



Programação em Ambiente Web I

Aula 10 – Acessibilidade

Prof. MSc. Cleyton Slaviero

cslaviero@gmail.com

Acessibilidade

”O poder da web está em sua universalidade. O acesso por todos, independente de qualquer deficiência é um aspecto essencial”

Tim Berners-Le

O que é acessibilidade na web?

Segundo a WAI (*Web Accessibility Initiative*)*

*”Acessibilidade na web significa que pessoas com deficiências podem **perceber, entender, navegar e interagir** além de poder **contribuir** para a web”**

* <https://www.w3.org/WAI/intro/accessibility.php>

A quem se destina?

- Deficientes auditivos
- Deficientes motores
- Deficientes visuais...

... Mas não somente para estes!

- Deficiência temporária
- Início de aprendizado
- Idade

A quem se destina?

- Fazer um site acessível é pensar em todos os diferentes públicos que podem acessar seu website

WAI e WCAG

- *WAI (Web Accessibility Initiative)*
 - Iniciativa da W3C que visa
 - Desenvolver guidelines para acessibilidade web
 - Desenvolver materiais de suporte para entender e aplicar acessibilidade na web
 - Desenvolver recursos, por meio de colaboração internacional
- *WCAG (Web Content Accessibility Guidelines)*
 - Recomendações para acessibilidade na web
 - *WCAG 1.0*
 - 14 recomendações
 - Era mais técnicos

WCAG 2.0

- Reformulou as recomendações
 - Voltado para o público geral (menos técnica)
- Princípios
 - Princípio 1: Perceptível
 - A informação e os componentes da interface do usuário tem de ser apresentados aos usuários em formas que eles possam perceber
 - Isso significa que qualquer usuário deve ser capaz e **perceber** a informação sendo apresentada
 - Exemplo: Alternativas em texto curto para o conteúdo não textual
 - Como?
 - Utilizando atributo **alt**

Princípio 1: Perceptível

- Tornar imagens acessíveis
 - Alguns usuários podem não ser capazes de ver imagens
 - Outros podem utilizar navegadores textuais
 - Podem até ter desativado o suporte a imagens
 - Podem estar usando "aceleradores" (google light web, opera)
 -

Princípio 1: Perceptível

- Exemplo com e sem texto alternativo
- Exemplo com texto que não represente corretamente o conteúdo não textual
- Outra coisa:
 - Alt é capturado pelos buscadores (e.g. google)
- Outros exemplos
 - Tipos diferentes de texto alternativo
- Cuidado com limitação de caracteres
 - Alt tem limite de aprox. 100 caracteres (conferir)
 - Podemos colocar uma descrição longa em outro ponto

Princípio 1: Perceptível

- E em elementos de formulário?
 - Atributo label + propriedade **for**
- **E se não puder usar label?**
 - Atributo **title**
 - Exemplo: campos de texto para números de telefone
- E se for um captcha?
 - W3C recomenda que se identifique o elemento como um captcha
 - Exemplo: ... alt='Digite as palavras da imagem'>
 - Utilizar outra forma de acessar
 - Um arquivo de áudio, uma série de perguntas
- E se o elemento textual tiver que ser ignorado pelas tecnologias assistivas?
 - Utilize um CSS para separar a imagem da estrutura da página
 - Background-image



Princípio 1: Perceptível

- Fornecer legendas
 - Alternativas para áudio e vídeo
 - Open ou closed-caption
 - Transcrição do áudio/vídeo

Princípio 1: Perceptível

- Conteúdo adaptável
 - Utilizar HTML+CSS
 - Organização --> HTML
 - Layout → CSS
 - Table **apenas** para informações tabulares
 - Utilizar **caption** para associar **tabelas de dados**
 - É exibido no browser
 - Utilizar **summary** na tabela para dar uma visão geral da tabela de dados
 - Não é exibido no browser!
 - Usar headers e ids e associar
 - Usar contraste adequado



Princípio 2: Operável

- Os componentes de interface de usuário e a navegação tem de ser operáveis
 - A interface de operação não pode exigir interações que um usuário não consiga executar
- Acessível por teclado
 - Como fazer isso?
 - Formulários e links já auxiliam isso

Princípio 2: Operável

- Links
 - Utilizar as propriedades adequadas
- Botões
 - Atributo value
 - Atributo alt, se o botão for uma imagem
- Formulários
 - Atributo label
 - Atributo title

Princípio 2: Operável

- Navegação
 - Ajudar os usuários a navegar e encontrar o conteúdo
- Exemplo
 - Adicionar um link na parte superior de cada página que vá para o conteúdo principal
 - `Pular para o conteúdo principal`
 - Agrupar blocos de material repetido de forma que possa ser ignorado
 - Separar cabeçalhos semanticamente
 - Agrupar links com lista

Princípio 3 – Compreensível

- A informação e a operação da interface de usuário devem ser compreensíveis
 - Usuário tem que compreender a informação na página
- Texto
 - Legível e compreensível
 - Exemplo
 - Definir o idioma da página

Princípio 4 - Robusto

- O conteúdo tem de ser robusto o suficiente para poder ser interpretado de forma concisa por diversos agentes do usuário, incluindo tecnologias assistivas
- Usuários devem ser capazes de acessar as tecnologias conforme elas evoluem
 - O conteúdo deve se manter acessível!

Princípio 4 - Robusto

- Compatibilidade
 - Maximizar a compatibilidade com as tecnologias atuais e futuras
 - Elementos devem conter
 - Marcações de início e fim corretas
 - Encaixados com as especificações
 - Não contém atributos duplicados
- Como garantir isso?
 - Validando a página
 - Exemplo:
 - <https://validator.w3.org/>
 - <https://html5.validator.nu/>

Outras guidelines

- eMAG
 - Modelo de Acessibilidade em Governo Eletrônico
 - Diretrizes brasileiras de acessibilidade
 - Possui uma série de recomendações a serem consideradas no acesso a sites do governo brasileiro
 - Está na versão 3.1
 - Desenvolvida com base na versão 2.0 do WCAG

Mais informações

- Validadores
 - <https://www.w3.org/WAI/ER/tools/>
- Diretrizes completas
 - <https://www.w3.org/Translations/WCAG20-pt-PT/>

Exercício

- A partir do documento HTML criado na última aula, verifique se ele atende às diretrizes do WCAG. Caso não atenda, verifique em que pontos ele pode ser melhorado e elabore as melhoras, pontuando aonde modificou (pode deixar uma versão antiga e a nova, lado a lado) , usando comentários para identificar o princípio seguido
 - Caso não tenha criado um documento anteriormente, pode ser utilizado um documento que você tenha criado em um exercício anterior