voltar ao sumário](#)

3.4 Apresentação / Design

Recomendação 4.1 - Oferecer contraste mínimo entre plano de fundo e primeiro plano

[Ver WCAG 2.0 Critério de Sucesso 1.4.3](#)

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

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

A relação de contraste pode ser encontrada dividindo-se o valor da luminosidade relativa da cor mais clara de um dos planos pelo valor da luminosidade relativa da cor mais escura do outro plano. A relação de contraste entre plano de fundo e primeiro plano de 3:1 é o nível mínimo de contraste recomendado pela ISO-9241-3. No entanto, levando-se em consideração a perda de percepção do contraste resultante da baixa acuidade visual, cromodeficiência ou perda de sensibilidade ao contraste devido ao envelhecimento, é recomendada aqui uma maior relação de contraste, de, no mínimo, 4,5:1. Existem ferramentas gratuitas disponíveis na Web que verificam a relação de contraste entre as cores do plano de fundo e do primeiro plano, referenciadas no [capítulo 4](#) deste documento e para uma [tabela de cores no anexo 1](#).

Exemplos de relações de contraste:

Contraste de 3:1 (abaixo do recomendado)



Primeiro plano: #3c3cff (RGB 60, 60, 255)

Plano de fundo: #b2b2b2 (RGB 178, 178, 178)

Contraste de 4,5:1 (mínimo recomendado pelo eMAG)



Primeiro plano: #0000a2 (RGB 0, 0, 162)

Plano de fundo: #939393 (RGB 147, 147, 147)

Contraste de 7:1 (mínimo recomendado para alto contraste)



Primeiro plano: #000065 (RGB 0, 0, 101)

Plano de fundo: #a3a3a3 (RGB 163, 163, 163)

[Voltar ao sumário](#)

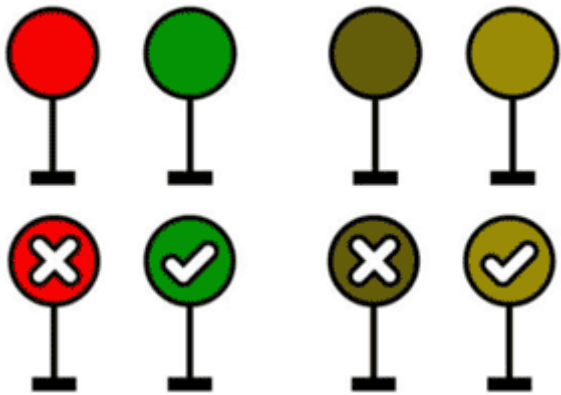


Acessível
em Libras

Recomendação 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

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



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

HTML - Exemplo incorreto

```
<p>Existem três procedimentos para
executar a tarefa:</p>
<ul>
  <li><a href="#">Procedimento
A</a></li>
  <li><a href="#"
class="recomendado">Procedimento B</a></li>
  <li><a href="#">Procedimento
C</a></li>
</ul>
```

CSS

```
a.recomendado {
  color: #FF0000;
}
```

HTML - Exemplo correto

```
<p>Existem três procedimentos para
executar a tarefa:</p>
<ul>
  <li><a href="#">Procedimento
A</a></li>
  <li><a href="#"
class="recomendado">Procedimento B (Recomendado)</a></li>
  <li><a href="#">Procedimento
C</a></li>
</ul>
```

CSS

```
a.recomendado {
  color: #FF0000;
}
```

Existem três procedimentos para executar a tarefa:

- Procedimento A
- Procedimento B (Recomendado)
- Procedimento C

Exemplo correto de utilização de cores nos elementos

[Voltar ao sumário](#)

Recomendação 4.3 – Permitir redimensionamento sem perda de funcionalidade

Ver WCAG 2.0 Critério de Sucesso 1.4.4

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.

Exemplo

Exemplo de sítio no qual, à medida que ocorre o redimensionamento, o leiaute se ajusta para que não ocorram “quebras”.



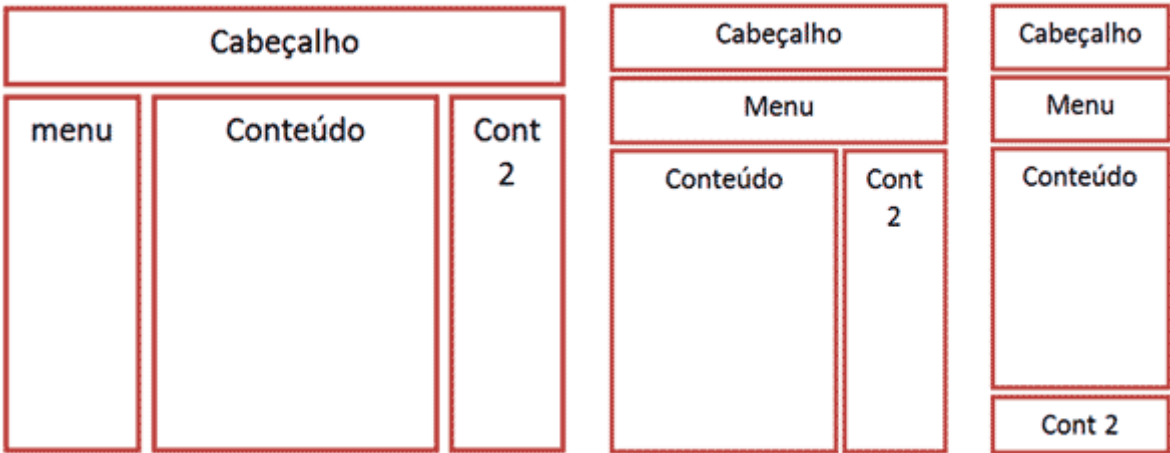


Exemplo de página em seu tamanho padrão



Exemplo de página redimensionada em 200% sem perda de funcionalidade

Além de permitir o redimensionamento sem perda de funcionalidade, o leiaute do sítio deve adequar-se à resolução de tela do dispositivo pelo qual está sendo acessado, já que, atualmente, não existe mais um padrão de resolução de tela para os computadores e há uma crescente utilização de dispositivos móveis. A melhor maneira de tratar as diferentes resoluções de tela é através do **design responsivo**, que trata especificamente da forma que o site será renderizado dependendo do tamanho em pixels da tela que o dispositivo possui e seu objetivo. O design responsivo trata a resolução de tela considerando faixas de resolução, através de diferentes folhas de estilo para tratamento das faixas de resolução de tela. Para tratar a folha de estilo, basta inserir no mesmo documento comandos de Media Queries contendo a resolução máxima de apresentação e as alterações desejadas logo depois.



Exemplo de leiaute responsivo, que se adapta conforme a resolução da tela

[Voltar ao sumário](#)

Recomendação 4.4 – Possibilitar que o elemento com foco seja visualmente evidente

[Ver WCAG 2.0 Critério de Sucesso 2.4.7](#)

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.



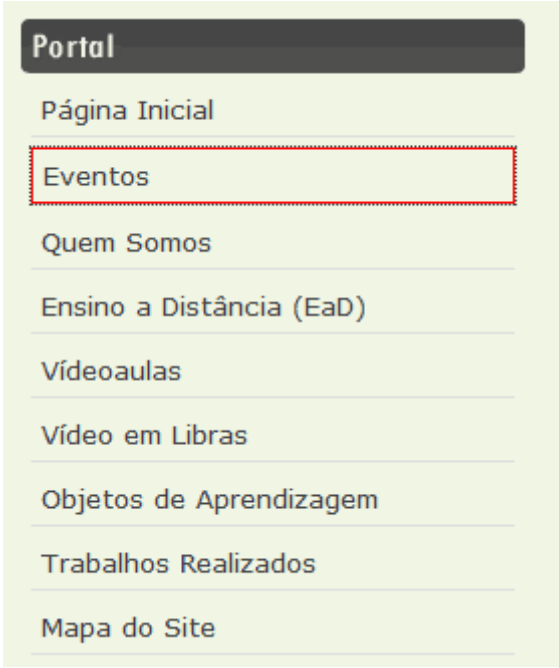
Acessível em Libras

Exemplo CSS

```
a:focus, a:hover {
border: 2px solid #F00;
}
```

HTML

```
<ul>
<li><a
href="/acessibilidade/index.php">Página Inicial</a></li>
<li><a
href="/acessibilidade/eventos.php">Eventos</a></li>
<li><a
href="/acessibilidade/quemsomos.php">Quem Somos</a></li>
<li><a
href="/acessibilidade/ead.php">Ensino a Distância (EaD)</a></li>
</ul></a>
```



Exemplo de foco visível em menu

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.

[Voltar ao sumário](#)

3.5 Multimídia

Recomendação 5.1 – Fornecer alternativa para vídeo

[Ver WCAG 2.0 Critérios de Sucesso 1.2.1, 1.2.2, 1.2.6 e 1.2.8](#)

Deve haver uma alternativa sonora ou textual para vídeos que não incluem faixas de áudio. Para vídeos que contêm áudio falado e no idioma natural da página, devem ser fornecidas legendas. Além de essencial para pessoas com deficiência visual, a alternativa em texto também é importante para usuários que não possuem equipamento de som, que desejam apenas realizar a leitura do material ou não dispõem de tempo para ouvir um arquivo multimídia.

Exemplo 1

Um vídeo mostra como produzir uma tecnologia assistiva de baixo custo. Não há áudio, mas o vídeo inclui uma série de números para representar cada etapa do processo. Nesse caso, junto ao vídeo, deve ser disponibilizado um arquivo com a alternativa de texto que indica o conteúdo do vídeo e a descrição de cada uma das etapas.



Vídeo contendo arquivo com alternativa em texto

Exemplo 2

Uma universidade oferece a opção de visualizar suas videoaulas com ou sem legendas.



Acessível
em Libras

Videoaulas:

- [A teoria da simplicidade \(com legenda\)](#)
- [A teoria da simplicidade \(sem legenda\)](#)
- [Dicas de HTML \(com legenda\)](#)
- [Dicas de HTML \(sem legenda\)](#)
- [Dicas de CSS \(com legenda\)](#)
- [Dicas de CSS \(sem legenda\)](#)



Vídeo com legenda

Além de alternativa em texto e legenda, é desejável que os vídeos com áudio apresentem alternativa na Língua Brasileira de Sinais (Libras).



Vídeo com alternativa em libras - interprete humano



Vídeo com alternativa em libras – interprete avatar 3D

É importante salientar que o player de vídeo deve ser acessível. Deve ser possível acessar os controles via teclado e os mesmos devem estar devidamente descritos, para serem interpretados pelos leitores de tela.

Elemento VIDEO em HTML5

Vídeos disponibilizados no site devem possuir botões de controles, especialmente para os controles de play, pause e stop. O HTML5 traz uma grande novidade para esta funcionalidade que é a elemento VIDEO, um padrão para reprodução de vídeo onde se tornou muito mais fácil controlar as funcionalidades do vídeo:



Acessível
em Libras

Exemplo HTML

JavaScript

```
<button onclick="playPause()">Play/Pause</button>
<button onclick="grande()">Grande</button>
<button onclick="pequeno()">Pequeno</button>
<button onclick="normal()">Normal</button>
<video id="video" width="320" height="240" controls>

var meuVideo=document.getElementById("video");
function playPause()
{
  if (meuVideo.paused)
    meuVideo.play();
  else
    meuVideo.pause();
}
function grande()
```

[Voltar ao sumário](#)

Recomendação 5.2 – Fornecer alternativa para áudio

[Ver WCAG 2.0 Critérios de Sucesso 1.2.1, 1.2.2 e 1.2.6](#)

Áudio gravado deve possuir uma transcrição descritiva. Além de essencial para pessoas com deficiência auditiva, a alternativa em texto também é importante para usuários que não possuem equipamento de som, que desejam apenas realizar a leitura do material ou não dispõem de tempo para ouvir um arquivo multimídia. Neste caso, também é desejável a alternativa em Libras.

Exemplo

Em um podcast o entrevistador faz perguntas a um especialista de saúde. Como essas informações são disponibilizadas ao usuário do sítio em um arquivo de áudio, deve ser fornecido um link para um arquivo com alternativa em texto, logo após o conteúdo em áudio.



Player de audio com descrição do podcast.

Uma apresentação prévia do conteúdo dos dois tipos de arquivo e de sua duração também é desejável.

[Voltar ao sumário](#)

Recomendação 5.3 – Oferecer audiodescrição para vídeo pré-gravado

[Ver WCAG 2.0 Critério de Sucesso 1.2.3, 1.2.5 e 1.2.7](#)

Vídeos que transmitem conteúdo visual que não está disponível na faixa de áudio devem possuir uma audiodescrição.

A audiodescrição consiste na descrição clara e objetiva de todas as informações apresentadas de forma visual e que não fazem parte dos diálogos. Essas descrições são apresentadas nos espaços entre os diálogos e nas pausas entre as informações sonoras.

Exemplo

Um vídeo de um malabarista se apresentando para um grupo de crianças inclui uma versão com audiodescrição. O narrador da audiodescrição descreve o número e o tipo de instrumentos que o malabarista utiliza, bem como as reações que as crianças têm durante a performance.



Vídeo com audiodescrição.

[Voltar ao sumário](#)

Recomendação 5.4 – Fornecer controle de áudio para som

[Ver WCAG 2.0 Critério de Sucesso 1.4.2](#)



Acessível
em Libras

Deve ser fornecido um mecanismo para parar, pausar, silenciar ou ajustar o volume de qualquer som que se reproduza na página.

Elemento audio do HTML5

Foi criado, no HTML5, o elemento AUDIO que é um padrão para reprodução de áudio e também deve receber controles de play, pause e stop.

HTML5

```
<button onclick="playPause()">Play/Pause</button>
<audio controls id="audio">
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Seu navegador não suporta a elemento audio.
</audio>
```

JavaScript

```
var meuAudio=document.getElementById("audio");
function playPause()
{
  if (meuAudio.paused)
    meuAudio.play();
  else
    meuAudio.pause();
}
```

[Voltar ao sumário](#)

Recomendação 5.5 – Fornecer controle de animação

[Ver WCAG 2.0 Critério de Sucesso 2.2.2](#)

Para qualquer animação que inicie automaticamente na página devem ser fornecidos mecanismos para que o usuário possa pausar, parar ou ocultar tal animação.



Gif animado de pessoas num ponto de ônibus com controle de animação.

[Voltar ao sumário](#)

3.6 Formulários

Recomendação 6.1 – Fornecer alternativa em texto para os botões de imagem de formulários

[Ver WCAG 2.0 Critério de Sucesso 1.1.1](#)

Ao serem utilizados botões do tipo imagem (input type="image"), que servem para o mesmo propósito do botão do tipo submit, deve ser fornecida uma descrição textual para o botão através do atributo alt, conforme o exemplo a seguir.

Exemplo 1



Imagem de botão Enviar

Código:

```
src="enviar.jpg" alt="enviar">
<input type="image" name="enviar"
```



Acessível
em Libras

Já para outros tipos de botões (reset e button), é preciso substituir o botão pela imagem que se deseja utilizar através do CSS. Nesse caso, para que o botão seja acessível, ele deve possuir um value descritivo, conforme o exemplo a seguir.

Exemplo 2




Imagem de botão Limpar

HTML

```
<input type="reset" name="limpar" value="Limpar" class="btLimpar">
```

CSS

```
input.btLimpar{
  background:transparent
  width:100px;
  height:47px;
  text-indent:-20000px;
  border:0;
}
```

[Voltar ao sumário](#)

Recomendação 6.2 – Associar etiquetas aos seus campos

[Ver WCAG 2.0 Critério de Sucesso 1.3.1 \(Técnica H44\)](#)

As etiquetas de texto (elemento 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.

Exemplo

```
id="nome">
name="sexo">
name="sexo">
```

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

<label>Sexo:</label>
<input type="radio" id="fem"

<label for="fem">Feminino</label>
<input type="radio" id="mas"

<label for="mas">Masculino</label>
```

[Ir ao exemplo navegável Recomendação 6.2](#)

[Voltar ao sumário](#)

Recomendação 6.3 – Estabelecer uma ordem lógica de navegação

[Ver WCAG 2.0 Critério de Sucesso 2.4.3](#)

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.

OBS: O atributo *tabindex* (ver recomendação 4) somente deverá ser utilizado quando existir real necessidade.

[Voltar ao sumário](#)

Recomendação 6.4 – Não provocar automaticamente alteração no contexto

[Ver WCAG 2.0 Critério de Sucesso 3.2.2](#)

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

Exemplo correto

```
Alegre</option>
Horizonte</option>
Janeiro</option>
Paulo</option>
```

```
<select id="cidade" name="cidade">
  <option value="POA">Porto
  <option value="BH">Belo
  <option value="RJ">Rio de
  <option value="SP">São
</select>
```

Exemplo incorreto

```
Alegre</option>
Horizonte</option>
Janeiro</option>
Paulo</option>
```

```
<select name="cidade" id="cidade"
onchange="location = this.options[this.selectedIndex].value;">
  <option value="POA.html">Porto
  <option value="BH.html">Belo
  <option value="RJ.html">Rio de
  <option value="SP.html">São
```


Recomendação 6.5 – Fornecer instruções para entrada de dados

[Ver WCAG 2.0 Critério de Sucesso 3.3.2](#)

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). A utilização de caracteres pré-definidos em áreas de entrada de texto só deve ocorrer se:

- O texto for incluído após a entrada de dados pelo usuário (por exemplo, sugerir um novo nome de usuário caso o escolhido já exista);
- A semântica do documento justifique a inclusão de texto pré-definido (por exemplo, uma loja virtual que só vende para determinado país já vem com o campo país preenchido);
- Os caracteres tenham sido fornecidos previamente pelo usuário (por exemplo, refinamento de busca).

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.

O seguinte exemplo indica que a data precisa ser inserida no formato dia (dd) – mês (mm) – ano (aaaa).

Exemplo 1

</label>
name="data">

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

Campos de preenchimento obrigatório devem ser claramente indicados. Uma solução para o caso de campos obrigatórios em formulários é inserir um asterisco em forma de imagem ao lado de cada campo obrigatório, fornecendo uma descrição no atributo alt para esta imagem, que poderia ser “campo obrigatório” ou, simplesmente, “obrigatório”.

Exemplo 2

alt="obrigatório"/>

</label>
<input type="text" name="nome" id="nome"

<label for="nome">Nome:

<input type="text" name="nome" id="nome"

Para quem enxerga, basta visualizar o asterisco para distinguir os campos obrigatórios. Com o leitor de tela, a informação do alt é lida para o usuário. No caso do exemplo de código acima, o leitor iria informar algo do tipo: “Campo de edição, nome, obrigatório”.

Exemplo 3

Outra solução é utilizar um elemento SPAN contendo a informação “obrigatório”, que será ocultada por CSS (ver Recomendação 6 – tabela com mecanismos para ocultar elementos e seus efeitos na acessibilidade) e, em seu lugar, inserimos a imagem de um asterisco.

HTML

</label>

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

CSS

images/icon_asterisk.gif;

label span{
background-image:

text-indent: -2000px;
font-size: 0;
width: 0.1 em;
height: 0.2em;
}

O HTML5 traz um novo atributo, o required, que especifica que um campo é obrigatório. Essa informação é lida pelo leitor de tela. Além disso, caso o usuário tente enviar o formulário sem preencher o campo obrigatório, uma informação do tipo “por favor, preencha esse campo” aparecerá na tela, além de ser interpretada pelo leitor de tela.

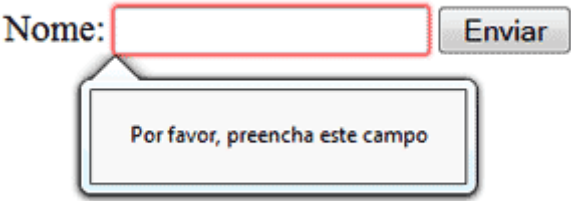


Acessível
em Libras

EM HTML 5

```
<label for="nome">Nome:</label>

<input type="text" name="nome" id="nome" required />
<input type="submit" value="Enviar" />
```



Uso do atributo "required"

Além do atributo required, o HTML5 apresenta, também, o placeholder. Ele é utilizado com os elementos INPUT e TEXTAREA para definir uma dica de preenchimento do campo. Quando o campo recebe o foco, a dica desaparece, mas é lida pelo leitor de tela.

Exemplo

```
seu e-mail">
<input type="email" placeholder="Digite
se">
<input type="submit" value="Inscrever-
```

Digite seu e-mail

Inscrever-se

Uso do atributo placeholder

Outros atributos do HTML5 bastante importantes para acrescentar informações aos campos do formulário são:

- autofocus: Utilizado para o foco do teclado ir diretamente ao campo que possua esse atributo quando a página com o formulário for carregada;
- maxlength: Determina um número máximo de caracteres que o campo pode ter;
- min e max: Determina o valor mínimo e máximo para o campo. Podem ser utilizados separadamente;
- pattern: Permite que sejam definidas expressões para validação do campo;
- novalidate: Para desativar a validação do formulário (utilizado com o elemento FORM);
- formnovalidate: Para desativar a validação dos dados de um formulário (utilizado com os elementos INPUT e BUTTON);
- draggable: Torna o campo arrastável (Drag and Drop);

Além de melhorar a semântica, os novos atributos e valores do HTML5 tornam o desenvolvimento mais fácil e o resultado mais acessível.

Exemplo

```
<form action="#" method="post">
<label>Nome:

<input name="nome" required placeholder="Digite seu nome" autofocus
us >

</label>
<label>E-mail:

<input name="email" type="email" placeholder="Digite seu e-
mail" required>
```

Renderização



Renderização de formulário acessível



[Voltar ao sumário](#)

Recomendação 6.6 – Identificar e descrever erros de entrada de dados e confirmar o envio das informações

[Ver WCAG 2.0 Critério de Sucesso 3.3.1](#)

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

O usuário envia o formulário e deixa de preencher campos obrigatórios ou preenche algum campo de maneira incorreta. O foco retorna ao início do formulário contendo o aviso de erro e links (âncoras) para os campos do formulário que apresentaram erro. Abaixo dos avisos, encontram-se os campos de formulário que contiveram erro de preenchimento e, também, um link que permite ao usuário expandir os demais campos do formulário.

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

Minimo 5 caracteres / sem espaços

▪ Nome completo

Endereço

▪ Logradouro

▪ CEP

Somente números

+ Expandir demais campos

Cadastrar

Limpar

Cancelar

Informação de erro em formulário

Após a validação dos dados, antes de enviar o formulário, uma tela de confirmação deverá aparecer, conforme no exemplo a seguir, permitindo que o usuário verifique e, se necessário, edite as informações antes de enviá-las.

Dados pessoais

Socioeconômico

Experiencia profissional

Necessidades especiais

Página de Confirmação

Formulário de Contato

Nome:

João de Almeida

E-mail:

joao.almeida@provedor.com.br

Assunto:

Informações

Mensagem:

Teste

Editar

Enviar

Página de Confirmação

Em HTML5

No HTML5 existem novos valores para o atributo type do elemento INPUT e, de acordo com a definição do input type, a validação do lado do cliente ocorre através do próprio HTML, já que esses valores basicamente definem o tipo de dado esperado pelo campo. Por exemplo, no código abaixo, temos um INPUT do tipo e-mail. Ao tentar enviar o formulário, se o tipo de texto inserido não for um endereço de e-mail, o navegador irá retornar uma mensagem no próprio campo, solicitando que o campo seja preenchido corretamente.



Acessível
em Libras

Exemplo

<input type="email">

<input type="submit" value="Inscrever-se">

se">

a

Inscrever-se

Por favor, forneça um endereço de email

Uso de novos valores type no INPUT

O HTML5 acrescentou os seguintes valores para o atributo type:

- email: Indica a entrada de um endereço de e-mail;
- tel: Indica a entrada de um número de telefone;
- search: Indica um controle input do tipo busca;
- url: Indica a entrada de endereço URL;
- datetime: Indica a entrada de data e hora;
- datetime-local: Indica a entrada de data e hora local;
- date: Indica a entrada de apenas uma data;
- month: Indica a entrada de ano e mês apenas;
- week: Indica a entrada de ano e dia da semana;
- time: Indica a entrada de um horário com hora, minuto, segundos e fração de segundos;
- number: Indica uma entrada numérica;
- range: Indica a entrada de um número contido dentro de um intervalo;
- color: Indica a entrada do código de uma cor.

[Voltar ao sumário](#)

Recomendação 6.7 – Agrupar campos de formulário

[Ver WCAG 2.0 Critério de Sucesso 1.3.1 \(Técnicas H71 e H85\).](#)

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 1

<form method="post" action="...">

<fieldset>

<legend>Dados

<label for="nome">O seu

<input type="text"

name="nome" id="nome">

Pessoais</legend>

Nome: </label>

No caso do elemento SELECT, pode ser utilizado o elemento OPTGROUP para agrupar os itens da lista de seleção.

Exemplo 2

<label for="instituto">Qual a sua

<select id="instituto" name="instituto">

<optgroup label="Rio Grande do

<option

value="ifrs">IFRS</option>

<option

value="ifsul">IFSUL</option>

instituição?</label>

Sul">

[Ir ao exemplo navegável Recomendação 6.7](#)

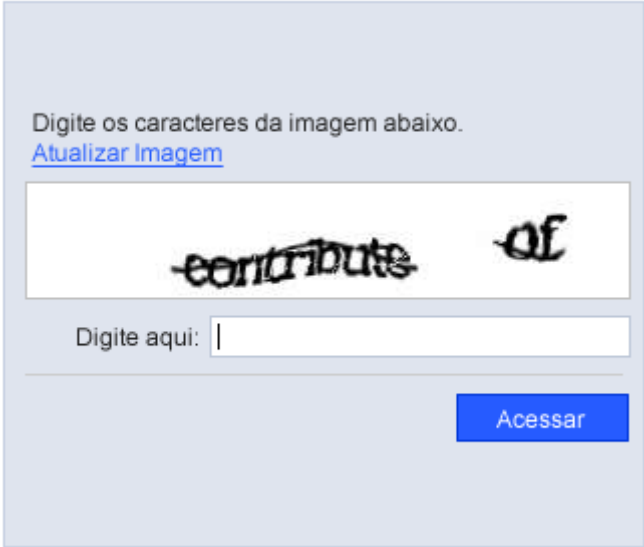
[Voltar ao sumário](#)

Recomendação 6.8 – Fornecer estratégias de segurança específicas ao invés de CAPTCHA

[Ver WCAG 2.0 Critério de Sucesso 1.1.1 \(Técnicas G143 e G144\).](#)



CAPTCHAs são utilizados para impedir que softwares automatizados, conhecidos como bots, executem ações que degradem a qualidade do serviço de um sistema, provocando danos em áreas e e-serviços de sítios em um curto espaço de tempo, podendo sobrecarregar servidores e deixar sítios indisponíveis por um dado período.



Exemplo de CAPTCHA com letras distorcidas

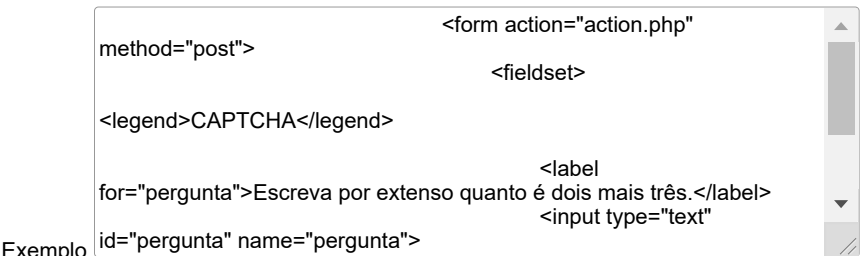
Recomenda-se uma combinação de diferentes estratégias para serviços mais seguros e acessíveis para substituir o uso de CAPTCHA, como por exemplo:

- Limites de conexão;
- Monitoramento;
- Consistência nas políticas de segurança;
- Uso de técnicas de desenvolvimento de serviços e formulários seguros.

Tal recomendação se deve ao fato de que o uso de CAPTCHA pode ocasionar problemas significativos a sítios e formulários:

- Usabilidade: O ônus de detecção de problemas e invasões é delegado a pessoa, ao invés do sistema. Como CAPTCHAs são projetados para serem difíceis de ler e entender, tornam os serviços que os utilizam muito mais difíceis de usar.
- Acessibilidade: Os CAPTCHAS são inacessíveis por sua natureza, não são lidos, nem interpretados por leitores de tela. Isso efetivamente torna o serviço inutilizável por alguns grupos de pessoas. Mesmo CAPTCHAs que oferecem versões em áudio não resolvem completamente o problema, pois muitas pessoas podem possuir deficiência auditiva e visual.
- Segurança: Desenvolver um CAPTCHA internamente costuma gerar CAPTCHAS inseguros, com falhas já mapeadas por spammers. No entanto, ao utilizar CAPTCHAS de terceiros há outros problemas a serem considerados:
 - 1. Privacidade: O serviço de CAPTCHA pode incluir cookies, coletar estatísticas e mapear o comportamento de navegação das pessoas que acessam o serviço. Isto introduz preocupações com a privacidade significativas.
 - 2. Performance: O uso de um serviço CAPTCHA incorre no desempenho do sítio. Se o serviço ficar indisponível, o mesmo acontece com o acesso ao serviço da página que utiliza o CAPTCHA.

Caso o uso de CAPTCHA seja estritamente necessário, o mesmo deverá ser fornecido em forma de pergunta simples de interpretação (CAPTCHA Humano), e este preferencialmente só deverá ser apresentado após pelo menos 2 tentativas de envio do formulário, por exemplo. Tais perguntas poderão ser respondidas apenas por um ser humano. No entanto, é preciso garantir que a pergunta não seja de difícil resolução, permitindo que a mesma possa ser respondida por pessoas de variadas culturas e níveis de instrução. Para tal, podem ser utilizadas perguntas de senso comum, como por exemplo, “qual é a cor do céu?” ou “o fogo é quente ou frio?”. Também podem ser utilizados testes matemáticos. No entanto, é preciso tomar cuidado para que esses testes não sejam facilmente “quebrados” por determinados programas. Uma alternativa é solicitar que o usuário escreva o resultado do teste matemático por extenso, como “escreva por extenso quanto é 2 + 3”, ou ainda “responda por extenso quanto é dois mais três”.



[Ir ao exemplo navegável](#) **Recomendação 6.8**

Para mais detalhes sobre CAPTCHA, verificar o documento Orientações para o uso do CAPTCHA no Governo Federal na seção do eMAG <http://www.governoeletronico.gov.br/eixos-de-atuacao/governo/acessibilidade/material-de-apoio>.



Acessível
em Libras

[Voltar ao sumário](#)

4. Elementos padronizados de acessibilidade digital no Governo Federal

Este capítulo demonstra os elementos padronizados de acessibilidade digital que devem estar presentes em todos os sítios do governo federal para facilitar o acesso ao cidadão.

Os elementos padronizados de acessibilidade são:

1. Teclas de atalho
2. Primeira folha de contraste
3. Barra de acessibilidade
4. Apresentação do mapa do sítio
5. 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>

[Voltar ao sumário](#)

4.1 Atalhos de teclado

Deverão ser disponibilizados atalhos por teclado para pontos estratégicos da página, permitindo que o usuário possa ir diretamente a esses pontos. Os atalhos deverão funcionar através de números precedidos da tecla padrão de cada navegador (Alt no Internet Explorer, Shift + Alt no Firefox, Shift + Esc no Opera, etc.). Os atalhos que deverão existir nas páginas do Governo Federal são os seguintes:

1. para ir ao conteúdo;
2. para ir ao menu principal;
3. para ir à caixa de pesquisa.

As dicas dos atalhos deverão ser disponibilizadas na barra de acessibilidade e na página sobre a acessibilidade do sítio, já comentada anteriormente. Para mais detalhes ver Recomendação 6.

[Voltar ao sumário](#)

4.2 Primeira folha de contraste

A opção alto contraste deve gerar uma página em que a relação de contraste entre o plano de fundo e os elementos do primeiro plano seja de, no mínimo 7:1 (contraste otimizado). Desta forma, a folha principal de autocontraste deve obedecer a seguinte configuração de cores:

1. **Cor de fundo** : independente da cor utilizada, ela deve ser alterada para preto (#000000) ;
2. **Cor de texto** : independente da cor utilizada, ela deve ser alterada para branco (#FFFFFF) ;
3. **Links** : O modo normal do link deve ser sublinhado (para que ele se diferencie do texto normal), assim como o modo hover e o modo active. O link deve ser alterado para amarelo (#FFF333) ;
4. **Ícones** : Todos os ícones devem ser brancos;
5. **Linhas e Contornos** : As linhas e os contornos de elementos devem ser alterados para branco.



Identidade digital - aplicação da folha de alto-contraste

É facultativo a alteração de fotos ou ilustrações complexas como infográficos. Para mais detalhes ver Recomendação 28.



Acessível
em Libras

[Voltar ao sumário](#)

4.3 Barra de acessibilidade

O sítio deverá conter uma barra de acessibilidade no topo de cada página contendo os seguintes itens:

- Alto contraste
- Atalhos (para Menu, Conteúdo e Busca)
- Acessibilidade (link para a página contendo os recursos de acessibilidade do sítio)

Exemplo de Barra de Acessibilidade

<div id="acessibilidade">
<ul id="atalhos">
Conteúdo [1]
Menu [2]
Busca [3]

<ul id="botoes">

Ir para o conteúdo Ir para o menu Ir para a busca Ir para o rodapé

Denominação do órgão

Nome principal

SUBORDINAÇÃO

ACESSIBILIDADE ALTO CONTRASTE MAPA DO SITE

Buscar no portal

Twitter YouTube Facebook

Perguntas frequentes Contato Serviços da [Denominação] Dados abertos Área de imprensa

Cabeçalho da identidade digital de governo (versão verde) com barra de acessibilidade no topo.

[Voltar ao sumário](#)

4.4 Apresentação do mapa do sítio

O mapa do sítio deve ser disponibilizado em forma de lista hierárquica (utilizando os elementos de lista do HTML), podendo conter quantos níveis forem necessários.

Mapa do Site

Uma visão geral do conteúdo disponível no site. Mantenha o ponteiro do mouse sobre o item por alguns segundos para visualizar sua descrição.

Destaques do Portal

Serviços

Perguntas frequentes

Contato

Serviços da [Denominação]

Dados abertos

Área de imprensa

Menu de Apoio

Conheça a identidade digital do governo

Manuais

Detalhe do mapa do sítio da Identidade Digital de Governo.

[Voltar ao sumário](#)

4.5 Página de descrição com os recursos de acessibilidade

Esta página apresenta os recursos de acessibilidade presentes no sítio, como as teclas de atalho disponíveis, as opções de alto contraste, detalhes sobre testes de acessibilidade realizados (validadores automáticos, leitores de tela e validação humana) no sítio e outras informações pertinentes a respeito de sua acessibilidade. O link para a página contendo os recursos de acessibilidade deve ser disponibilizado na barra de acessibilidade.

Um modelo funcional da página pode ser visto em:<http://portalpadrao.gov.br/acessibilidade>

O texto e links considerados na Identidade Digital de Governo:

Acessibilidade

Este portal segue as diretrizes do eMAG (Modelo de Acessibilidade em Governo Eletrônico), conforme as normas do Governo Federal, em obediência ao Decreto 5.296, de 2.12.2004.

O termo acessibilidade significa incluir a pessoa com deficiência na participação de atividades como o uso de produtos, serviços e informações. Alguns exemplos são os prédios com rampas de acesso para cadeira de rodas e banheiros adaptados para deficientes.

Na internet, acessibilidade refere-se principalmente às recomendações do WCAG (World Content Accessibility Guide) do W3C e no caso do Governo Brasileiro ao eMAG (Modelo de Acessibilidade em Governo Eletrônico). O eMAG está alinhado as recomendações internacionais, mas estabelece padrões de comportamento acessível para sites governamentais.

Na parte superior do portal existe uma barra de acessibilidade onde se encontra atalhos de navegação padronizados e a opção para alterar o contraste. Essas ferramentas estão disponíveis em todas as páginas do portal.



Acessível em Libras

Os atalhos padrões do governo federal são:

- Teclando-se Alt + 1 em qualquer página do portal, chega-se diretamente ao começo do conteúdo principal da página.
- Teclando-se Alt + 2 em qualquer página do portal, chega-se diretamente ao início do menu principal.
- Teclando-se Alt + 3 em qualquer página do portal, chega-se diretamente em sua busca interna.
- Teclando-se Alt + 4 em qualquer página do portal, chega-se diretamente ao rodapé do site.

Esses atalhos valem para o navegador Chrome, mas existem algumas variações para outros navegadores:

- Quem prefere utilizar o Internet Explorer é preciso apertar o botão Enter do seu teclado após uma das combinações acima. Portanto, para chegar ao campo de busca de interna é preciso pressionar Alt+3 e depois Enter.
- No caso do Firefox, em vez de Alt + número, tecele simultaneamente Alt + Shift + número.
- Sendo Firefox no Mac OS, em vez de Alt + Shift + número, tecele simultaneamente Ctrl + Alt + número.
- No Opera, as teclas são Shift + Escape + número. Ao teclar apenas Shift + Escape, o usuário encontrará uma janela com todas as alternativas de ACCESSKEY da página.

Ao final desse texto, você poderá baixar alguns arquivos que explicam melhor o termo acessibilidade e como deve ser implementado nos sites da Internet.

Leis e decretos sobre acessibilidade:

- Decreto nº 5.296 de 02 de dezembro de 2004 ([link externo](#))
- Decreto nº 6.949, de 25 de agosto de 2009 ([link externo](#)) - Promulga a Convenção Internacional sobre os Direitos das Pessoas com Deficiência e seu Protocolo Facultativo, assinados em Nova York, em 30 de março de 2007
- Decreto nº 7.724, de 16 de Maio de 2012 ([link externo](#)) - Regulamenta a Lei Nº 12.527, que dispõe sobre o acesso a informações.
- Modelo de Acessibilidade de Governo Eletrônico ([link externo](#))
- Portaria nº 03, de 07 de Maio de 2007 - formato .pdf (35,5Kb) ([link externo](#)) - Institucionaliza o Modelo de Acessibilidade em Governo Eletrônico – e-AG

Dúvidas, sugestões e críticas:

No caso de problemas com a acessibilidade do portal, favor acessar a Página de contato.

Dicas, links e recursos úteis:

- [Acessibilidade Legal](#) ([link externo](#))
- [Acesso Digital](#) ([link externo](#))

[Voltar ao sumário](#)

5. Práticas desaconselhadas

Listamos aqui algumas práticas que devem ser desencorajadas no desenvolvimento de sítios e serviços eletrônicos no governo federal.

Algumas práticas, apesar de comuns, configuram-se não só como empecilhos para o acesso de pessoas com deficiência, mas também, o acesso por dispositivos móveis.

- Uso de animações e aplicações FLASH;



Acessível
em Libras

- Uso de CAPTCHAS em formulários;
- 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, strike, u, b, etc.

O uso de qualquer uma dessas práticas tem um impacto negativo significativo na experiência de uso do usuário.

[Voltar ao sumário](#)

6. Glossário

Acessibilidade: Significa permitir o acesso por todos, independente do tipo de usuário, situação ou ferramenta.

Acessibilidade à Web: Criar ou tornar as ferramentas e páginas Web acessíveis a um maior número de usuários, inclusive pessoas com deficiência.

Agente de Usuário: Qualquer software que recupera e apresenta conteúdo Web para seus usuários, possibilitando que o usuário interaja com o conteúdo Web.

Atributo: Os atributos servem para definir uma propriedade de um elemento (X)HTML.

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart): consiste em um meio automatizado de gerar desafios que, hipoteticamente, apenas seres humanos possam resolver. O tipo mais comum requer que o usuário identifique letras distorcidas em um arquivo de imagem.

CSS (Cascading Style Sheets – Folhas de estilo em cascata): Linguagem que descreve a apresentação, visual ou sonora, de um documento.

Leitor de tela: Software que fornece informações através de síntese de voz sobre os elementos exibidos na tela do computador. Esses softwares interagem com o sistema operacional, capturando as informações apresentadas na forma de texto e transformando-as em resposta falada através de um sintetizador de voz. Para navegar utilizando um leitor de tela, o usuário faz uso de comandos pelo teclado. O leitor de tela também pode transformar o conteúdo em informação tátil, exibida dinamicamente em Braille por um hardware chamado de linha ou display Braille, servindo, em especial, a usuários com surdocegueira.

Navegador: Programa que permite ao usuário consultar e interagir com o material publicado na Web.

Navegador textual: Tipo de navegador baseado em texto, diferente dos navegadores com interface gráfica, onde as imagens são carregadas. O navegador textual pode ser utilizado com leitores de tela por pessoas com deficiência visual e, também, por pessoas que acessam a Internet com conexão lenta.

Padrões Web (Web Standards): Padrões Web são regras de tecnologia definidas pelo W3C e outros órgãos da área utilizados para criar e interpretar o conteúdo Web.

Podcast: Arquivo ou uma série de arquivos de áudio digital, geralmente em formato MP3 ou AAC (este pode conter imagens estáticas e links), publicada através de podcasting (forma de publicação de arquivos de mídia digital pela Internet, através de um feed RSS).

Tag/Elemento: Código usado para marcar o início e o fim de um elemento (X)HTML.

Tecnologia assistiva: refere-se ao conjunto de artefatos disponibilizados às pessoas com deficiência, que contribui para prover-lhes uma vida mais independente, com mais qualidade e possibilidades de inclusão social. O propósito das tecnologias assistivas reside em ampliar a comunicação, a mobilidade, o controle do ambiente, as possibilidades de aprendizado, trabalho e integração na vida familiar, com os amigos e na sociedade em geral.

W3C (World Wide Web Consortium): Consórcio de âmbito internacional com a finalidade de desenvolver especificações, guias e ferramentas para Web.

WAI: (Accessibility Initiative - Iniciativa de Acessibilidade na Web): Tem como objetivo desenvolver protocolos que melhorem a interoperabilidade e promovam a evolução da WWW.

WCAG (Web Content Accessibility Guidelines: Recomendações de Acessibilidade para Conteúdo Web): Abrange diversas recomendações com a finalidade de tornar o conteúdo Web mais acessível.

Webcast: Transmissão de áudio e vídeo por meio da Internet ou redes corporativas para um grupo de pessoas, simultaneamente.

XHTML (Extensible HyperText Markup Language): É a linguagem HTML reformulada para seguir as regras do XML.

XML (Extensible Markup Language): Uma linguagem de marcação semelhante à HTML, mas que permite a descrição de dados definidos por elementos de programação.

[Voltar ao sumário](#)

7. Anexo 1 – Tabela de Contraste de Cores

As tabelas a seguir mostram as cores que, em fundo preto ou branco, apresentam uma relação de contraste de no mínimo 4,5:1.



Acessível
em Libras

#090	#093	#096	#099	#09C	#09F
#0C0	#0C3	#0C6	#0C9	#0CC	#0CF
#0F0	#0F3	#0F6	#0F9	#0FC	#0FF
#390	#393	#396	#399	#39C	#39F
#3C0	#3C3	#3C6	#3C9	#3CC	#3CF
#3F0	#3F3	#3F6	#3F9	#3FC	#3FF
					#66F
#690	#693	#696	#699	#69C	#69F
#6C0	#6C3	#6C6	#6C9	#6CC	#6CF
#6F0	#6F3	#6F6	#6F9	#6FC	#6FF
			#969	#96C	#96F
#990	#993	#996	#999	#99C	#99F
#9C0	#9C3	#9C6	#9C9	#9CC	#9CF
#9F0	#9F3	#9F6	#9F9	#9FC	#9FF
					#C0F
			#C39	#C3C	#C3F
#C60	#C63	#C66	#C69	#C6C	#C6F
#C90	#C93	#C96	#C99	#C9C	#C9F
#CC0	#CC3	#CC6	#CC9	#CCC	#CCF
#CF0	#CF3	#CF6	#CF9	#CFC	#CFF
#F00	#F03	#F06	#F09	#F0C	#F0F
#F30	#F33	#F36	#F39	#F3C	#F3F
#F60	#F63	#F66	#F69	#F6C	#F6F
#F90	#F93	#F96	#F99	#F9C	#F9F
#FC0	#FC3	#FC6	#FC9	#FCC	#FCF
#FF0	#FF3	#FF6	#FF9	#FFC	#FFF

Tabela 1 – Cores contrastantes com a cor preta

#000	#003	#006	#009	#00C	#00F
#030	#033	#036	#039	#03C	#03F
#060	#063	#066	#069	#06C	#06F
#300	#303	#306	#309	#30C	#30F
#330	#333	#336	#339	#33C	#33F
#360	#363	#366	#369	#36C	#36F
#600	#603	#606	#609	#60C	#60F
#630	#633	#636	#639	#63C	#63F
#660	#663	#666	#669	#66C	#66F
#900	#903	#906	#909	#90C	#90F
#930	#933	#936	#939	#93C	#93F
#960	#963	#966			
#C00	#C03	#C06	#C09	#C0C	
#C30	#C33	#C36	#C39		

Tabela 2 – Cores contrastantes com a cor branca



Acessível
em Libras