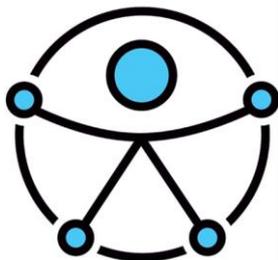


<http://bit.ly/dicasacessibilidade>



Dicas de Acessibilidade na Web #PraCegoVer



Monica Craveiro de Menezes

Quem sou?



Monica Craveiro de Menezes

Carioca, Bacharel em Sistemas de Informação e Matemática, Engenheira de Software na CI&T, ex-aluna do [des]programe, {reprograma} e DigitalHouse> e Professora de Programação, além de Participante e Palestrante em diversas comunidades de TI.



@mocraveirodev

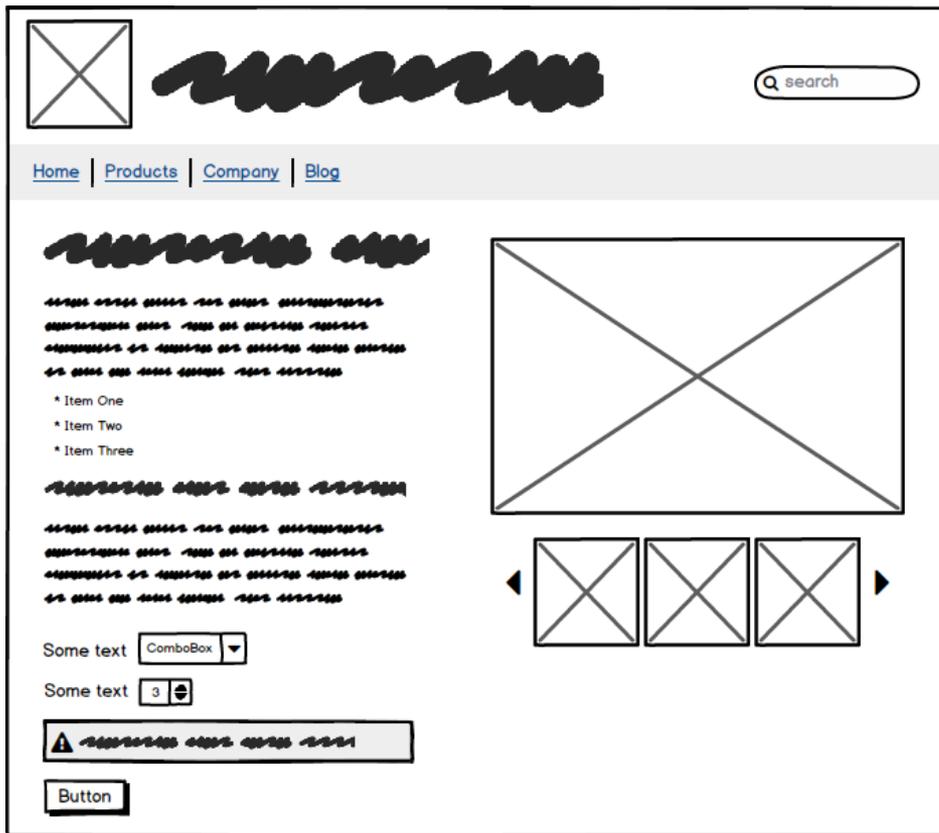
e só por último vai ler isso aqui

**VOCÊ VAI LER
ISSO AQUI PRIMEIRO**

depois vai ler isso aqui
e depois isso aqui

Quão importante é o layout, o tamanho das fontes e suas cores para que a gente consiga ter um entendimento completo de algo?

O que define um website para você?

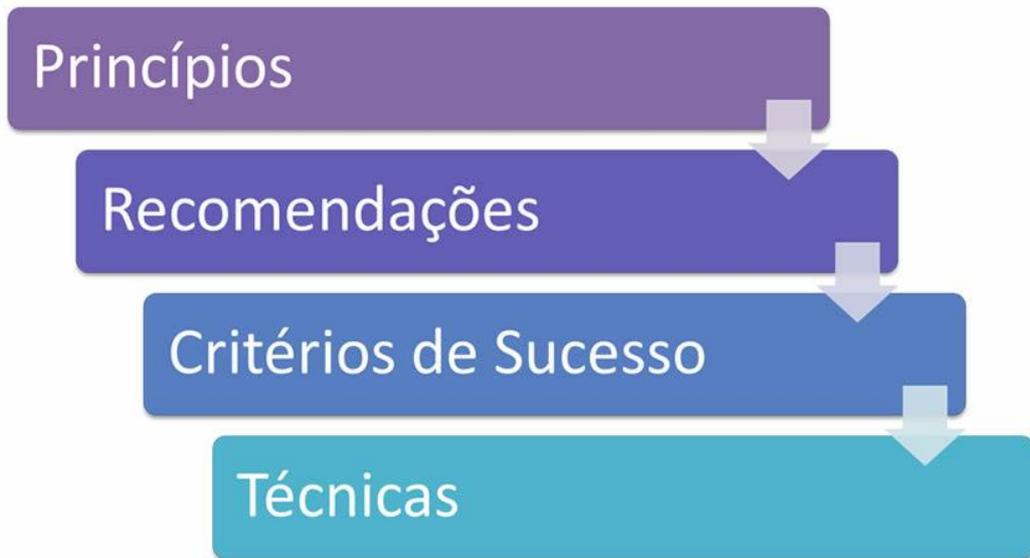


Claramente, não é possível saber o que está escrito, uma vez que as palavras estão borradas. No entanto, há dicas visuais que nos auxiliam a entender quais informações estão disponíveis e onde elas estão.

As dicas visuais são de extrema importância para o usuário. Pensando nisso, como podemos melhorar a experiência de usuários com deficiência visual?

**Criando um site
acessível!**
**Mas como podemos
validar se o site é
acessível ou não?**

Uma das formas de validar se o usuário está tendo uma boa experiência é através do **WCAG**, uma documentação criada pelo WAI (Web Accessibility Initiative) para conteúdos na web. Este documento é estruturado da seguinte forma:



Princípios

Perceptível



A informação e os componentes da interface do usuário têm de ser apresentados aos usuários em formas que eles possam perceber.

Operável



Os componentes de interface de usuário e a navegação têm de ser operáveis.

Recomendações



- 1.1 Fornecer alternativas textuais para qualquer conteúdo não textual
- 1.2 Fornecer alternativas para multimídia
- 1.3 Criar conteúdo que possa ser apresentado de modos diferentes sem perder informação ou estrutura
- 1.4 Tornar mais fácil aos usuários a visualização e audição de conteúdos, incluindo as separações das camadas da frente e de fundo



- 2.1 Fazer com que todas as funcionalidades estejam disponíveis no teclado
- 2.2 Prover tempo suficiente para os usuários lerem e usarem o conteúdo
- 2.3 Não projetar conteúdo de uma forma conhecida por causar ataques epiléticos
- 2.4 Prover formas de ajudar os usuários a navegar, localizar conteúdos e determinar onde se encontram

Princípios

Compreensível



A informação e a operação da interface de usuário têm de ser compreensíveis.

Robusto



O conteúdo tem de ser robusto o suficiente para poder ser interpretado de forma concisa por diversos agentes do usuário, incluindo recursos de tecnologia assistiva.

Recomendações



- 3.1 Tornar o conteúdo de texto legível e compreensível
- 3.2 Fazer com que as páginas da Web apareçam e funcionem de modo previsível
- 3.3 Ajudar os usuários a evitar e corrigir erros



- 4.1 Maximizar a compatibilidade entre os atuais e futuros agentes do usuário, incluindo os recursos de tecnologia assistiva

Para cada recomendação existem **critérios de sucesso**, que são pontos específicos que devem ser atingidos e, para cada critério de sucesso, estão disponíveis **técnicas específicas**, com exemplos de como o objetivo do critério pode ser atingido e testado.

Cada critério de sucesso é indicado por um nível de conformidade, que pode ser A, AA ou AAA:

Nível A: barreiras mais significativas de acessibilidade. Não garante um site altamente acessível;

Nível AA: estar em conformidade com todos os critérios de sucesso de nível AA garante um site bastante acessível.

Nível AAA: o nível de conformidade triplo A é bastante meticuloso, ou seja, visa garantir um nível otimizado de acessibilidade. A maioria dos critérios de sucesso de nível AAA referem-se a situações bastante específicas, normalmente objetivando refinar os critérios de sucesso de nível AA. Muitos sites não possuem conteúdo que se aplica aos critérios de sucesso de nível AAA.

A chave é uma semântica HTML bem escrita!

É muito importante tomar cuidado e entender qual deve ser a experiência do usuário quando for implementar ou atualizar um componente. Dessa forma, você consegue sempre utilizar os elementos corretos, permitindo que os leitores de tela possam fazer uma leitura correta e permitindo ao usuário uma experiência mais completa.

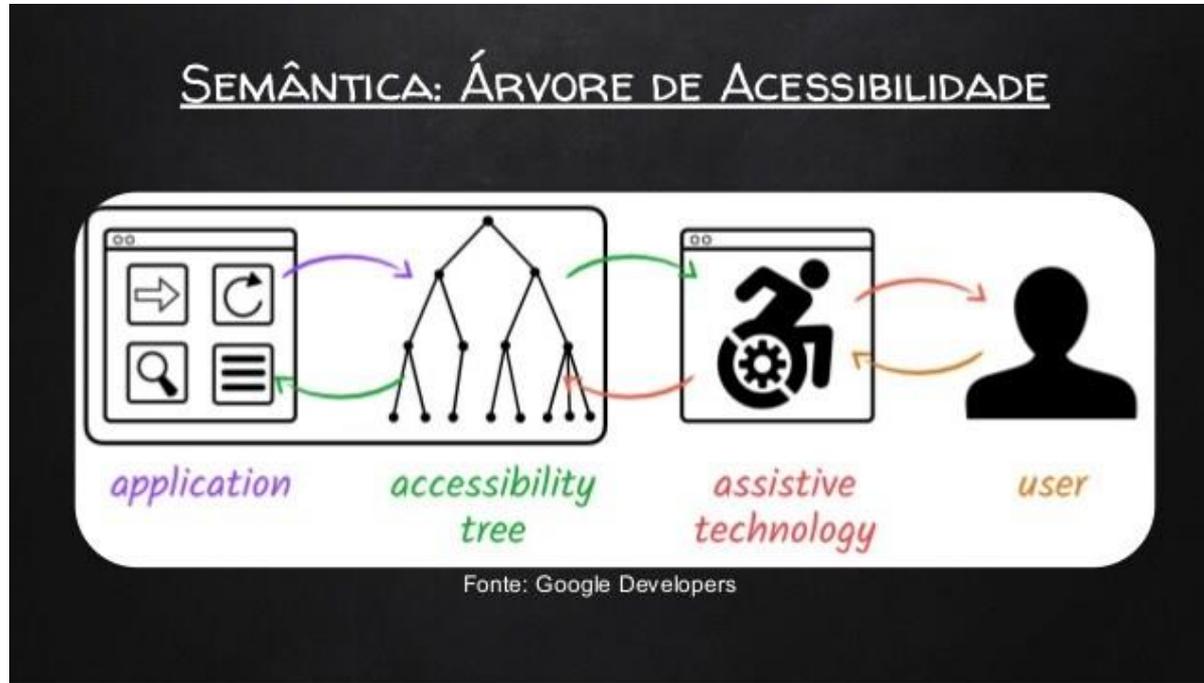
```
<!-- Uso incorreto das tags HTML -->
<div class="button" tabindex="0" role="button">
  <span>Clique aqui!</span>
</div>

<a class="button" role="button">Clique aqui!</a>

<span class="button" tabindex="0" role="button">Clique aqui!</span>

<!-- Uso correto das tags HTML -->
<button>Clique aqui!</button>
```

A árvore de acessibilidade herda os componentes da árvore HTML e, por isso, é muito importante utilizar corretamente as tags, roles e atributos aria.



A primeira dica é: os elementos **** e **<div>** são elementos que não possuem sentido semântico. Portanto, os leitores de tela não leem esses elementos. Para que sejam lidos, deve-se adicionar `tabindex="0"`.

Mas espere! Não é pra sair adicionando `tabindex="0"` em todo lugar!

Quando for criar um novo componente, atente-se ao uso correto das tags semânticas, são elas:

<a>

<button>

<nav>

<embed>

<label>

<select>

<main>

<h1>, <h2>, ... , <h6>

<textarea>

<input>

<p>

<video>

<audio>

<figure>

<section>

<header>

<footer>

<time>

<article>

<aside>

**A seguir temos alguns
exemplos de estruturas,
atributos mais
importantes e boas
práticas.**

Linguagem da página

- Definir corretamente o idioma em que estará o site usando o atributo **lang**.

Título da página

- O título do site deve descrever claramente seu propósito ou sobre qual tópico se refere.

Formas de navegação

- Sempre ofereça diversas formas de navegar pela página;
- Todas as funcionalidades da página devem ser operáveis usando um teclado;
- Em *TouchScreens*, sempre ofereça uma alternativa de toque único para ações complexas com os dedos, tipo movimento de pinça.

Pontos de Referência

- Use apenas para demarcar blocos importantes de conteúdo para facilitar a navegação até eles;
- Não esquecer de utilizar o atributo **role** ou **aria** caso não haja um elemento semântico correspondente.

Contraste

- Atente-se para não prejudicar a percepção dos componentes.

Foco

- Deve ser óbvio em qual elemento está o foco durante a navegação;
- Cuidado ao manipular o foco via código para não criar “armadilhas” de navegação.

Títulos

- Os títulos definem a estrutura do site;
- Evite pular os níveis de título, siga a hierarquia do `<h1>` até `<h6>`;
- Os títulos devem estar aninhados de acordo com sua hierarquia.
- Devem ser claros e descritivos.

Listas

- Devem estar dentro dos elementos `` ou ``;
- Cada item da lista deve estar dentro do elemento ``.

Links

- Deve-se usar os elementos **<a>** ou **<button>**;
- Assegure-se de que são visualmente distintos, descritivos, concisos e únicos;
- Caso necessário, use **aria-label** ou **aria-labelledby** para distinguí-los.

Tabelas

- Atentar-se à formatação das tabelas para não dificultar a leitura e interação com os elementos;
- Colocar um título para a tabela utilizando o elemento **<caption>** e **<table>** para identificar a região da tabela;
- Utilizar **<th scope="col">** para identificar o título da coluna;
- Utilizar **<th scope="row">** para identificar o título da linha;
- Utilizar **<td>** para identificar as demais células da tabela;
- Caso necessário, utilize **<thead>**, **<tbody>** e **<tfoot>** para identificar as regiões da tabela.

Botões

- Evite o uso de elementos: **<div>**, **<a>** e **** para criar botões;
- Se for realmente necessário criar utilizando uma tag, utilize a **role="button"** no elemento.
- Caso haja *label* associada, garanta que seja igual ou similar ao texto do botão.

Carousel

- Botões de play/pause devem estar evidentes e com contraste de cores;
- A navegação entre os slides deve ser acessível pelo teclado;
- Ter um botão de *skip* para pular o conteúdo do carousel.

Menus

- Se houver mais de um menu na mesma página cada um deve ter um **aria-label** específico;
- Se o menu utilizar um hamburger ou outro ícone que o sinalize, adicione uma **role="button"** no elemento e também um **aria-label** compatível, além do **aria-expanded** que dará ao usuário a informação se o menu está expandido ou colapsado.

Formulários

- Jamais utilizar avanço automático entre os campos de preenchimento do usuário ou reorganizar a ordem dos elementos através do CSS;
- Dar suporte para autocompletar informações repetitivas e para erros;
- Cada campo deve conter cores e tipografia com contraste de cores além de labels específicas e compatíveis;
- Colocar **label** visível, **aria-label/aria-labelledby** e **role** correta para todos os campos. Caso necessário, forneça instruções ou exemplos para os campos.

TabIndex

O **tabindex** é um atributo global que permite que os elementos da DOM sejam lidos ou não pelos leitores de tela. Há três valores possíveis:

- **tabindex="-1"**: retira o elemento interativo da árvore do DOM, mas continua acessível através por JS ou por clique do mouse.
- **tabindex="0"**: inclui um elemento que não tem foco nativo na árvore do DOM.
- **tabindex="1" (ou qualquer valor positivo)**: permite reordenar a ordem de foco dos elementos na árvore do DOM.

Observações importantes:

1. Evite utilizar o tabindex com valores positivos, mudar a ordem que uma página é lida atrapalha a experiência do usuário;
2. Evite utilizar tabindex em elementos não interativos para torná-los interativos.
3. Elementos que possuem foco nativo não precisam de tabindex.

Imagens

Texto Alternativo

É a descrição textual do conteúdo da imagem. Deve ser incluída no atributo **alt**.

Deve ser curta e finalizada com ponto final (.).

Evite usar **“Foto de ...”** ou **“Gráfico para ...”** para não ficar repetitivo.

Para imagens decorativas, ou seja, que não tem conteúdo, função ou que tenham algum texto relacionado, não é necessário que tenham um texto alternativo. utilize o atributo **alt=""**.

É muito importante que todas imagens possuam texto alternativo.

Aria-label

O aria-label é um atributo que define uma string para rotular um determinado componente, ele deve ser usado quando o rótulo deste componente não for visível.

```
<nav class="menu" aria-label="Navegação com itens">
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
  </ul>
</nav>
```

Aria-labelledby

Quando o rótulo do componente estiver visível para o usuário, basta utilizar o `aria-labelledby`, este atributo recebe o `id` do elemento que rotula.

```
<nav class="menu" aria-labelledby="id-titulo-menu">
  <h2 class="titulo-menu" id="id-titulo-menu"> Menu com Itens</h2>
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
  </ul>
</nav>
```

Roles

Para listas:

- **role="list"** para o elemento que define a lista [**** ou ****];
- **role="listitem"** para cada elemento da lista [****].

Para abas:

- **role="tablist"** para o elemento que define a lista de abas;
- **role="tab"** para cada aba.

Para links:

- **role="link"** no elemento.

Para botões:

- **role="button"** no elemento.

Boas práticas - HTML

- Verificar se todos os elementos foram abertos e fechados corretamente;
- Se os elementos estão em suas hierarquias corretas de acordo com as especificações;
- se não há *IDs* duplicados;
- Use um *linter*.

Boas práticas – CSS

- Não utilize **visibility:hidden**! Os leitores de tela leem esses elementos da mesma forma, para escondê-los na página e para os leitores, utilize o **display:none**;
- Caso o elemento esteja visível no página, mas você entende que ele não deve ser lido pelo leitor de telas, utilize `aria-hidden='true'` no html do elemento;
- Evite utilizar o **outline:none** para o estado de foco dos elementos, essa propriedade é muito importante para as pessoas que utilizam o site através da navegação do teclado, dessa forma elas conseguem saber em que posição estão no site;
- Evite alterar a ordem dos elementos;
- Evite incluir conteúdo com CSS.

Boas práticas – Javascript

- Todo conteúdo inserido por *script* deve ser acessível;
- Os *scripts* não podem impedir a navegação por leitores de tela nem por teclado;
- Os *scripts* devem funcionar independente do tipo de dispositivo utilizado.

Axe Tools

Um plugin interessante para utilizar na hora do desenvolvimento do seu site e que ajuda na validação das regras de acessibilidade é o Axe Tools, ele é uma extensão do chrome e além de fazer a varredura da página sugere quais alterações devem ser feitas, quando necessário.

[Link para adicionar a extensão](#)



axe DevTools - Web Accessibility Testing

Offered by: www.deque.com

★★★★☆ 87 | [Developer Tools](#) | 👤 100,000+ users

🟢 Runs offline

Leitores de Tela

Leitores de tela são *softwares* de tecnologia assistiva que fornecem informações através de síntese de voz sobre os elementos exibidos na tela.



[NVDA](#)



[Jaws](#)



[Screen Reader](#)



[TalkBack](#)



[VoiceOver](#)



[Dosvox](#)

Alguns links para aprofundar o assunto...

- <https://www.w3.org/TR/wai-aria-1.1/>
- <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles>
- https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Techniques
- <https://developer.mozilla.org/pt-BR/docs/Learn/Accessibility/HTML>
- <https://www.devmedia.com.br/html-semantic-conheca-os-elementos-semanticos-da-html5/38065>
- <https://www.w3.org/TR/UNDERSTANDING-WCAG20/conformance.html#uc-levels-head>
- <http://emag.governoeletronico.gov.br/cursoconteudista/desenvolvimento-web/recomendacoes-de-acessibilidade-wcag2.html>
- <https://mwpt.com.br/acessibilidade-digital/boas-praticas/>
- <https://www.pedrodias.net/ux/dicas-de-acessibilidade-web>
- <http://www.acessibilidadelegal.com/>

Estão preparados para criar sites mais acessíveis?

Obrigada!

Monica Craveiro de Menezes



@mocraveirodev