

Tecnologias assistivas - leitores de tela

Diogo da Silva Lima - diogo.lima@navi.ifrn.edu.br

Núcleo Avançado de Inovação Tecnológica - NAVI

Softwares mais utilizados - sistema operacional - documentação/manual de instalação

- Orca - Linux - <https://help.gnome.org/users/orca/stable/>
- NVDA - Windows - <https://www.nvaccess.org/>
- Virtual Vision - Windows - <https://www.virtualvision.com.br/>
- Voice ove - Mac - <https://www.apple.com/accessibility/mac/vision/>

Por que deixar sites acessíveis?

- Ideia do padrão W3C de trazer a informação para todas as pessoas, sendo assim o desenvolvimento como um todo terá retorno às tecnologias da informação e à sociedade dessas pessoas com dificuldade de acesso à informação;
- Democratização da informação;

No Brasil, como é gerenciado a acessibilidade de sítios?

- Baseado no WCAG - Web Content Accessibility Guidelines;
- Padrão eMAG e cartilha de codificação;
 - No meu repositório do Github tem a cartilha de indicações do governo (que é traduzida do padrão WCAG da W3C);
 - “o eMAG é o documento que norteia o desenvolvimento de sítios e portais acessíveis no âmbito do governo federal.”

Quem é obrigado a seguir o padrão eMAG?

- Os órgãos do SISP - Sistema de Administração dos Recursos de Tecnologia da Informação;
- Qualquer empresa/sociedade pode aderir ao padrão;

Então, do que se trata exatamente a acessibilidade?

- Tentar adaptar, isto é, melhorar a interpretação do conteúdo para alvos como:
 - pessoas com baixa habilidade motora;
 - Deficientes visuais e auditivos;
 - Cegos e surdos, etc;
- Essas adaptações vêm do padrão web internacional W3C;

Após tentar deixar meu sítio acessível, como posso o avaliar?

- O desenvolvedor pode normalmente testar com a pessoa com baixa habilidade de acesso a qual se deseja adaptar o sítio (por exemplo, um deficiente visual que usa sites com as tecnologias assistivas) e faça o teste, está bom para esta pessoa?
- Após a avaliação virtual pode ser feita pela plataforma do governo: ASES - Avaliador E Simulador De Acessibilidade Em Sítios;
 - Link para o sítio: <http://asesweb.governoeletronico.gov.br/ases/>

DESENVOLVIMENTO

Dicas básicas de boas práticas de programação para web que é indicado para sítios acessíveis:

- Utilize SEMPRE tags semânticas, isso significa que você não deve cair na tentadora técnica de usar as famigeradas “divs” vazias de significado. A seguir uma lista de tags semânticas (as normalmente mais utilizadas, entretanto deixarei também links para uma maior exploração):
 - **header**: identificar um cabeçado (parte de cima do site);
 - **navbar**: menu de itens linkados com outras partes do site;
 - **main**: identificação de que é a parte principal do site, ou seja, o conteúdo de fato relevante e informativo;
 - **section**: normalmente o main deve ser dividido em várias dessas tags, indicando as repartições dos conteúdos;
 - **aside**: leitura complementar, algo a mais dentro do conteúdo principal;
 - **article**: algo que pode ser interpretado de maneira isolada dentro do site (sem a necessidade, obrigatória, das outras partes do site);
 - **details**: informações adicionais que o usuário pode mostrar ou esconder;


```
<details>
                <summary>O que fica visível</summary>
                <!-- depois que clicado em um button:-->
                <p>o que aparece depois</p>
              </details>
```

 - **figcaption**: representa a descrição de uma legenda ou imagem;
 - **figure**: elemento independente: imagem ou ilustração etc;
 - **footer**: conteúdo de rodapé;
 - **address**: indicando endereço;
 - **time**: indicando data e hora;
 - Utilize sempre da hierarquia de tags, por exemplo NUNCA utilize a tag h2 sem antes em um contexto desencadeador de uma h1;

- Separe bem o que é comportamento, conteúdo e estilo bem - arquivos diferentes;
- dicas de conteúdo que deve ficar oculto para os videntes; porém legível para leitores de tela:

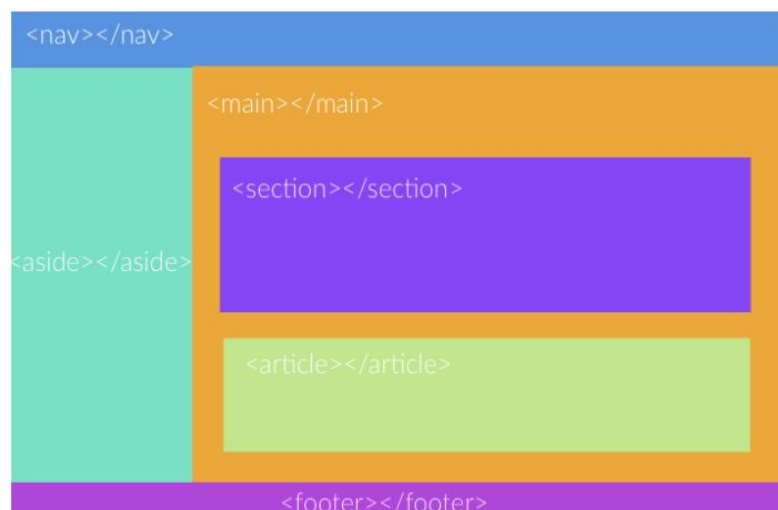
CSS	Efeito na tela	Efeito na acessibilidade
<code>visibility: hidden;</code>	O elemento fica oculto, mas continua a ocupar o espaço que normalmente ocuparia	O conteúdo é ignorado pelos leitores de tela
<code>display: none;</code>	O elemento fica oculto e não ocupa espaço	O conteúdo é ignorado pelos leitores de tela
<code>height: 0; width: 0; overflow: hidden;</code>	O elemento fica oculto e não ocupa espaço	O conteúdo é ignorado pelos leitores de tela
<code>text-indent: -999em;</code>	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
<code>position: absolute; left: -999em;</code>	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

ATENÇÃO!!!!!!

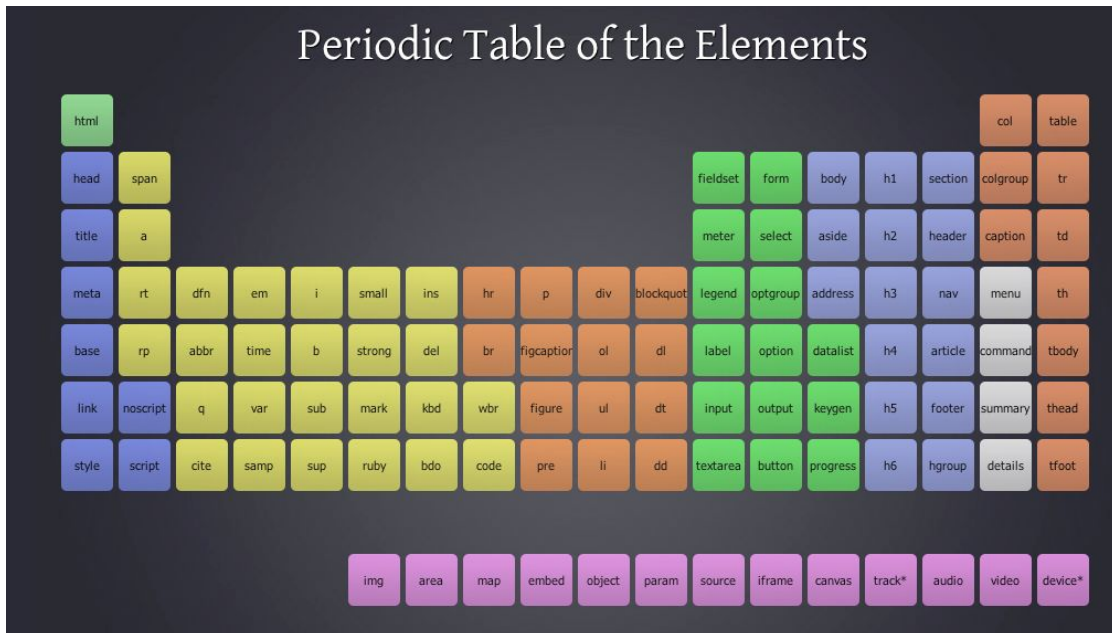
Vale ressaltar que nesse caso, utiliza-se (de maneira a facilitar) a classe "sr-only" e "sr-only focusable" do Bootstrap para ocultar dos videntes no site, sem ocupar espaço e ser legível para os leitores de tela

Link para o sítio do framework e do elemento: <https://getbootstrap.com/>
<https://getbootstrap.com/docs/4.3/utilities/screen-readers/>

- Dica de como deixar semanticamente desencadeado o sítio por partes:
 - header, onde irá hospedar o navbar e logo, se houver; main, conteúdo de fato principal; footer, demarcar informações para contato, endereço, etc;
 - Veja:



- Utilize bem a hierarquia de tags:



- Estruturação semântica exemplo:

```
<header role="banner">
  <!-- Cabeçalho -->
</header>
<main role="main">
  <!-- Conteúdo principal -->
  <article role="article">
    <!-- Artigo do site -->
  </article>
  <aside role="complementary">
    <!-- Conteúdo auxiliar -->
  </aside>
</main>
<footer role="contentinfo">
  <!-- Rodapé -->
</footer>
```

- Perceba a utilização persistente do atributo “*role*” que identifica o que cada parte é: complementary, já representado pela tag de significado como

conteúdo a mais para o site, reforçado pelo atributo *role* como sendo algo complementar;

- Quando utilizar FORMULÁRIOS sempre os cerque com fieldset para que o leitor entenda que ali é um grupo;
- Quando utilizar imagens SEMPRE utilize o atributo “alt”, veja:



○

■ ``

- Tudo que está dentro do atributo alt, será lido pelos screen readers: tente ser fiel ao conteúdo descritivo da imagem SEM SER EXTENSO para não atrapalhar o raciocínio da pessoa com deficiência visual;
- Apesar de tags como está não possuir fechamento, utilize, no final, “/>”;



○

■ ``

- **Não utilizar elementos em desuso (FONT, CENTER) - tentar manter as divisões de estilização, comportamento e conteúdo em diferentes camadas, isto é, documentos diferentes;**
- Em validações:
 - Deve ser posto a descrição de áudio - devagar e com poucos caracteres - ou, no local da imagem para a verificação, colocar perguntas básicas como “em qual mês se comemora o natal?”; “qual o primeiro mês do ano?”;
- Quanto ao desempenho do site e URL:
 - Tente deixar uma URL amigável e não muito extensa, porque o leitor ler todas as letras e caracteres da URL, exatamente todos mesmo, até os pontos: veja o quanto um deficiente visual pode sofrer com suas passagens de parâmetro via GET? repense;
 - reduza a quantidade de requisições http através de imagens e vídeos extremamente pesados pois reduz o desempenho do sítio;
- Evitar o uso dos frames:

- Utilizado primitivamente a divisão da página em frames, porém hoje está em desuso (isso entra, primeiro, como “não utilizar elementos em desuso” deste documento);
- Serviços de busca tem dificuldade de localizar o conteúdo, o que muitos usuários têm incertezas ao que desejam imprimir;
- Dificulta pessoa que não utilizam navegadores gráficos: têm dificuldades de navegar;
- Evitar o uso de pup-up e partes PISCANTES:
 - **Pessoas que utilizam leitores de tela, tem dificuldades em lê-los, normalmente se perdem;**
 - Pessoas com epilepsia podem ter crises quando em contato com cores vibrantes demais piscando (ou simplesmente piscando);
 - Lembre-se que seu site deve ser agradável para que a democratização da informação aconteça;
- Utilizar, no HTML, tags reduzidas e com encadeamento semântico, por exemplo:
 - main> maincontent> maincontenttext;
 - no CSS, div#main;
 - UTILIZAR O CSS ORIENTADO A OBJETOS;
- **Identificação real do idioma:**
 - Se estiver utilizando português e definido na tag html e meta que o idioma é *pt-br*, top; caso mude para inglês, por exemplo, abra um atributo “*lang=en*” dentro da tag, de preferência utilize span para que o leitor de tela identifique corretamente o idioma que irá falar;
- **É IMPORTANTE UTILIZAR ÂNCORAS COM ATALHOS DISPONIBILIZADOS NO TECLADO:**
 - Construa uma barra de acessibilidade que leve a partes relevantes do sítio com o atalho no teclado
 - Ir para seção tal
 - O leitor lerá: “link ir para seção tal, alt +1”

Para a real experiência com exemplo real de sites acessíveis, veja meu repositório de indicações de acessibilidade, no github:

link: <<https://github.com/diogolimas/Indica-esDeAcessibilidade>>

meu usuário no Github: diogolimas

DOCUMENTAÇÃO PRO MODELO W3C

<https://www.w3.org/TR/wai-aria-practices/#radiobutton>

link do css do w3c, modelo, padrão, web:

<https://www.w3.org/StyleSheets/TR/2016/base.css>

Padrão w3C: trabalhando com grupos de nota

Seguir a cartilha de codificação de acessibilidade do governo:

<http://epwg.governoeletronico.gov.br/cartilha-codificacao>

<https://www.governodigital.gov.br/documentos-e-arquivos/e-pwg-Codificacao.pdf>

Material de apoio quanto a dicas e validação de acessibilidade do governo:

<https://www.governodigital.gov.br/transformacao/cidadania/acessibilidade/material-d-e-apoio>

<https://www.governodigital.gov.br/transformacao/compras/orientacoes/identidade-digital-do-governo/epwg-padroes-web-em-governo-eletronico/cartilha-de-codificacao>

Fonte:

<http://emag.governoeletronico.gov.br/>

<http://www.w3c.br/Padroes/>