

Capítulo

4

Personas e Acessibilidade Web: Desenvolvendo para Diferentes Usuários

Matheus A. S. Santos, Beatriz L. Silva, Marina F. Souza, Bruna C. R. Cunha

Abstract

Web accessibility is essential for digital inclusion, yet technical guidelines and tools do not, by themselves, guarantee accessible experiences. The chapter presents the fundamentals of Web accessibility: legal and regulatory aspects, WCAG, semantic HTML, ARIA, Assistive Technology, and evaluation tools. As its central focus, it discusses the creation and analysis of personas representing different disability profiles, used to align technical requirements with users' real needs and anticipate barriers in Web interfaces before implementation. A Barrier Walkthrough example illustrates this application, identifying problems and proposing solutions. Finally, it discusses the limitations of personas and of technical compliance itself, reinforcing that compliance is necessary but not sufficient, and that the participation of people with disabilities remains indispensable.

Resumo

A acessibilidade Web é essencial para a inclusão digital, mas diretrizes e ferramentas técnicas não garantem, por si só, experiências acessíveis. O capítulo apresenta os fundamentos da acessibilidade Web: aspectos legais e normativos, WCAG, HTML semântico, ARIA, Tecnologia Assistiva e ferramentas de avaliação. Como eixo central, discute a criação e análise de personas com diferentes perfis de deficiência, usadas para aproximar requisitos técnicos das necessidades reais dos usuários e antecipar barreiras em interfaces Web antes da implementação. Um exemplo de Percurso de Barreira ilustra essa aplicação, identificando problemas e propondo soluções. Por fim, discutem-se os limites das personas e da própria conformidade técnica, reforçando que ela é necessária, mas não suficiente, e que a participação de pessoas com deficiência continua indispensável

4.1. Introdução

Segundo a Pesquisa Nacional por Amostra de Domicílios (PNAD) Contínua 2022, cerca de 8,9% da população brasileira com dois anos ou mais possui algum tipo de deficiência,

correspondendo a aproximadamente 18,6 milhões de pessoas [14]. Esses dados evidenciam que as pessoas com deficiência representam uma parcela significativa da população brasileira e reforçam a necessidade de promover condições de acesso e participação em diferentes espaços da sociedade, incluindo os ambientes digitais. Por trás desse número, no entanto, existe uma diversidade de perfis (pessoas cegas, com baixa visão, surdos, com deficiência motora ou cognitiva, entre outros) cujas necessidades específicas frequentemente não são consideradas de forma concreta ao longo do processo de desenvolvimento, o que motiva a proposta deste capítulo de aproximar essa diversidade da prática de design e código por meio de personas.

Em um contexto no qual a tecnologia e a internet desempenham um papel fundamental no acesso à informação, à educação, aos serviços e à comunicação, a acessibilidade na Web torna-se um aspecto essencial para a promoção da inclusão digital. Nesse sentido, a acessibilidade Web busca reduzir ou eliminar barreiras que possam dificultar a interação de pessoas com diferentes tipos de deficiência com conteúdos e serviços disponibilizados na internet. Diretrizes como as *Web Content Accessibility Guidelines* (WCAG) estabelecem recomendações para tornar conteúdos Web mais acessíveis e constituem uma das principais referências para o desenvolvimento de interfaces acessíveis; entretanto, diretrizes técnicas por si só nem sempre traduzem, na prática do desenvolvedor, a experiência real de quem enfrenta essas barreiras, lacuna que a construção de personas busca preencher, ao dar rosto e contexto de uso a critérios que, de outra forma, permanecem abstratos.

Apesar da existência dessas diretrizes e da crescente importância dos ambientes digitais, a acessibilidade ainda apresenta limitações significativas nos sites brasileiros. De acordo com um relatório da BigDataCorp em parceria com o Movimento Web para Todos, que analisou a acessibilidade de sites ativos no Brasil por meio de testes automatizados, dos 26,3 milhões de sites analisados, apenas 2,9% foram considerados acessíveis, resultado que representa uma redução em relação aos 3,3% identificados em 2023 [15]. Além disso, o mesmo levantamento aponta que, mesmo com uma leve melhora frente ao ano anterior, 96% dos sites de notícias analisados ainda apresentam algum tipo de falha de acessibilidade [15].

Esses dados evidenciam que a acessibilidade ainda não é incorporada de maneira consistente ao desenvolvimento de projetos Web, e que o cumprimento formal de diretrizes não é, isoladamente, suficiente para garantir experiências verdadeiramente inclusivas. A presença de barreiras de acessibilidade pode comprometer a autonomia das pessoas com deficiência e dificultar seu acesso a informações e serviços disponíveis no ambiente digital. Dessa forma, torna-se necessário compreender como essas barreiras são produzidas na prática, e é justamente para exercitar esse olhar, situado no usuário real e não apenas na norma, que a criação de personas com diferentes perfis de deficiência ocupa um lugar central neste capítulo.

Este capítulo tem por objetivo apresentar, de forma integrada, (a) os fundamentos da acessibilidade Web e sua relação com a inclusão digital, incluindo aspectos legais e normativos brasileiros; (b) as principais diretrizes das *Web Content Accessibility Guidelines* (WCAG) e sua aplicação no desenvolvimento de interfaces acessíveis; (c) práticas de desenvolvimento inclusivo, abordando HTML semântico, formulários acessíveis, navega-

ção por teclado e uso adequado de atributos ARIA; (d) uma visão prática sobre recursos de TA e ferramentas automatizadas de avaliação de acessibilidade; e, como eixo central do capítulo, (e) atividades baseadas na criação e análise de personas com diferentes perfis de deficiência, utilizadas para exercitar a identificação de barreiras e a proposição de soluções de design e desenvolvimento a partir de necessidades concretas de usuários.

4.2. Acessibilidade Web, Direitos e Legislação

A trajetória da acessibilidade digital está intimamente ligada à evolução da própria tecnologia computacional. Nas décadas de 1970 e 1980, a popularização dos computadores pessoais (PCs) representou uma mudança significativa: dispositivos antes restritos a técnicos e a grandes estruturas compartilhadas passaram a ser operados diretamente por um único usuário. Essa transformação impulsionou as primeiras discussões mais amplas sobre acessibilidade digital.

Em 1990, surgiu a World Wide Web (WWW), que, em sua versão inicial, era limitada e restrita a poucos usuários [60]. A partir de 1994, no entanto, sua expansão se tornou expressiva, consolidando-a como um dos principais meios de acesso à informação na atualidade, à frente inclusive de mídias tradicionais como a televisão e o rádio. Nesse mesmo ano, foi criado o World Wide Web Consortium (W3C), organização responsável pelo desenvolvimento de padrões e diretrizes voltados à construção de uma Web baseada nos princípios de acessibilidade, internacionalização, privacidade e segurança.

À medida que a Web se expandia, tornou-se necessário discutir formas de garantir o acesso democrático a pessoas com diferentes necessidades. Nesse contexto, em 1997 o W3C lançou a Web Accessibility Initiative (WAI) [1], iniciativa internacional voltada à promoção da acessibilidade na Web. No ano seguinte, em 1998, os Estados Unidos aprovaram a Section 508 [2], que passou a exigir que órgãos federais oferecessem conteúdos digitais acessíveis. Esse processo teve continuidade em 1999, quando a WAI publicou a primeira versão das *Web Content Accessibility Guidelines* (WCAG) [3], documento que estabeleceu as bases técnicas ainda hoje utilizadas para orientar a construção de conteúdos acessíveis na Web.

4.2.1. Direitos e Legislação no Contexto Brasileiro

No Brasil, o primeiro marco legal totalmente dedicado ao tema foi a Lei nº 10.098, sancionada em 2000 [7], conhecida como Lei de Acessibilidade. Essa norma estabeleceu critérios gerais para a eliminação de barreiras nos meios de comunicação e informação, ainda que voltada, em grande parte, à acessibilidade física e urbana. Em 2004, o Decreto nº 5.296 [8] regulamentou essa lei e, em seu artigo 47, determinou explicitamente que os portais e sítios eletrônicos da administração pública deveriam ser tornados acessíveis, representando um dos primeiros dispositivos legais brasileiros a tratar diretamente da acessibilidade digital.

Em 2009, o Brasil promulgou a Convenção da ONU sobre os Direitos das Pessoas com Deficiência, por meio do Decreto nº 6.949 [9]. Por ter sido incorporada com status de emenda constitucional, a Convenção consolidou o acesso à informação digital como um direito fundamental no ordenamento jurídico brasileiro. Já em 2012, o Decreto nº 7.724 [10], que regulamenta a Lei de Acesso à Informação (LAI), reforçou essa diretriz

ao determinar que os sítios eletrônicos públicos garantissem o acesso a dados de forma clara e acessível a cidadãos com deficiência.

O maior marco da acessibilidade digital no país, no entanto, viria em 2015, com a promulgação da Lei nº 13.146 [11], conhecida como Lei Brasileira de Inclusão. Em seu artigo 63, a lei tornou obrigatória a acessibilidade em sítios e aplicativos mantidos por órgãos governamentais e por empresas privadas com sede no Brasil, consolidando um percurso de mais de uma década de avanços legislativos em direção à garantia do acesso digital como direito.

4.2.2. WCAG, eMAG e ABNT NBR 17225

Atualmente, a acessibilidade digital é orientada por um conjunto de padrões técnicos, internacionais e nacionais, que traduzem princípios legais em diretrizes práticas de implementação. No plano internacional, a principal referência é a WCAG (*Web Content Accessibility Guidelines*), publicada pelo W3C por meio da WAI. Sua primeira versão, a WCAG 1.0, foi lançada em 1999 [3], com 14 diretrizes voltadas a páginas estáticas. Em 2008, a WCAG 2.0 [4] ampliou substancialmente o escopo do documento, introduzindo os quatro princípios que sustentam a norma até hoje: conteúdo perceptível, operável, compreensível e robusto. Novas versões vieram em 2018 (WCAG 2.1 [5], com foco em dispositivos móveis e baixa visão) e em 2023 (WCAG 2.2 [6], a mais recente), sempre de forma aditiva e compatível com as versões anteriores, consolidando-se como o principal parâmetro técnico de acessibilidade adotado por governos e organizações em todo o mundo.

No Brasil, o primeiro padrão técnico nacional foi o eMAG (Modelo de Acessibilidade em Governo Eletrônico) [12], desenvolvido a partir de 2004 com base no estudo comparativo de catorze normas internacionais, entre elas a própria WCAG 1.0. Disponibilizado para consulta pública em 2005 e institucionalizado pela Portaria nº 3 de 2007, o eMAG tornou-se obrigatório nos sítios e portais do governo federal, passando por revisões que culminaram na versão 3.1, de 2014, já alinhada à WCAG 2.0.

Mais recentemente, em março de 2025, a ABNT publicou a NBR 17225 [13], primeira norma técnica brasileira dedicada especificamente à acessibilidade em conteúdos e aplicações Web. Alinhada à WCAG 2.2 e ao artigo 63 da Lei Brasileira de Inclusão, a norma reúne 146 diretrizes de alcance nacional, aplicáveis tanto a órgãos públicos quanto a empresas privadas, e representa o padrão técnico mais abrangente já produzido no país sobre o tema.

Juntos, esses três documentos, um internacional e dois nacionais, formam a base técnica que orienta, hoje, a construção de sites e aplicações digitalmente acessíveis.

4.3. Diretrizes WCAG

Conforme apresentado no capítulo anterior, a WCAG constitui uma das principais referências para a acessibilidade Web [6]. Sua estrutura é organizada em quatro níveis hierárquicos: os princípios, que representam os pilares fundamentais da acessibilidade; as diretrizes, que detalham objetivos gerais dentro de cada princípio; os critérios de sucesso, que traduzem essas diretrizes em requisitos testáveis e objetivos; e os níveis de confor-

midade (A, AA e AAA), atribuídos a cada critério conforme seu grau de exigência. Essa estrutura hierárquica permite que a WCAG seja utilizada tanto como referência conceitual quanto como checklist técnico ao longo do processo de desenvolvimento.

4.3.1. Princípios POUR

Os princípios POUR (*Perceivable, Operable, Understandable, Robust*) constituem os quatro pilares fundamentais da WCAG e orientam, de forma geral, o design de conteúdos acessíveis. Mais do que garantir a conformidade técnica com a norma, esses princípios buscam assegurar que o conteúdo seja genuinamente utilizável por pessoas com diferentes tipos de deficiência.

Tabela 4.1. Princípios de acessibilidade da WCAG.

Princípio	O que significa	Como aplicar
Perceptível	As informações e os componentes da interface devem ser apresentados de forma que possam ser percebidos pelos usuários.	Fornecer alternativas em texto para imagens, legendas para conteúdos em vídeo e garantir contraste adequado entre texto e fundo.
Operável	Os componentes de navegação e da interface devem ser operáveis por diferentes meios de interação.	Permitir a navegação completa por teclado, evitar conteúdos que possam causar convulsões e fornecer tempo suficiente para leitura e interação.
Compreensível	As informações e a operação da interface devem ser apresentadas de maneira clara e de fácil entendimento.	Utilizar linguagem clara, manter o comportamento das páginas previsível e fornecer instruções e mensagens para prevenção e correção de erros.
Robusto	O conteúdo deve ser compatível com diferentes navegadores, dispositivos e recursos de TA, atuais e futuras.	Utilizar HTML semântico e código válido, garantindo que o conteúdo possa ser corretamente interpretado por leitores de tela e outros recursos de TA.

4.3.2. Critérios de Sucesso

Dentro de cada princípio, a WCAG estabelece diretrizes e, dentro de cada diretriz, critérios de sucesso: regras práticas e testáveis, identificadas por numeração própria (como 1.1.1, 1.4.3 ou 2.1.1), que especificam exatamente o que deve ser implementado no código ou no design para que o conteúdo funcione corretamente com leitores de tela, navegação por teclado e demais recursos de TA. A Tabela 4.2 apresenta alguns exemplos de critérios de sucesso associados a cada princípio.

Tabela 4.2. Exemplos de critérios de sucesso da WCAG.

Princípio	Critério	Exemplo
Perceptível	1.1.1	Conteúdo Não Textual: toda imagem informativa deve possuir um texto alternativo (atributo <code>alt</code>) que descreva seu propósito.
Perceptível	1.4.3	Contraste (Mínimo): o contraste entre texto e plano de fundo deve ser de, no mínimo, 4,5:1 para texto padrão.
Operável	2.1.1	Teclado: toda funcionalidade da página deve poder ser operada exclusivamente por teclado, sem depender do <i>mouse</i> .
Operável	2.4.7	Foco Visível: quando um elemento recebe foco via teclado, deve haver uma indicação visual clara desse foco.
Compreensível	3.3.1	Identificação de Erro: quando um erro é detectado em um formulário, o item com problema deve ser identificado e descrito ao usuário em texto.
Robusto	4.1.2	Nome, Função, Valor: componentes de interface (como botões e campos) devem ter seu nome e função corretamente expostos a recursos de TA, geralmente por meio de HTML semântico ou atributos ARIA.

4.3.3. Níveis de Conformidade

Cada critério de sucesso está associado a um nível de conformidade, A, AA ou AAA, que indica seu grau de exigência e o impacto de seu descumprimento sobre a experiência do usuário.

Tabela 4.3. Níveis de conformidade da WCAG.

Nível	Descrição
A (Básico)	Remove as barreiras mais críticas de acessibilidade. Sem o atendimento a esse nível, parte dos usuários pode não conseguir utilizar o site ou acessar determinados conteúdos.
AA (Recomendado)	Resolve as principais barreiras de acessibilidade encontradas em situações comuns de uso. É o nível geralmente adotado como referência em leis, normas e diretrizes de acessibilidade, incluindo a legislação brasileira.
AAA (Avançado)	Representa o nível mais elevado de conformidade com as diretrizes de acessibilidade. Pode ser indicado para públicos ou conteúdos específicos, ainda que sua aplicação integral seja, muitas vezes, difícil em sites complexos.

4.4. HTML Semântico e Recursos de Acessibilidade

HTML semântico refere-se ao uso de elementos que descrevem o *significado* e o papel do conteúdo dentro da página, e não apenas sua aparência visual [30]. Em vez de estruturar um documento inteiro com elementos genéricos como `<div>` e `<span>` – prática conhecida informalmente como “*div soup*” –, o HTML5 disponibiliza elementos como `<header>`, `<nav>`, `<main>`, `<article>`, `<section>`, `<aside>`, `<footer>` e `<figure>`, cada um comunicando explicitamente sua função dentro do documento [31]. Essa separação entre estrutura e apresentação é um dos princípios fundadores da própria Web, reforçado desde as primeiras recomendações do W3C sobre acessibilidade [1] e formalizado nas primeiras versões das diretrizes de conteúdo Web acessível [3].

O uso de marcação semântica não é apenas uma boa prática estética: navegadores, motores de busca e, principalmente, tecnologias assistivas (leitores de tela, linhas Braille,

softwares de reconhecimento de voz) dependem dessa semântica para construir uma *árvore de acessibilidade* (*accessibility tree*) que representa corretamente a página para o usuário [32].

4.4.1. Importância da Estrutura Correta da Página

A estrutura correta de uma página HTML contribui para que seu conteúdo seja organizado de maneira lógica e possa ser interpretado adequadamente por diferentes agentes de usuário e mecanismos de busca [30, 32]. O uso de elementos semânticos permite representar a função e o significado de cada região do documento em vez de sua mera aparência visual, definindo claramente áreas como cabeçalho (`<header>`), navegação (`<nav>`), conteúdo principal (`<main>`) e rodapé (`<footer>`) [30, 33]. Essa separação entre estrutura e apresentação torna a organização do código mais transparente e previsível, tanto para navegadores quanto para recursos de TA [30, 32].

Essa estruturação também é determinante para a navegabilidade do documento. Leitores de tela, por exemplo, utilizam elementos semânticos e marcas estruturais (*landmarks*) como pontos de referência para sinalizar as regiões da página, permitindo que o usuário salte diretamente entre elas [30, 33]. Essa forma de navegação é particularmente relevante na prática: levantamentos periódicos sobre o uso de leitores de tela mostram que navegar por landmarks e por cabeçalhos está entre os métodos mais utilizados pelos usuários para explorar rapidamente uma página, ao lado da leitura sequencial e da navegação por listas de links [19]. Do mesmo modo, uma organização hierárquica adequada dos títulos possibilita compreender as relações de subordinação entre as diferentes seções, funcionando como um mapa mental do conteúdo [30, 33].

Do ponto de vista da engenharia de software, a adesão ao HTML semântico também impacta a manutenção do código ao longo do ciclo de vida do projeto. Quando os elementos são empregados conforme sua finalidade nativa, elimina-se a necessidade de encadeamentos excessivos de contêineres genéricos (a prática de “*div soup*”) ou do uso de classes, scripts e estilos adicionais apenas para simular um significado que o próprio elemento nativo já expressa por padrão [30, 33]. Isso reduz a quantidade de código a ser mantido, facilita a leitura por outros desenvolvedores e diminui o risco de que mudanças futuras quebrem o comportamento esperado por recursos de TA, já que a semântica passa a fazer parte da própria estrutura do documento, e não de convenções implícitas de estilo ou de comportamento adicionadas via JavaScript.

Portanto, a utilização adequada dos elementos estruturais do HTML ultrapassa a mera organização de código: ela estabelece a base para que o conteúdo seja interpretado de forma consistente por uma grande diversidade de navegadores, dispositivos e tecnologias de interação [30, 32]. A Tabela 4.4 resume os principais elementos estruturais do HTML5 e o papel que cada um assume perante recursos de TA.

Tabela 4.4. Principais elementos semânticos do HTML5 e seu papel de acessibilidade.

Elemento	Função	Landmark/papel implícito
<header>	Cabeçalho da página ou de uma seção.	banner (quando associado ao documento inteiro).
<nav>	Bloco de links de navegação principal ou secundária.	navigation.
<main>	Conteúdo principal e único da página.	main.
<article>	Conteúdo autocontido e reutilizável (post, notícia, comentário).	article.
<section>	Agrupamento temático de conteúdo, geralmente com título próprio.	region (quando rotulado).
<aside>	Conteúdo relacionado, mas tangencial ao fluxo principal.	complementary.
<footer>	Rodapé da página ou de uma seção.	contentinfo (quando associado ao documento inteiro).
<figure>	Conteúdo ilustrativo (imagem, gráfico, código) com legenda associada.	figure.

4.4.2. Estrutura e Elementos Acessíveis

Nesse contexto, destacam-se alguns elementos e práticas que contribuem para a construção de páginas HTML acessíveis. A seguir, são apresentados exemplos de implementações não acessíveis e acessíveis, acompanhados de uma breve explicação sobre sua importância.

Títulos e Hierarquia.

Os elementos <h1> a <h6> devem refletir a hierarquia lógica do conteúdo, sem saltos de nível (por exemplo, de <h2> direto para <h4>). Leitores de tela permitem que o usuário navegue de título em título, de modo que uma hierarquia mal formada compromete a compreensão geral da página [5, 12].

Links e Botões.

Elementos de *link* (<a>) devem ser usados para navegação, enquanto <button> deve ser usado para ações dentro da própria página. O texto do link deve ser descritivo por si só (evitando construções como “clique aqui”), pois muitos usuários de leitor de tela navegam ouvindo apenas a lista de links da página [6, 2]. Elementos interativos criados a partir de <div> ou com eventos de clique, sem função ou papel semântico associado, não recebem foco de teclado nem são anunciados corretamente por recursos de TA.

Como exemplo, um botão implementado com um elemento genérico <div> não possui, por padrão, o comportamento e a semântica de um botão:

```
<div onclick="salvar()">Salvar</div>
```

Código 4.1. Não acessível: div como botão

A utilização do elemento nativo <button> fornece semântica e suporte à interação por teclado de forma nativa:

```
<button type="button" onclick="salvar()">Salvar</button>
```

Código 4.2. Acessível: elemento nativo

Formulários.

Cada campo de formulário deve possuir um `<label>` explicitamente associado (via atributo `for/id` ou envolvendo o campo), e grupos de campos relacionados devem usar `<fieldset>` e `<legend>`. Mensagens de erro e campos obrigatórios devem ser comunicados de forma programática (por exemplo, com `aria-describedby` e `aria-required`), e não apenas por cor [5, 12].

Um campo sem um rótulo associado dificulta a identificação de sua finalidade por recursos de TA:

```
<p>Nome :</p>
<input type="text" name="nome">
```

Código 4.3. Não acessível: campo sem rótulo associado

A associação explícita entre `<label>` e `<input>` permite que recursos de TA identifiquem corretamente o campo:

```
<label for="nome">Nome:</label>
<input type="text" id="nome" name="nome" autocomplete="name">
```

Código 4.4. Acessível: label associado ao campo

Imagens e Texto Alternativo.

Toda imagem com valor informativo precisa de um atributo `alt` descritivo; imagens puramente decorativas devem usar `alt=""` para serem ignoradas por leitores de tela. Imagens complexas, como gráficos, infográficos e diagramas, podem exigir uma descrição estendida para transmitir informações que não podem ser adequadamente apresentadas apenas no atributo `alt`. A descrição estendida fornece detalhes adicionais sobre o conteúdo da imagem, como valores, relações, tendências ou outras informações relevantes para sua compreensão. Ela pode ser apresentada como texto adjacente à imagem ou associada por meio de elementos como `<figure>` e `<figcaption>` [3, 2].

Uma imagem sem texto alternativo não fornece aos usuários de leitores de tela informações sobre seu conteúdo:

```

```

Código 4.5. Não acessível: sem texto alternativo

Quando a imagem possui valor informativo, o atributo `alt` deve apresentar uma descrição adequada ao seu conteúdo. Para imagens simples, essa descrição pode ser suficiente:

```

```

Código 4.6. Acessível: com texto alternativo descritivo

Em imagens complexas, entretanto, o atributo `alt` pode ser insuficiente para transmitir todas as informações apresentadas. Nesses casos, recomenda-se fornecer uma descrição estendida. Por exemplo, um gráfico pode ter um texto alternativo resumido e uma descrição adicional com os valores e as principais tendências:

```
<figure>
  
  <figcaption>
    As vendas foram de 120 unidades em janeiro,
    150 em fevereiro, 180 em março, 210 em abril
    e 250 em maio, indicando crescimento contínuo.
  </figcaption>
</figure>
```

Código 4.7. Acessível: imagem complexa com descrição estendida

Assim, o atributo `alt` fornece uma identificação resumida da imagem, enquanto a descrição estendida disponibiliza informações adicionais necessárias para compreender seu conteúdo.

Tabelas de Dados.

Tabelas usadas para apresentar dados (e não para fins de *layout*) devem expor sua estrutura de forma programática, e não apenas visualmente. O elemento `<caption>` fornece um título acessível para a tabela, enquanto o atributo `scope` em células de cabeçalho (`<th scope="col">` ou `<th scope="row">`) associa cada célula de dado ao seu respectivo cabeçalho de coluna ou linha, permitindo que leitores de tela anunciem esse contexto ao navegar célula a célula [31, 6].

Uma tabela sem cabeçalhos adequadamente definidos dificulta a compreensão da relação entre os dados:

```
<table>
  <tr><td>Ms</td><td>Vendas</td></tr>
  <tr><td>Maro</td><td>R$ 12.000</td></tr>
</table>
```

Código 4.8. Não acessível: tabela sem `caption` nem `scope`

A utilização de `<caption>` e de `scope` permite estruturar semanticamente a tabela e estabelecer a relação entre cabeçalhos e dados:

```
<table>
  <caption>Vendas mensais por periodo</caption>
  <tr><th scope="col">Ms</th><th scope="col">Vendas</th></tr>
  <tr><th scope="row">Maro</th><td>R$ 12.000</td></tr>
</table>
```

Código 4.9. Acessível: `caption` e `scope` associando cabeçalhos

Navegação por Teclado.

Todo elemento interativo deve ser alcançável e operável apenas com teclado (Tab, Shift+Tab, Enter, Espaço), sem “armadilhas de foco” que impeçam o usuário de

sair de um componente. A ordem de tabulação deve seguir a ordem visual/lógica do conteúdo (evitando valores positivos de `tabindex`), o indicador de foco deve permanecer visível, e recomenda-se a inclusão de um link de “pular para o conteúdo principal” no início da página [5, 12].

4.4.3. Recursos de ARIA

Quando os elementos nativos do HTML não são suficientes para descrever um componente de interface – como abas, *modais*, *comboboxes* ou barras de progresso –, a especificação WAI-ARIA (*Accessible Rich Internet Applications*) oferece atributos que comunicam papel, estado e propriedades a recursos de TA [28]. A diferença entre esses dois casos é a de papel *implícito* e papel *explícito*: elementos nativos como `<button>` ou `<nav>` já carregam, por definição da especificação HTML, um papel, um estado e um comportamento de teclado reconhecidos automaticamente por leitores de tela; já um `<div>` não possui papel algum até que este seja atribuído explicitamente via `role`, e mesmo assim o ARIA só descreve *o que* o componente é, sem implementar *como* ele deve se comportar diante do teclado ou do foco [28, 29]. Entre os atributos mais utilizados estão:

Tabela 4.5. Principais atributos ARIA e sua finalidade.

Atributo	Finalidade
<code>role</code>	Define o papel semântico de um elemento (ex.: <code>role="dialog"</code> , <code>role="alert"</code>).
<code>aria-label</code> / <code>aria-labelledby</code>	Fornecem um nome acessível quando não há texto visível suficiente.
<code>aria-describedby</code>	Associa uma descrição adicional a um elemento.
<code>aria-live</code>	Anuncia atualizações dinâmicas de conteúdo (ex.: mensagens de erro carregadas via JavaScript).
<code>aria-expanded</code>	Indica se um componente expansível (menu, acordeão) está aberto ou fechado.
<code>aria-hidden</code>	Oculto um elemento da árvore de acessibilidade, sem afetar sua exibição visual.
<code>aria-current</code>	Indica o item atual dentro de um conjunto (ex.: página ativa em um menu de navegação).

É importante observar a chamada “primeira regra do ARIA”: sempre que existir um elemento HTML nativo com a semântica e o comportamento necessários, ele deve ser preferido a recriar esse comportamento com `<div>/<span>` e atributos ARIA, já que ARIA descreve papéis e estados, mas não implementa comportamento (como resposta a teclado) por si só [28, 29, 6].

4.5. Tecnologia Assistiva e Interação com a Web

O Estatuto da Pessoa com Deficiência (Lei N° 13.146, de 2015) define Tecnologia Assistiva (TA) como:

“(...) produtos, equipamentos, dispositivos, recursos, metodologias, estratégias, práticas e serviços que objetivem promover a funcionalidade, relacionada à atividade e à participação da pessoa com deficiência ou com mobilidade reduzida, visando à sua autonomia, independência, qualidade de vida e inclusão social.” [11]

Alternativamente, a TA pode ser entendida como o conjunto de serviços e recursos que contribuem para oferecer ou ampliar habilidades funcionais de pessoas com deficiência e, consequentemente, facilitar a vida dessas pessoas [23].

No contexto digital e na Web, os recursos de TA funcionam como pontes de mediação entre o usuário e a interface, tornando o formato do conteúdo adequado para o usuário e permitindo diferentes alternativas para a interação com ele [1].

4.5.1. Conceitos e Ferramentas de Tecnologia Assistiva

Na Web, os recursos de TA manifestam-se em diferentes formas de hardware e software para atender a diversas necessidades funcionais. Algumas delas estão listadas a seguir:

- **Leitores de Tela (*Screen Readers*):** Softwares que convertem texto e elementos estruturais de uma interface gráfica em fala sintetizada ou em braille tátil. Exemplos populares incluem o NVDA (*NonVisual Desktop Access*), JAWS e o VoiceOver (nativo em ecossistemas Apple) [26, 35, 34, 1].
- **Softwares de Ampliação de Tela:** Recursos voltados para pessoas com baixa visão que aumentam porções da tela e ajustam o contraste de cores, como o ZoomText [36, 1].
- **Dispositivos de Entrada Alternativos:** Teclados adaptados, acionadores de pressão, sistemas de rastreamento ocular (*eye-tracking*) e navegação por comando de voz, que atendem pessoas com restrições motoras severas [1].
- **Ferramentas de Tradução Automática para Libras:** Tradutores virtuais como o VLibras, que convertem textos em língua portuguesa para a Língua Brasileira de Sinais por meio de avatares tridimensionais [27].

4.5.2. Relação entre Tecnologia Assistiva e Acessibilidade Web

Existe uma relação de interdependência direta entre a Tecnologia Assistiva e a Acessibilidade Web. Enquanto a TA provê o meio técnico para que o usuário interaja com o conteúdo, a Acessibilidade Web determina como o conteúdo Web deve ser desenvolvido para que ele seja utilizável por pessoas com deficiência [1].

Sem a aplicação das diretrizes de acessibilidade recomendadas pelo W3C através das *Web Content Accessibility Guidelines* (WCAG), o desempenho dos recursos de TA é comprometido [5]. Por exemplo, um leitor de tela não consegue descrever o propósito de um botão interativo se o desenvolvedor utilizar apenas uma imagem sem texto alternativo ou sem rótulos estruturais adequados.

4.5.3. Como Recursos de TA Interpretam uma Página Web

A interpretação de uma página por um recurso de TA não ocorre por meio do processamento visual da interface gráfica renderizada na tela, mas sim pela navegação através da estrutura do documento subjacente. O processo ocorre através das seguintes etapas e mecânicas:

1. **Construção do DOM (*Document Object Model*):** Quando o navegador carrega o documento HTML, XML ou SVG, ele monta uma árvore estrutural de elementos. O DOM possui métodos que permitem acesso à árvore, e, consequentemente, que possibilitam a sua modificação [24].
2. **Árvore de Acessibilidade (*Accessibility Tree*):** A partir do DOM, a API do sistema operacional cria uma representação paralela chamada *Árvore de Acessibilidade*. Essa árvore simplifica o código e expõe apenas informações relevantes. Seus objetos apresentam quatro propriedades principais [25]:
 - **Nome (*Name*):** O rótulo legível do elemento (ex.: "Enviar" e "Ler mais").
 - **Descrição (*Description*):** A descrição do elemento, caso se pretenda informar uma descrição para além do nome do objeto (ex.: a descrição de uma tabela pode explicar que tipo de informação está presente no objeto).
 - **Papel (*Role*):** A função do elemento na interface (ex.: *button*, *heading*, *checkbox*).
 - **Estado (*State*):** A condição atual do elemento (ex.: *expanded*, *checked*, *disabled*).

Além dessas propriedades, o objeto também pode informar "o que ele faz". Por exemplo, uma entrada de texto pode receber informações do teclado [25].

3. **Uso de Especificações ARIA:** Quando os elementos HTML nativos não são suficientes para expressar a semântica de componentes dinâmicos (como menus retráteis ou *modais*), utilizam-se os atributos WAI-ARIA (*Accessible Rich Internet Applications*) para enriquecer a árvore de acessibilidade [28].

4.6. Ferramentas de Avaliação de Acessibilidade

A avaliação de acessibilidade consiste no processo de verificar se um site ou aplicação Web atende aos critérios estabelecidos por normas como a WCAG [6], identificando barreiras que possam impedir ou dificultar o uso por pessoas com deficiência. Esse processo pode ser conduzido de diferentes formas, sendo a avaliação automática uma das mais utilizadas por sua rapidez e baixo custo de aplicação. Atualmente, existem diversas ferramentas que auxiliam desenvolvedores a testar a acessibilidade de páginas Web com base nos critérios de sucesso da WCAG, entre as quais se destacam o WAVE, o axe DevTools e o Lighthouse.

Essas ferramentas realizam uma varredura automatizada do código da página e conseguem identificar problemas objetivos e programaticamente verificáveis, como a ausência de texto alternativo em imagens, contrastes de cor insuficientes entre texto e plano de fundo, ausência de rótulos em campos de formulário, uso incorreto de hierarquia de cabeçalhos e elementos interativos que não podem ser acessados via teclado. Um levantamento realizado com o WebAIM Million, que analisa anualmente a acessibilidade de um milhão dos principais sites do mundo, aponta que a grande maioria das páginas analisadas apresenta falhas detectáveis por ferramentas automáticas, evidenciando a utilidade dessas ferramentas para identificar rapidamente problemas técnicos recorrentes [18].

Entretanto, a avaliação automática não substitui a avaliação humana, uma vez que apresenta limitações significativas. Um estudo conduzido pelo Government Digital Service do Reino Unido testou treze ferramentas automáticas de avaliação contra uma página com 142 barreiras de acessibilidade previamente documentadas: a ferramenta com melhor desempenho identificou 40% dos problemas, enquanto as demais apresentaram taxas de detecção ainda menores [17]. De forma semelhante, o relatório da Deque Systems, baseado em mais de 2.000 auditorias, abrangendo mais de 13.000 páginas ou estados de páginas e aproximadamente 300.000 problemas de acessibilidade, identificou que, em média, 57,38% dos problemas registrados foram detectados por testes automatizados. Entretanto, a análise também identificou problemas automatizáveis em apenas 16 dos 50 critérios de sucesso da WCAG 2.1 de níveis A e AA, evidenciando que determinados aspectos da acessibilidade ainda requerem avaliação manual. Entre os critérios analisados, problemas relacionados à ordem de foco, navegação por teclado e visibilidade do foco apresentaram predominância de detecção manual [16].

Por esse motivo, a literatura da área recomenda uma combinação entre avaliação automática e avaliação manual, sendo a primeira utilizada para identificar rapidamente problemas técnicos objetivos, e a segunda, conduzida por especialistas ou pelas próprias pessoas com deficiência, responsável por avaliar aspectos qualitativos e de experiência que nenhuma ferramenta é capaz de mensurar sozinha.

4.6.1. WAVE Web Accessibility Evaluation Tools

O WAVE consiste em um conjunto de ferramentas de avaliação que objetivam aumentar a acessibilidade de conteúdo Web. Entre as suas funções, destacam-se a identificação de erros de acessibilidade e de conformidade com as Diretrizes de Acessibilidade para Conteúdo Web (WCAG), bem como a facilitação de avaliação humana do conteúdo Web. O WAVE está disponível através de extensões de navegador e de Interfaces de Programação de Aplicações (APIs). As extensões de navegador estão disponíveis para Chrome, Firefox e Edge, e permitem testar acessibilidade diretamente no navegador. Já as APIs possibilitam a coleta de dados de teste de acessibilidade em múltiplas páginas Web. Adicionalmente, elas possuem mecanismos de teste que podem ser integrados à infraestrutura do sistema. Ao final do processo, o WAVE retorna um relatório de avaliação de Impacto de Acessibilidade (AIM), que fornece dados dos testes, pontuação AIM e resultados de testes manuais especializados. A pontuação AIM visa medir o impacto no usuário comparativamente, em relação páginas Web em geral e considerando regras e diretrizes definidas por testadores humanos. [22].

4.6.2. axe DevTools

O axe DevTools é um verificador de acessibilidade no Google Chrome direcionado para designers, desenvolvedores e testadores. A ferramenta permite a avaliação de acessibilidade em soluções Web e *mobile*. O axe DevTools foi a primeira ferramenta de acessibilidade a usar automação e Inteligência Artificial (IA). Há três ferramentas principais: (i) axe MCP Server (ou Servidor de Modelo de Contexto e Protocolo axe): conecta o motor de testes de acessibilidade Web a Ambientes de Desenvolvimento (IDEs), auxiliando no desenvolvimento do código; (ii) Assistente axe: um chatbot de IA, treinado com uma base robusta de conhecimento em acessibilidade; e (iii) Extensão Axe DevTools: utiliza

Testes Guiados Inteligentes (IGTs) baseados em IA dentro da extensão do navegador para identificar erros de acessibilidade que a automação não foi capaz de identificar. Além disso, a ferramenta também oferece auditorias realizadas por especialistas que avaliam a acessibilidade e padrões de conformidade do sistema. [21]

4.6.3. Lighthouse

O Lighthouse é uma ferramenta de código aberto que objetiva a melhoria da qualidade de páginas Web automaticamente. Para utilizá-lo, é possível fornecer uma URL à ferramenta - através das ferramentas do desenvolvedor do Chrome, da linha de comando, de um módulo Node ou de uma interface da Web - para a realização da auditoria. O Lighthouse, em seguida, executa uma série de auditorias na página Web e, a partir dos resultados obtidos, gera um relatório sobre o desempenho dessa página. Para cada auditoria, há uma explicação de sua importância e, se necessário, do que se precisa fazer para corrigi-la. O Lighthouse apresenta duas ferramentas que o personalizam de acordo com as necessidades do desenvolvedor: (i) pacotes de pilhas: o Lighthouse consegue detectar em qual plataforma o site foi criado e apresenta recomendações específicas, estabelecidas e definidas por especialistas, com base na pilha utilizada; e (ii) plug-ins: permitem a adição de novas categorias no relatório através da implementação de módulos de nó. [20]

4.7. Prática de Criação com Personas

A criação de personas é uma técnica consolidada do design centrado no usuário, introduzida por Cooper [43] e posteriormente sistematizada como método de pesquisa e comunicação de requisitos por Pruitt e Grudin [44]. Uma persona é um indivíduo fictício, porém realista, que representa as características, necessidades, comportamentos, objetivos e expectativas de um grupo de usuários-alvo. Sua principal contribuição não é descritiva, mas comunicativa: personas traduzem dados de pesquisa em uma linguagem narrativa que as equipes de desenvolvimento conseguem incorporar às decisões cotidianas de projeto, funcionando como substituto viável quando o acesso direto aos usuários reais é inviável ou o recrutamento é particularmente difícil [55].

No contexto da acessibilidade, essa técnica assume um papel específico. Personas que representam pessoas com deficiência são utilizadas sobretudo nas fases de pesquisa e design, com o objetivo de antecipar barreiras que, de outro modo, só seriam identificadas após a implementação. A ausência desse tipo de representação faz com que produtos e serviços digitais sejam projetados sem considerar barreiras que afetam pessoas com deficiência, ao passo que sua inclusão nas etapas de planejamento e design permite que a acessibilidade seja tratada de forma proativa, e não reativa [55]. Personas também podem ser empregadas como instrumento de conscientização sobre desenho universal e sobre as necessidades de acessibilidade desse público, incluindo o uso de recursos de TA [45].

Um aspecto frequentemente negligenciado na construção dessas personas é a caracterização do recurso de TA utilizado e do nível de proficiência do usuário com ela. Conforme discutido na Seção 4.5, a experiência de navegação de um usuário iniciante em leitores de tela difere substancialmente da de um usuário experiente na mesma página. Uma persona que registre apenas o diagnóstico, sem descrever o contexto de uso, o dispositivo, a tecnologia assistiva e o grau de familiaridade com ela, tende a produzir

análises de barreiras genéricas e pouco acionáveis.

4.7.1. Limitações e riscos da técnica

A literatura de Interação Humano-Computador documenta que personas são uma ferramenta de construção de empatia controversa, que pode produzir efeitos prejudiciais ao design de produtos acessíveis quando mal construída ou empregada de forma inadequada [53]. Quatro riscos merecem atenção explícita.

O primeiro diz respeito à **natureza estática do artefato**. Edwards, Sum e Brannan [47], a partir de um estudo qualitativo com nove participantes que possuíam múltiplas deficiências ou uma deficiência associada a outras identidades marginalizadas, apontam que personas são tipicamente documentos fixos, enquanto as identidades de pessoas com deficiência são dinâmicas e em constante transformação. Como consequência, personas não se adaptam nem respondem como uma pessoa real durante o processo de design, o que pode reforçar estereótipos em vez de dissolvê-los.

O segundo é o **problema da cobertura**. O número de personas empregado em um projeto é inerentemente limitado, o que torna impossível representar adequadamente a diversidade e a complexidade das experiências vividas [47]. Uma persona “usuário cego” não representa a população de pessoas cegas; representa, no máximo, um recorte específico dela.

O terceiro risco refere-se a atividades de **simulação de deficiência**, frequentemente associadas a dinâmicas de sensibilização e por vezes utilizadas como complemento à construção de personas. Nario-Redmond, Gospodinov e Cobb [49] avaliaram experimentalmente esse tipo de atividade e verificaram que as simulações deixaram os participantes mais confusos, constrangidos, desamparados e mais vulneráveis à percepção de que poderiam se tornar deficientes. Embora a preocupação empática tenha aumentado, as atitudes relativas à interação com pessoas com deficiência não melhoraram; em um dos experimentos, os participantes julgaram-se menos competentes, expressaram mais pena e maior desconforto de interação após a simulação. Os autores concluem que simular deficiências promove sofrimento e falha em melhorar atitudes, minando esforços de inclusão mesmo quando os próprios participantes relatam ter compreendido melhor a experiência da deficiência. Por essa razão, este minicurso não adota dinâmicas de simulação como recurso pedagógico.

O quarto risco é de ordem conceitual. Bennett e Rosner [48] analisam criticamente a fase de “empatia” do processo de design centrado no ser humano e argumentam que atividades de construção de empatia podem, paradoxalmente, afastar pessoas com deficiência dos processos nos quais se pretendia incluí-las, ao substituir a participação efetiva pela representação feita por terceiros. A implicação prática é direta: personas informam decisões de projeto, mas não substituem a participação de usuários reais.

4.7.2. Estratégias de mitigação

Três estratégias respondem a essas limitações.

A primeira é a **co-criação**. Neate et al. [46] investigaram o uso de personas co-criadas com pessoas com doença de Parkinson, demência e afasia, e observaram que a

técnica incentivou usuários com necessidades diversas a se engajarem no co-design, além de produzir benefícios adicionais de empoderamento dentro de um processo de design mais acessível. No estudo, as personas não eram fixas: podiam ser modificadas à medida que o escopo do projeto se tornava mais definido. Os autores registram ainda que, ao trabalhar apenas com os co-designers presentes, a equipe fica limitada à experiência existente na sala, e as personas co-criadas servem justamente para ampliar esse alcance.

A segunda é a **fundamentação em dados empíricos**. Recomenda-se que personas de acessibilidade partam de pesquisa real com usuários, seja por meio de entrevistas com pessoas com deficiência, seja pelo aproveitamento de evidências provenientes de estudos de acessibilidade, testes de usabilidade e organizações de defesa de direitos [55]. Um exemplo direto de aplicação dessa estratégia em contexto próximo ao deste minicurso é o de Heitmeier et al. [53], que descrevem uma abordagem de co-design aninhado, incluindo pessoas com deficiência em múltiplas etapas e funções do processo, aplicada especificamente à melhoria de materiais didáticos de acessibilidade digital em instituições de ensino superior, um contexto educacional diretamente análogo ao deste minicurso.

A terceira, aplicável quando não há acesso direto a usuários, é o **apoio em guias de recomendações derivados de revisão sistemática da literatura**. No contexto brasileiro, o GAIA constitui exemplo relevante para o perfil cognitivo: Britto e Pizzolato [50] derivaram suas recomendações de uma revisão de 17 trabalhos publicados entre 2005 e 2015, identificando inicialmente 107 recomendações agrupadas em 10 categorias por meio de diagrama de afinidades, posteriormente normalizadas em um conjunto final de 28 recomendações únicas voltadas ao Transtorno do Espectro Autista.

4.7.3. Exemplo de Aplicação com o Percurso de Barreira

Apresenta-se um cenário de avaliação com um uso prático de Percurso de Barreira (*Barrier Walkthrough*). Este exemplo demonstra como o uso de personas apoia o desenvolvimento de interfaces de usuário, configurando-se como uma ferramenta de inspeção para o desenvolvedor.

Considere a persona apresentada na Figura 4.1, baseada no modelo de [62]. Lucas, persona desenvolvida para este exemplo, é uma pessoa cega desde a infância. Ele é desenvolvedor de *software*, apresentando facilidade com tecnologia. Lucas utiliza os seguintes recursos de TA: leitor de tela, teclado e linha Braille, sendo proficiente em Braille. As principais formas de interação de Lucas com soluções *web* são: auditiva, teclado e toque. Ele apresenta um perfil independente, paciente, adaptável, objetivo e um pouco cauteloso. As suas necessidades específicas são compatibilidade total com leitor de tela e navegação fluida por teclado.

Tarefa Avaliada: O objetivo é avaliar o fluxo em um **aplicativo de e-commerce de eletrônicos**, em que a persona, Lucas, precisa encontrar e realizar a compra de um **combo de mouse e teclado** para uso residencial. A avaliação realiza o percurso do usuário e confronta as exigências do sistema com as capacidades do usuário.

Durante a execução do percurso, a interface é analisada considerando as características, necessidades e formas de interação descritas na persona de Lucas. Sempre que o sistema exigir uma habilidade incompatível com essas características, uma barreira de



Figura 4.1. Exemplo de persona PCD.

acesso é identificada, registrada e catalogada.

Definiu-se um percurso com quatro passos sequenciais da jornada de compra no *e-commerce*. A análise considera as ações esperadas de um usuário que navega pela interface utilizando um leitor de tela e teclado, conforme as características definidas na persona.

- Passo 1 — Seleção e Validação do Produto:** Lucas navega até a página do combo de *mouse* e teclado. Ele tenta verificar se o item está disponível para o seu endereço. A disponibilidade do produto, porém, é indicada exclusivamente por meio de uma informação visual baseada em cores (a bolha colorida), sem uma alternativa textual equivalente acessível ao leitor de tela. Como essa informação não é comunicada pelo leitor de tela, Lucas assume que há um erro na página ou que o produto pode estar indisponível.
- Passo 2 — Leitura dos Detalhes Técnicos:** Ao tentar entender as especificações do teclado (*layout* das teclas, conexões), Lucas busca a descrição nas imagens do produto. Como as imagens não possuem texto alternativo adequado, o leitor de

tela informa apenas que se trata de uma imagem, sem fornecer detalhes sobre seu conteúdo. Ele precisa confiar apenas no título curto do produto, sem conseguir validar os detalhes técnicos necessários para o seu uso como desenvolvedor.

3. **Passo 3 — Revisão no Carrinho de Compras:** Após adicionar o item, Lucas acessa o carrinho para conferir o valor. Ele decide ajustar o pedido, mas o leitor de telas anuncia apenas a palavra genérica “botão” repetidas vezes. Trata-se do botão de exclusão, representado apenas pelo ícone de uma lixeira, sem um nome acessível associado. Sem saber qual botão remove o item ou avança para o pagamento, o fluxo é interrompido por falta de semântica.
4. **Passo 4 — Tentativa de *Checkout* e Cadastro:** Ao tentar digitar o endereço, Lucas espera que o aplicativo auxilie sua digitação de forma ágil. No entanto, a ausência do uso adequado do atributo `autocomplete` nos campos do formulário o obriga a interagir campo por campo de forma lenta e manual, resultando em uma quebra na eficiência do sistema.

A Tabela 4.6 ilustra esse caminho prático, conectando a persona às falhas de usabilidade e acessibilidade encontradas na interface do *e-commerce*. Para simplificar o preenchimento do modelo de persona (Figura 4.1), as barreiras associadas à navegação por teclado foram agrupadas sob a condição de “sistemas que não permitem navegação por teclado”.

Tabela 4.6. Matriz de Tradução de Barreiras na Compra de Eletrônicos.

Atributo da Persona	Barreira e Impacto Gerado (Interface)
Ausência de Interação Visual	Nível de estoque definido por cor: Na tela de informações do produto (combo de <i>mouse</i> e teclado), a disponibilidade do item é indicada apenas por um círculo colorido, sem texto alternativo ou legenda. Como a informação é apresentada exclusivamente de forma visual, ela não é comunicada pelo leitor de telas, impossibilitando que o usuário saiba qual é a disponibilidade do produto.
Perfil Tecnológico / Ágil	Formulário sem marcação semântica e sem preenchimento automático: Ao preencher os dados de entrega, os campos não possuem identificação semântica adequada e não utilizam corretamente o atributo <code>autocomplete</code> (por exemplo, no campo de CEP). Isso impede que o navegador ofereça preenchimento automático, além de forçar o usuário a digitar manualmente dados repetitivos de forma sequencial e lenta. Como consequência, a eficiência do sistema é reduzida consideravelmente.
Imagens do produto	Imagens sem texto alternativo: As imagens dos produtos (<i>mouse</i> e teclado) não possuem texto alternativo. Isso faz com que essas imagens se tornem “invisíveis” para o leitor de telas e prejudica a experiência do usuário.
Acessibilidade através do teclado	Navegação através do teclado não é possível: Lucas utiliza leitor de tela associado a teclado. Isso faz com que, para que a sua experiência como usuário seja positiva, o sistema precise ser inteiramente navegável por teclado. Além disso, é necessário que exista ordem lógica na página, foco visível e a garantia de que o usuário consegue entrar e sair de qualquer componente apenas com o teclado.
Uso de Leitor de Tela	Botão de remover item sem rótulo: No carrinho de compras, o botão para excluir um item é representado apenas pelo ícone de uma lixeira, sem um nome acessível associado (por exemplo, via atributo <code>aria-label</code> ou texto equivalente). O leitor de telas reproduz apenas a palavra genérica “botão”, o que impede a continuidade do fluxo de pagamento com segurança.

Ao final do levantamento de barreiras, as dificuldades mapeadas evoluem de falhas técnicas abstratas para a compreensão de seu impacto real na experiência de compra de um usuário. Isso permite que a equipe de desenvolvimento priorize correções de *UI/UX* com o objetivo de eliminar os bloqueios que impedem a inclusão de diferentes tipos de usuário.

4.8. Limitações da WCAG e Requisitos Além da Conformidade

A Seção 4.6 estabeleceu que ferramentas automatizadas identificam apenas parte dos problemas de acessibilidade e que a avaliação manual é necessária para complementá-las. Esta seção avança um argumento distinto e mais amplo: mesmo a conformidade plena e corretamente verificada com as diretrizes não garante, por si só, uma experiência acessível. A distância entre conformidade técnica e experiência real de uso constitui um objeto de investigação próprio, com evidência empírica acumulada há quase duas décadas.

4.8.1. Duas definições de acessibilidade

Essa distância tem origem em uma ambiguidade conceitual explicitada por Rømen e Svanæs [39]. Os autores observam que a própria WCAG define acessibilidade de forma minimalista: “content is accessible when it may be used by someone with a disability”, ou

seja, basta que seja *possível* a um usuário com deficiência utilizar o conteúdo. Essa definição, que os autores chamam de *WCAG-accessibility*, contrasta com a definição adotada pela norma ISO 9241-171:2008, que trata acessibilidade como usabilidade para pessoas com deficiência, isto é, a capacidade de “achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use”. Essa segunda definição, chamada pelos autores de *ISO-accessibility*, é substancialmente mais exigente: um site pode “poder ser usado” por uma pessoa com deficiência (satisfazendo a WCAG) e, ainda assim, apresentar problemas de usabilidade severos que impeçam esse mesmo usuário de atingir seus objetivos concretos [39]. É precisamente essa diferença, entre poder ser usado e poder ser usado com eficácia, eficiência e satisfação, que a evidência empírica a seguir documenta.

4.8.2. Evidências empíricas de cobertura parcial

Três estudos são particularmente relevantes para dimensionar essa distância.

Petrie e Kheir [37] investigaram a relação entre acessibilidade e usabilidade por meio de testes com doze participantes, seis pessoas cegas e seis pessoas sem deficiência visual, como grupo de controle, em dois sites comerciais de telefonia móvel (Orange e T-Mobile, Reino Unido). Os problemas encontrados pelos dois grupos formaram conjuntos que se intersectam em aproximadamente 14% quando os dados dos dois sites são combinados (17% no site Orange e 10,6% no T-Mobile) [37]. Mais significativo do que o percentual de sobreposição é o segundo achado do estudo: a correlação entre a severidade atribuída pelos participantes cegos aos problemas e a classificação de importância fornecida pelas diretrizes de usabilidade do *U.S. Department of Health and Human Services* (HHS) foi *negativa* e estatisticamente significativa ($r = -0,244$ a $-0,273$, $p \leq 0,05$), ou seja, problemas que as diretrizes classificam como mais importantes tenderam a ser avaliados pelos próprios usuários como *menos* severos. Não houve, tampouco, concordância significativa entre as avaliações de participantes ou pesquisadores e os níveis de prioridade da WCAG [37].

Rømen e Svanæs [39] conduziram testes de usabilidade controlados com sete pessoas com deficiência (três com deficiência visual, dois com deficiência motora e dois com dislexia) e um grupo de controle de seis pessoas, em dois sites de administrações municipais norueguesas. Do total de 80 problemas distintos identificados, 47 (58,75%) foram experimentados apenas por pessoas com deficiência. Desses 47 problemas de acessibilidade, apenas 13 (27%) poderiam ter sido identificados pela aplicação da WCAG 1.0, e apenas 15 (32%) pela WCAG 2.0, uma melhoria marginal de apenas 5 pontos percentuais entre versões. Combinando as duas versões da norma, a cobertura sobe para 38%. Compensando o número reduzido de participantes por meio de testes de confiança, os valores permaneceram baixos: há 95% de confiança de que a conformidade com a WCAG 1.0 fique abaixo de 42%, com a WCAG 2.0 abaixo de 49%, e com as duas versões combinadas abaixo de 50% [39]. O dado mais contundente do estudo, porém, refere-se à severidade: dos seis problemas classificados como críticos, aqueles que impedem completamente a realização de uma tarefa, apenas um seria identificado pela WCAG 1.0 e apenas dois pela WCAG 2.0; mesmo combinando as duas versões, três dos seis problemas críticos permaneceriam sem cobertura normativa [39]. Os autores concluem que a aplicação das diretrizes da WAI não é suficiente para garantir a acessibilidade de um site e recomendam

que versões futuras sejam fundamentadas e validadas empiricamente, além de defenderem a ampliação da definição de acessibilidade adotada pela WAI de modo a incorporar a noção de ISO-accessibility descrita acima.

O estudo de maior escala é o de Power, Freire, Petrie e Swallow [38], que analisou os problemas encontrados por 32 pessoas cegas em avaliações baseadas em tarefas realizadas em 16 sites, gerando 1.383 instâncias de problemas. Apenas 50,4% dessas instâncias eram cobertas por critérios de sucesso (*Success Criteria*, SC) da WCAG 2.0. Mais relevante para o argumento desta seção é o segundo resultado: dos problemas cobertos pela norma, 16,7% — equivalentes a 8,4% de todos os problemas relatados — corresponderam a casos em que a técnica recomendada pela WCAG 2.0 *havia sido implementada corretamente* e, ainda assim, o problema persistiu para o usuário [38]. Um exemplo concreto ilustra o mecanismo: na categoria “destino do link não é claro”, que respondeu por 117 problemas (8,5% do total), 35% dos casos ocorreram em páginas que haviam implementado corretamente o critério de sucesso 2.4.4 (*Link Purpose*), e os usuários ainda assim não conseguiam determinar para onde o link levava. Os próprios autores concluem que “as técnicas suficientes para este critério [...] não são, de fato, suficientes” [38]. Os autores identificam ainda seis categorias de problemas com cobertura *nula* (0%) pela WCAG 2.0, que juntas correspondem a 18,5% de todos os problemas relatados: conteúdo encontrado onde não era esperado pelo usuário, conteúdo não encontrado onde era esperado, páginas lentas para carregar, ausência de alternativa a formatos de documento (como PDF), arquitetura de informação excessivamente complexa, e links quebrados [38]. Com base nesses achados, os autores defendem a transição de uma abordagem baseada em problemas para uma abordagem baseada em princípios de design para a acessibilidade Web.

4.8.3. Escopo das diretrizes: um contraponto necessário

Esses resultados não devem ser interpretados como evidência de falha das diretrizes, e sim como delimitação de seu escopo. Os próprios Rømen e Svanæs reconhecem essa defesa possível da WCAG: “in defence of WCAG one could argue that WCAG in its very definition of accessibility never promised that conformance to its guidelines would result in usability for disabled users [...], it only promised that the resulting sites ‘may be used by someone with a disability’ ” [39]. De modo semelhante, o próprio grupo de trabalho da WCAG registrou uma resposta ao estudo de Power et al., argumentando que parte dos problemas relatados constitui questões de usabilidade apresentadas como questões de acessibilidade, e que há concordância entre revisores de que os problemas apontados como não cobertos pela WCAG 2.0 não seriam algo que diretrizes de acessibilidade pudessem ou deveriam abordar especificamente [56].

A questão pertinente, portanto, não é se a WCAG falha, mas onde termina o escopo legítimo de uma norma verificável e onde começa a responsabilidade de projeto. Essa reformulação, entre WCAG-accessibility e ISO-accessibility, é o que sustenta o restante desta seção.

4.8.4. A testabilidade como origem estrutural das lacunas

A principal restrição de escopo da WCAG é metodológica e está documentada pelo próprio W3C. Todo critério de sucesso da WCAG precisa ser testável, de modo que autores possam determinar objetivamente se estão ou não em conformidade, e essa exigência é apontada pela própria força-tarefa de acessibilidade cognitiva como uma das razões pelas quais determinadas necessidades de usuários não foram incorporadas às diretrizes [42].

O efeito dessa restrição é assimétrico entre perfis de deficiência. Necessidades de pessoas com deficiência visual, auditiva ou motora tendem a se traduzir com relativa facilidade em condições binárias verificáveis no código; necessidades relacionadas a compreensão, memória, atenção e carga cognitiva, não. Como resultado, apenas três dos novos critérios de sucesso introduzidos na WCAG 2.1 podem ser diretamente atribuídos ao trabalho da *Cognitive and Learning Disabilities Accessibility Task Force* (COGA): os critérios 1.3.5 (Identify Input Purpose), 1.3.6 (Identify Purpose) e 2.2.6 (Timeouts) [57, 58, 59].

O reconhecimento mais explícito dessa lacuna está no documento *Making Content Usable for People with Cognitive and Learning Disabilities* [40], produzido pelo próprio W3C. O documento declara que suas orientações e padrões constituem material suplementar aos requisitos da WCAG, que segui-los não é exigido para conformidade, e que eles endereçam barreiras de acessibilidade não contempladas na WCAG 2.x normativa. Existe, portanto, um corpo consolidado de orientação produzido pelo próprio organismo normatizador que permanece deliberadamente fora do escopo da conformidade. Guias específicos por perfil, como o GAIA [50] no contexto brasileiro, cumprem função análoga.

4.8.5. O modelo de conformidade e sua revisão em curso

Uma segunda limitação decorre do modelo de conformidade adotado pela WCAG 2.x, que é binário e organizado por página: cada critério de sucesso é aprovado ou reprovado, sem gradação intermediária que reflita a severidade do impacto sobre o usuário ou o progresso alcançado.

Essa limitação é reconhecida pelo grupo de trabalho responsável pela próxima geração das diretrizes. O documento explicativo da WCAG 3.0 registra que o grupo tem explorado classificações de severidade que possam contribuir para pontuação e priorização, bem como classificações adjetivais que permitam que resultados de teste vão além de “aprovado” ou “reprovado”, indicando progresso em direção a uma meta ou sua superação, em uma escala exemplificada como *Fail*, *Progress*, *Pass*, *Better* e *Exceptional* [41]. Cabe ressaltar que a WCAG 3.0 encontra-se em estágio de rascunho de trabalho (*Working Draft*) e não constitui norma vigente nem referência para conformidade legal; a WCAG 2.2 [6] permanece a recomendação em vigor.

4.8.6. Conformidade performática: o caso dos *overlays*

A distância entre conformidade declarada e acessibilidade efetiva manifesta-se de forma particularmente evidente no caso dos *overlays* de acessibilidade, soluções de terceiros, tipicamente em JavaScript, que prometem corrigir automaticamente barreiras de acessibilidade a partir da inserção de um único trecho de código na página, frequentemente comercializadas também como proteção contra litígio.

A evidência disponível é consistentemente desfavorável. Na pesquisa da WebAIM com 758 profissionais de acessibilidade, conduzida em janeiro de 2021, 67% dos respondentes classificaram essas ferramentas como pouco ou nada eficazes; entre respondentes com deficiência, essa proporção sobe para 72%, e apenas 2,4% as consideraram muito eficazes [51]. Em estudo revisado por pares que combinou questionário e entrevistas semiestruturadas com pessoas cegas e com baixa visão, Makati, Tigwell e Shinohara [52] relatam que a maioria dos participantes considerou os overlays inúteis, e que eles frequentemente agravavam os problemas de acessibilidade em vez de resolvê-los; os participantes relataram com frequência que os overlays entravam em conflito com suas próprias recursos de TA, reduzindo a funcionalidade e aumentando a frustração. Os autores defendem o abandono de correções superficiais em favor de práticas de design centrado no usuário que incorporem a acessibilidade desde a concepção do produto [52].

O caso é ilustrativo porque explicita o risco de tratar a acessibilidade como problema de conformidade a ser resolvido no final do ciclo de desenvolvimento, e não como requisito de qualidade incorporado desde a concepção.

4.8.7. Necessidades de acesso conflitantes

Uma limitação final não é atribuível à WCAG nem a qualquer conjunto de diretrizes, mas à natureza do problema. Lidar com necessidades de acesso variadas e, por vezes, conflitantes tornou-se uma consideração central no design para acessibilidade: quando um recurso beneficia algumas pessoas ao mesmo tempo em que restringe outras, não existe critério normativo capaz de resolver o impasse. Tang e Piper [54], a partir de um estudo de campo de seis meses com uma organização sem fins lucrativos de composição mista de usuários, argumentam pela reformulação dessa questão: em vez de tratar necessidades conflitantes como problemas técnicos a serem eliminados pelo design, os autores defendem repensar esses “conflitos” como locais-chave para revelar estruturas de poder e criar oportunidades de responsabilização e reparação [54].

Esse é precisamente o tipo de decisão que a atividade com personas descrita na Seção 4.7 procura tornar visível: há escolhas de projeto que exigem deliberação e priorização explícitas, e que nenhuma lista de verificação pode substituir.

4.8.8. Síntese

A discussão apresentada não invalida as diretrizes existentes, tampouco autoriza sua relativização. Considerando o quadro normativo brasileiro discutido na Seção 4.2, em particular o artigo 63 da Lei Brasileira de Inclusão [11] e a ABNT NBR 17225 [13], a conformidade constitui obrigação legal e piso mínimo. O que a evidência empírica indica é que ela é condição necessária, mas não suficiente, a distinção entre WCAG-accessibility e ISO-accessibility de Rømen e Svanæs [39] sintetiza precisamente esse limite.

Quatro implicações práticas decorrem dessa discussão: (i) a conformidade deve ser tratada como ponto de partida, e não como objetivo final; (ii) a avaliação deve ser conduzida em camadas, combinando verificação automatizada, avaliação manual especializada e testes com usuários reais; (iii) orientações suplementares específicas por perfil, como as produzidas pela COGA [40] e pelo GAIA [50], devem complementar a norma naquilo que ela deliberadamente não cobre; e (iv) pessoas com deficiência devem partici-

par do processo de desenvolvimento, e não apenas de sua etapa final de validação.

4.9. Conclusão

Este capítulo percorreu, do início ao fim, o caminho que a acessibilidade Web trilhou até se tornar o que é hoje: partiu das primeiras discussões dos anos 1970 e 1980 sobre computação pessoal, passou pela criação do W3C e da WAI nos anos 1990, atravessou décadas de avanços legislativos no Brasil até chegar ao artigo 63 da Lei Brasileira de Inclusão e à recém publicada ABNT NBR 17225, e desembocou nas diretrizes técnicas da WCAG, no HTML semântico, nos atributos ARIA, nas recursos de TA e nas ferramentas automatizadas de avaliação apresentadas ao longo do texto. Esse percurso mostra que a acessibilidade digital não é um tema recente nem um modismo passageiro, mas o resultado de décadas de construção coletiva, técnica e legal, em direção a uma Web mais justa.

Ainda assim, todo esse arcabouço de normas e recursos só ganha sentido pleno quando encontra o usuário real, e é exatamente esse encontro que a criação de personas proporciona. Ao dar rosto, contexto e tecnologia assistiva a critérios que, isolados, permanecem abstratos, as personas transformam diretrizes técnicas em decisões concretas de projeto e permitem enxergar, antes mesmo da implementação, barreiras que de outra forma só apareceriam depois que o produto já estivesse no ar, quando corrigi-las custa mais caro e afeta mais pessoas. Por isso, entre tudo o que foi discutido neste capítulo, a construção e a análise de personas com diferentes perfis de deficiência funcionam como o verdadeiro eixo central: é o ponto em que a norma deixa de ser checklist e passa a ser empatia orientada por evidência.

Ao mesmo tempo, é preciso reforçar que conformidade técnica não é sinônimo de acessibilidade real. Passar em uma ferramenta automatizada, atingir o nível AA da WCAG ou seguir à risca uma norma como a NBR 17225 garante um piso mínimo, necessário e legalmente exigido, mas não garante, por si só, que a pessoa com deficiência consiga usar o site com autonomia, eficiência e satisfação. A distância entre o que a norma exige e o que o usuário de fato vive é o espaço que caberá sempre às equipes de desenvolvimento e design preencherem, e é justamente aí que a escuta de pessoas reais, a construção cuidadosa de personas e a avaliação manual complementar às ferramentas automáticas se tornam indispensáveis.

Por fim, cabe uma reflexão dirigida às próprias empresas e organizações que constroem produtos digitais. A acessibilidade ainda costuma ser tratada como item secundário, adicionado às pressas no fim do desenvolvimento ou terceirizado a soluções superficiais que prometem resolver tudo com uma linha de código, quando na verdade deveria ser incorporada desde a concepção do produto, como requisito de qualidade e não como obrigação a ser cumprida no último momento. Investir em acessibilidade não é apenas questão de conformidade legal ou de evitar litígios: é reconhecer que milhões de pessoas dependem de experiências digitais bem construídas para estudar, trabalhar, se comunicar e exercer sua autonomia. Quanto mais cedo as empresas entenderem que acessibilidade é parte do produto e não um adicional, mais perto a Web estará de cumprir sua promessa original: ser, de fato, para todos.

Agradecimentos

Este trabalho contou com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), por meio de bolsa de mestrado, processo nº 88887.371070/2026-00.

Referências

- [1] W3C (1997) “Web Accessibility Initiative (WAI)”, World Wide Web Consortium, <https://www.w3.org/WAI/>, acesso em: 4 set. 2026.
- [2] UNITED STATES CONGRESS (1998) “Section 508 of the Rehabilitation Act”, 29 U.S.C. § 794d, Washington, D.C.
- [3] CHISHOLM, W.; VANDERHEIDEN, G.; JACOBS, I. (1999) “Web Content Accessibility Guidelines (WCAG) 1.0”, W3C Recommendation, 5 May, <https://www.w3.org/TR/WAI-WEBCONTENT/>, acesso em: 3 set. 2026.
- [4] CALDWELL, B.; COOPER, M.; GUARINO REID, L.; VANDERHEIDEN, G. (2008) “Web Content Accessibility Guidelines (WCAG) 2.0”, W3C Recommendation, 11 December, <https://www.w3.org/TR/WCAG20/>, acesso em: 3 set. 2026.
- [5] KIRKPATRICK, A.; O CONNOR, J.; CAMPBELL, A.; COOPER, M. (2018) “Web Content Accessibility Guidelines (WCAG) 2.1”, W3C Recommendation, 5 June, <https://www.w3.org/TR/WCAG21/>, acesso em: 3 set. 2026.
- [6] KIRKPATRICK, A.; O CONNOR, J.; CAMPBELL, A.; COOPER, M. (2023) “Web Content Accessibility Guidelines (WCAG) 2.2”, W3C Recommendation, 5 October, <https://www.w3.org/TR/WCAG22/>, acesso em: 3 set. 2026.
- [7] BRASIL (2000) “Lei nº 10.098, de 19 de dezembro de 2000”, Diário Oficial da União, Brasília.
- [8] BRASIL (2004) “Decreto nº 5.296, de 2 de dezembro de 2004”, Diário Oficial da União, Brasília.
- [9] BRASIL (2009) “Decreto nº 6.949, de 25 de agosto de 2009”, Diário Oficial da União, Brasília.
- [10] BRASIL (2012) “Decreto nº 7.724, de 16 de maio de 2012”, Diário Oficial da União, Brasília.
- [11] BRASIL (2015) “Lei nº 13.146, de 6 de julho de 2015 (Lei Brasileira de Inclusão)”, Diário Oficial da União, Brasília.
- [12] BRASIL. Ministério do Planejamento, Desenvolvimento e Gestão (2014) “Modelo de Acessibilidade em Governo Eletrônico (eMAG), versão 3.1”, Departamento de Governo Eletrônico, Brasília.
- [13] ABNT (2025) “NBR 17225: Acessibilidade em conteúdos e aplicações web”, Associação Brasileira de Normas Técnicas, Rio de Janeiro.

- [14] INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE) (2023) “Pessoas com deficiência têm menor acesso à educação, ao trabalho e à renda”, Agência de Notícias, 7 jul., <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/37317-pessoas-com-deficiencia-tem-menor-acesso-a-educacao-ao-trabalho-e-a-renda>, acesso em: 5 set. 2026.
- [15] MORO, P.; HENRIQUES, M. (2025) “Falta de acessibilidade digital ainda é desafio para pessoas com deficiência: pesquisa da UFSM investiga consequências da falta de acessibilidade em veículos de comunicação”, Agência de Notícias UFSM, 7 ago., <https://www.ufsm.br/2025/08/07/falta-de-acessibilidade-digital-ainda-e-desafio-para-pessoas-com-deficiencia>, acesso em: 5 set. 2026.
- [16] DEQUE SYSTEMS (2021) “The Automated Accessibility Coverage Report”, <https://www.deque.com/automated-accessibility-coverage-report/>, acesso em: 2 set. 2026.
- [17] GOVERNMENT DIGITAL SERVICE (GDS) (2018) “Accessibility tools audit results”, GDS Accessibility Team, United Kingdom, <https://alphagov.github.io/accessibility-tool-audit/>, acesso em: 3 set. 2026.
- [18] WEBAIM (2026) “The WebAIM Million: An Annual Accessibility Analysis of the Top 1,000,000 Home Pages”, <https://webaim.org/projects/million/>, acesso em: 4 set. 2026.
- [19] WEBAIM (2024) “Screen Reader User Survey”, <https://webaim.org/projects/screenreadersurvey/>, acesso em: 5 set. 2026.
- [20] Lighthouse (2026) “Introduction to Lighthouse”, <https://developer.chrome.com/docs/lighthouse/overview>, acesso em: 4 set. 2026.
- [21] Axe DevTools (2026) “Axe DevTools - The ultimate digital accessibility testing solution”, <https://www.deque.com/axe/devtools/>, acesso em: 4 set. 2026.
- [22] WAVE (2026) “WAVE web accessibility evaluation tool”, <https://wave.webaim.org/>, acesso em: 4 set. 2026.
- [23] Bersch, R. (2017) “Introdução à Tecnologia Assistiva”, CVI-Compreender Centro de Acessibilidade Assistiva, Porto Alegre, RS.
- [24] MDN Web Docs (2023) “Document Object Model (DOM)”, Mozilla Developer Network, https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model, acesso em: 4 set. 2026.
- [25] MDN Glossary (2026) “Accessibility tree”, Mozilla Developer Network, https://developer.mozilla.org/en-US/docs/Glossary/Accessibility_tree, acesso em: 5 set. 2026.
- [26] NV Access (2023) “About NVDA”, NV Access, <https://www.nvaccess.org/about-nvda/>, acesso em: 4 set. 2026.

- [27] VLibras (2023) “Suíte VLibras”, Governo do Brasil, <https://www.gov.br/governodigital/pt-br/vlibras>, acesso em: 4 set. 2026.
- [28] W3C (2023) “Accessible Rich Internet Applications (WAI-ARIA) 1.2”, World Wide Web Consortium, <https://www.w3.org/TR/wai-aria-1.2/>, acesso em: 4 set. 2026.
- [29] W3C (2024) “ARIA Authoring Practices Guide (APG)”, World Wide Web Consortium, <https://www.w3.org/WAI/ARIA/apg/>, acesso em: 5 set. 2026.
- [30] MDN Web Docs (2024) “Semântica – Glossário do MDN Web Docs”, Mozilla, <https://developer.mozilla.org/pt-BR/docs/Glossary/Semantics>, acesso em: 5 set. 2026.
- [31] MDN Web Docs (2024) “Referência de elementos HTML”, Mozilla, <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element>, acesso em: 5 set. 2026.
- [32] W3C (2024) “HTML Specification: Semantics, structure, and APIs of HTML documents”, World Wide Web Consortium, <https://www.w3.org/TR/html52/semantics.html>, acesso em: 5 set. 2026.
- [33] Google Developers (2024) “Learn HTML: Semantic HTML”, Google, <https://web.dev/learn/html/semantic-html>, acesso em: 5 set. 2026.
- [34] Apple (2026) “Manuel de Uso do VoiceOver”, Apple, <https://support.apple.com/pt-br/guide/voiceover/welcome/mac>, acesso em: 5 set. 2026.
- [35] Tecassistiva (2026) “Jaws Screen Reader”, Tecassistiva, <https://tecassistiva.com.br/produtos/jaws/>, acesso em: 5 set. 2026.
- [36] Vispero (2026) “ZoomText Screen Magnification”, Vispero, <https://vispero.com/zoomtext-screen-magnifier-software/>, acesso em: 5 set. 2026.
- [37] PETRIE, H.; KHEIR, O. (2007) “The relationship between accessibility and usability of websites”, In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*, ACM, New York, p. 397–406. <https://doi.org/10.1145/1240624.1240688>
- [38] POWER, C.; FREIRE, A. P.; PETRIE, H.; SWALLOW, D. (2012) “Guidelines are only half of the story: accessibility problems encountered by blind users on the web”, In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, ACM, New York, p. 433–442. <https://doi.org/10.1145/2207676.2207736>
- [39] RØMEN, D.; SVANÆS, D. (2012) “Validating WCAG versions 1.0 and 2.0 through usability testing with disabled users”, *Universal Access in the Information Society*, v. 11, n. 4, p. 375–385. <https://doi.org/10.1007/s10209-011-0259-3>

- [40] W3C (2021) “Making Content Usable for People with Cognitive and Learning Disabilities”, W3C Group Note, <https://www.w3.org/TR/coga-usable/>, acesso em: 4 set. 2026.
- [41] W3C (2025) “Explainer for W3C Accessibility Guidelines (WCAG) 3.0”, <https://w3c.github.io/wcag3/explainer/>, acesso em: 4 set. 2026.
- [42] W3C COGA TASK FORCE “COGA FAQ”, Cognitive and Learning Disabilities Accessibility Task Force, <https://www.w3.org/WAI/GL/task-forces/coga/wiki/COGA-FAQ>, acesso em: 4 set. 2026.
- [43] COOPER, A. (1999) *The Inmates Are Running the Asylum*, Macmillan Publishing, Indianapolis.
- [44] PRUITT, J.; GRUDIN, J. (2003) “Personas: practice and theory”, In: *Proceedings of the 2003 Conference on Designing for User Experiences (DUX '03)*, ACM, New York, p. 1–15.
- [45] SCHULZ, T.; FUGLERUD, K. S. (2012) “Creating Personas with Disabilities”, In: *Computers Helping People with Special Needs (ICCHP 2012)*, LNCS 7383, Springer, Berlin, p. 145–152. https://doi.org/10.1007/978-3-642-31534-3_22
- [46] NEATE, T.; BOURAZERI, A.; ROPER, A.; STUMPF, S.; WILSON, S. (2019) “Co-Created Personas: Engaging and Empowering Users with Diverse Needs Within the Design Process”, In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*, ACM, New York, p. 1–12. <https://doi.org/10.1145/3290605.3300880>
- [47] EDWARDS, E. J.; SUM, C. M.; BRANHAM, S. M. (2020) “Three Tensions Between Personas and Complex Disability Identities”, In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20)*, ACM, New York, p. 1–9. <https://doi.org/10.1145/3334480.3382931>
- [48] BENNETT, C. L.; ROSNER, D. K. (2019) “The Promise of Empathy: Design, Disability, and Knowing the ‘Other’”, In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*, ACM, New York, Paper 298. <https://doi.org/10.1145/3290605.3300528>
- [49] NARIO-REDMOND, M. R.; GOSPODINOV, D.; COBB, A. (2017) “Crip for a Day: The Unintended Negative Consequences of Disability Simulations”, *Rehabilitation Psychology*, v. 62, n. 3, p. 324–333. <https://doi.org/10.1037/rep0000127>
- [50] BRITTO, T. C. P.; PIZZOLATO, E. B. (2016) “GAIA: uma proposta de um guia de recomendações de acessibilidade de interfaces Web com foco em aspectos do Autismo”, *Revista Brasileira de Informática na Educação*, v. 26, n. 2, p. 102–123.
- [51] WEBAIM (2021) “Survey of Web Accessibility Practitioners #3”, <https://webaim.org/projects/practitionersurvey3/>, acesso em: 2 set. 2026.

- [52] MAKATI, T.; TIGWELL, G. W.; SHINOHARA, K. (2024) “The Promise and Pitfalls of Web Accessibility Overlays for Blind and Low Vision Users”, In: *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '24)*, ACM, New York, Article 38, 12 p. <https://doi.org/10.1145/3663548.3675650>
- [53] HEITMEIER, K.-A.; KERSKEN, V.; PISKOREK, P.; BÖHM, A.-K.; EGGER, N.; LANG, M.; ZIMMERMANN, G. (2023) “Persona Co-Design for Improving Digital Accessibility”, In: *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA '23)*, ACM, New York, Article 227, 7 p. <https://doi.org/10.1145/3544549.3585857>
- [54] TANG, X.; PIPER, A. M. (2026) “Designing for Collective Access: In Search of a Solution to Accessible Communication in a Mixed-Ability Non-Profit”, In: *Proceedings of the 2026 Designing Interactive Systems Conference (DIS '26)*, ACM, New York, p. 4400–4415. <https://doi.org/10.1145/3800645.3813075>
- [55] SECTION508.GOV “Sample Personas for Users With Disabilities”, U.S. General Services Administration, <https://www.section508.gov/develop/sample-personas/>, acesso em: 4 set. 2026.
- [56] CARLSON, L. “Review: Guidelines are only half of the story”, WCAG Working Group Wiki, W3C, https://www.w3.org/WAI/GL/wiki/Review:_Guidelines_are_only_half_of_the_story, acesso em: 2 set. 2026.
- [57] W3C (2023) “Understanding Success Criterion 1.3.5: Identify Input Purpose”, Web Accessibility Initiative (WAI), <https://www.w3.org/WAI/WCAG22/Understanding/identify-input-purpose.html>, acesso em: 4 set. 2026.
- [58] W3C (2018) “Understanding Success Criterion 1.3.6: Identify Purpose”, Web Accessibility Initiative (WAI), <https://w3c.github.io/wcag21/understanding/identify-purpose.html>, acesso em: 4 set. 2026.
- [59] W3C (2018) “Understanding Success Criterion 2.2.6: Timeouts”, Web Accessibility Initiative (WAI), <https://www.w3.org/WAI/WCAG22/Understanding/timeouts.html>, acesso em: 4 set. 2026.
- [60] BERNERS-LEE, T. (1999) “Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor”, *HarperSanFrancisco*.
- [61] JANSEN, B. J.; JUNG, S.-G; NIELSEN, L.; GUAN, K. W.; SALMINEN, J. (2022) “How to Create Personas: Three Persona Creation Methodologies with Implications for Practical Employment”, *Pacific Asia Journal of the Association for Information Systems*, v. 14, n. 3, p. 1–28
- [62] JANSEN, B. J.; JUNG, S.-G; NIELSEN, L.; GUAN, K. W.; SALMINEN, J. (2022) “How to Create Personas: Three Persona Creation Methodologies with Implications

for Practical Employment”, *Pacific Asia Journal of the Association for Information Systems*, v. 14, n. 3, p. 1–28

- [63] BILLESTRUP, J.; STAGE, J.; BRUUN, A.; NIELSEN, L.; NIELSEN, K. S. (2014) “Creating and Using Personas in Software Development: Experiences from Practice”, In: *Sauer, S., Bogdan, C., Forbrig, P., Bernhaupt, R., Winckler, M. (eds) Human-Centered Software Engineering. HCSE 2014. Lecture Notes in Computer Science, vol 8742*), Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44811-3_16