

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BA – IFBA

PROGRAMAÇÃO WEB - APOSTILA DE ACESSIBILIDADE  
PROFESSOR: ADDSON ARAÚJO DA COSTA

VALENÇA – BA

# Sumário

|                                                                     |    |
|---------------------------------------------------------------------|----|
| 1. Introdução.....                                                  | 4  |
| 2. Direito a Acessibilidade.....                                    | 4  |
| 3. Perceptível.....                                                 | 5  |
| 3.1 Alternativas para conteúdos não textuais.....                   | 5  |
| → Campos de entrada.....                                            | 5  |
| → Mídias de áudio e vídeo.....                                      | 5  |
| → Testes/avaliação.....                                             | 5  |
| → Experiência sensorial.....                                        | 5  |
| → Elementos de decoração e formatação.....                          | 6  |
| → Captcha.....                                                      | 6  |
| 3.2 Alternativas para áudio e vídeo pré gravados.....               | 7  |
| 3.3 O conteúdo deve poder ser apresentado de diferentes formas..... | 7  |
| 3.4 Facilitar a visão e audição dos conteúdos.....                  | 8  |
| 4. Operável.....                                                    | 9  |
| 4.1 Acessível via teclado.....                                      | 9  |
| 4.2 Tempo suficiente.....                                           | 9  |
| 4.3 Convulsões.....                                                 | 9  |
| 4.4 Navegável.....                                                  | 10 |
| 5. Compreensível.....                                               | 10 |
| 5.1 Legível.....                                                    | 10 |
| 5.2 Previsível.....                                                 | 10 |
| 5.3 Assistência de Entrada.....                                     | 11 |
| 6. Robusto.....                                                     | 11 |
| 6.1 Compatível.....                                                 | 11 |
| EXERCÍCIO.....                                                      | 11 |
| REFERÊNCIAS.....                                                    | 12 |

**Lista de Figuras**

Figure 1: Google reCaptcha.....6

## 1. Introdução

Antes de falarmos de HTML 5, precisamos entender a evolução desta linguagem ao longo dos anos. Logo de sua criação o objetivo era fornecer uma linguagem fácil para que os navegadores conseguissem traduzir em um aspecto gráfico usando pouca largura de banda de internet e pouco processamento.

A solução foi enviar os dados através de arquivos de texto usando a linguagem XML como base (ela que define o que é uma tag, atributo e como devemos escrever). Toda esta parte de abertura e fechamento correto de tags e atributos, chamamos de sintaxe da linguagem.

Usando a sintaxe do XML, foram definidos tags específicas com significado, por exemplo, a tag **B** significa negrito, **TABLE** significa tabela, e assim por diante. Toda essa relação que dá significado ao código, chamamos de semântica. A semântica que nos impede, por exemplo, de colocar uma tag **TR** fora de uma tag **TABLE**, pois não podemos ter uma linha da tabela fora da tabela.

A internet era muito lenta, então a medida que a internet foi ficando mais robusta, mais funcionalidades foram sendo agregadas até o ponto que hoje conseguimos ter jogos 3D rodando diretamente no navegador, o que criou a necessidade de tags com significados, ou seja, semânticas coerentes aos seus novos usos.

Além disso, a internet passou a ser parte da vida das pessoas em todo o seu dia a dia, trazendo as preocupações com o direito a acessibilidade que se tornou imprescindível na criação de páginas de internet na última década. Dito isto, esta apostila busca introduzir o HTML 5 junto as questões de acessibilidade.

## 2. Direito a Acessibilidade

Segundo a LEI Nº 13.146, de 6 de julho de 2015 instituída pela presidenta Dilma Rousseff, “É obrigatória a acessibilidade nos sítios da internet mantidos por empresas com sede ou representação comercial no País ou por órgãos de governo, para uso da pessoa com deficiência, garantindo-lhe acesso às informações disponíveis, conforme as melhores práticas e diretrizes de acessibilidade adotadas internacionalmente”.

Sobre as diretrizes de acessibilidade adotadas internacionalmente, surge destaque a Web Content Accessibility Guidelines (WCAG) que em português significa Diretrizes de Acessibilidade para o Conteúdo da Web que é definida pelo World Wide Web Consortium (W3C) que é o mesmo grupo responsável por padronizar toda a internet incluindo as linguagens HTML, CSS, entre outras.

Segundo o W3C, “As Diretrizes de Acessibilidade para Conteúdo Web (WCAG) 2.0 definem a forma de como tornar o conteúdo da Web mais acessível para pessoas com deficiência. A acessibilidade abrange uma vasta gama de deficiências, incluindo visual, auditiva, física, de fala, intelectual, de linguagem, de aprendizagem e neurológica”. A WCAG mais atual é a versão 2.1 que adicionou novos pontos a serem considerados no desenvolvimento, mas por tamanho do conteúdo, essa apostila está restrita a versão 2.0.

A WCAG tem quatro princípios, são eles: que os conteúdos web devem ser perceptíveis, operáveis, compreensíveis e robustos. Os tópicos dessa apostila vão seguir essa sequência. Dentre esses princípios, estão distribuídas doze diretrizes que estudaremos abaixo.

### 3. Perceptível

Esse princípio afirma que “As informações e os componentes da interface do usuário devem ser apresentados em formas que possam ser percebidas pelo usuário.”. Para isso, deve existir alternativas em texto para os conteúdos não textuais, alternativas para mídias de som e vídeo, entre outros.

#### 3.1 Alternativas para conteúdos não textuais

Essa diretriz afirma que deve-se “fornecer alternativas textuais para qualquer conteúdo não textual, para que possa ser transformado em outras formas de acordo com as necessidades dos usuários, tais como impressão com tamanho de fontes maiores, braille, fala, símbolos ou linguagem mais simples”. Sobre essa diretriz, surgem algumas características a serem consideradas, são elas:

##### → ***Campos de entrada***

Tags input devem ter nome condizente com o que se espera que seja informado no campo. Isso pode ser feito usando o atributo name do input. Além disso, é interessante manter consistência e definir o ID com mesmo valor do name.

Para leitores de tela se faz uso também do atributo aria-label que denomina como o campo será rotulado por leitores de tela e tecnologias assistivas.

```
<input type="text" name="cpf" aria-label="cpf"/>
```

##### → ***Mídias de áudio e vídeo***

Deve ser fornecido uma descrição do conteúdo apresentado para que seja possível saber o conteúdo do vídeo. Para isso, pode-se utilizar o atributo aria-label, mostrado no item anterior, na tag vídeo para que seja lido o título do vídeo pela tecnologia assistiva. Arquivos de áudio podem ser inseridos na tag vídeo

```
<video controls aria-label="Filme: A fantástica fábrica de chocolates">  
  <source src="example.mp4" type="video/mp4">  
</video>
```

##### → ***Testes/avaliação***

Casos onde a percepção visual é importante e fornecer uma descrição completa invalide o uso, por exemplo, em uma avaliação, deve-se oferecer uma alternativa em texto com uma descrição do elemento, por exemplo, em imagens podemos usar o atributo "alt" e uma legenda.

No código abaixo vemos a tag Figure responsável por agrupar a imagem propriamente com sua legenda, a tag "figcaption" e o uso do atributo "alt" da imagem responsável por fornecer uma descrição da imagem.

```
<figure>  
    
  <figcaption>Fig.1 – Figura da cidade de Trulli na Itália </figcaption>  
</figure>
```

##### → ***Experiência sensorial***

Se o objetivo do conteúdo não textual for criar uma experiência sensorial, por exemplo, com sons de fundo, efeitos visuais, entre outros, deve-se fornecer uma descrição textual do que está acontecendo.

O Bootstrap tem uma classe chamada "visually-hidden" que oculta o conteúdo do item para todos, exceto para usuários em leitores de tela, o que pode nos ajudar a adicionar informação exclusivas para PcD.

Assim podemos colocar links, instruções sobre alternativas para navegação no site, dentre outros itens que só serão visíveis por PcD explicando a relação entre a estrutura do site e sua forma de navegação. Um uso é colocar informação dessa forma informando sobre a experiência sensorial que está sendo inserida no site.

Além disso, ao usar o atributo "role" do XHTML com o valor alert, o leitor de tela irá ler esse item o quanto antes para o usuário mesmo sem ele navegar entre itens do site.

```
<h2 class="visually-hidden" role="alert">Texto para leitores de tela</h2>
```

### → **Elementos de decoração e formatação**

Estes elementos devem ser colocados de forma invisível aos leitores de tela. Uma opção é fazer bastante uso de CSS sem inserir tags desnecessárias no corpo do site, deixando o corpo do site apenas com informações úteis para serem lidas.

Sobre essa questão, considere não usar tags de formatação junto ao texto, eliminando uso, por exemplo, de tag para negrito, itálico, entre outros, deixando que o CSS resolva tudo relativo a formatação.

Usar imagens de fundo ao invés de inserir tags de imagens quando essas imagens são decoração entre outras possibilidades que deixem o código html o mais limpo possível. Já quando as imagens fazem parte do conteúdo, deve-se usar a tag figure com a figcaption conforme exemplificado acima.

### → **Captcha**

Se for um captcha, isto é, elemento cuja finalidade é provar que o usuário não é um computador, deve ser fornecido uma descrição informando a necessidade do captcha e também alternativas para diferentes deficiências, como, por exemplo, figuras e poder escutar o som referente a figura. Note ao lado como funciona o reCaptcha, oferecido pelo google. Uma alternativa gratuita é o hCaptcha.

Note a presença de um botão para escutar o conteúdo da imagem na parte inferior junto a um botão de informação. Além disso, na parte superior, existe uma descrição informando em modo texto o que o usuário deve fazer.

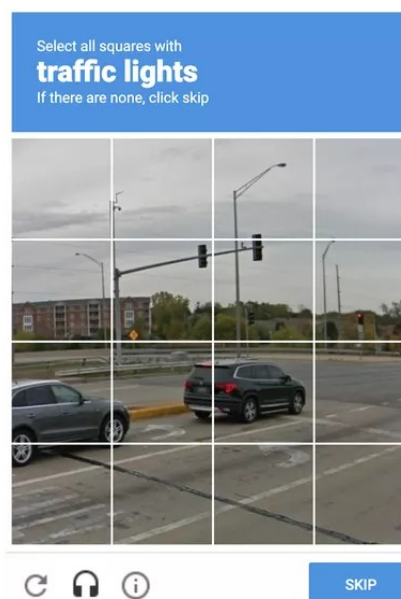


Figure 1: Google reCaptcha

### 3.2 Alternativas para áudio e vídeo pré gravados

Esta diretriz afirma que deve-se “fornecer alternativas para mídias baseadas em tempo”. Estas mídias devem fornecer um link para uma descrição textual com tudo que foi dito e exibido. No exemplo abaixo vemos as tags video, responsável por criar um player de vídeo, as tags source responsáveis por informar onde encontrar alternativas aos formatos de arquivo e as tags track responsáveis por fornecer subtitles, captions e descriptions que representam, respectivamente, legendas, transcrições e descrições.

```
<video controls>
  <source src="example.mp4" type="video/mp4">
  <source src="example.webm" type="video/webm">
  <track kind="subtitles" src="legendas_en.vtt" srclang="en" label="legendas em inglês">
  <track kind="captions" src="efeitos_en.vtt" srclang="en" label="legendas de efeitos sonoros em inglês">
  <track kind="descriptions" src="subtitles_en.vtt" srclang="en" label="descrição de cenas em inglês">
</video>
```

As legendas oferecem alternativa em texto para as falas. As transcrições oferecem alternativa em texto para informar sons que não são falas, como por exemplo barulhos de tiros, explosões, etc. Por último, as descrições oferecem descrição para elementos visuais que estão ocorrendo no vídeo. Para reproduzir arquivos de áudio com acessibilidade, podemos usar a tag video para exibir áudio.

Além disso, o vídeo pode conter uma audiodescrição que nada mais é que o conteúdo das legendas, transcrições e descrições, em forma de áudio. Esse áudio deve ser embutido no arquivo de vídeo de tal forma que possa escolher qual escutar de forma semelhante a escolha de um idioma de um vídeo.

### 3.3 O conteúdo deve poder ser apresentado de diferentes formas

Esta diretriz afirma que deve-se “criar conteúdo que pode ser apresentado de diferentes maneiras (por exemplo um layout simplificado) sem perder informação ou estrutura”.

Para isso podemos ter versões de layout simplificados para pessoas com deficiência (PcD), o que pode ser criado usando a tecnologia ARIA, porém não é o foco dessa apostila, assim, será mostrado apenas o básico sobre ARIA onde for necessário.

Além disso, devemos ter uma sequência de conteúdo fácil de entender ou de ser explicada em modo texto, pois, apenas em usar as tags semânticas corretamente já fará o leitor de tela conseguir identificar a estrutura do conteúdo informando quando é uma lista, um link, uma seção, o que dará a capacidade ao leitor de tela nomear os itens corretamente, possibilitando esta forma além da possibilidade visual.

Para que a organização de conteúdos esteja clara, é importante usar tags semânticas, isto é, tags que têm significado. Um exemplo comum no HTML 4 é organizar os tópicos usando as tags h1 a h6, pois elas indicam a estrutura do conteúdo em níveis, usar parágrafos quando o conteúdo for textual, entre outros mecanismos, uma vez que o leitor de tela permitirá uma navegação usando essa estrutura.

Além das tags do HTML 4 temos tags novas criadas no HTML 5, chamadas de tags semânticas, são elas:

| Tag     | Significado                                                                                                                                                                                                                                                                                                               |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Header  | Significa cabeçalho e é normalmente usado para atribuir as informações do topo da página com logotipo e menu de navegação                                                                                                                                                                                                 |
| Footer  | Significa rodapé e é normalmente usado para inserir informação de copyright, formas de contato e localização                                                                                                                                                                                                              |
| Nav     | Significa a abreviação de navigation (do inglês, navegação) é usada para criar o menu de navegação do site                                                                                                                                                                                                                |
| Main    | Significa conteúdo principal e nela não deve ser inserido informações desnecessárias tal como menu, logotipos, banners, formulários de pesquisa, etc. Deve conter apenas o conteúdo objetivo do site                                                                                                                      |
| Section | Uma seção é um agrupamento de conteúdo, é colocado dentro da tag main para agrupar itens de um conteúdo e deve conter um elemento h1, h2, h3, ... h6 de tal modo a estabelecer um título que descreve tudo que pode ser encontrado nessa seção                                                                            |
| Article | Significa artigo e é usada para agrupar um conteúdo independente e completo como, por exemplo, artigos de jornal, postagens em fórum ou blog, por serem itens independentes e também deve conter um elemento h1, h2, h3, ... h6 de tal modo a estabelecer um título que descreve tudo que pode ser encontrado nessa seção |
| Aside   | Significa um conteúdo a parte como, por exemplo, uma curiosidade, uma informação adicional ao conteúdo sendo lido. Algo que vem a trazer mais detalhes ao conteúdo, mas não traz prejuízo a leitura do conteúdo principal, caso o conteúdo aside não seja lido                                                            |

```

<html>
  <head> </head>
  <body>
    <header>
      <nav> </nav>
    </header>
    <main>
      <section>
        <article> </article>
        <aside> </aside>
      </section>
      <section> </section>
      <section> </section>
    </main>
    <footer> </footer>
  </body>
</html>

```

### 3.4 Facilitar a visão e audição dos conteúdos

Esta diretriz afirma que deve-se “facilitar a audição e a visualização de conteúdo aos usuários, incluindo a separação entre o primeiro plano e o plano de fundo”. Para isso, a cor não deve ser o único meio para transmitir informações, indicar uma ação ou distinguir um elemento visual. Podemos usar formatos, posição, entre outros.

Aqui uma boa idéia é usar espaço negativo, muito contraste e alterar o tipo de layout quando se muda entre seções diferentes de conteúdo a fim de tornar claro a existência de uma transição sem necessidade de recorrer ao uso de cores.



Essa melhoria ajudará quem tem perdas parciais de visão ou mesmo quem está em um ambiente ensolarado e tenta enxergar a tela de um telefone, pois com muito contraste ficará melhor de visualizar.

Além disso, caso um som toque por mais de 3 segundos no site, deve haver um meio para pausá-lo, pará-lo e controlar volume de forma independente ao controle global do sistema, o que no HTML 5 é providenciado pelo uso da tag video, mostrada acima.

## 4. Operável

Esse princípio afirma que "Os componentes de interface de usuário e a navegação devem ser operáveis", isto é, deve ser possível acessar todos os itens de conteúdo da interface.

### 4.1 Acessível via teclado

Esta diretriz afirma que devemos "fazer com que toda funcionalidade fique disponível a partir de um teclado". Notemos aqui que para o site ser acessível via teclado, o usuário deve ser capaz de entender o significado do elemento que está sendo lido, portanto deve ter sido usado as tags semânticas corretamente. Uma vez que as tags semânticas foram feitas para esse uso, o que inclui desde as tags h1, h2...h6 até as tags novas do HTML5.

Além disso, o leitor de tela irá buscar apenas por elementos significativos, evitando olhar DIVs ou elementos mal estruturados, assim, pode ser necessário informarmos quais elementos devem receber foco e estão sendo desconsiderados. Para isso, podemos usar o atributo tabindex com valor de zero, informando ao leitor de tela para incluir o elemento na leitura mantendo a ordem que ele é apresentado no site. Por outro lado, colocar um número negativo fará o elemento ser desconsiderado na leitura.

```
<h3 class="text-center" tabindex="0">
  Apostilas
</h3>
```

### 4.2 Tempo suficiente

Esta diretriz afirma que devemos "fornecer aos usuários tempo suficiente para ler e utilizar o conteúdo". Caso exista limitação de tempo, o usuário deve poder remover o limite de tempo, ajustar ou prolongar o limite antes dele ser atingido em até 10 vezes o limite de tempo original. Isso é importante em acesso a sites de banco ou avaliações online, por exemplo. As exceções a esse item é quando é em tempo real, ou o tempo decorrido na atividade é importante ser o valor estabelecido ou levaria mais de 20 horas, pois nesses casos não haveria o que fazer.

Além disso, conteúdos em movimento devem poder ser pausados. Por exemplo, conteúdos em carousel entre outras animações que durem mais de 5 segundos, pois dependendo da PcD, pode ser necessário mais tempo para entender o item. Para isso, o carousel do Bootstrap tem valor default para girar a cada 5 segundos exatos além de permitir pausar quando se coloca o mouse sobre a figura.

### 4.3 Convulsões

Esta diretriz afirma que não devemos "...criar conteúdo de uma forma conhecida por causar convulsões". A página WEB não contém qualquer conteúdo que produza mais de três flashes por segundo, uma vez que pode causar convulsões em pessoas propensas.

Para que sejam criados flashes em páginas web, uma forma seria usando javascript ou JQuery para alterar cores de fundo, por exemplo. Ainda não estudamos JQuery, mas deve-se evitar componentes prontos que possam causar esse comportamento.

#### **4.4 Navegável**

Esta diretriz afirma que devemos “Fornecer maneiras de ajudar os usuários a navegar, localizar conteúdos e determinar onde se encontram”. Para isso, deve existir uma forma de ajudar os usuários a navegar como, por exemplo, um mecanismo para ignorar blocos de conteúdo que são repetidos em várias páginas web, ajudando os usuários. Uma forma de fazer isso é usando o atributo tabindex com valor negativo, como explicado acima, pois, assim, o elemento será ignorado pelo leitor de tela.

Deve ser fácil definir onde os usuários se encontram. Para isso, a página web deve ter títulos, isto é, nome da página e cabeçalhos h1, h2, .. h6 que descrevam o tópico ou assunto abordado em seu conteúdo tornando mais simples a navegação.

A ordem de foco na página deve ser fácil de entender, e para escolhermos essa ordem podemos fazer uso ainda do atributo tabindex, tornando mais fácil localizar conteúdos que devem estar indexados usando a tag "nav" pois o leitor de tela reconhecerá automaticamente como sendo o elemento de navegação. Este deve estar no topo do site por ser o primeiro local a ser lido.

A finalidade dos links devem poder ser determinadas apenas através da tag responsável pelo link, uma forma é usar um atributo aria-label para melhor descrever sua finalidade.

### **5. Compreensível**

Esse princípio afirma que "a informação e a operação da interface de usuário devem ser compreensíveis".

#### **5.1 Legível**

Esta diretriz afirma que devemos “tornar o conteúdo do texto legível e compreensível”. Aqui é importante definir corretamente o idioma da página web, para isso podemos usar o atributo lang no html. Isto torna possível a pronúncia correta de palavras, permite também tradução correta por leitores de tela assim como na impressão em braile evita que sejam feitas contrações erradas de palavras. Para português, o valor correto é “pt” ao invés de “pt-br”.

```
<html lang="pt"> ... </html>
```

#### **5.2 Previsível**

Esta diretriz afirma que devemos “fazer com que as páginas web apareçam e funcionem de modo previsível”. Ao visualizar um componente, por exemplo, ele não deve fazer alterações automaticamente exceto se o usuário foi avisado.

Por exemplo, um carousel quando recebe foco, não deve modificar slides sem avisar ou sem ter recebido um comando do usuário para isso, o que não é o comportamento padrão. Assim, faz-se necessário ajustes para que o carousel fique previsível para tecnologias assistivas, contudo isto vai além do escopo desta apostila.

### 5.3 Assistência de Entrada

Esta diretriz afirma que deve-se “ajudar os usuários a evitar e corrigir erros”. Se um erro de entrada for automaticamente detectado, o item que apresenta erro é identificado e o erro é descrito para o usuário em texto.

Uma forma de descrever o erro e ele ser imediatamente lido pelo leitor de tela é usar uma div, visualmente escondida com o atributo role com o valor alert, pois ele será imediatamente lido pelo leitor de tela, contudo para inserir o elemento no html enquanto o usuário manipula o item seria necessário JQuery, que foge do conteúdo desta apostila.

```
<h2 class="visually-hidden" role="alert">Texto para leitores de tela</h2>
```

Além disso, sempre que desejado que o usuário insira dados, devem ser fornecidos rótulos ou instruções. Para que isso seja possível, podemos utilizar o atributo aria-label como explicado anteriormente.

## 6. Robusto

Este princípio afirma que “o conteúdo deve ser robusto o suficiente para poder ser interpretado de forma confiável por uma ampla variedade de agentes de usuário, incluindo tecnologias assistivas”.

### 6.1 Compatível

Esta diretriz afirma que devemos “Maximizar a compatibilidade entre os atuais e futuros agentes de usuário, incluindo tecnologias assistivas”. Para isso, é importante que o código HTML esteja corretamente validado para que as tecnologias assistivas não tenham dificuldade em interpretá-lo.

Além disso, todos os elementos de interface de usuário, isto é, elementos com que o usuário seria capaz de ver ou interagir, devem ser possíveis de identificar seu nome e funcionalidade através de tecnologias assistivas como, por exemplo, leitores de tela. Para isso pode-se identificar esses nomes e funcionalidades usando o atributo aria-label, mais uma vez.

## EXERCÍCIO

Agora que entendemos um pouco mais sobre acessibilidade e HTML 5, crie um site seguindo todas as regras pedidas no último exercício e com acessibilidade. Você pode reusar o site anterior de sua autoria, caso deseje.

A entrega será dois vídeos explicativos. O primeiro deve mostrar o comportamento do leitor de tela lendo o conteúdo como o site era antes de aplicado a acessibilidade onde deve-se usar apenas o teclado para acessar todo o conteúdo. O segundo vídeo deve mostrar como ficou a leitura de tela após o site se tornar acessível novamente apenas usando teclado para navegar.

No segundo vídeo, quando o leitor de tela se comportar diferente do que era no primeiro vídeo, o aluno deve mostrar o que foi alterado no código para causar esse efeito.

Apesar de ser entregue dois vídeos, será considerado como uma única atividade uma vez que para gravar o primeiro vídeo será necessário apenas um código já desenvolvido e um leitor de tela que fará toda a leitura da tela.

## REFERÊNCIAS

WCAG 2.0. 2021. Disponível em: <<https://www.w3.org/Translations/WCAG20-pt-br/>>