

Ministério do Planejamento, Orçamento e Gestão
Secretaria de Logística e Tecnologia da Informação
Departamento de Governo Eletrônico
www.governoeletronico.gov.br



Padrões Brasil e-Gov

**Recomendações para Codificação
de páginas, sítios e portais**

Versão 2.0
Maio de 2008

© Copyright 2008, Departamento de Governo Eletrônico (DGE) – Secretaria de Logística e Tecnologia da Informação (SLTI) – Ministério do Planejamento, Orçamento e Gestão (MPOG).

Todos os direitos reservados. Reprodução permitida, mesmo parcial, por qualquer processo, desde que citada a fonte.

www.governoeletronico.gov.br

Brasil. Ministério do Planejamento, Orçamento e Gestão. Secretaria de Logística e Tecnologia da Informação. Departamento de Governo Eletrônico.

Padrões Brasil e-Gov : Recomendações para codificação de páginas, sítios e portais / Ministério do Planejamento, Orçamento e Gestão. Secretaria de Logística e Tecnologia da Informação. Departamento de Governo Eletrônico –

Brasília : MP, 2008.

51p.

1. Tecnologia da informação 2. Sistema de informação 3. Administração pública I. Título.

CDU 658.011.56

Índice

1.	Introdução	4
1.1	Objetivos desse Guia	5
1.2	Responsabilidade e manutenção	5
1.3	Diretrizes	6
2.	Padrões web	7
2.1	Código em camadas	9
2.2	HTML e XHTML	10
2.3	CSS (cascating style sheets)	12
3.	Recomendações	15
3.1	Recomendações gerais	15
3.2	Elementos do cabeçalho	19
3.3	Corpo	22
3.4	Folhas de estilo	25
3.5	Uso de scripts e cookies	28
3.6	Arquivos em outros formatos e uso de plugins	29
4.	Gerenciadores de conteúdo	31
4.1	Escolha do sistema de gerenciamento	31
5.	Orientações gerais sobre testes	33
5.1	Validação	33
5.2	Resoluções de vídeo	33
5.3	Navegadores e sistema	33
5.4	Outros dispositivos	34
5.5	Teste em camadas	35
5.6	Teste de acessibilidade	36
6.	Problemas mais comuns	37
7.	Lista de verificação	40
8.	Lista de formatos de arquivos e plugins	41
9.	Recursos	45
10.	Leitura complementar	48
11.	Glossário	49

Agradecimentos

Fabio Augusto Barbosa Gameleira
Luis Flavio Loreto da Rocha
Luiz Agner
Maysa Alves
Miriam Hitomi Simofusa
Nohab Santos Carvalho Rocha
Thiago Prado de Campos

Pela contribuição, através da Consulta Pública, com sugestões, esclarecimentos e correções para o presente documento.

A participação de todos foi inestimável.

DGE – Departamento de Governo Eletrônico

1. Introdução

As Tecnologias de Informação e Comunicação (TICs) são instrumentos cada vez mais importantes no desenvolvimento de ferramentas que melhorem a prestação de serviços e informações para os cidadãos. A adoção dessas tecnologias permite o aumento da eficácia, da eficiência e da transparência governamental.

A adoção de meios eletrônicos para a prestação dos serviços governamentais exige que os sítios e portais desenvolvidos e mantidos pela administração pública sejam fáceis de usar, relevantes e efetivos. Somente por meio da eficiência é possível aumentar a satisfação dos usuários de serviços eletrônicos e conquistar, gradativamente, uma parcela cada vez maior da população.

É, portanto, responsabilidade da administração pública oferecer ao cidadão a melhor experiência possível de acesso ao governo eletrônico, respeitando inclusive, as particularidades da população atingida.

Entre os vários aspectos relacionados ao uso da Internet para a prestação de serviços por meios eletrônicos, destaca-se a questão da codificação de páginas dos sítios e portais que devem seguir os padrões web e de boas práticas de implementação para garantir seu desenvolvimento estruturado e evolutivo.

Sítios e portais que seguem recomendações de codificação são mais ágeis e versáteis, além de oferecer facilidades para a portabilidade das mesmas soluções para outros meios eletrônicos como aparelhos celulares, computadores de mão entre outros. A solução dos problemas mais comuns de formatação gera ganhos imediatos e, em geral, custa muito pouco para resolver.

O Departamento de Governo Eletrônico (DGE) publica esse Guia para orientar e facilitar o desenvolvimento de sítios e portais governamentais seguindo as melhores práticas da web. Trata-se de uma primeira versão que reflete um acúmulo de conhecimento e experiências difundidas internacionalmente até o momento. Assim, esse documento se oferece como uma referência para o debate e contribuições que deverão ser incorporadas às próximas versões.

1.1 Objetivos desse Guia

O objetivo desse Guia é detalhar as recomendações de boas práticas em codificação, que orientem as equipes no desenvolvimento de sítios, portais e serviços de governo eletrônico com o propósito de torná-los identificáveis, portáteis, relevantes, acessíveis e efetivos à população.

Esse Guia não tem o objetivo do detalhamento do código. Não citamos, por exemplo, a lista de atributos considerados obsoletos pela W3C (World Wide Web Consortium) porque o documento ficaria muito extenso e perderia rapidamente sua atualidade. A documentação produzida e mantida pela W3C é diversas vezes referenciada nesse Guia e deve ser consultada sempre que necessário. Esse guia foi concebido para ser prático e de fácil consulta.

A codificação constituiu-se de um conjunto de arquivos com função de conteúdo, apresentação e comportamento. É o esqueleto, a estrutura que dá suporte aos aspectos da página relacionados à apresentação, organização, navegação e usabilidade.

Além de apresentar recomendações que sigam os padrões web preconizados pela W3C e de boas práticas recomendadas por outros grupos, esse guia apresenta orientações para testes e escolha de gerenciadores de conteúdo.

1.2 Responsabilidade e manutenção

A manutenção desse documento é responsabilidade do Departamento de Governo Eletrônico - DGE. Questões sobre dúvidas ou informações devem ser enviadas para padroesgov@governoeletronico.gov.br

1.3 Diretrizes

As orientações desse Guia seguem quatro diretrizes:

- **Páginas leves**

As páginas devem ter preferencialmente até 50 kb, somados código, conteúdo e imagens. Recomenda-se que as páginas não ultrapassem 70kb de tamanho;

- **Separação da forma do conteúdo**

Separe a estrutura e o conteúdo da página de sua codificação e visual. O posicionamento, a forma e a cor devem ser criados a partir de folhas de estilo (CSS);

- **Páginas em conformidade com os padrões web**

Padrões web são especificações abertas que prevêm a acessibilidade desses documentos ao maior grupo de indivíduos possível; e,

- **Páginas independentes de navegador e plataforma**

A utilização de elementos, atributos ou comandos proprietários podem comprometer, prejudicar e até impedir a visualização e o perfeito funcionamento da página em outros navegadores ou sistemas operacionais.

2. Padrões web

Padrões web são especificações abertas estabelecidas pelo W3C, utilizadas para criar e interpretar o conteúdo web. Essas tecnologias são desenvolvidas prevendo a acessibilidade desses documentos ao maior grupo de indivíduos possível.

Quando um documento segue os padrões web, significa que:

- É escrito em uma linguagem de marcação válida, mais comumente em HTML ou XHTML;
- Utiliza folhas de estilo (CSS) para a apresentação;
- Está estruturado de forma semântica;
- Funciona em qualquer navegador ou dispositivo que acesse a rede.

Estruturar um documento de forma semântica significa utilizar os elementos da linguagem de acordo com a função para as quais foram criados. Ao utilizar uma marcação semântica, o documento se torna compreensível para qualquer navegador ou dispositivo incluindo aqueles baseados em texto.

Alguns desenvolvedores, principalmente os preocupados com a parte visual da página, pensam que *'funcionar em qualquer navegador'* quer dizer *'tem o mesmo visual em qualquer navegador'* que é uma forma errônea de pensar o desenvolvimento de páginas e aplicativos para web.

Documentos web são visualizados a partir de uma vasta gama de navegadores, versões, resoluções de tela e cor. Usuários podem modificar as características dos navegadores de acordo com suas necessidades e preferências. Esta é uma característica que está unida estruturalmente ao conteúdo web. Não há necessidade que o conteúdo tenha o mesmo visual em todas as plataformas, o que é **absolutamente necessário** é que o **conteúdo, a aplicação seja acessível, fácil de ler e que funcione no maior número de plataformas e dispositivos.**

Desenvolver e evoluir páginas, sítios e portais aplicando os padrões web requer um esforço inicial para o abandono de velhas práticas, mas também oferece uma série de benefícios tais como:

- **Simplifica o desenvolvimento e a manutenção**

Os ciclos de desenvolvimento e manutenção são mais curtos, pois o código é menor, sem tabelas de estrutura e aninhadas ou elementos desnecessários;

- **Independência de “legado”**

Utilizar e conhecer os padrões web permite que os membros da equipe compreendam o código independente de quem o tenha escrito;

- **Estabelece métricas consistentes**

O desenvolvimento seguindo os padrões web tem parâmetros consistentes de qualidade de código;

- **Compatibilidade com navegadores futuros e outros dispositivos**

Utilizar os padrões web reduz o risco das páginas não serem funcionais em outros dispositivos ou navegadores futuros;

- **Carregamento e apresentação de página mais ágil**

A utilização de padrões web incorre em menos código escrito e os navegadores conseguem montar a página de forma mais ágil;

- **Melhora na acessibilidade**

Separar a estrutura da apresentação auxilia a interpretação do conteúdo por leitores de tela e dispositivos alternativos de navegação;

- **Melhor posicionamento nos resultados de busca**

Assim como a separação do conteúdo da apresentação facilita a acessibilidade também o torna mais representativo aos motores de busca;

- **Facilidade de evoluir e adaptar**

A adoção de padrões web facilita a transposição para outras mídias, como a impressão das páginas e outros dispositivos como PDAs e celulares apenas utilizando outra folha de estilo;

- **Diminuição nos custos de hospedagem, manutenção e banda**

O redesenho de páginas seguindo os padrões web tem um impacto no tamanho/peso dos arquivos e uma diminuição no tempo de carregamento e número de arquivos requisitados, reduzindo os custos com hospedagem e banda.

2.1 Código em camadas

Desenvolver códigos seguindo os padrões web significa utilizar as linguagens seguindo os objetivos para as quais elas foram desenvolvidas. Dessa forma, o código é organizado em camadas de conteúdo, de apresentação e de comportamento. Nesse Guia trataremos da camada de conteúdo e da camada de apresentação.

Camada de conteúdo

O desenvolvimento deve sempre começar por esta camada, assim, todo o restante do processo de desenvolvimento vai depender de como o conteúdo está estruturado. Caso o conteúdo não esteja bem estruturado é provável que as camadas posteriores apresentem problemas e dificuldades no desenvolvimento que seriam facilmente resolvidas se a camada de conteúdo tivesse sido formatada corretamente.

Para desenvolver essa camada são utilizadas linguagens de estruturação como HTML, XHTML, WML ou XML. O conteúdo é dividido em suas partes lógicas com seus blocos de informações, utilizando os elementos corretos na sua marcação.

Uma forma de testar se a camada de conteúdo foi estruturada corretamente é abrir o documento, sem estilos, no navegador e verificar se é compreensível com uma estrutura seqüencial que permita a leitura de forma natural.

Para uma melhor compreensão, a partir deste ponto denominaremos os arquivos de conteúdo simplesmente como documento HTML, mesmo que este tenha conteúdo XHTML, WML ou XML.

Camada de apresentação

A camada de apresentação é a parte visual do conteúdo. Ela deve ser construída após a camada de conteúdo. A camada de apresentação utiliza as linguagens CSS e XSLT.

O desenvolvimento da camada de apresentação deve primar pela simplicidade. Deve-se evitar utilizar propriedades que causem problemas em dispositivos ou alterar a camada de conteúdo apenas para satisfazer uma necessidade específica da apresentação. O resultado da apresentação deve ser testado em diversos navegadores, principalmente os mais usados. Para maiores detalhes, veja o capítulo *Orientações gerais sobre testes*.

Camada de Comportamento

É utilizada para modificar o comportamento dos elementos presentes na camada de conteúdo, visando melhorar a experiência do usuário. Essa camada é opcional e geralmente, não suportada por dispositivos mais antigos. A página ou aplicação deve funcionar na ausência desse suporte.

Essa camada é construída utilizando linguagens de script (javascript) e modelos de objeto (DOM – Document Object Model).

2.2 HTML e XHTML

O HTML (HyperText Markup Language), linguagem de marcação de hipertexto, é a linguagem usada para a codificação de documentos hipertexto.

O HTML é composto por elementos semânticos, que por sua vez são compostos por tags (etiquetas) que são os comandos de codificação da linguagem. Cada elemento HTML tem sua função estrutural como cabeçalhos, parágrafos, quebras de linha, entre outros.

Um elemento geralmente é composto de tag de abertura, conteúdo e tag de fechamento. A tag de fechamento é opcional.

```
Ex: <p>Isto é um parágrafo.</p>
```

No exemplo acima, <p> é uma tag de abertura de parágrafo e </p> uma tag de fechamento. Há elementos que não utilizam tags de fechamento, como o IMG , por exemplo. Todos os elementos podem ter atributos. Os atributos definem uma propriedade do elemento e são colocados sempre na tag de abertura.

Ex: o atributo alt que indica a substituição da imagem pelo texto alternativo.

```
.
```

O XHTML (Extensible HyperText Markup Language), é a reformulação do HTML 4 para XML 1.0. Alguns pontos devem ser observados relativos a diferenças entre o XHTML e o HTML:

- **Deve-se usar caixa baixa e aspas em todos os valores**

Ex:

correto: ``

errado: ``

- **Todos elementos devem ser fechados**

Incluindo os elementos que em HTML não possuem tag de fechamento:

Ex:

correto: ``

errado: ``

correto: `<p> Todos os elementos devem ser fechados.</p>`

errado: `<p>Todos os elementos devem ser fechados.`

- **Nenhum atributo pode ser abreviado**

Em HTML, alguns atributos podem ser minimizados, mas em XHTML isso não é permitido.

Ex:

correto: `<input type="checkbox" id="checkbox1" name="checkbox1" checked="checked" />`

errado: `<input type="checkbox" id="checkbox1" name="checkbox1" checked>`

- **Não utilizar elementos e atributos em desuso**

Essa regra vale tanto para a adoção do XHTML 1.0 Strict quanto o HTML 4.01 Strict. Alguns elementos em desuso: FONT, CENTER - e atributos: alink, width, height (para alguns elementos) e background. Para uma lista completa, consulte o site da W3C.

2.3 CSS (cascating style sheets)

O CSS, folhas de estilo em cascata, deve ser utilizado para controlar todo o leiaute de um documento. O suporte ao CSS varia de navegador para navegador, mas de forma geral, todos suportam o CSS1, apesar de algumas diferenças.

Sintaxe CSS

Uma regra CSS consiste em um seletor com uma ou mais declarações. O seletor determina qual elemento HTML é afetado pela regra. Cada declaração consiste em uma propriedade e seu valor. O bloco de declaração é encapsulado por chaves "{ }" e cada declaração é encerrada por um ponto-e-vírgula ";" .

```
Ex: h1 { font-weight:bold; color: #fc0; }
```

Onde, h1 é o seletor que significa que a regra afeta o elemento <h1>, a regra contém duas declarações que indicam que o elemento está em negrito e tem a cor laranja.

Media Types

Através dos *media type*, é possível criar uma folha de estilos para cada dispositivo que o usuário for acessar, otimizando sua apresentação. A especificação prevê diversos dispositivos, nem todos com implementação plena:

- all – todos os tipos de dispositivos;
- screen – computadores;
- print – impressoras;
- handheld – PDAs, mobiles palmtops;
- projection – apresentações, slides;
- aural - sintetizadores de voz;
- braille - dispositivos táteis em braille;
- embossed - impressoras de Braille;
- tv – televisões;
- tty - terminais, dispositivos portáteis.

Atualmente, apenas as media types, **screen** e **print** funcionam perfeitamente. As demais se encontram em estudo e implementação.

Uma forma de permitir a melhor acessibilidade dos conteúdos é evitar o atributo *media all*. O mais correto é utilizar a *media screen* quando a folha de estilos for desenvolvida para apresentação no computador.

Chamadas de CSS

Há três formas de chamar ou anexar uma folha de estilo em um documento HTML. A indicada pelos padrões e boas práticas é a folha externa. As folhas internas e estilos inline (estilos dentro do código) devem ser evitados, pois misturam no documento HTML a apresentação e a estrutura.

- **Externa:** a folha de estilo se encontra em um ou mais arquivos separados. É o mais recomendado, por possuir diversas vantagens: a folha de estilos é carregada apenas uma vez, o tamanho do documento HTML é menor, e é preciso editar apenas um arquivo para mudar a apresentação de todo o sitio.

Os arquivos contendo o CSS são chamados no cabeçalho do documento na seguinte forma:

```
<link rel="stylesheet" type="text/css" media="screen" href="style.css" />
```

ou utilizando a regra @import rule dentro do elemento <style>:

```
<style type="text/css">@import url("style.css") screen;</style>
```

- **Interna:** a folha de estilos é descrita no elemento <style> dentro do cabeçalho <head>.

```
<style type="text/css">h1 {font-weight:bold;}</style>
```

- **Inline:** a regra CSS encontra-se no corpo do documento HTML.

```
<h1 style="font-weight:bold;">Nosso título</h1>
```

A chamada de várias folhas de estilo para diferentes dispositivos pode ser feita das seguintes formas:

- **Externa 1** – importando um arquivo CSS usando o @import, onde primeiro vem o nome do arquivo e depois o dispositivo a que ele se destina.

```
@import url("impressao.css") print
```

- **Externa 2** – Por meio da tag LINK

```
<link rel="stylesheet" type="text/css" media="print" href="impressao.css" />
```

- **Interna** – Utilizando o @media. Nesse caso, pode ser construído tanto dentro do documento HTML (não recomendado) quanto dentro do documento CSS.

No HTML

```
<style>
```

```
@media screen { p {font-family:verdana,sans-serif; font-size:1em;}}
```

```
@media print { p {font-family:times,serif; font-size:10pt;}}
```

```
</style>
```

No CSS

```
@media screen {p {font-family:verdana,sans-serif; font-size:1em;}}
```

```
@media print { p { font-family:times,serif; font-size:10pt;}}
```

3. Recomendações

As recomendações oferecem um grupo abrangente de boas práticas para o desenvolvimento da codificação, levando em conta as necessidades de acessibilidade e usabilidade. As recomendações estão separadas em gerais, para o cabeçalho, para o corpo e apresentação do documento e, para utilização de outros formatos.

3.1 Recomendações gerais

1. Utilize arquivos externos para as Folhas de estilo e javascript

O CSS e o javascript devem estar sempre em arquivos externos a serem chamados pelas páginas do sítio/portal.

Os arquivos CSS podem ser chamados das seguintes formas:

```
<link rel="stylesheet" type="text/css" media="screen" href="style.css" />
ou
<style type="text/css"> @import url("style.css") screen;</style>
```

Já os arquivos javascript devem ser chamados da seguinte forma:

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

2. Limite as requisições HTTP

Muitas requisições HTTP (a quantidade de arquivos chamados: imagens, folhas de estilo, scripts, vídeos, entre outros) reduzem o desempenho do navegador.

Limite o número de requisições. Folhas de estilo e script externos são carregadas apenas uma vez, ficando armazenadas no cache do navegador. Com o cache carregado, transfere-se apenas o conteúdo e eventuais novas imagens.

3. Todas as páginas devem ter recursos de impressão amigável

As páginas devem fornecer recursos de impressão amigável, seja via javascript ou na utilização de uma CSS voltada para impressão. Deve-se verificar a real necessidade de

usuário, retirando da impressão elementos não relevantes como banners e menus. O texto, conteúdo principal, nunca deve ser cortado ou de alguma forma prejudicado na impressão.

O CSS tem uma folha própria para impressão, através da *media type print*, prevendo margens, mudança do valor de pixel para pontos, mudança de fonte e quebras de página. Para a criação de folhas de estilo para impressão, consulte a especificação CSS no sítio do W3C.

4. Evite a utilização do recurso quadros (frame)

Páginas utilizando quadros (frame) eram comuns no início da web e tinham como função economizar o uso de arquivos como cabeçalhos e menus. Com o avanço das novas linguagens e novos aplicativos esse recurso tornou-se obsoleto.

Alguns motivos para o não-uso de quadros:

- A impressão é dificultada, porque muitos usuários não sabem selecionar o quadro certo que querem imprimir;
- Os serviços de busca têm mais dificuldade em indexar páginas com quadros. A página acaba perdendo pontos em sua localização;
- Dificulta a acessibilidade do sítio. Pessoas que não usam navegadores gráficos têm mais dificuldade em navegar em páginas com quadros.

5. Evite o uso de pop-ups

Pop-ups são janelas de diversos tamanhos que abrem com avisos ou publicidade. Esse recurso deve ser evitado a todo o custo em sítios governamentais.

Alguns motivos para o não-uso de pop-ups:

- Os navegadores mais modernos bloqueiam pop-ups. Sendo assim, se algum informe importante for através de pop-ups, não é visto;
- Pop-ups não são acessíveis. Usuários que usam programas leitores de tela podem se perder, sem saber onde estão ou se realmente entraram no sítio pretendido;
- Alguns pop-ups mais modernos costumam burlar o sistema anti-pop-ups dos navegadores, carregando na própria página, atrapalhando a leitura do conteúdo. Esse procedimento é considerado invasivo por muitos usuários.

Informações importantes devem sempre estar em destaque dentro da página. Na diagramação, reserve um espaço para os destaques que podem ser desativados, caso não estejam sendo usados. Nunca use pop-ups para avisos importantes, como recadastramento de usuários, para carregar aplicações ou novas instâncias da página.

6. Documente o código

O código deve ser documentado de forma que qualquer membro da equipe consiga entendê-lo. Blocos de informação devem ser separados com comentários.

```
Comentário (X)HTML
<!-- isso é um comentário html -->
Comentário CSS
/* isso é um comentario css */
```

7. URLs devem ser amigáveis

Cada página de um sítio deve ser identificada com sua URL única e compreensível.

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

Além de facilitar a memorização do endereço para o usuário, a primeira opção permite que mecanismos de busca indexem a página. Muitos mecanismos de busca não indexam páginas que contenham pontos de interrogação "?" ou outro caractere como um "&" ou "=" na sua URL.

8. As URLs devem funcionar sem o "www"

Aconselha-se que a URL funcione sem o www. Muitos navegadores já desconsideram o trígama e deixam o usuário digitar apenas o nome do domínio.

9. Evite elementos ou atributos proprietários, em desuso ou obsoletos

Atributos em desuso (deprecated) ou obsoletos (obsolete) segundo a W3C tendem a não ser suportados nos dispositivos futuros. Consulte o sítio do W3C para a lista

completa de atributos em desuso ou obsoletos.

Atributos proprietários costumam funcionar apenas nos dispositivos para os quais foram desenvolvidos. Devem ser usados como subterfúgios para resolver problemas pontuais do próprio navegador e não devem ser utilizados como a solução padrão para o código.

10. Evite a utilização desnecessária de elementos HTML e classes

Mantenha o código o mais limpo e sintético possível. Utilize os atributos "class" e "id" com parcimônia, assim como os elementos SPAN e DIV.

Ex: Criação de um parágrafo diferenciado.

Nesses casos, deve-se utilizar os seletores contextuais no código CSS:

Correto:

No HTML:

```
<div id="main">
  <div>
    <p> O texto. </p>
  </div>
</div>
```

Na folha de estilos:

```
div#main p { /* regras */ }
```

Errado:

```
<div id="main">
  <div class="maincontent">
    <p class="maincontenttext">O texto.</p>
  </div>
</div>
```

11. Evite a utilização de tabelas para layouts

A implementação atual de folhas nos navegadores torna desnecessária a utilização de tabelas para diagramação. O uso de HTML para estruturar a página e CSS para

formatar a apresentação visual beneficia a acessibilidade da página, inclusive em dispositivos móveis, além de diminuir consideravelmente o tamanho do código.

3.2 Elementos do cabeçalho

Entende-se por elementos do cabeçalho a parte inicial do código, não aparente nos navegadores, que inclui o DOCTYPE, a declaração de linguagem e demais elementos encontrados dentro do elemento head - tags <head></head>.

12. Declare o doctype correto da página

Todo documento HTML e XHTML deve declarar o doctype correto para ser considerado válido. O doctype informa ao navegador o tipo de documento que deve ser usado ao carregar a página. É também por meio da declaração de doctype que as ferramentas de validação analisam o código da página e indicam correções.

Os tipos de doctype existentes são:

- **Strict:** exclui totalmente atributos em desuso e apresentação como cor de fundo de página e tamanho de fontes. Toda a estrutura visual da página deve ser feita a partir de folhas de estilo. A recomendação da W3C é que sempre se dê preferência ao Doctype Strict.

Ex: **HTML 4.01 Strict**

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

- **Transitional:** recomendado para projetos onde há uma transição e dificuldades em curto prazo com sistemas legados. É mais tolerante em relação à utilização de elementos e atributos de apresentação.

Ex: **HTML 4.01 Transitional**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

- **Frameset:** usado quando se utiliza quadro (frames) nos sítios. O uso de quadros não é recomendado e deve ser desestimulado no desenvolvimento de novos sítios.

Ex:

HTML 4.01 Strict Frameset

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

Para a lista completa de DOCTYPES, consulte o sítio da W3C.

13. Declare o idioma utilizado

O idioma principal da página deve ser declarado. Além de auxiliar na acessibilidade do conteúdo, a indicação correta da linguagem auxilia a indexação certa nos buscadores.

A declaração de idioma pode ser realizada de várias formas:

Em HTML 4.01

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

Em XHTML

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt-br" >
```

Como meta etiqueta

```
<meta http-equiv="Content-Language" content="pt-br" />
```

Note que a declaração de idioma também atende as recomendações da **Cartilha Técnica do e-Mag – Modelo de acessibilidade em Governo Eletrônico**.

Além da linguagem, pode-se indicar a localização da página e o país-alvo por meio das seguintes meta-etiquetas:

```
<meta name="target_country" content="br" />
```

```
<meta name="country" content="Brazil" />
```

14. O título deve ser relevante e presente em todas as páginas

O título de uma página é o enunciado entre as tags <title></title> e deve ser um dos elementos obrigatórios do cabeçalho. É importante que o título esteja de acordo com o conteúdo. O título define o rótulo padrão dos favoritos e é a chamada na listagem dos motores de busca. Recomenda-se:

- Evitar expressões redundantes como “bem-vindo ao sítio do ministério x” ou “sítio do ministério x”, “página”, “homepage”, entre outros;
- Não utilize qualquer tipo de slogan no início do título. Se for necessário, utilize após o nome do órgão ou projeto separado por hífen:

Ex: <title>Projeto Verdeamarelo – Mais desenvolvimento e cidadania</title>

- Evite o emprego de siglas de instituições ou de seus departamentos, projetos ou programas;
- Utilizar, de preferência, o assunto da página seguido pelo nome do órgão separado por hífen:

Ex: <title>Contato - Ministério do Planejamento Orçamento e Gestão </title>

- O título deve ter no máximo 60 caracteres;
- Não “polua” o título com palavras-chave. A inserção de palavras-chave antes ou após o título principal da página pode confundir os usuários que utilizam leitores de tela. Utilize as tags meta para palavras-chave.

Importante: Alguns buscadores apenas mostrarão os primeiros 50 – 60 caracteres do título, enquanto outros mostrarão o título completo.

15. Descreva a codificação de caracteres da página

De acordo com a W3C, o conjunto de caracteres mais adequados para documentos XML e HTML 4.0 é Unicode (utf-8). É importante que a codificação de caracteres para qualquer documento seja claramente indicada, de modo a que os dispositivos do cliente possam facilmente mapear a codificação. A indicação Unicode (utf-8) poderá ser feita das seguintes formas:

- Usar o parâmetro 'charset' no Content-Type do cabeçalho HTTP:

```
Content-Type: text/html; charset=utf-8
```

- Em XHTML pode-se usar o pseudo-atributo para codificação na declaração XML no início do documento ou a declaração textual no início da entidade:

```
<?xml version="1.0" encoding="utf-8" ?>
```

- Para HTML ou XHTML pode-se usar a tag <meta> dentro do cabeçalho <head> do documento. Sendo que em XHTML, deve-se colocar uma barra "/" no final:

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

3.3 Corpo

16. Utilize os elementos corretos para a marcação do código

Utilize os elementos de marcação de acordo com a sua finalidade. Para marcar títulos utilize <h1>, <h2>, <h3>, <h4>, <h5> ou <h6> dependendo do nível do título ou subtítulo, sendo <h1> o título mais alto. Use o elemento <p> para demarcar os parágrafos. Não crie classes de CSS para apresentar um elemento em um formato não condizente com seu objetivo.

Ex: Um título de nível dois.

Correto: <div id="main">
 <h2> O subtítulo</h2>
 </div>

Errado: <div id="main">
 <p class="titulosec"> O subtítulo </p>
 </div>

17. A página deve possuir apenas um elemento H1

Toda página deve possuir necessariamente um e somente um elemento H1. O conteúdo do cabeçalho deve guardar semelhança com o conteúdo do elemento TITLE do cabeçalho.

18. Marque listas de itens e objetos de forma adequada

Existem três tipos básicos de listas: ordenada, não-ordenada e de definição. Deve-se escolher o tipo de lista a partir da informação que se quer inserir.

- **Lista ordenada (ordered list – tag OL):** sempre que a lista necessitar de uma ordem, como a classificação em um concurso público, a lista ordenada é a escolha correta.

Ex:

```
<ol>
  <li>Primeiro colocado</li>
  <li>Segundo colocado</li>
  <li>Terceiro colocado</li>
</ol>
```

- **Lista sem ordem (unordered list – tag UL):** utilizar quando a necessidade for listar itens sem uma ordem de colocação. É a forma ideal para a codificação de menus e barras de navegação.

Ex:

```
<ul>
  <li>Quem somos</li>
  <li>Projetos</li>
  <li>Serviços</li>
</ul>
```

- **Lista de definições (definition list – tags DL, DT e DD):** tem por objetivo exibir uma lista de itens e suas respectivas definições. A lista de definições possui duas partes distintas. O termo a ser definido ou descrito (tag DT), e a descrição propriamente dita (tag DD). É o formato ideal para glossários.


```
Ex.:
<dl>
  <dt>Navegador</dt>
  <dd>Programa que permite ao usuário consultar e interagir com o
  material publicado na Internet. </dd>
  <dt>Padrões web (web standards)</dt>
  <dd>Padrões web são regras de tecnologia definidos pelo W3C e outros
  órgãos da área, utilizados para criar e interpretar o conteúdo web.</dd>
</dl>
```

Para a formatação visual na folha de estilos, além dos elementos comuns as demais tags, as listas possuem a propriedade list-style-type (tipo de estilos de lista). Os atributos variam de acordo com o tipo de lista.

Ex: A lista ordenada tem como padrão a lista numérica em ordem crescente, mas é possível mudar de acordo com o atributo dado:

list-style-type: lower-alpha; (a, b, c, d, e, ...)

list-style-type: upper-roman ; (I, II, III, IV, V,...)

list-style-type: decimal-leading-zero ; (01, 02, 03, ...)

list-style-type: none (nenhum estilo de lista aparece)

Para mais informações, consulte a especificação CSS no sítio da W3C.

19. Tabelas devem ser utilizadas para dados tabulares

Para tornar os dados das tabelas acessíveis é importante utilizar os diversos componentes disponíveis como o cabeçalho <th> e o atributo de sumário.

Para maiores detalhes da marcação acessível em tabelas, consulte a **Cartilha Técnica do e-MAG**.

20. Formate os formulários de forma correta e acessível

O formulário deve ser construído de forma lógica utilizando os elementos disponíveis corretamente. Os elementos <label>, <fieldset> e <legend> devem ser utilizados.

Um formulário bem estruturado recebe, sem problemas, sua formatação via CSS.

Para maiores detalhes da marcação acessível em formulários, consulte a **Cartilha Técnica do e-MAG**.

3.4 Folhas de estilo

21. A folha de estilos deve ser externa

Apesar de permitido pelo W3C, o uso de estilos *inline* (estilos dentro do código) para o HTML 4.01 e para o XHTML 1.0 é desaconselhado, porque mistura as camadas de estrutura e apresentação, dificultando a manutenção do código.

A folha de estilos (CSS) deve estar sempre em um ou mais arquivos à parte do código, mantendo a separação entre estrutura e apresentação.

22. A página deve ser compreendida e usável com o CSS desabilitado

Alguns recursos como o uso de imagens no lugar de texto ou de menus dinâmicos podem incorrer na perda de acessibilidade e compreensão quando o CSS é desabilitado. Assegure-se que a página é compreensível e usável com o CSS desabilitado.

23. Nomeie classes e IDs pela sua função, não pela apresentação

O nome de classes e IDs devem ser pensados de acordo com a sua finalidade e não por sua localização na diagramação ou suas características. Isso evita que ao mudar o visual da página, o sentido do nome da classe ou ID se perca.

Ex: Uma div que contenha uma lista vertical de menu:

Correto: #menu {/* regra */}

Errado: #colunaesquerdaverical {/* regra */}

24. Ordem e organização

Organize a folha de estilos de forma que faça sentido aos diversos desenvolvedores do projeto. Crie blocos de regras lógicas, separados por comentários no início e no final de cada um. Uma lista lógica de regras pode ser separada da seguinte forma:

- Regras para estrutura geral do código (*, html, body, cabeçalho, rodapé, entre outros);
- Regras para navegação;
- Regras para as divs na ordem que aparecem no código;
- Regras gerais para seletores (H1, H2, ...);
- Regras para links;
- Regras CSS para listas;
- Regras CSS para elementos dentro das divs;
- Regras CSS para classes gerais;
- Outras regras.

Dentro de cada regra, defina uma ordem de propriedades (ex: alfabética) e mantenha essa ordem para todos os seletores.

Abrevie as regras CSS para as propriedades que assim as admitem. Algumas propriedades como *margin* e *padding* permitem que se abrevie seus valores, seguindo sempre a ordem de topo, direita, fundo e esquerda:

Ex: Dada a declaração:

```
#conteúdo {
    margin-top: 10px;
    margin-right: 8px;
    margin-bottom: 0;
    margin-left: 20px;
}
```

Ela pode ser abreviada da seguinte forma:

```
#conteúdo {margin: 10px 8px 0 20px; }
```

Especifique sempre unidades para valores não iguais a zero. O CSS requer que se especifique as unidades (*pixels*, *ems*, porcentagem, entre outros) em propriedades como *padding*, *width*, *height* e *font-size*. A exceção é quando esse valor é igual a zero (ex: *padding=0*;). Nesse caso, zero é zero e não importa que unidade esteja sendo utilizada.

Links e suas pseudo-classes devem ter regras CSS declaradas em ordem correta para funcionarem corretamente:

```
1º a:link { /* regra */};
2º a:visited { /* regra */};
2º a:hover { /* regra */};
4ºa:active { /* regra */}.
```

Seletores semelhantes devem ser agrupados sempre que for possível:

Ex: Todos os cabeçalhos em negrito com uma linha de borda:

```
h1, h2, h3, h4, h4, h5, h6 {
    border: 1px solid #000;
}
```

Utilize a herança CSS e o elemento cascata para simplificar a folha de estilos.

Ex:

```
h1, h2, h3, h4, h4, h5, h6 {
    border: 1px solid #000;
    color: #000;
    text-align:right;
}
h1 { color: #0000cd; }
```

Dessa forma, os títulos H1 passam a ter a cor azul, mas mantendo todas as características das declarações anteriores. Para que o efeito de cascata funcione é importante que a regra que altera a declaração ocorra **depois** da regra geral.

25. Declare famílias de fonte alternativas

Nem sempre o sistema/navegador que o usuário utiliza possui a fonte escolhida, devendo-se fornecer alternativas. Um conjunto de três fontes, terminando com uma genérica é o ideal. Deve-se testar o desenho com as fontes alternativas para verificar se há problemas de leitura.

Ex: Declare a família de fontes no início do documento, dessa forma não é necessário declará-la em nenhum outro seletor:

```
body {font-family: Verdana, Helvetica, Sans-serif; }
```

Ps: Outros seletores que podem ser utilizados: html e * (seletor universal)

26. Utilize preferencialmente unidades de tamanho relativas

Evite o uso de tamanhos de fontes em unidades absolutas “px” ou “pt” nas folhas para mídias digitais, dando preferência pelo uso de tamanhos relativos tipo “em” e “%”.

Para o tamanho padrão de texto utilize 1em ou 100%, use tamanhos menores apenas para informações de copyright ou semelhante.

27. Utilize preferencialmente letras para nomear classes e IDs

Nomeie classes utilizando apenas letras. O uso de números, *underscore* e caracteres especiais, quando não proibidos, são permitidos com restrições. O hífen pode ser utilizado com parcimônia e não pode ser utilizado no início do nome da classe ou ID.

3.5 Uso de scripts e cookies

28. Não utilize scripts que não ofereçam um benefício relevante ao conteúdo

O uso de scripts deve ser evitado se não tiver uma função relevante ao funcionamento da página: textos animados, janelas pop-ups, efeitos especiais, entre outros, devem ser evitados.

29. O documento deve ser acessível mesmo com o script desabilitado

A camada de comportamento deve ser opcional, pois pode não funcionar em todos os dispositivos.

30. Evite soluções proprietárias e teste o script em diversos navegadores

O script deve ser testado em diversos navegadores para atestar sua funcionalidade. Evite soluções proprietárias que funcionem apenas em um navegador.

31. Forneça uma alternativa ao conteúdo script utilizando o atributo

NOSCRIP

Esse atributo pode conter um link para uma alternativa acessível do conteúdo. Para maiores detalhes sobre alternativa acessíveis consulte, a **Cartilha Técnica do e-MAG**.

32. O conteúdo e o propósito de um *cookie* deve ser sempre informado ao usuário

O motivo da instalação de um *cookie* deve ser sempre avisado ao usuário. O usuário deve ter a opção de recusar um *cookie* se achar conveniente.

33. O usuário pode recusar o uso de um *cookie* sem afetar a usabilidade central do conteúdo

A recusa da instalação de um *cookie* não deve afetar o objetivo central do sítio. O *cookie* não deve ser utilizado como prerrogativa para acesso da página principal do sítio ou tarefas banais, como uma escolha de seção.

3.6 Arquivos em outros formatos e uso de plugins

34. Avalie a real necessidade do uso de um arquivo para baixar

Publicações complexas e extensas, com muitos gráficos e dados tabulares, costumam ser dispostas em formatos prontos para impressão, como o PDF. No entanto, é importante lembrar que muitos desses formatos não são acessíveis. Muitas vezes o arquivo tem um tamanho grande demais e demora a ser baixado pelo usuário. Avalie com cuidado essa solução e sempre prefira disponibilizá-la em HTML.

35. O nome do arquivo deve ser relacionado ao seu conteúdo

O nome do arquivo a ser baixado deve estar de acordo com o conteúdo do documento. Nomes genéricos como "planilha01" ou apresentação.ppt devem ser evitados. Nomes compostos devem ser separados por hífen "-", nunca deixe espaços em branco ou *underscore*.

36. Evite o uso de formatos proprietários ou não acessíveis

Toda informação contida em um formato proprietário ou não acessível deve estar também disponível em outra opção acessível e não-proprietária.

37. Forneça alternativa em texto para vídeo e áudio

Para garantir a acessibilidade, deve haver descrições dos arquivos de áudio e vídeo. Para maiores detalhes, consulte a **Cartilha Técnica do e-MAG**.

38. Informe o tamanho e o formato do arquivo a ser baixado

Permite que o usuário programe a baixa de arquivos e tenha uma idéia de quanto tempo a tarefa demorará a ser concluída.

39. Avalie a real necessidade do uso de um *plugin*

Plugins necessitam de instalação e manutenção por parte do usuário. Nem sempre essa manutenção é fácil de ser realizada, mesmo por pessoas experientes. Grande parte dos *plugins* não é acessível, necessitando que sejam criadas versões alternativas acessíveis.

40. Avise e forneça um endereço de onde o *plugin* deve ser baixado

Avise claramente ao usuário a necessidade do *plugin*. Na mesma página deve-se fornecer o endereço web para sua obtenção. A URL fornecida deve ser da empresa que mantém o programa. *Plugins* e leitores não devem ser hospedados em páginas do governo, pois criam tráfego desnecessário.

41. Nenhuma instalação deve ser necessária para acessar a página inicial ou executar tarefas banais

A exemplo dos cookies, nenhuma instalação de *plugin* deve ser necessária para acessar a página inicial ou executar qualquer tarefa rotineira ou banal (ex: enviar um formulário de contato) do sítio.

4. Gerenciadores de conteúdo

O uso de gerenciadores de conteúdo, também conhecidos por CMS (*Content Management System*) oferece muitas vantagens no desenvolvimento, gerenciamento e manutenção de um sítio em comparação com a criação manual de páginas.

- **Facilidade no gerenciamento de permissões:** um bom gerenciador de conteúdo permite a criação de perfis diferenciados de usuário de acordo com suas necessidades e responsabilidades. Alguns usuários podem ser responsáveis apenas pela publicação de notícias, outros podem administrar seções completas de um sítio, entre outros;
- **Manutenção da árvore de navegação:** a maior parte dos gerenciadores de conteúdo é capaz de criar estruturas lógicas que podem ser utilizadas como menus e são atualizadas automaticamente pelo gerenciador;
- **Ferramentas e extensões:** muitos gerenciadores de conteúdo já vêm com ferramentas como sistemas de busca e de criação de URLs amigáveis. Também é comum existirem extensões que podem ser adicionadas para atender necessidades específicas como a criação de fóruns;
- **Consistência visual:** por ser baseada em modelos, a manutenção da consistência visual é facilitada.
- **Conformidade com os padrões web e acessibilidade:** os gerenciadores costumam vir com 'modelos de fábrica' de acordo com os padrões da W3C e, em alguns casos, de acordo com as regras de acessibilidade internacionais.

4.1 Escolha do sistema de gerenciamento

Antes da escolha do sistema a equipe deve fazer um levantamento de suas necessidades. Existem diversos gerenciadores de conteúdo com diferentes níveis de complexidade, tamanhos e suporte. Um gerenciador de conteúdo "x" que serve para o desenvolvimento de um portal, pode ser complexo demais para a manutenção de um sítio promocional. Além dessa análise, outros fatores a serem levados em conta são:

O gerenciador cria páginas de acordo com os padrões web?

As perguntas “O gerenciador está de acordo com os padrões web?” e “Seus modelos nativos estão validados?” são fundamentais para verificar se o gerenciador de conteúdo está de acordo com os padrões web;

O gerenciador cria páginas HTML acessíveis?

É importante lembrar que, apesar do gerenciador vir com estas características nativas, as mesmas podem ser perdidas na confecção dos modelos ou customizações no sistema;

O gerenciador possui todos os elementos necessários à criação do sítio e/ou possui extensões que supram a necessidade do administrador?

É importante saber que recursos serão necessários no sítio: chats, ferramentas de fóruns e, se o gerenciador dispõem delas;

O gerenciador de conteúdo possui uma boa rede de suporte? Fóruns e páginas especializadas onde se possam esclarecer as dúvidas?

Antes de adotar qualquer ferramenta é interessante fazer um estudo do material de apoio. As FAQs são fáceis de serem compreendidas? O fórum é ativo e as respostas às dúvidas são condizentes? Qual o apoio em português?

O gerenciador permite a criação de múltiplos perfis?

O gerenciador permite a criação de perfis de publicação de acordo com as áreas, seções e níveis de responsabilidade (ex: usuário, editor, administrador)?

Qual a complexidade do sistema?

Qual a curva de aprendizado do sistema? Qual a opinião de usuários sobre o uso do sistema?

O gerenciador é capaz de criar URLs estáveis e amigáveis?

As URLs são estáveis, permanecendo a mesma com a alteração do conteúdo ou inserção de novos? O sistema possui um dispositivo para criação de URLs amigáveis?

O gerenciador responde de forma rápida? Qual a sua capacidade de requisições?

Qual a carga de requisições que o sistema agüenta? Qual a resposta do sistema em cada intervalo de número de requisições?

5. Orientações gerais sobre testes

Testar a apresentação, a estrutura e o funcionamento no sítio ou portal em diversos ambientes é uma condição necessária para garantir a melhor experiência de uso possível para o usuário.

5.1 Validação

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

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

Além das ferramentas de validação para verificação do código, existem outras que aferem a acessibilidade do conteúdo.

Para uma lista de alguns validadores disponíveis, consulte o capítulo **Recursos**.

5.2 Resoluções de vídeo

Recomenda-se que seja testado o comportamento da página, no mínimo, em três resoluções: a padrão, atual do mercado, uma mais abaixo e outra mais acima.

5.3 Navegadores e sistema

Deve-se testar o CSS e o HTML nos navegadores mais modernos para depois testar em outros mais antigos. Outra regra a ser seguida é testar a página em navegadores que sigam os padrões web, como o Firefox e Ópera, para depois, testar em navegadores com suporte deficiente aos padrões. A seguir, uma sugestão de ordem de teste em navegadores:

Firefox e Ópera em Windows

São os dois navegadores mais próximos dos padrões web. Se o desenho e estrutura da página funcionam nesses dois navegadores, poucos ajustes precisarão ser realizados nas configurações a seguir.

Internet Explorer 7.0 e 6 em Windows XP

São os navegadores mais utilizados do mercado e possuem muitas não-conformidades com os padrões web. A Microsoft recomenda que os ajustes de CSS que precisem ser realizados devem ser feitos por meio de comentários condicionais (*conditional comments*).

Firefox em Linux/Ubuntu

A possibilidade de erros nesse teste é pequena, já que o Firefox foi testado anteriormente no Windows. Os erros mais comuns costumam acontecer na camada de comportamento.

Safári em Mac

Apesar do número menor de usuários em Mac, o teste nessa plataforma deve ser realizado quando possível.

Há casos em que não é possível ter toda a gama de opções de sistemas e navegadores disponíveis para teste. No entanto, é possível contar com ferramentas que simulam sistemas e navegadores, disponíveis na internet. Dentre estas ferramentas de apoio estão os desktops virtuais, programas que emulam por imitação um outro sistema operacional além de sítios que prestam serviço de impressão de telas (*screenshots*) dos mais diversos navegadores e sistemas.

Para uma lista com algumas ferramentas, veja o capítulo **Recursos**.

5.4 Outros dispositivos

Dispositivos móveis

A tendência é um aumento da oferta do número de serviços governamentais via celular. Depois da TV, o celular é o dispositivo eletrônico de maior uso. A adoção de um desenho para serviços eletrônicos prevendo a sua utilização deve ser levada em conta.

Uma forma de ofertar o serviço de forma diferenciada para esse novo meio é o uso de folhas de estilo (CSS).

Para uma consulta sobre folhas de estilo próprias para cada meio, consulte o **Capítulo 3 Padrões Web – 3.4 CSS**. Para ferramentas de teste, veja o capítulo **Recursos**.

Visualização de impressão

Esse é um teste relativamente simples, mas que não deve ser negligenciado. Deve-se verificar se o conteúdo central do documento foi impresso satisfatoriamente, se não há desperdício de tinta ou papel para o usuário.

5.5 Teste em camadas

O teste em camadas é utilizado para verificar a acessibilidade e usabilidade do conteúdo em algumas configurações que podem ser definidas pelo usuário.

Páginas com imagens desabilitadas: alguns usuários mantêm o CSS, mas desabilitam as imagens para um carregamento mais rápido. Devem-se desabilitar as imagens do navegador e verificar se o entendimento e navegação do conteúdo estão intactos.

Navegador sem suporte ao CSS

Alguns navegadores mais antigos não possuem suporte ao CSS. Para testar essa situação, desabilite o CSS do navegador e verifique se o entendimento e navegação do conteúdo estão intactos.

Página sem suporte a *Javascript*

Por questões de segurança, muitos usuários mantêm o *javascript* desabilitado. Navegadores por texto e por voz muitas vezes não compreendem *javascript*. Deve-se desabilitar o javascript e verificar se o entendimento e navegação do conteúdo estão intactos.

5.6 Teste de acessibilidade

Assim como as ferramentas de validação de código, as ferramentas de validação de acessibilidade não garantem a qualidade do que é descrito. Por exemplo, a ferramenta atesta a existência de texto alternativo – o atributo “alt” – nas imagens, mas não garante se o que está escrito no texto alternativo condiz com a imagem ou é sua melhor descrição. No entanto, a validação da acessibilidade é útil na verificação de não-conformidades e para o diagnóstico de erros. Para uma listagem desses validadores, veja o capítulo **Recursos**.

Programas leitores de tela

Os usuários com necessidades especiais utilizam-se de programas capazes de ler e interpretar diretamente a tela do computador. De acordo com o modelo de acessibilidade em Governo Eletrônico - e-MAG, deve-se utilizar esses programas para testes finais de acessibilidade. Sempre que possível, recomenda-se a utilização de usuários com necessidades especiais para efetuar testes nas páginas do sítio.

Navegadores de texto

Esse tipo de navegador exibe todo o conteúdo das páginas na Internet em formato texto, sem imagem, sons ou animações, e pode ser usado tanto por usuários portadores de deficiência visual como por usuários que acessam a Internet com conexão discada. Pode ainda, ser utilizado em conjunto com o leitor de tela.

Análise de contraste

É importante verificar se o jogo de cores da página proporciona contraste suficiente para leitura do conteúdo. Além do conhecimento das escalas de contraste e do bom-senso, alguns testes podem ser realizados. Para uma listagem de algumas dessas ferramentas, veja o capítulo **Recursos**.

Visualização em letras grandes

A exemplo do contraste, deve-se prestar atenção ao funcionamento da página no caso de alteração do tamanho das letras. Muitos usuários usam uma configuração de texto maior para facilitar a leitura. Para verificar problemas nesses casos, pode-se utilizar o menu de exibição do tamanho de texto no navegador que está sendo utilizado.

6. Problemas mais comuns

Destacamos abaixo alguns dos problemas mais comuns encontrados durante o desenvolvimento. O conhecimento das origens desses problemas auxilia na prevenção e diagnóstico.

Diferenças do “Modelo de caixa” entre navegadores

O modelo de caixa (box model) é como se comportam as propriedades de espaço (padding), borda (border), largura (width) e altura (height) dos elementos. O modelo de caixa se aplica a qualquer seletor que utilize essas propriedades, independente delas terem sido declaradas ou não.

Alguns dos problemas mais freqüentes na apresentação dos documentos referem-se a diferenças do cálculo da largura e altura dos elementos. No modelo de caixa instituído pela W3C, as propriedades espaço e borda são somados à largura e à altura, sendo esse modelo entendido dessa forma por diversos navegadores. No entanto, em navegadores como o Internet Explorer 5.* / Win, as medidas de espaço e borda não são incluídas na altura e/ou na largura.

Ex: No CSS que segue:

```
div.caixa {  
  width:300px;  
  padding:20px;  
  border:10px solid;  
}
```

De acordo com a W3C a largura total do elemento é 360px:

$$10\text{px} + 20\text{px} + 300\text{px} + 20\text{px} + 10\text{px} = 360\text{px}$$

No entanto, o Internet Explorer 5.* / Win, interpreta a medida total como 300px, e a largura final do conteúdo como 240px.

$$300\text{px} - 10\text{px} - 20\text{px} - 20\text{px} - 10\text{px} = 240\text{px}$$

Há diversas formas de se contornar o problema como, por exemplo, utilizando comentários condicionais, hacks ou evitando declarar a largura, espaço ou borda em um elemento.

Cuidados com a acessibilidade ao utilizar imagens no lugar de texto

Substituir textos escritos em títulos e cabeçalhos por imagens utilizando-se para tanto das propriedades *text-indent* e *background-image* do CSS, visando uma melhoria estética é um recurso bastante utilizado no desenho de páginas.

No entanto, esse recurso costuma apresentar problemas nos casos em que no navegador, as imagens estão desabilitadas, mas a folha de estilo está ativa. A forma encontrada até o momento é a adição da tag SPAN vazia e o uso do atributo "title".

Código HTML

```
<h1 id="cabecalho" title="Governo Federal">  
<span></span>Governo Federal</h1>
```

CSS

```
#cabecalho {  
    width: 329px;  
    height: 25px;  
    position: relative;  
}  
#cabecalho span {  
    background: url(tituloemimagem.gif) no-repeat;  
    position: absolute;  
    width: 100%;  
    height: 100%;  
}
```

Para evitar que o mesmo problema ocorra nos leitores de tela, deve-se utilizar a *media screen* na chamada da folha de estilos.

Lapso na renderização do IE

Por vezes, quando se utiliza a regra @import para CSS externo, o Internet Explorer por um lapso de tempo 'pisca' o conteúdo sem a renderização do CSS.

Esse problema é facilmente contornado utilizando o elemento LINK ou SCRIPT dentro do cabeçalho <HEAD></HEAD>. Outras duas formas de solucionar o lapso da renderização são:

1 - Adicionando o elemento LINK

A solução a seguir consiste em adicionar outra folha de estilo por meio do elemento LINK.

Pode ser uma folha de estilo para outra mídia como impressão ou simplesmente uma folha de estilos em branco:

```
<head>  
<title>My Page</title>  
<link rel="stylesheet" type="text/css" media="print" href="print.css">  
<style type="text/css" media="screen">@import "style.css";</style>  
</head>
```

2 - Adicionando o elemento SCRIPT

Outra solução existente é inserir o elemento SCRIPT. Essa solução é recomendada apenas se a página já possui um script funcionando:

```
<head>  
<title>My Page</title>  
<script type="text/javascript"> </script>  
<style type="text/css" media="screen">@import "style.css";</style>  
</head>
```


7. Lista de verificação

A lista de verificação é um recurso para avaliar se as recomendações estão sendo seguidas no desenvolvimento, evolução ou manutenção do sítio ou portal:

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

8. Lista de formatos de arquivos e plugins

O objetivo desse capítulo é de apresentar uma lista resumida dos formatos de arquivos e plugins mais utilizados. Não é nosso objetivo incluir todos os itens existentes.

ARQUIVOS DE TEXTO

Portable Document File (.pdf)

Arquivos PDF são criados utilizando um aplicativo proprietário da Adobe. Para leitura é utilizado o programa Adobe Acrobat Reader de uso gratuito. Documentos salvos em PDF costumam guardar quase a integridade de sua formatação original. O PDF possui diversos níveis de proteção contra cópias e alterações indevidas, sendo o formato mais seguro para documentos oficiais. No entanto, o formato não é considerado acessível, o que gera a necessidade de dispor de versão alternativa.

Texto (.txt)

É o formato mais simples de texto. Documentos salvos em formato texto serão legíveis, porém não terão qualquer formatação além da quebra de linhas e parágrafos.

Microsoft Word (.doc)

Formato nativo do Microsoft Word, processador de texto proprietário da Microsoft. Existem alguns leitores gratuitos da Microsoft, mas não há garantias de compatibilidade de versões e integridade da formatação. É o formato menos indicado para disponibilizar um arquivo texto para web.

Rich Text Format (.rtf)

É um formato de arquivo que permite a troca de textos entre diferentes processadores de texto, em diferentes sistemas. Esse formato é preferível aos formatos .DOC.

Open Document Format (ODF)

O Open Document Format (ODF) possui formato aberto e é utilizado por vários processadores como BOffice, Open Office e sistema web como o *Google Docs and Spreadsheets*.

O padrão OpenDocument é de acesso público. Isso significa que pode ser implementado em qualquer sistema, seja ele de código aberto ou proprietário. As extensões de arquivos usadas pelos documentos do OpenDocument são:

- **.odt** para Processadores De Texto (**t**ext)
- **.ods** para Planilhas Eletrônicas (**s**preadsheets)
- **.odp** para Apresentações em Slides (**p**resentations)
- **.odg** para Editor de imagens (**g**raphics)
- **.odf** para Equações Matemáticas (**f**ormulae)

TABULAÇÃO DE DADOS

Sempre que possível, prefira disponibilizar dados em formato HTML ou PDF.

Comma-Separated Values (.csv)

É o formato simples de salvar informação tabular e pode ser importada em qualquer aplicação de tabulação de dados como Microsoft Excel ou aplicações de banco de dados. Um CVS é um formato simples de tabulação onde vírgulas (*comma* em inglês) separam cada coluna de informação e as linhas são separadas por quebras.

Open Document Spreadsheet (.ods)

Formato aberto de planilha do Open Document Format (ODF), presente em suítes como BOffice, Open Office e sistema web como o *Google Docs and Spreadsheets*.

Microsof Excel (.xls)

Formato proprietário do Microsoft Excel.

FORMATOS DE APRESENTAÇÃO

Sempre que possível, prefira disponibilizar apresentações em formato HTML ou PDF.

Open Document Presentation (.odp)

Formato aberto de apresentação, presente em suítes como BOffice, Open Office e sistema web como o *Google Docs and Spreadsheets*. Ainda pouco utilizado.

Microsoft Power Point (.ppt .pps)

Formatos proprietários de apresentações da Microsoft.

FORMATOS DE IMAGEM

Graphics Interchange Format (.gif)

O formato GIF utiliza uma paleta de 256 cores, permitindo o uso de fundos transparentes e de animação. O formato é mais indicado para ilustrações de cores chapadas e a traço, como logotipos e ilustrações, com um número limitado de cores.

Portable Network Graphics (.png)

É um formato livre, utilizado em alternativa ao GIF, recomendado pela W3C. O formato PNG não tem limitação de 256 cores e permite retirar o fundo de imagens com o uso do canal alfa, definindo o nível de opacidade, porém não comporta animação. Sua indicação é a mesma do formato GIF, exceto no caso de animações, com mais flexibilidade no uso de cores e transparências.

Joint Photographic Experts Group (.jpg ou .jpeg)

O formato JPEG is é o mais indicado em caso de fotografias, podendo salvar a informação em mais de 16 milhões de cores. Por outro lado, o JPEG não é indicado para imagens com cores chapadas e a traço devido a criação de diversos “artefatos” em torno das linhas e áreas chapadas.

FORMATOS DE COMPRESSÃO

ZIP (.zip)

O ZIP é um dos formatos de compressão mais popular. Um arquivo ZIP contém um ou mais arquivos que podem estar organizados em pastas e subpastas. Ao descompactar o arquivo ZIP, o usuário tem acesso aos arquivos na sua formatação e organização original. Existem diversos programas que criam e extraem conteúdos de arquivos ZIP. Alguns são gratuitos outros pagos.

Roshal ARchive (.rar)

O RAR é um formato proprietário de compactação de arquivos. Tem como principais características a alta taxa de compressão e o suporte a arquivos grandes (mais de 4 Gb).

MULTIMÍDIA

Independentemente da escolha do formato multimídia, devem ser fornecidas versões acessíveis para o conteúdo: trilhas de áudio e legenda para vídeos, versões alternativas em HTML, entre outras.

Flash/ Shockwave (.swf)

Formato de aplicações multimídia proprietário da Adobe. O formato permite a criação de páginas interativas e animadas. O plugin para visualização é gratuito. No entanto, o formato ainda possui pouco suporte à acessibilidade, não devendo ser utilizado em banners ou páginas com informações essenciais – como um acesso ao um serviço.

Quicktime (.mov)

Formato proprietário da Apple Computer, utilizado para armazenar seqüências de vídeo pelos programas/plug-ins QuickTime e iTunes.

Moving Picture Experts Group (MPG/MPEG)

Formato de compressão de áudio e vídeo mantido pela International Organization for Standardization. Sua especificação mais conhecida e utilizada é o MP3.

Real Player (RM/RA)

Formato proprietário da Real Networks Inc para seu tocador Real Player.

Windows Media Player (.wmv/.wma)

Formato proprietário da Microsoft para áudio e vídeo no PC - Computador pessoal (do inglês personal computer). É baseado em uma coleção de Codecs - dispositivo que codifica/decodifica sinais e podem ser usados pelo Windows Media Player.

Active Streaming Format (.asf)

Formato de arquivo da Microsoft para executar vídeo digital via Internet (streaming).

Musical Instrument Digital Interface (.mid ou .midi)

Formato padrão de troca de informações musicais entre instrumentos musicais, sintetizadores e computadores.

Audio Video Interleave (.avi)

Formato encapsulador de áudio e vídeo proprietário da Microsoft.

9. Recursos

Validadores de código

Validador (X)HTML - <http://validator.w3.org/>

Validador CSS - <http://jigsaw.w3.org/css-validator>

Validador de Links - <http://validator.w3.org/checklink>

Validadores de acessibilidade

ASES (WAI e e-MAG) - <http://www.governoeletronico.gov.br/>

Da Silva (WAI e e-MAG) - <http://www.acessobrasil.org.br/>

Cynthia Says (WAI) - <http://www.cynthiasays.com/>

Lift (WAI) - <http://www.usablenet.com/>

TAW (WAI) - <http://www.tawdis.net/>

Hera (WAI) - <http://www.sidar.org/hera/>

Web@Examinator (WAI) - <http://www.acesso.unic.pt/examinator.php>

Leitores de tela

Dosvox/Webvox (desenvolvido no Brasil) - <http://intervox.nce.ufrj.br/>

Virtual Vision (desenvolvido no Brasil) - <http://www.micropower.com.br/>

Jaws for Windows - <http://www.freedomscientific.com/>

Emacspeak - <http://emacspeak.sourceforge.net/>

Gnopernicus - <http://www.baum.ro/gnopernicus.html>

Orça - <http://live.gnome.org/Orca>

Dispositivos móveis

Opera Mini - <http://www.operamini.com/>

Opera Mini Simulator - <http://www.operamini.com/features/simulator>

Navegador tipo Texto

Ópera (simulador) - <http://www.opera.com>

"Exibir > Estilo > Modo usuário > Emular navegador de texto".

Lynx (navegador tipo texto) - <http://lynx.browser.org/>

Lynx Viewer (simulador) - <http://www.delorie.com/web/lynxview.html>

Serviços de impressões de navegador

BrowserCam (pago / trial de 24h) - <http://www.browsercam.com/>

Browsershots (gratuito / screenshots expostas ao público) - <http://browsershots.org/>

Desktops virtuais

PearPC (Windows) - <http://pearpc.sourceforge.net/>

VMWare (Linux) - <http://www.vmware.com/>

IEs4Linux (Linux – necessita Wine) - <http://www.tatanka.com.br/ies4linux/>

Parallels (Mac) - <http://www.parallels.com/en/products/workstation/mac>

MS Virtual PC - programa para virtualização da Microsoft

<http://www.microsoft.com/windows/products/winfamily/virtualpc/default.mspx>

Visualizadores de arquivos e plugins

Adobe online accessibility resource - <http://access.adobe.com/>

Adobe Acrobat Reader - <http://www.adobe.com/br>

Flash player - <http://www.macromedia.com/software/flash/>

Shockwave player - <http://www.macromedia.com/shockwave/>

Quicktime player - <http://www.apple.com/quicktime/download/>

RealPlayer Basic player - <http://www.real.com/>

Análise de performance

YSlow - <http://developer.yahoo.com/yslow/>

Análise de performance de carregamento da página e dicas para melhorá-la.

EDITORES (X)HTML E CSS

Bluefish (Linux) - <http://bluefish.openoffice.nl/index.html>

NVU (Windows/Linux /Mac) - <http://www.nvu.com> Possui editor visual.

PSPad (Windows) - <http://www.pspad.com/en/>

Possui dicionário em português-br e integração com Topstyle (editor CSS).

TopStyle Lite (Windows) -

<http://www.newsgator.com/NGOLProduct.aspx?ProdId=TopStyle&ProdView=lite>

Extensões Firefox

Extensões (*extensions*) são pequenos programas que adicionam funcionalidades ao Firefox. Para ter acesso as extensões, basta ir a Ferramentas > Complementos > Extensões > Mais extensões.

Navegador Firefox - <http://br.mozdev.org/firefox>

Web Developer - <https://addons.mozilla.org/firefox/60>

Várias ferramentas de desenvolvimento e teste.

HTML validator - <https://addons.mozilla.org/firefox/249>

Validador HTML baseado no TIDY, permite correções no código.

ColorZilla - <https://addons.mozilla.org/firefox/271>

Exibe a informação da cor selecionada.

HTML Tidy Library Project - <http://tidy.sourceforge.net>

Converte HTML para XHTML

Firefox Accessibility Extension

<https://addons.mozilla.org/en-US/firefox/addon/1891?>

Firebug - <http://www.getfirebug.com/>

ferramentas para análise código, folhas de estilo, testes e debug de javascript.

DOM Inspector - <http://www.mozilla.org/projects/inspector/>

Ferramenta para visualizar e editar o DOM de um documento.

10. Leitura complementar

Organizações e especificações

Codificação de caracteres - <http://www.w3.org/International/O-charset.pt-br.php>

CSS Paged media - <http://www.w3.org/TR/REC-CSS2/page.html>

Especificação HTML 4.01 - <http://www.w3.org/TR/html401>

Especificação XHTML 1.0 - <http://www.w3.org/TR/xhtml1>

Especificação XHTML 1.1 - <http://www.w3.org/TR/xhtml11>

Especificação CSS 1.0 - <http://www.w3.org/TR/CSS1>

Especificação CSS 2.1 - <http://www.w3.org/TR/CSS21>

Especificação CSS 3.0 (em construção) - <http://www.w3.org/Style/CSS/current-work>

The Web Standards Project - <http://www.webstandards.org/>

Tutoriais W3C - <http://www.w3.org/2002/03/tutorials>

World Wide Web Consortium (W3C) - <http://www.w3.org>

Governo Eletrônico Brasileiro

e-MAG - <http://www.governoeletronico.gov.br/acoese-projetos/e-MAG/>

Resolução nº7 - <http://www.governoeletronico.gov.br/o-gov.br/legislacao>

Sítios em português CSS para Webdesign - <http://www.maujor.com/>

O básico da Web - <http://www.obasicodaweb.com/>

Revolução etc - <http://www.revolucao.etc.br/>

Sítios em inglês

A List Apart - <http://www.alistapart.com>

CSS Discuss - <http://www.css-discuss.org/>

CSS Play - <http://www.cssplay.co.uk/>

CSS Zen Garden - <http://www.csszengarden.com/>

W3 schools - <http://www.w3schools.com/>

Goggle Doctype - <http://code.google.com/doctype/>

Sítios em espanhol

Anieto2K - <http://www.anieto2k.com/>

Estándares y accesibilidad - <http://www.estandaresyaccesibilidad.com/>

Ser accesible - <http://www.seraccesible.net/>

Usable y accesible - <http://olgacarreras.blogspot.com/>

11. Glossário

Acessibilidade

Significa permitir o acesso por todos, independente do tipo de usuário, situação ou ferramenta. Significa ainda, criar ou tornar as ferramentas e páginas web acessíveis a um maior número de usuários, inclusive pessoas portadoras de deficiências.

CoDec

Acrônimo de Codificador/Decodificador. Dispositivo de *hardware* ou *software* que codifica/decodifica sinais. Existem dois tipos de CoDecs: Sem perdas (*lossless*) e Com perdas (*lossy*). Os CoDecs sem perdas codificam o som ou imagem sem alterar a qualidade original. Esse tipo de CoDec normalmente é utilizado em rádios e emissoras de televisão. Os CoDecs geram arquivos com perda de qualidade mas com maiores taxas de compressão. Esse tipo de CoDec é o mais utilizado para arquivos a serem utilizados na internet.

CSS (*Cascading Style Sheets* – Folhas de estilo em cascata)

Linguagem que descreve a apresentação, visual ou sonora, de um documento.

Hack

Hacks são subterfúgios utilizados pelos desenvolvedores como forma de contornar problemas apresentados na apresentação de uma página devido à deficiência de um dado navegador entender os padrões web.

Navegador

Programa que permite ao usuário consultar e interagir com o material publicado na Internet.

Padrões web (*Web standards*)

Padrões web são regras de tecnologia definidos pelo W3C e outros órgãos da área, utilizados para criar e interpretar o conteúdo web.

Parser

Um *Parser* é um programa de computador (ou apenas um componente de um programa) que analisa a estrutura de uma entrada segundo uma determinada gramática formal.

URI (*Uniform Resource Identifier - Identificador Unificado de Recurso*)

Basicamente é uma string (conjunto de caracteres) que seguem uma certa sintaxe e é usado para definir identificar um recurso na web. O tipo mais comum de URI é a URL.

URL (*Uniform Resource Locator - Localizador Unificado de Recurso*)

Um tipo específico de URI usado para definir a localização de um recurso na web. Geralmente dizemos que a URL é o endereço de uma página web. Um exemplo de URL: `http://www.exemplo.com/pagina/`

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.

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 a HTML, mas que permite a descrição de dados definidos por elementos de programação.