

- Back-End(<https://imasters.com.br/back-end>)
 - Mobile(<https://imasters.com.br/mobile>)
 - Front End(<https://imasters.com.br/front-end>)
 - DevSecOps(<https://imasters.com.br/devsecops>)
 - Design & UX(<https://imasters.com.br/design-ux>)
 - Data(<https://imasters.com.br/data>)
 - APIs e Microsserviços(<https://imasters.com.br/apis-microsservicos>)
 - IoT e Makers(<https://imasters.com.br/iot-makers>)

ACESSIBILIDADE

PATROCINADOR:

29 SET, 2014

Falando sobre aplicações e conteúdo dinâmico acessível

[f \(hGps://www.facebook.com/sharer?u=https://imasters.com.br/acesibilidade/falando-sobre-aplicacoes-e-conteudo-dinamico-aceessivel\)](https://www.facebook.com/sharer?u=https://imasters.com.br/acesibilidade/falando-sobre-aplicacoes-e-conteudo-dinamico-aceessivel)

100 visualizações
[t \(hGps://twiGer.com/share?url=https://imasters.com.br/acesibilidade/falando-sobre-aplicacoes-e-conteudo-dinamico-aceessivel\)](https://twitter.com/share?url=https://imasters.com.br/acesibilidade/falando-sobre-aplicacoes-e-conteudo-dinamico-aceessivel)

[in \(hGps://www.linkedin.com/shareArticle?url=https://imasters.com.br/acesibilidade/falando-sobre-aplicacoes-e-conteudo-dinamico-aceessivel\)](https://www.linkedin.com/shareArticle?url=https://imasters.com.br/acesibilidade/falando-sobre-aplicacoes-e-conteudo-dinamico-aceessivel) COMPARTILHE!

REINALDO FERRAZ
([HTTPS://IMASTERS.COM.BR/PERFIL/REINALDONIC-BR](https://imasters.com.br/perfil/reinaldonic-br))
Tem 34 artigos publicados com 37433 visualizações desde 2012



REINALDO FERRAZ ([HTTPS://IMASTERS.COM.BR/PERFIL/REINALDONIC-BR](https://imasters.com.br/perfil/reinaldonic-br))

34

Formado em desenho e computação gráfica e pós graduado em design de hipermídia pela Universidade Anhembi Morumbi em São Paulo. Trabalha com desenvolvimento web desde 1998. Coordena as iniciativas de acessibilidade na Web do NIC.br e projetos relacionados a Open Web Platform, Digital Publishing e Web das Coisas. Representante do NIC.br em grupos de trabalho do W3C internacional em Acessibilidade na Web, Digital Publishing e Web das Coisas. Apaixonado por acessibilidade, usabilidade, padrões web, HTML, CSS e café sem açúcar.

LEIA MAIS ([HTTPS://IMASTERS.COM.BR/PERFIL/REINALDONIC-BR](https://imasters.com.br/perfil/reinaldonic-br))

26 NOV, 2018

Thing description: um manifesto para aplicações de Internet das Coisas ([hGps:/ imasters.com.br/iot-makers/thing-description-um-manifesto-para-aplicacoes-de-internet-das-coisas](https://imasters.com.br/iot-makers/thing-description-um-manifesto-para-aplicacoes-de-internet-das-coisas))

9 OUT, 2018

Recursos de acessibilidade na Web para designers, desenvolvedores e produtores de conteúdo da WAI ([hGps:/ imasters.com.br/acesibilidade/recursos-de-acesibilidade-na-web-para-designers-desenvolvedores-e-produtores-de-conteudo-da-wai](https://imasters.com.br/acesibilidade/recursos-de-acesibilidade-na-web-para-designers-desenvolvedores-e-produtores-de-conteudo-da-wai))

10 ABR, 2018

Quando a primeira versão das WCAG (Diretrizes para Conteúdo acessível na Web) foi publicada, em maio de 1999, não era possível criar conteúdos dinâmicos ou manipular elementos multimídia na Web sem um plug-in ou aplicativo proprietário. Nessa época, as diretrizes eram suficientes para tornar conteúdo acessível, afinal a Web era composta apenas formulários, tabelas, imagens e outros elementos estáticos. Nove anos depois surgiram as WCAG 2.0, criadas com base na versão 1.0, que ampliou as possibilidades para tornar o conteúdo acessível. Mas a Web demandava mais...

Conteúdos dinâmicos ainda tinham barreiras sérias de acessibilidade, inclusive porque as tecnologias assistivas não conseguiam acessar modais ou áreas que eram atualizadas sem um “refresh” completo da página. Para tornar esse conteúdo dinâmico acessível na Web o W3C publicou a recomendação WAI-ARIA 1.0, que se tornou uma [Recomendação do W3C em março de 2014](http://www.w3.org/TR/wai-aria/) ([http:// www.w3.org/TR/wai-aria/](http://www.w3.org/TR/wai-aria/)).

WAI-ARIA (Accessible Rich Internet Applications) define a forma de tornar conteúdo, principalmente aplicações Web, acessíveis para pessoas com deficiências. Essa documentação auxilia especialmente conteúdo dinâmico e interfaces de usuário com controles avançados. WAI-ARIA funciona com as tecnologias já existentes, como HTML e SVG e proporciona uma forma de aplicar os requisitos das WCAG para aplicações ricas na Web.

Aplicações web complexas tornam-se inacessíveis quando as tecnologias assistivas não podem determinar a semântica de partes de um documento ou quando o usuário não é capaz de navegar de forma eficaz em todas as partes do site ou aplicação. WAI-ARIA divide a semântica em roles (papéis), states e properties (estados e propriedades).

Roles servem para identificar o elemento ou a aplicação na interface e estão divididas em quatro tipos:

- **Abstract Roles:** Usadas para ontologias. Não devem ser utilizadas para conteúdo
- **Widget Roles:** Utilizadas para definir interfaces de widgets, como alert, dialog, slider, tab, etc.
- **Document Structure Roles:** descrevem as estruturas que organizam o conteúdo em uma página, por exemplo article, group, img, region, etc. Estruturas de documentos geralmente não são interativos.
- **Landmark Roles:** são regiões da página destinadas a marcação de navegação, por exemplo banner, main, navigation, etc.

A lista completa de roles está [neste link](http://www.w3.org/TR/wai-aria/roles#roles_categorization) ([http:// www.w3.org/TR/wai-aria/roles#roles_categorization](http://www.w3.org/TR/wai-aria/roles#roles_categorization)).

Cada uma das “roles” pode ser manipulada e definida por diversos “states and properties”, por exemplo, um link definido como item de um checkbox pode ser definido como “true” ou “false” para que o usuário tenha um retorno acessível do resultado da ação.

```
<li role="menucheckbox" aria-checked="true">Link ativo</li>
```

A lista completa de states and properties para cada role [está aqui](http://www.w3.org/TR/wai-aria/states_and_properties) ([http:// www.w3.org/TR/wai-aria/states_and_properties](http://www.w3.org/TR/wai-aria/states_and_properties)).

Uma das principais diferenças entre as WCAG e ARIA é que a primeira utiliza os elementos semânticos do HTML para tornar o conteúdo acessível. Já WAI-ARIA possibilita a mudança da semântica de um elemento para torna-lo acessível. Eu explico.

Neste exemplo, o desenvolvedor precisa criar um cabeçalho que é um botão. Para esse caso a recomendação é:

- Não faça isso: `<h1 role=button>botão</h1 >`
- Faça isso: `<h1 ><button>botão</button ></h1 >`
- Ou então isso: `<h1 >botão</h1 >`

Porque se `role=button` for colocado dentro do `H1`, a estrutura de acessibilidade dos objetos tratará esse elemento como `<button>botão</button >` e perderá sua característica de cabeçalho.

Por esse motivo, WAI-ARIA deve ser utilizado somente onde não for possível utilizar a semântica do HTML para tornar o conteúdo acessível. Sempre dê prioridade para as WCAG.

A Web está em constante evolução e conforme ela evolui, navegadores e tecnologias assistivas devem acompanhar esse processo. Por esse motivo o W3C tem uma preocupação imensa com a acessibilidade. A Web foi criada com o intuito de ser acessível desde sua concepção e esse princípio deve acompanhar sua evolução. E nós, criadores e mantenedores dos códigos que circulam pela Web, temos uma responsabilidade imensa de manter esse legado de uma Web acessível para as atuais e futuras gerações, afinal, dentro de algumas décadas nós poderemos ser os usuários de tecnologias assistivas que vão acessar esse conteúdo na Web. E ele deve ser acessível. Para o nosso próprio bem.