

Capítulo

2

Avaliação de Acessibilidade e Usabilidade em RIA

Renata Pontin M. Fortes¹, Humberto Lidio Antonelli¹ e André de Lima Salgado¹

¹Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo (ICMC/USP)

Resumo

A popularidade de aplicações de Internet atingiu escalas abrangentes. Como consequência, uma grande diversidade de soluções foram criadas baseadas em funcionalidades Web. Aplicações de Internet Ricas (RIAs) é um termo importante adotado para avanços tecnológicos em software desenvolvidos para a Web, e que confere a aplicações Web meios para prover aos usuários uma experiência similar à experiência de uso de desktops. RIAs possuem, usualmente, capacidades mais amplas em comparação com aplicações tradicionais de hipertexto, especialmente considerando elementos interativos de interfaces. Novas possibilidades que emergiram a partir de RIAs foram essenciais para suportar aspectos de Web 2.0, como a participação e colaboração. Como entre outras aplicações, o desenvolvimento de RIAs usáveis e acessíveis é aspecto valioso e fundamental para as equipes de desenvolvimento. Algumas das novas funcionalidades de interação RIA disponíveis na Web ainda não são acessíveis por usuários com deficiências ou mobilidade reduzida. Por esta razão, este minicurso visa apresentar alguns dos principais conceitos usualmente aplicados para avaliação de usabilidade e acessibilidade RIA; ele abrange perspectivas sobre referenciais práticos e teóricos, a partir de Padrões de Qualidade até implementação de conteúdos para aplicações Web. A abordagem deste minicurso cobriu práticas sobre principais técnicas de codificação RIA, e roteiros para métodos de avaliação de acessibilidade e usabilidade como Avaliação Heurística e Testes com Usuários. Este minicurso foi desenvolvido para novatos e profissionais que desejam especializar suas habilidades para o desenvolvimento de aplicações RIA usáveis e acessíveis.

Palavras-chave: RIA, Acessibilidade, Usabilidade, Roteiro, Avaliação

Abstract

Popularity of Internet applications has reached wide scales. In consequence, a wide diversity of solutions has been created based on Web features. Rich Internet Application (RIA) is a important term adopted for technological advances in software developed for Web, which refers to Web applications aimed at providing users with a desktop similar experience. RIAs usually have wider capabilities in comparison with traditional hypertext applications, specially regarding interactive elements of their interfaces. New possibilities emerged from RIA were essential to support relevant aspects of Web 2.0, such as participation and collaboration. As among other applications, developing accessible and usable RIAs is a valuable and fundamental aspect for development teams. Some RIAs new interaction features available on Web are still not accessible and usable for people with disabilities or reduced mobility. For this reason, this mini-course aimed at presenting some of the main concepts usually used on accessibility and usability evaluation of RIA; it is an overview of perspectives about practices and theoretical references, from recent Standards for Quality up to the implementation resources of Web applications. The approach of this mini-course covered practices on main RIA coding techniques, and a Guide for methods of usability and accessibility evaluation as Heuristic Evaluation and Test with Users. Moreover, this mini-course was developed aiming newcomers and professionals that want to specialize their skills on the development and evaluation of usable and accessible RIAs.

Keywords: RIA, Accessibility, Usability, Guide, Evaluation

2.1. Introdução

A popularização do acesso à Internet possibilitou que grande parte das tarefas diárias (compras, movimentações financeiras, comunicação entre as pessoas, etc.) pudessem ser desempenhadas com auxílio de aplicações Web, tornando-as indispensáveis na rotina de várias pessoas. Assim, o projeto de interfaces para sites e aplicações da Web também tornou-se essencial no desenvolvimento de qualquer produto que apoie a diversidade de tarefas realizadas na Internet. A necessidade de que, na Web, interfaces sejam usáveis e acessíveis, se consolida e proporciona cada vez mais pessoas exigindo uma experiência que lhes satisfaça e atenda suas demandas de eficiência e efetividade, durante a navegação e realização de suas tarefas do cotidiano. No contexto da importância de aplicações Web para o cotidiano dos usuários, estudos têm mostrado opções de enriquecimento de interfaces para aplicações na Web, na direção de propiciar interações semelhantes às existentes em ambientes *Desktop*, por meio de *Rich Internet Applications* (RIA, que pode ser traduzido como Aplicações de Internet Rica) [Almeida and Baranauskas 2012], [Fernandes et al. 2013], [Carvalho et al. 2016].

Este minicurso aborda um tema de grande relevância técnica e social para profissionais e pesquisadores envolvidos com o desenvolvimento de aplicações Web em geral, e em especial, de RIAs. É cada vez mais importante promover o uso de princípios de *design* inclusivo no desenvolvimento de sistemas interativos — e em particular para sistemas Web. Desenvolver sistemas Web acessíveis e usáveis é importante para atingir uma parcela significativa de usuários que possuam alguma deficiência (estimados em cerca de 23,9% da população brasileira, segundo o IBGE (2010) e para atender a requisitos legais.

Realizar avaliações de diversos tipos, priorizando as demandas da variedade de aplicações Web e seus domínios, é reconhecida como atividade fundamental para atingir resultados efetivos no desenvolvimento de sistemas que todas as pessoas realmente possam usar de maneira satisfatória [ISO/IEC 25066 2016]. O desafio de identificar os recursos necessários para acompanhar a evolução tecnológica que emergiu com as aplicações Web, reconhecidas por terem interfaces “ricas”, se mostra com grande potencial para condução de estudos a fim de prover aprofundamentos no tema [Bozzon et al. 2006], [Fernandes et al. 2012]. Ao passo que novas tecnologias e possibilidades são implementadas para aplicações Web, torna-se evidente a necessidade de atualização e/ou criação de métodos apropriados para avaliação de tais tecnologias [Doush et al. 2013], [Fernandes 2013], [Watanabe et al. 2012].

O objetivo deste minicurso é disseminar conhecimento sobre as práticas, geralmente utilizadas, para avaliação de usabilidade e acessibilidade, que podem ser aplicadas a interfaces RIA, considerando-se o uso de recursos de Tecnologia Assistiva (TA). Sobre tudo, são apresentadas as práticas relacionadas a planejamento e organização de testes com usuários, a planejamento e organização de inspeções com especialistas por Avaliação Heurística (AH) e a práticas de revisão de *guidelines* próprias para aplicações ricas (RIA). Nesse sentido, este minicurso abordou como foco principal a utilização correta da especificação WAI-ARIA para desenvolvimento de RIAs mais usáveis e acessíveis, bem como o uso de metodologias para a avaliação do produto desenvolvido em relação a diferentes perfis de usuários.

Este capítulo apresentou uma introdução ao conteúdo preparado para todo o minicurso desenvolvido. As seções seguintes foram organizadas como segue: *Termos e Definições* (Seção 2.2), *Planejamento de Avaliação de Interfaces RIA* (Seção 2.3), *Técnicas de Avaliação de Acessibilidade para RIA* (Seção 2.4) e, ao final, *Conclusões* (Seção 2.5), seção na qual apresentamos uma breve discussão sobre os conhecimentos vistos.

2.2. Termos e Definições

Os conceitos apresentados neste minicurso tornaram-se cada vez mais sólidos como requisitos fundamentais para o *design* e avaliação de uma “boa” interação de usuários com interfaces Web e diferencial profissional para desenvolvedores Web, seguindo a crescente evolução de tecnologias e recursos na Web.

Nas próximas subseções são descritos: princípios fundamentais de interação, com o objetivo de fornecer conceitos básicos sobre Interação Humano-Computador (IHC); definições sobre *User Experience* (UX), usabilidade, acessibilidade e características de soluções RIA. Em seguida, apresentamos o tópico sobre Avaliação de interfaces, detalhando principais tipos de métodos: baseados em usuários e baseados em inspeção.

2.2.1. Princípios Fundamentais de Interação

Este minicurso foi idealizado para ser ministrado a profissionais que desejam aprofundamento em conhecimentos relacionados à área de *design* de interfaces Web usáveis e acessíveis e, também, para estudantes interessados em especializar seus estudos nesta área. Para tanto, entende-se ser necessário fornecer conceitos fundamentais de IHC para que profissionais possam revisá-los e estudantes possam conhecê-los. Estes conceitos não

são específicos das áreas de *Web design* (projeto de Web), mas são amplamente utilizados em diversas áreas do conhecimento como ferramenta para melhorar a compreensão sobre o que deve ser observado na interação entre pessoas e interfaces diversas.

A partir da experiência prévia dos autores com ministração de cursos e aulas sobre conteúdos relacionados a IHC e recursos Web, os autores priorizam, neste minicurso, o conteúdo sobre princípios fundamentais de interação, popularizados por Norman [2013]. Norman apresentou tais princípios na primeira versão de seu livro, então intitulado “*Psicologia do Dia a Dia*” e, posteriormente, intitulado “*Design do Dia a Dia*” [Norman 1988], [Norman 2013].

Os conteúdos apresentados neste minicurso foram retirados da versão revisada do livro “*Design do Dia a Dia*”, publicado no ano de 2013. A escolha deste material ocorreu pela relevância da publicação e do autor na área de IHC, bem como da importância do autor na área de desenvolvimento Web, sendo sócio de um dos principais grupos de consultoria da área de interfaces Web, o *Nielsen and Norman Group*¹. Adicionalmente, destaca-se a necessidade de difundir os conteúdos recém apresentados por Norman (note a data de publicação deste minicurso) durante a versão revisada de seu livro [Norman 2013].

Neste contexto, abordamos o conceito de *Significantes*, apresentado por Norman em “*Living with Complexity*” (Vivendo com a complexidade) [Norman 2010] e destacado pelo mesmo autor na versão revisada de “*Design do Dia a Dia*”, a fim de esclarecer confusões que, segundo Norman, surgiram no mundo do *design* sobre o uso de outro termo importante em princípios de interação: *Affordances*.

Os sete **Princípios Fundamentais de Interação** apresentados por Norman são os seguintes: *Descoberta*, *Feedback*, *Modelo Conceitual*, *Affordances*, *Significantes*, *Mapeamento* e *Restrições*. A seguir, apresentamos breve descrição e exemplificação para cada um destes princípios.

(1) Descoberta - O primeiro princípio destacado por Norman é chamado *Descoberta*. O princípio de *Descoberta* refere-se à necessidade que usuários possuem de, quando interagindo pela primeira vez com uma nova interface, compreender como ela funciona.

O princípio de *Descoberta* é o estado de compreensão do usuário sobre como uma interface funciona, o que ela é capaz de fazer e quais operações que o usuário pode executar com ela [Norman 2013, p. 10].

(2) Feedback - O princípio de *Feedback* pode ser traduzido em Português por parecer (ou retorno). Para profissionais e estudantes de áreas relacionadas à computação, este termo é comumente utilizado a fim de descrever saídas esperadas para algum processamento. No contexto apresentado por Norman, *Feedback* refere-se à existência contínua de informações sobre os resultados das ações conduzidas pelo usuário na interface.

Preferencialmente, o *Feedback* deve aparecer o quão imediato possível após a ação executada pelo usuário na interface [Norman 2013, p. 72]. A Figura 2.1 é uma ilustração

¹nngroup.com

sobre a ocorrência de *Feedback* após a execução da ação do usuário digitar “www” e na interface em questão, é então apresentada essa sequencia de caracteres.

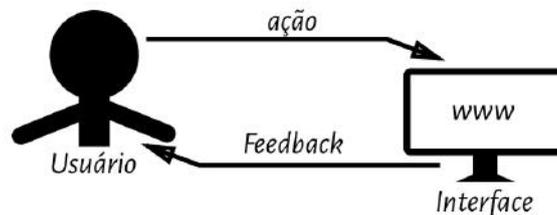


Figura 2.1. Ilustração sobre *Feedback* ocorrendo após a execução de uma ação pelo usuário com a interface.

(3) Modelo Conceitual - O princípio seguinte é chamado por Norman de *Modelo Conceitual*. *Modelo Conceitual* representa uma explicação de como a interface funciona. Segundo Norman, tais explicações são geralmente simplificadas. Quando tais modelos estão na mente do usuário, são chamados *Modelos Mentais*.

Repare que modelos conceituais são explicações existentes, independente de estarem na mente dos usuários [Norman 2013, p. 25-26]. A Figura 2.2 apresenta uma amostra da diversidade de ícones encontrados no site Flaticon² a partir da busca pelo termo “menu”, em setembro de 2016. Essa variedade de ícones pode refletir a variedade de modelos mentais de usuários quanto ao padrão de funcionamento de menus. Nestes casos, o importante é que os *designers* (profissionais do *design*) conheçam o público de usuários da interface em desenvolvimento a fim de identificar a predominância de *Modelos Mentais* entre os mesmos.



Figura 2.2. Exemplo da diversidade de ícones existentes para representação de menu em software.

(4) Affordances - Dentre os princípios, segundo Norman, o termo de mais destaque entre profissionais da área de *design* foi *Affordances*. Em seu livro, Norman mostrou que *Affordances* são relacionamentos possíveis entre as propriedades da interface e do usuário em questão, ressaltando que o termo vem da palavra *afford* (do Inglês permitir, "ser para") [Norman 2013, p. 11].

Por exemplo, se determinados usuários se deparam com alguma barreira visual que lhes impeça a visualização de uma aplicação Web, então, para esses usuários interagindo com esta aplicação, não existe o *Affordance* “estar visível” da aplicação, sendo necessário o

²Licenciado por Flaticon - www.flaticon.com - "Designed by the authors of this minicourse from Flaticon"

desenvolvimento de outras possibilidades de interação (por exemplo, código acessível a leitores de tela).

De maneira similar, quando uma interface possui propriedades que impedem algum tipo de relacionamento com as propriedades dos usuários, ocorre a existência de *Anti-affordances* - previnem determinadas interações [Norman 2013, p. 11]. Alguns óculos estereoscópicos possuem lentes diferentes para cada lado dos óculos, cada uma exercendo um diferente tipo de filtro para a percepção do usuário; os impedimentos causados por tais filtros são exemplos de *Anti-affordances*.

(5) Significantes - O termo *Significantes* (do Inglês *Signifiers*) foi introduzido por Norman na versão revisada de seu livro [Norman 2013]. Norman entendeu que era preciso introduzir um termo que auxiliasse a comunidade de *designers* a evitar confusões com o uso do termo *Affordances*.

Segundo Norman, os *designers* precisavam de um termo para designar quais partes uma interface podem ser clicadas, tocadas, etc., e por erro utilizavam o termo *Affordances*. Neste sentido, Norman relatou que alguns *designers* acabavam utilizando afirmações erradas como “coloquei um *Affordance* aqui para indicar onde o usuário poderia digitar o texto”. Portanto, para satisfazer a necessidade de um termo que referisse ao local onde a interação deve ocorrer, ele introduziu o termo *Significantes* (em Inglês *Signifiers*).

Assim, *Significantes* indicam onde a ação do usuário deve ocorrer [Norman 2013, p. 13-14]. Norman ressaltou que, durante o processo de *design*, *Significantes* são mais importantes que *Affordances* [Norman 2013, p. 19]. No exemplo mostrado na Figura 2.3, é possível verificar que a aplicação RIA (Google Docs³) apresenta uma paleta de cores, na qual cada cor significa onde a ação de seleção de cor (*Significantes*) deve ocorrer para seleção da cor específica para texto à esquerda.



Figura 2.3. Imagem do editor de textos Google Docs, ferramenta de seleção de cores de texto.

(6) Mapeamento - O princípio de Mapeamento refere-se ao relacionamento entre diferentes elementos da interface, assim como a correspondência de causa-efeito entre eles. Um exemplo prático de *Mapeamento* está no mecanismo de rolagem de alguns *mouses* que podem variar a direção (para cima ou para baixo) da rolagem de páginas Web conforme o sistema operacional utilizado, causando problemas de mapeamento para usuários que migram entre sistemas de padrões de mapeamento distintos (vide Figura 2.4).

³www.google.com/docs/about/

Mapeamentos podem ser observados também em aplicações que gerenciam casas inteligentes, para que usuários controlem de maneira satisfatória a iluminação de um ambiente específico a partir de seu celular ou *tablet* [Norman 2013, p. 20-21].



Figura 2.4. Foto de um *mouse* para exemplificar *Mapeamento* segundo seu mecanismo de rolagem.

(7) Restrições - O princípio *Restrições* pode ser explicado a partir de quatro vertentes: *Restrições Físicas*, *Restrições Culturais*, *Restrições Semânticas* e *Restrições Lógicas*.

Restrições Físicas implicam na impossibilidade de determinadas interações devido a limites físicos da interface. Por exemplo, usuários podem identificar facilmente qual a conexão do computador eles devem conectar o cabo de vídeo e qual devem conectar o cabo de rede, por causa da impossibilidade física de conectar cabo de rede na entrada de vídeo (e vice-versa) [Norman 2013, p. 125].

Restrições Culturais implicam em questões diferentes entre culturas diferentes [Norman 2013, p. 128-129]. Por exemplo, algumas culturas religiosas podem preferir aplicações de recomendações de refeições que não contenham recomendações sobre determinados alimentos, ou que apresentem indicações para casos em que ocorra a existência de tais alimentos.

Restrições Semânticas variam as possibilidades de ações de acordo com o significado da situação ou tempo [Norman 2013, p. 129-130]. Como exemplo, consideremos carros totalmente autônomos, estes ainda precisam do farol para iluminar a pista? É possível que, nestes casos, o valor semântico dos faróis mude: ao invés de iluminar o caminho (necessário para carros guiados por motoristas humanos), passam a servir para alertar motoristas de carros próximos, sobre a presença do carro autônomo.

Por fim, as *Restrições Lógicas* estão relacionadas com tentativas e erros. Quando usuários estão em dúvida sobre o efeito de suas ações com diferentes elementos da interface, eles podem optar por realizar diversas tentativas. A cada tentativa errada, o número de alternativas diminuirá de maneira lógica [Norman 2013, p. 130]. Ressalta-se que isto não implica na possibilidade de construir interfaces com inúmeras possibilidades de dúvidas, pois o usuário poderá ficar insatisfeito com a aplicação e, até, deixar de usá-la.

2.2.2. *User Experience* - UX

A definição para o termo Experiência de Usuário (do Inglês *User Experience* - UX) ainda é bastante discutida na literatura [Bevan et al. 2016]. Um dos trabalhos bem aceitos na literatura mostrou UX como:

- a experiência envolvendo os sentimentos dos usuários, assim como os aspectos afetivos e hedônicos relacionados com sua interação com interfaces [Law et al. 2009].

A definição de UX foi apresentada neste minicurso devido à importância do termo na área, e à proximidade deste minicurso com práticas atuais de mercado. De fato, para compreender e considerar UX, tem-se que observar essencialmente as interações dos usuários, mas as questões emocionais dos usuários ampliam sobremaneira o escopo dos possíveis atributos para se avaliar.

O foco deste minicurso foi direcionado aos conceitos relativos a Usabilidade e Acessibilidade, a partir dos quais, tem-se uma perspectiva mais delimitada e relacionada com as tecnologias utilizadas em RIAs.

2.2.3. Usabilidade e Acessibilidade

A **usabilidade** é compreendida como um dos principais atributos de qualidade de interfaces. A ISO/IEC 25066 (2016) apresenta a seguinte definição para usabilidade:

“O grau em que um produto pode ser usado por usuários específicos para atingir objetivos específicos com efetividade, eficiência e satisfação em um contexto específico de uso”.

Neste contexto, podemos observar com mais detalhes esta definição, a partir dos seguintes aspectos: **usuário** se refere à “*pessoa que interage com*” a interface; **objetivo** se refere à “*saída desejada*”; **efetividade** se refere a “*acurácia e a completude com as quais usuários alcançam objetivos*”; **eficiência** se refere à efetividade e adequada utilização de recursos; **satisfação** se refere a “*ausência de desconforto, e atitudes positivas*” a medida que os usuários utilizam a interface; **contexto específico de uso** se refere à combinação entre perfil dos usuários, tarefas e equipamentos envolvidos e; **tarefas** foram apresentadas como “*atividades requisitadas para atingir um objetivo*”⁴.

A **acessibilidade** é mostrada pela ISO/IEC 25066 (2016) a partir de características similares às de usabilidade, porém visando pessoas de uma população com maior diversidade de características e capacidades. Trabalhos recentes mostraram que a acessibilidade é um conceito essencial que visa garantir o acesso e uso a todos os usuários, independente de suas limitações. Assim, acessibilidade na Web corresponde a possibilitar que qualquer usuário, utilizando qualquer agente (software ou hardware que recupera e apresenta conteúdo Web) possa entender e interagir com o conteúdo Web.

⁴Conteúdo disponibilizado online pelo portal ISO no endereço: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25066:ed-1:v1:en>

Acessibilidade incorpora ainda a ideia de que todas as pessoas têm o direito de serem incluídas na sociedade, independente de deficiências, localização geográfica, barreiras de linguagem ou outra barreira [Thatcher et al. 2002].

Outra definição para acessibilidade bastante aceita na área foi mostrada pela *Web Accessibility Initiative* (WAI), organização criada pelo W3C, na qual a acessibilidade na Web consiste em [W3C 2005]:

*“websites e softwares que atendam às diferentes necessidades, preferências e situações dos usuários. Especificamente, a acessibilidade permite que pessoas com deficiência possam perceber, compreender, navegar e **interagir** com a Web, e assim podem contribuir com a mesma.”*

2.2.4. Rich Internet Applications

A Web foi elaborada com o objetivo de oferecer um repositório multiplataforma e interoperável de dados e informações. Para permitir que o conteúdo pudesse ser acessado por quaisquer conjuntos de software e hardware que os usuários estivessem operando, foram especificadas a linguagem de marcação para os conteúdos *HyperText Markup Language* (HTML) e o *HyperText Transport Protocol* (HTTP), como um mecanismo para transmitir arquivos textuais entre diferentes plataformas e sistemas operacionais conectados em uma rede [Berners-Lee 1990].

Desde sua criação, a Web tem passado por um constante processo evolutivo, proporcionando espaços mais dinâmicos e flexíveis de interação, de modo que usuários sem conhecimentos técnicos da área de computação também participem e modifiquem os conteúdos disponíveis. A partir desses novos usuários, surgiu o conceito de Web 2.0, que estabelece que o seu conteúdo não é mais proprietário e disponibilizado por pequenos grupos de pessoas. Esse conteúdo deve ser decidido por toda a população presente na Web, que passou a desempenhar co-autoria, além de consumir os serviços e conteúdos disponibilizados [van Wamelen and de Kool 2008].

A capacidade de contribuir com o conteúdo Web e interagir com os demais usuários da Internet mudou drasticamente o panorama da Web em um curto período de tempo, deixando de ser um ambiente passivo, caracterizado por páginas estáticas, para se tornar um ambiente ativo e passível de rápidas atualizações, repleto de páginas dinâmicas e interativas [Gehtland et al. 2006]. Novas aplicações Web, tais como *blogs*, *Social Bookmarking*, *wikis*, *podcasts*, *RSS feeds*, serviços online como *Facebook*⁵ e *Gmail*⁶ proporcionaram melhorias expressivas sobre os sites estáticos. Com isso, a Web tornou-se um dos principais canais de informação e comunicação entre as pessoas.

As características de maior interatividade são, muitas vezes, implementadas utilizando tecnologias abertas de Internet, tais como o HTML, CSS, JavaScript, DOM, AJAX, entre outras, nas chamadas *Rich Internet Applications* (RIA). Vale observar que HTML é a linguagem de marcação padrão da Web, responsável pela representação dos dados e estrutura dos conteúdos disponibilizados; CSS é a linguagem de folhas de estilo para definir-se

⁵<http://www.facebook.com>

⁶<http://www.gmail.com>

a formatação e layout das páginas Web; JavaScript é uma linguagem de programação que propicia a lógica de processamento dos dados (consistências, validação por ex.) no lado do cliente. As tecnologias DOM e AJAX, por serem especialmente importantes no escopo deste minicurso, serão descritas adiante.

Para Lawton (2008), a principal finalidade de RIAs é proporcionar uma experiência de acesso às aplicações Web de forma semelhante à que ocorre com as aplicações *desktop*. Segundo [Hooshmand et al. 2016], a utilização das características de RIA tornou-se padrão no desenvolvimento aplicações Web modernas.

Segundo Mesbah, Deursen e Roest (2012), alguns dos motivos para ampla adoção de RIAs são:

- não requerem a instalação no lado do cliente;
- todo mundo utiliza a versão mais recente;
- possibilidade de interação a partir de qualquer lugar com acesso à Internet, e;
- novas oportunidades de construção colaborativa e de comunidades.

RIAs combinam funcionalidades de interfaces do usuário de aplicações *desktop* com o amplo alcance do desenvolvimento de aplicações Web, além do melhor da interatividade e comunicação multimídia [Casteleyn et al. 2014]. O resultado final é uma aplicação que oferece uma experiência de usuário mais intuitiva, ágil e eficaz. Atualmente, parte significativa dos *websites* são RIAs ou apresentam componentes (*widgets*) que implementam as características de uma RIA. Os editores colaborativos, redes sociais e acompanhamento ao vivo de elementos são alguns exemplos populares de RIAs [Hooshmand et al. 2016].

As novas possibilidades de interação enriquecidas, que surgiram a partir de RIAs, são essenciais para apoiar os princípios relevantes da Web 2.0, como a participação e colaboração. Por outro lado, existe uma preocupação crescente sobre a acessibilidade e usabilidade dessas aplicações, uma vez que esses novos recursos de interação disponíveis na Web nem sempre são acessíveis ou são usados de maneira eficiente por todas as pessoas [Mori et al. 2011],[Casteleyn et al. 2014].

O aumento da interatividade tem exigido cada vez mais dos agentes de usuários (navegadores Web) e de recursos de TA, uma vez que a interação em RIAs deve se manter acessível aos usuários com deficiências ou com limitações temporárias [Machado Neto et al. 2014]. As alterações e atualizações na estrutura de uma aplicação Web geradas dinamicamente pelo JavaScript, no entanto, nem sempre são percebidas pelos usuários que interagem por meio de recursos de TA [Almeida and Baranauskas 2012],[Carvalho et al. 2016].

Tradicionalmente, tais recursos de TA apresentam informações, considerando que a página Web possui uma estrutura linearizada de conteúdo; porém as RIAs, muitas vezes, não possuem uma estrutura linearizada, permitindo que a estrutura da página Web (inclusive a árvore gerada por meio do *Document Object Model* – DOM) seja modificada durante a interação [Doush et al. 2013],[Fernandes et al. 2013].

A estrutura de dados “árvore”, gerada pelo DOM, é utilizada pelos navegadores Web para representar toda a estrutura da página Web, ou seja, ela corresponde à estrutura de dados referente aos elementos e conteúdos HTML que são armazenados em RAM, durante a renderização de uma página no *browser*/navegador.

Assim, quando essa árvore é alterada com o uso do JavaScript, altera-se também a apresentação da página Web, seja a sua estrutura ou seu conteúdo. Essa alteração é uma das características de RIAs, que se não implementada de maneira correta pode ocasionar problemas de acessibilidade e usabilidade, em especial para os usuários de recursos de TA⁷.

2.2.5. Avaliação

A literatura apresenta diversos métodos para a avaliação de usabilidade e/ou acessibilidade de interfaces. Segundo a ISO/IEC 25066 (2016), tais métodos podem ser agrupados em duas categorias: *Métodos Baseados em Usuários* e *Métodos Baseados em Inspeção*.

As subseções seguintes apresentam os exemplos de métodos abordados neste minicurso, segundo sua respectiva categoria.

2.2.5.1. Métodos Baseados em Usuários

A ISO/IEC 25066 (2016) apresenta *Métodos Baseados em Usuários* como métodos que envolvem a participação de uma amostra representativa de usuários reais da interface. Testes envolvem a observação de usuários reais na realização de tarefas pré-estabelecidas a partir da interface considerada na avaliação. Para tanto, técnicas e ferramentas adicionais podem ser consideradas, como o protocolo *Think Aloud* (do Inglês “pensando em voz Alta”), ferramentas de gravação de vídeo e de *Eye-tracking* (do Inglês “Perseguindo o Olhar”) [Ericsson and Simon 1980], [Dix et al. 2003], [Preece et al. 2015].

Este minicurso preparou a apresentação de um modelo genérico para planejamento e condução de testes com usuários, apresentados na Seção 2.3.2. Este modelo de planejamento genérico foi definido, considerando-se os principais cuidados relativos a realização de testes com usuários com deficiência [Freire et al. 2013].

2.2.5.2. Métodos Baseados em Inspeção

De acordo com a ISO/IEC 25066 (2016), *Métodos Baseados em Inspeção* envolvem o julgamento de inspetores com base em critérios pré-determinados. A norma mostra que estes inspetores podem variar de especialistas em usabilidade, usuários finais do sistema a outros tipos de profissionais.

Este minicurso aborda dois dos métodos de inspeção de usabilidade e/ou acessibilidade mais populares na literatura: Avaliação Heurística e Revisão de *Guidelines* [Følstad et al. 2012, Petrie and Power 2012, Martins et al. 2014]. A seguir estão descritos cada um destes métodos.

⁷<https://developer.mozilla.org/>

- (a) **Avaliação Heurística** – O método de Avaliação Heurística (AH) foi desenvolvido por Nielsen e Molich (1990) . A AH envolve o julgo de avaliadores especialistas em usabilidade sobre a conformidade entre a interface avaliada e um conjunto de princípios amplos sobre usabilidade (ditos heurísticas). Inicialmente, Nielsen e Molich desenvolveram um conjunto de 9 heurísticas que, posteriormente, foram refinadas e reorganizadas em 10 heurísticas por Nielsen em trabalhos posteriores [Nielsen 1994a], [Nielsen 1994b]. No Apêndice A encontra-se a lista das 10 heurísticas de Nielsen.
- (b) **Revisão de *Guidelines*** – O método de Revisão de *Guidelines* envolve a verificação de conformidade entre a interface avaliada e um conjunto de diretrizes (do Inglês *Guidelines*) sobre usabilidade e/ou acessibilidade. Ao contrário de heurísticas, *guidelines* representam situações conhecidas e específicas que, caso não atendidas, ocorrem na existência de problemas de usabilidade e/ou acessibilidade [Paz and Pow-Sang 2016].

Neste minicurso, abordaremos um conjunto de diretrizes da *Accessible Rich Internet Applications* (WAI-ARIA 1.0)⁸ que busca promover a acessibilidade de RIAs em complemento a *Web Content Accessibility Guidelines* (WCAG 2.0)⁹. Por meio da WAI-ARIA, é possível aumentar a semântica do conteúdo que está sendo apresentado visualmente (por exemplo, identificar que uma lista de itens está sendo usada como um menu ou que uma lista de links está sendo usada como uma lista em formato de árvore), bem com identificar regiões da página Web em que ocorrem atualizações automáticas ou por meio da interação do usuário (por exemplo, validações automáticas, atualizações do placar de um jogo, galeria de fotos apresentadas em um *widget*, entre outros) de maneira que o navegador Web possa avisar a API de acessibilidade sobre as atualizações dinâmicas no momento em que ocorrem.

Manter a API de acessibilidade atualizada, especialmente sobre as atualizações constantes que ocorrem em RIAs, é extremamente importante do ponto de vista dos usuários que utilizam recursos de TA para interagir com a aplicação, uma vez que sua interação pode ser seriamente prejudicada.

2.3. Planejamento de Avaliação de Interfaces RIA

A organização de avaliações de interfaces RIA possui peculiaridades que as tornaram mais sensíveis que a organização de avaliações em sistemas Web tradicionais. Em todo contexto de avaliações de interfaces, é importante conhecer as características do perfil de usuários que deseja-se abranger [Preece et al. 2015, p. 265]. Nestes casos, usuários diferem entre si, porém características em comum podem ser levantadas por técnicas apropriadas de compreensão do perfil de usuário (como questionários e entrevistas). Entretanto, RIAs podem ser ainda mais sensíveis que aplicações tradicionais devido às amplas possibilidade de personalização de conteúdo do DOM feita pelos usuários (mais detalhes apresentados na Seção 2.4) [Doush et al. 2013],[Fernandes et al. 2013].

As diferenças entre usuários do grupo desejado são ressaltadas em RIAs pela sua própria natureza adaptativa às preferências e usos dos usuários [Doush et al. 2013],[Fer-

⁸<https://www.w3.org/TR/wai-aria/>

⁹<https://www.w3.org/TR/WCAG20/>

mandes et al. 2013]. Assim, é intuitivo compreender que avaliações de interfaces RIA com base em pouca automatização precisam, conseqüentemente, de organizações bem estruturadas para minimizar a perda de informações importantes sobre usabilidade e acessibilidade da aplicação. Diante da necessidade de elaboração de um planejamento para avaliação de interfaces RIA, foi utilizada a perspectiva prática organizacional. Nesse contexto, preparou-se dois roteiros (passo-a-passo) de atividades para a organização de dois dos principais métodos de avaliação: **AH** e **Testes com usuários** [Følstad et al. 2012], [Martins et al. 2014]. As seções seguintes apresentam cada um desses modelos, e respectivas descrições.

2.3.1. Roteiro para Preparação de Avaliações Heurísticas

Esta seção apresenta uma proposta de roteiro, cujo objetivo é didático, para organização de inspeções de interfaces por AH. O passo-a-passo foi desenvolvido inicialmente a partir da tabela “Atividades do método de avaliação heurística” apresentada por Barbosa e Da Silva (2010, p. 318). Em seguida, essa tabela foi adaptada para contemplar abordagens adicionais apresentadas na literatura [Nielsen 1995],[Preece et al. 2015].

O roteiro está estruturado nas seguintes etapas: *Pré-inspeção* - Tabela 2.1; *Inspeção* - Tabela 2.2 e *Pós-inspeção* - Tabela 2.3. Para cada etapa, preparou-se uma tabela explicativa contendo o nome da etapa, os passos identificados para a organização da mesma segundo a literatura consultada e a finalidade principal de cada passo elencado. Cabe ressaltar que os passos nas linhas hachuradas e marcados com um asterisco (*), à frente de sua descrição, são considerados pelos autores como de maior importância.

Tabela 2.1: Passos da Etapa Pré-inspeção para organização de AHs e descrição de suas finalidades.

Etapa 1 – Pré-inspeção	Documentação das características dos usuários: visa o conhecimento sobre os usuários.
	Documentação das características da aplicação: visa o conhecimento sobre a aplicação.
	Documentação de cenário: visa o conhecimento sobre usuários utilizando a aplicação.
	(*)Descrição das tarefas sob avaliação: visa o conhecimento sobre tarefas abordadas durante a avaliação.
	Descrição de escopo da aplicação: visa o conhecimento das partes da aplicação que serão envolvidas na avaliação.
	(*)Convite a avaliadores: recomenda-se múltiplos avaliadores (pelo menos 3) especialistas no domínio da aplicação ou em IHC.
	(*)Definição do(s) observador(es): podem ser profissionais já envolvidos no projeto, com conhecimento sobre a aplicação.
	Definição de canal de comunicação: listas de e-mails, telefones, etc., para comunicação entre avaliadores e organizadores.
	(*)Agendamento da avaliação: definição de data, horário e local para a avaliação segundo disponibilidade dos avaliadores.
	Preparação e verificação de aparato: preparar local da avaliação e verificar funcionamento de todo o aparato necessário (computadores, rede, etc.)
(*)Definição do conjunto de heurísticas: escolher o conjunto de heurísticas mais adequado à aplicação. Recomenda-se envolver os avaliadores neste passo.	

Tabela 2.2: Passos da Etapa Inspeção para organização de AHs e descrição de suas finalidades.

Etapa 2 – Inspeção	Compreensão de escopo aplicação: observadores disponíveis para sanar possíveis dúvidas dos avaliadores quanto a aplicação. Avaliadores percorrem a interface para compreender seu funcionamento.
	(*)Inspeção: inspeção da interface segundo conjunto de heurísticas considerado. Cada problema levantado pode estar relacionado a uma ou mais heurísticas do conjunto considerado.
	Anotações por observadores: observadores podem anotar os problemas mencionados pelos avaliadores, deixando-os mais dedicados à inspeção.

Tabela 2.3: Passos da Etapa Pós-inspeção para organização de AHs e descrição de suas finalidades.

Etapa 3 – Pós-inspeção	(*)Revisão colaborativa de existência de problemas: avaliadores se reúnem para discutir a existência dos problemas levantados durante a inspeção.
	(*)Atribuição colaborativa de severidade: avaliadores discutem e definem juntos a severidade de cada problema levantado.
	Definição de sugestões: os avaliadores listam as sugestões de melhorias na interface, citando os respectivos problemas a corrigir.
	(*)Elaboração de relatório final: avaliadores elaboram o relatório final de avaliação contendo a lista de problemas (descrição, heurística(s) afetada(s), elemento da interface envolvido, severidade, etc.) e soluções levantadas (quando identificadas).
	Manutenção do canal de comunicação: manter a comunicação entre organizadores e avaliadores após a avaliação para possíveis questionamentos e explicações.

Informações detalhadas sobre a execução de AHs podem ser encontradas no portal *Nielsen and Norman Group*¹⁰. A seção seguinte apresenta roteiro similar voltado para o planejamento e organização de sessões de avaliação por Testes com Usuários.

2.3.2. Roteiro para preparação de Testes com Usuários

Assim como feito para o roteiro de preparação de AHs, os autores deste minicurso prepararam um roteiro passo-a-passo, com finalidade didática, para a organização de testes com usuários. Similarmente, o roteiro de organização de testes com usuários foi elaborado com base na tabela de atividades de testes com usuários apresentada por Barbosa e Da Silva (2010, p. 342). Foi também realizada uma adaptação, para contemplar experiências prévias dos autores com organização de testes com usuários, e também a partir de referências adicionais obtidas na literatura [Preece et al. 2015],[Usability.gov 2016a],[Usability.gov 2016b], [Nielsen 2001].

O roteiro sobre Testes com Usuários também está organizado em três sessões:

¹⁰<https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>

Pré-teste, Teste e Pós-teste. Assim como o roteiro para AHs, apresentamos uma tabela explicativa contendo nome da etapa, passos para a organização da mesma segundo a literatura consultada e descrição da finalidade principal de cada passo elencado pelos autores. Estas tabelas também apresentam passos marcados com um asterisco (*) a frente, significando os passos considerados pelos autores como de maior importância em cada etapa. Assim, preparamos as seguintes tabelas: Tabela 2.4 apresenta conteúdo da etapa *Pré-teste*; Tabela 2.5 apresenta conteúdo da etapa *Teste* e; Tabela 2.3 apresenta conteúdo da etapa *Pós-teste*.

Tabela 2.4: Passos da Etapa Pré-teste para organização de Testes com Usuários e descrição de suas finalidades.

Etapa 1 – Pré-teste	(*)Descrição das tarefas sob avaliação: visa o conhecimento sobre tarefas abordadas durante a avaliação.
	(*)Definição do(s) observador(es): podem ser profissionais já envolvidos no projeto, com conhecimento sobre a aplicação e sobre IHC.
	Treinamento e formação do(s) observador(es): análises de custo benefícios podem indicar se formação de observadores dentro da empresa é mais viável comparado à contratação de observadores disponíveis no mercado.
	(*)Definição de perfil dos participantes: Conhecimento sobre os usuários.
	(*)Agendamento individual da avaliação: Definição de data, horário e local para a avaliação segundo disponibilidade dos usuários.
	Definição de métricas de avaliação: Taxa de sucesso, taxa de erros, tempo necessário para completar a tarefa e/ou satisfação dos usuários.
	Descrição de escopo da aplicação: Conhecimento das partes da aplicação que serão envolvidas na avaliação.
	Definição de questões de teste: O que se deseja responder com os testes.
	Documentação de cenário: Conhecimento sobre usuários utilizando a aplicação.
	(*)Revisão de ética: Atender à regulamentações sobre aspectos éticos de pesquisas envolvendo pessoas.
Preparação e verificação de aparato: Preparar local da avaliação e verificar funcionamento de todo o aparato necessário (computadores, rede, etc.)	
Definição de canal de comunicação: Listas de e-mails, telefones, etc., para comunicação entre participantes e organizadores.	

Tabela 2.5: Passos da Etapa Teste para organização de Testes com Usuários e descrição de suas finalidades.

Etapa 2 – Teste	(*)Verificação de Consentimento: documentar ciência e concordância dos usuários sobre participação nos testes.
	Acompanhamento de cuidado e conforto: garantir que todos os participantes recebam o conforto e cuidado digno, não expondo usuários à situações prejudiciais de nenhuma natureza.
	Configuração de software de apoio: configurar softwares de apoio caso utilizados, como eye-trackers, gravadores de voz, câmeras, gravação de tela, etc.
	(*)Teste piloto: realizar testes pilotos com um número reduzido de usuários a fim de obter indícios sobre a efetividade e eficiência dos testes, evitando custos desnecessários.

(*)**Teste:** condução dos testes como planejados, considerando possíveis adaptações levantadas durante o teste piloto.

(*)**Anotações por observador(es):** observadores devem tomar notas sobre os problemas de usabilidade ocorridos durante as interações dos participantes com a aplicação nos testes.

Tabela 2.6: Passos da Etapa Pós-teste para organização de Testes com Usuários e descrição de suas finalidades.

Etapa 3 – Pós-teste	Questionários pós-avaliação: coleta de informações adicionais sobre satisfação e/ou outras características da interação.
	Análise de software de apoio: análise dos dados coletados por software de apoio. Deve ser feita por profissionais com conhecimento sobre IHC e sobre a ferramenta.
	Análise de correlação entre dados: analisar correlação entre dados de diferentes fontes (notas de observadores e software de apoio).
	(*) Elaboração de relatório final: observadores elaboram o relatório final de avaliação contendo a lista de problemas observados e soluções possíveis (quando identificadas).
	Manutenção do canal de comunicação: manter a comunicação entre organizadores e avaliadores após a avaliação para possíveis questionamentos e explicações.

Informações detalhadas sobre a execução de Testes com Usuários podem ser encontradas no portal do governo Americano, *Usability.gov*¹¹. É importante também destacar que os cuidados durante a realização dos testes com usuários com deficiências devem ser criteriosamente considerados [Freire et al. 2013].

A seção seguinte descreve técnicas para avaliações de interface RIA que dependem de menor participação de usuários, por estarem especialmente relacionadas a aspectos tecnológicos.

2.4. Técnicas de Avaliação de Acessibilidade para RIA

De acordo com o relatório de setembro/2016 do SecuritySpace¹², mais de 66% de todos os *websites* são implementados utilizando JavaScript. Entretanto, o desenvolvimento dessas aplicações, quando não são consideradas as questões de acessibilidade e usabilidade, afeta dramaticamente a capacidade das pessoas com deficiência acessarem o conteúdo da Web [Steen-Hansen and Fagernes 2015].

As aplicações de interfaces gráficas mais antigas utilizavam APIs de acessibilidade próprias que interoperavam com recursos de TA, a fim de solucionar ou reduzir minimamente os diversos problemas de acesso e utilização dessas aplicações. Contudo, as implementações atuais de RIAs, que utilizam puramente DHTML¹³, não suportam a utilização de tais APIs. Para tanto, o W3C, por meio da *Web Accessibility Initiative*

¹¹<https://www.usability.gov/how-to-and-tools/methods/running-usability-tests.html>

¹²http://www.securityspace.com/s_survey/data/man.201608/techpen.html

¹³*Dynamic* (dinâmico em Português) HTML, ou DHTML¹³, é a união das tecnologias: HTML, JavaScript e uma linguagem de apresentação, como folhas de estilo, aliada a um (DOM), para permitir que uma página

(WAI), tem trabalhado na implementação de recomendações para desenvolvimento de RIAs acessíveis.

Nesse sentido, uma das principais atividades para melhorar a acessibilidade de *websites* é a utilização de métodos que revelem problemas e barreiras de acessibilidade para que possam ser resolvidos durante o desenvolvimento. Existem métodos que podem ser utilizados, incluindo métodos de inspeção, realizados por especialistas, ou testes envolvendo usuários com deficiência, conforme descritos na Seção 2.3.

Os métodos de inspeção manual por especialistas têm um papel importante no processo de avaliação de *websites*. Para tanto, este minicurso orienta os desenvolvedores na inspeção de aplicações Web, tendo como base o conjunto de documentos que integram a *Accessible Rich Internet Applications* (WAI-ARIA). Esse conjunto de documentos define uma maneira para tornar as aplicações Web e seus respectivos conteúdos mais acessíveis a pessoas com deficiência, e visa auxiliar, especialmente, a criação de conteúdo dinâmico e controles avançados nas interfaces, as quais são desenvolvidas com Ajax, HTML, JavaScript ou tecnologias relacionadas.

A especificação técnica da WAI-ARIA é uma recomendação oficial do W3C desde março/2014, voltada principalmente para desenvolvedores de *browsers*, recursos de TA e outros agentes do usuário; desenvolvedores de tecnologias Web; e desenvolvedores de ferramentas de avaliação de acessibilidade. Um dos objetivos para se adotar a WAI-ARIA, durante o desenvolvimento das páginas Web, é fornecer semânticas adequadas para elementos dinâmicos (*widgets*), de modo a torná-los acessíveis, utilizáveis e interoperáveis com o uso de recursos de TA. Por meio dessa especificação, é possível identificar os tipos de *widgets* e suas estruturas, fornecendo uma ontologia das funções correspondentes que podem ser incorporadas ao conteúdo.

Em outras palavras, os elementos da linguagem de marcação utilizada que estejam definidos por meio de determinado *role* podem ser interpretados como elementos específicos, independentemente de qualquer semântica herdada da linguagem hospedeira de execução, como por exemplo do HTML. A Figura 2.5 ilustra o funcionamento de um acesso a uma RIA que inclui a marcação da WAI-ARIA.

Como mostrado na Figura 2.5, é possível observar que o relacionamento entre o navegador Web e os recursos de TA ocorre por meio de uma API de acessibilidade. Esse relacionamento possibilita o compartilhamento das informações comuns (*roles*, estados, eventos notificações, relações de informações e descrições) dos dados da aplicação que precisam ser processados quando o usuário requer meios especiais para acesso.

O navegador Web é encarregado de transmitir as informações relevantes para essa API de acessibilidade, conforme o usuário interage com a aplicação Web, e elas podem ser usadas por qualquer recurso de TA para transmitir o conteúdo de maneira acessível ao usuário.

Nesse contexto, para que de fato a acessibilidade possa ser alcançada, a especificação ARIA define um conjunto de atributos que os elementos HTML que compõem a implementação de uma determinada *widget* deve incluir. Esses atributos adicionam dados

Web seja modificada dinamicamente na própria máquina cliente, sem necessidade de novos acessos ao servidor web.

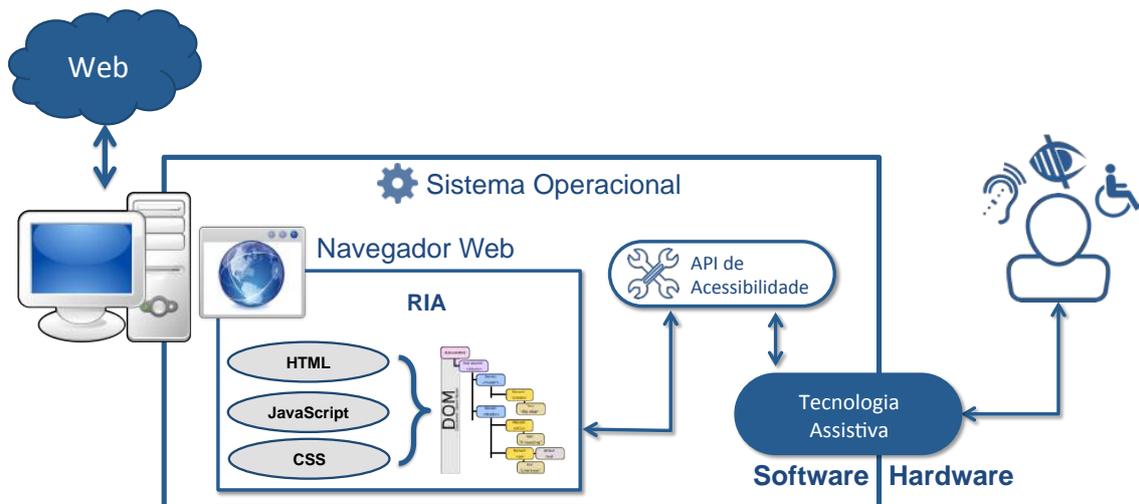


Figura 2.5. Esquema de interação em RIAs por pessoas com deficiência

semânticos aos elementos de uso genérico (tais como `<div>` e ``) e disponibilizam informações sobre o comportamento das *widgets* aos agentes de usuário e recursos de TA.

Esse conjunto de atributos da WAI-ARIA compreende as seguintes extensões de tecnologias Web que devem ser incorporadas às tecnologias utilizadas em RIAs, atualmente [W3C 2014]:

- **Role:** permite que o desenvolvedor anote um determinado elemento HTML com informações semânticas sobre o seu comportamento como um componente de interface por meio do atributo `role`. É importante ressaltar que este minicurso não tem como finalidade a descrição de toda a especificação WAI-ARIA. Portanto, será introduzida a contextualização e apresentação dos principais conceitos, abordando apenas alguns dos tipos de *roles*. A lista completa está disponível na especificação da WAI-ARIA, em que é possível encontrar a finalidade de cada *role*, as propriedades e estados aplicáveis a ela, bem como sua relação com as demais. Segundo a especificação da WAI-ARIA, os *roles* são categorizados da seguinte maneira:
 - **Roles Abstratos** – são utilizados apenas para definição de conceitos abstratos da taxonomia, servindo como base para construção de todos os demais *roles*. Portanto, os desenvolvedores não devem fazer uso desses *roles*;
 - **Roles de componentes de interface (*widgets*)** – são usados para definir componentes de interface de usuário (por exemplo, `alert`, `menutitem`, `tooltip` e `treeitem`) ou partes de componentes compostos de interface de usuário (por exemplo, `combobox` e `treegrid`). Além disso, alguns desses *roles* podem ser classificados como regiões de conteúdo “ao vivo”, em que ocorre atualização automática de conteúdo como resultados de eventos externos, ou seja, não depende da interação do usuário. Exemplos são: `marquee`, `status` e `timer`;
 - **Roles de estrutura de documento** – são utilizados para definir estruturas que organizam uma página Web. Exemplos: `article`, `heading` e `note`;

- **Roles de pontos de referência (landmarks)** – são regiões em uma página Web compostas por pontos importantes, as quais um usuário pode querer acessá-las rapidamente. Exemplos são: `main`, `navigation` e `search`;
- **Estados e Propriedades:** são usados para declarar atributos de um elemento. Assim, os recursos de TA e sistemas operacionais são capazes de manipular o elemento, mesmo quando ele é alterado de forma assíncrona, por vezes, sem a interação do usuário. Os atributos utilizados para representar estados e propriedades se referem a funcionalidades similares, uma vez que ambos proveem informações específicas sobre um objeto e fazem parte da natureza de uma *role*. De acordo com a especificação WAI-ARIA, os estados e propriedades são tratados como atributos prefixados com `aria-`. No entanto, os dois conceitos (estados e propriedades) são mantidos separados, visto que os valores de propriedades são menos prováveis de serem alterados dinamicamente em comparação com os valores de estados, os quais são frequentemente alterados em função da interação dos usuários.

A seguir é apresentado um exemplo de implementação em HTML do *widget Tab*, utilizando os atributos da especificação WAI-ARIA. Assim, os diferenciais dessa implementação que obedecem à especificação WAI-ARIA estão na inclusão dos atributos: `role`, `aria-controls` e `aria-labelledby`.

A apresentação visual permanecerá a mesma caso os atributos da WAI-ARIA não sejam definidos, porém o uso desses atributos melhora a interação dos usuários de recursos de TA, uma vez que fornecem informações adicionais que não estariam acessíveis sem a WAI-ARIA.

Além disso, é importante ressaltar que apenas a especificação dos atributos no código HTML não garante a acessibilidade, visto que deve-se também implementar os controles de interação dinâmicos em uma linguagem de *script*, conforme descrito na especificação WAI-ARIA.

```
1 <!-- Tabs em que cada link no topo abre uma tab de conteúdos
   em baixo -->
2 <ul class="tablist" role="tablist">
3   <li id="tab1" class="tab" aria-controls="panel1" role="tab"
   >Frutas</li>
4   <li id="tab2" class="tab" aria-controls="panel2" role="tab"
   >Vegetais</li>
5 </ul>
6
7 <div id="panel1" aria-labelledby="tab1" role="tabpanel">
8   <h3>Frutas</h3>
9   <ul>
10    <li>Maça</li>
11    <li>Laranja</li>
12    <li>Banana</li>
13  </ul>
14 </div>
15
```

```
16 <div id="panel2" aria-labelledby="tab2" role="tabpanel">
17   <h3>Vegetais</h3>
18   <ul>
19     <li>Batata</li>
20     <li>Cenoura</li>
21     <li>Cebola</li>
22   </ul>
23 </div>
```

A implementação de RIAs, seja na totalidade do *website* ou em partes específicas, requer que os desenvolvedores definam o comportamento dos componentes dinâmicos, dadas as múltiplas possibilidades de interação por parte do usuário. Além disso, é necessário associar os componentes de interface com um atributo *role*, que mapeie uma funcionalidade semelhante para a qual cada componente foi projetado, e atribuir os respectivos estados e propriedades que são requisitos dessa *role*, utilizando a linguagem de *script* apropriada.

Todos os elementos dinâmicos devem ter um esquema bem definido de navegação por teclado (por meio da gerência de foco na aplicação Web), considerando-se a possibilidade de utilização de TA. Finalmente, esses elementos dinâmicos devem ser incluídos na estrutura geral da página Web, considerando a região em que o usuário interagiu, e deverão ser apresentadas em meio aos demais elementos existentes [W3C 2014].

Vale destacar que o processo de desenvolvimento de RIAs é bastante especializado, possibilitando uma grande variedade de recursos que podem ser explorados nas diversas formas de interação da Web 2.0. A adição dos atributos da especificação WAI-ARIA nos elementos da página Web não implica em alterações visuais. Assim, testar RIAs por observação pode ser frustrante. No entanto, há algumas técnicas que podem ser utilizadas.

O uso de um ou mais recursos de TA que tenham dependência da marcação WAI-ARIA para apresentar o conteúdo de uma RIA pode ser utilizado para verificar sua acessibilidade. Os sistemas operacionais atualmente oferecem, em sua maioria, um leitor de tela nativo, porém existem outras alternativas gratuitas, como por exemplo, o NVDA¹⁴. Ademais, existem diversas soluções comerciais de leitores de tela que oferecem versões limitadas, mas gratuitas para testes, como o JAWS¹⁵.

O uso de leitores de tela, no entanto, pode ser lento a princípio e cansativo a longo prazo. Em geral, os leitores de tela oferecem um recurso para habilitar a saída textual dos textos que seria pronunciado pelo sintetizador de voz, tornando-se possível analisar o conteúdo sem a necessidade de ficar ouvindo a leitura repetidas vezes. Por exemplo, no NVDA, basta selecionar o menu *Ferramentas* e depois habilitar a opção *Exibidor de fala*. Outra estratégia é utilizar um software para emulação de leitores de tela, como Fangs¹⁶, disponível para o navegador Mozilla Firefox.

¹⁴<http://www.nvaccess.org>

¹⁵<http://www.freedomscientific.com/Products/Blindness/JAWS>

¹⁶<https://addons.mozilla.org/pt-br/firefox/addon/fangs-screen-reader-emulator/>

Cabe ressaltar que, antes de verificar se os recursos de TA estão apresentando o seu conteúdo corretamente, é altamente recomendada a verificação se o código-fonte atende aos padrões de linguagem de marcação, folhas de estilo e *scripts*. A WAI-ARIA proporciona orientações gerais que auxiliam na implementação acessível de RIAs, mas se mostra desprovida de formas para se garantir o cumprimento de suas especificações de implementação tecnológica [Watanabe 2014]. Portanto, o desenvolvedor deve ser responsável pela condução de inspeções manuais e testes com usuários para avaliação da acessibilidade e usabilidade das aplicações em conformidade com as recomendações da especificação WAI-ARIA.

Recomendações gerais para uso da especificação WAI-ARIA

O entendimento claro de quando e como adicionar os atributos definidos da especificação WAI-ARIA é tão importante quanto compreender quais são os *roles*, propriedades e estados existentes, uma vez que nem sempre é necessário o uso desses atributos na página Web. Para tanto, o W3C [W3C 2015] definiu cinco recomendações básicas para utilização:

Regra 1: Dar preferência ao uso de elementos ou atributos nativos da linguagem HTML, os quais apresentem a semântica e o comportamento desejado, em vez de utilizar os atributos da WAI-ARIA. Contudo, existem algumas exceções em que o uso apenas do HTML não é possível:

- o elemento ou atributo definido pela linguagem HTML ainda não está implementado por navegadores Web ou esse não é suportado por recursos de TA;
- as restrições do *design* visual excluem a utilização de um elemento nativo em particular, pois esse não pode ser personalizado como desejado;
- o recurso desejado não está disponível no HTML.

Regra 2: Não alterar a semântica de elementos nativos do HTML, a menos que seja estritamente necessário. Entretanto, se um elemento HTML estático for usado como base de um elemento interativo com a marcação WAI-ARIA adequada, será necessário adicionar *scripts* com todo o comportamento dinâmico desejado, incluindo a interação apenas pelo teclado.

Regra 3: Todos os controles interativos da especificação WAI-ARIA devem ser usáveis pelo teclado. Em outras palavras, em uma *widget* que permite ao usuário clicar, arrastar, tocar, rolar, entre outras maneiras de interação, deve-se fornecer a mesma capacidade de interação nessa *widget* apenas com o uso do teclado. Para isso, é necessário a criação de *scripts* que respondam aos pressionamentos de teclas ou combinações de teclas, quando aplicável.

Regra 4: Não utilizar `role=presentation` ou o estado `aria-hidden="true"` em um elemento visível e que pode receber foco, uma vez que isto pode resultar em um estado de foco no “nada”. Exemplo de mau uso:

```
1 | <button aria-hidden="true">press me</button>
```

Regra 5: Todos os elementos interativos devem ter um nome acessível. As APIs de acessibilidade de cada plataforma fornecem a propriedade de nome acessível ou equivalentes, de modo que esse é concatenado com a função do elemento de interface de usuário e apresentado pelo recurso de TA. Um exemplo é a identificação de um determinado campo de formulário por meio do elemento *label*, conforme a seguir:

```
1 | <label for="nomeusu">Usuário</label>
2 | <input type="text" id="nomeusu">
```

2.5. Conclusões

Este minicurso abordou um tema de grande relevância técnica e social para profissionais e pesquisadores envolvidos com o desenvolvimento de aplicações Web em geral, e em especial, de RIAs. Desenvolver sistemas Web acessíveis e usáveis é importante para atingir uma parcela significativa de usuários com deficiência. Para tanto, foram apresentados os principais conceitos que têm sido geralmente utilizados para avaliação de acessibilidade e usabilidade em RIA - uma visão geral das perspectivas, desde Normas / Padrões de Qualidade até os recursos para implementação de aplicações Web.

A experiência dos autores no campo do ensino mostrou-lhes que o desenvolvimento de interfaces RIA usáveis e acessíveis é desafio também no campo do ensino. A recente existência dos conteúdos RIA ainda gera incertezas quanto à aplicabilidade de métodos tradicionais de avaliação de interface para avaliação das mesmas. A grande possibilidade de adaptação de tecnologias RIA para preferências individuais de usuários, como mostrado ao longo deste minicurso, reforça a sensibilidade de tais tecnologias à qualidade da organização de métodos de avaliação pouco automatizáveis.

Assim, neste minicurso, os autores organizaram roteiros práticos para dois dos métodos mais tradicionais entre os de avaliação de usabilidade e acessibilidade a fim de assegurar a boa organização dos mesmos por praticantes, também objetivando maximizar o aproveitamento e performance de tais métodos. Tais roteiros foram desenvolvidos com base em uma compilação de recomendações identificadas na literatura. Assim, este minicurso foi oportunidade de difundir a importância de avanços no desenvolvimento e validação de roteiros similares para a literatura da área.

2.6. Agradecimentos

Agradecemos ao suporte dado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e a Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo n. 2015/09493-5.

Referências

- [Almeida and Baranauskas 2012] Almeida, L. D. A. and Baranauskas, M. C. C. (2012). Accessibility in rich internet applications: People and research. In *Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems, IHC '12*, pages 3–12, Porto Alegre, Brazil, Brazil. Brazilian Computer Society.
- [Barbosa and da Silva 2010] Barbosa, S. D. J. and da Silva, B. S. (2010). *Interação humano-computador*. Elsevier.

- [Berners-Lee 1990] Berners-Lee, T. (1990). Information management: A proposal. W3C archive.
- [Bevan et al. 2016] Bevan, N., Carter, J., Earthy, J., Geis, T., and Harker, S. (2016). New ISO Standards for Usability, Usability Reports and Usability Measures. In Kurosu, M., editor, *Human-Computer Interaction. Theory, Design, Development and Practice*, number 9731 in Lecture Notes in Computer Science, pages 268–278. Springer International Publishing. DOI: 10.1007/978-3-319-39510-4_25.
- [Bozzon et al. 2006] Bozzon, A., Comai, S., Fraternali, P., and Carughi, G. T. (2006). Capturing ria concepts in a web modeling language. In *Proceedings of the 15th International Conference on World Wide Web, WWW '06*, pages 907–908, New York, NY, USA. ACM.
- [Carvalho et al. 2016] Carvalho, L. P., Ferreira, L. P., and Freire, A. P. (2016). Accessibility evaluation of rich internet applications interface components for mobile screen readers. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing, SAC '16*, pages 181–186, New York, NY, USA. ACM.
- [Casteleyn et al. 2014] Casteleyn, S., Garrig'os, I., and Maz'on, J.-N. (2014). Ten years of rich internet applications: A systematic mapping study, and beyond. *ACM Trans. Web*, 8(3):18:1–18:46.
- [Dix et al. 2003] Dix, A., Finlay, J., Abowd, G. D., and Beale, R. (2003). *Human Computer Interaction*. Pearson Education Limited, 3rd edition.
- [Doush et al. 2013] Doush, I. A., Alkhateeb, F., Maghayreh, E. A., and Al-Betar, M. A. (2013). The design of {RIA} accessibility evaluation tool. *Advances in Engineering Software*, 57:1 – 7.
- [Ericsson and Simon 1980] Ericsson, K. A. and Simon, H. A. (1980). Verbal reports as data. *Psychological Review*, 87(3):215–251.
- [Fernandes 2013] Fernandes, N. (2013). Towards Web Accessibility Repair. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility, W4A '13*, pages 9:1–9:2, New York, NY, USA. ACM.
- [Fernandes et al. 2013] Fernandes, N., Batista, A. S., Costa, D., Duarte, C., and Carriço, L. (2013). Three Web Accessibility Evaluation Perspectives for RIA. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility, W4A '13*, pages 12:1–12:9, New York, NY, USA. ACM.
- [Fernandes et al. 2012] Fernandes, N., Costa, D., Neves, S., Duarte, C., and Carriço, L. (2012). Evaluating the Accessibility of Rich Internet Applications. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility, W4A '12*, pages 13:1–13:4, New York, NY, USA. ACM.
- [Følstad et al. 2012] Følstad, A., Law, E., and Hornbæk, K. (2012). Analysis in practical usability evaluation: A survey study. In *Proceedings of the SIGCHI Conference on*

Human Factors in Computing Systems, CHI '12, pages 2127–2136, New York, NY, USA. ACM.

- [Freire et al. 2013] Freire, A. P., de Lara, S. M. A., and de Mattos Fortes, R. P. (2013). Avaliação Da Acessibilidade De Websites Por Usuários Com Deficiência. In *Proceedings of the 12th Brazilian Symposium on Human Factors in Computing Systems*, IHC '13, pages 348–351, Porto Alegre, Brazil, Brazil. Brazilian Computer Society.
- [Gehtland et al. 2006] Gehtland, J., Galbraith, B., and Dion, A. (2006). *Pragmatic Ajax: A Web 2.0 Primer*. Pragmatic Bookshelf, USA, 1 ed. edition.
- [Hooshmand et al. 2016] Hooshmand, S., Mahmud, A., Bochmann, G. V., Faheem, M., Jourdan, G.-V., Couturier, R., and Onut, I.-V. (2016). D-ForenRIA: Distributed Reconstruction of User-Interactions for Rich Internet Applications. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, pages 211–214, Republic and Canton of Geneva, Switzerland. ACM Press.
- [IBGE 2010] IBGE (2010). Censo demográfico 2010. Technical report, Rio de Janeiro, RJ.
- [ISO/IEC 25066 2016] ISO/IEC 25066 (2016). ISO/IEC 25066:2016(en) Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for Usability — Evaluation Report.
- [Law et al. 2009] Law, E. L.-C., Roto, V., Hassenzahl, M., Vermeeren, A. P., and Kort, J. (2009). Understanding, Scoping and Defining User Experience: A Survey Approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 719–728, New York, NY, USA. ACM.
- [Lawton 2008] Lawton, G. (2008). New ways to build rich internet applications. *Computer*, 41(8):10–12.
- [Machado Neto et al. 2014] Machado Neto, O. J., Cunha, B. C., Pimentel, M. G., and Fortes, R. (2014). Capítulo - Tecnologia Assistiva: pesquisas, oportunidades e exemplos de recursos de tecnologia assistiva multimídia interativos. In *Livro de minicursos do XX Simpósio Brasileiro de Sistemas Multimídia em Web*, João Pessoa, PB, pages 1–18, João Pessoa, PB. SBC.
- [Martins et al. 2014] Martins, A. I., Queirós, A., Silva, A. G., and Rocha, N. P. (2014). Usability Evaluation Methods: A Systematic Review. *Human Factors in Software Development and Design*, page 250.
- [Mesbah et al. 2012] Mesbah, A., van Deursen, A., and Roest, D. (2012). Invariant-based automatic testing of modern web applications. *IEEE Transactions on Software Engineering*, 38(1):35–53.
- [Mori et al. 2011] Mori, G., Buzzi, M. C., Buzzi, M., Leporini, B., and Penichet, V. M. R. (2011). *Collaborative Editing for All: The Google Docs Example*, pages 165–174. Springer Berlin Heidelberg, Berlin, Heidelberg.

- [Nielsen 1994a] Nielsen, J. (1994a). Enhancing the Explanatory Power of Usability Heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 152–158, New York, NY, USA. ACM.
- [Nielsen 1994b] Nielsen, J. (1994b). Heuristic evaluation. In NIELSEN, J.; MACK, R. L., editor, *Usability inspection methods*, pages 25 –62.
- [Nielsen 1995] Nielsen, J. (1995). How to conduct a heuristic evaluation. Nielsen Norman Group: Evidence-Based User Experience Research, Training, and Consulting. Url: <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>. Acessado em 2 de setembro de 2016.
- [Nielsen 2001] Nielsen, J. (2001). Usability metrics. Nielsen Norman Group: Evidence-Based User Experience Research, Training, and Consulting. Url: <https://www.nngroup.com/articles/usability-metrics/>. Acessado em 2 de setembro de 2016.
- [Nielsen and Molich 1990] Nielsen, J. and Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256. ACM.
- [Norman 1988] Norman, D. A. (1988). *The psychology of everyday things*. Basic books.
- [Norman 2010] Norman, D. A. (2010). *Living with complexity*. MIT press.
- [Norman 2013] Norman, D. A. (2013). *The design of everyday things: Revised and expanded edition*. Basic books.
- [Paz and Pow-Sang 2016] Paz, F. and Pow-Sang, J. A. (2016). A systematic mapping review of usability evaluation methods for software development process. *International Journal of Software Engineering and Its Applications*, 10(1):165–178.
- [Petrie and Power 2012] Petrie, H. and Power, C. (2012). What do users really care about?: a comparison of usability problems found by users and experts on highly interactive websites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2107–2116. ACM.
- [Preece et al. 2015] Preece, J., Sharp, H., and Rogers, Y. (2015). *Interaction design: beyond human-computer interaction*. John Wiley & Sons.
- [Steen-Hansen and Fagernes 2015] Steen-Hansen, L. and Fagernes, S. (2015). Achieving accessible rich internet applications. *Norsk Informatikkonferanse (NIK)*.
- [Thatcher et al. 2002] Thatcher, J., Bohman, P., Burks, M., Henry, S. L., Regan, B., Swirenga, S., and Urban, M. (2002). *Constructing Accessible Web Sites*. Glasshaus.
- [Usability.gov 2016a] Usability.gov (2016a). Planning a usability test. Usability.gov: Improving the User Experience (U.S. Department of Health & Human Services). Url: <https://www.usability.gov/how-to-and-tools/>

methods/planning-usability-testing.html. Acessado em 2 de setembro de 2016.

[Usability.gov 2016b] Usability.gov (2016b). Running a usability test. Usability.gov: Improving the User Experience (U.S. Department of Health & Human Services). Url: <https://www.usability.gov/how-to-and-tools/methods/running-usability-tests.html>. Acessado em 2 de setembro de 2016.

[van Wamelen and de Kool 2008] van Wamelen, J. and de Kool, D. (2008). Web 2.0: a basis for the second society? In *Proceedings of the 2nd international conference on Theory and practice of electronic governance*, volume 1 of *ICEGOV '08*, pages 349–354, New York, NY, USA. ACM.

[W3C 2005] W3C (2005). Introduction to Web Accessibility. W3C. Url: <https://www.w3.org/WAI/intro/accessibility.php>. Acessado em 05 jun. 2016. World Wide Web Consortium.

[W3C 2014] W3C (2014). Accessible rich internet applications (wai-aria) 1.0. W3C. Url: <https://www.w3.org/TR/wai-aria/>. Acessado em 05 jun. 2016. World Wide Web Consortium.

[W3C 2015] W3C (2015). Notes on using aria in html. W3C. Url: <https://www.w3.org/TR/aria-in-html/>. Acessado em 05 jun. 2016. World Wide Web Consortium.

[Watanabe 2014] Watanabe, W. M. (2014). *Avaliação automática de acessibilidade em RIA*. Tese de Doutorado em Ciências de Computação e Matemática Computacional, Instituto de Ciências Matemáticas e de Computação de São Carlos (ICMC/USP), Universidade de São Paulo, São Carlos - SP.

[Watanabe et al. 2012] Watanabe, W. M., Fortes, R. P. M., and Dias, A. L. (2012). Using Acceptance Tests to Validate Accessibility Requirements in RIA. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility*, W4A '12, pages 15:1–15:10, New York, NY, USA. ACM.

Apêndice A: As 10 Heurísticas de Nielsen

Como indicado na Tabela 2.1, a definição do conjunto de heurísticas considerado para a avaliação é de grande importância. O conjunto tradicional das dez heurísticas de Nielsen está disponível online no portal do grupo de consultoria Nielsen & Norman Group. A seguir, apresentamos uma tradução feita a partir das descrições mostradas por Nielsen:

- 1. Visibilidade do estado do sistema** - O sistema deveria sempre manter os usuários informados sobre o que está acontecendo, usando de *feedback* em tempo apropriado.
- 2. Compatibilidade entre o sistema e o mundo real** - O sistema deveria falar a linguagem do usuário, com palavras, frases e conceitos familiares ao usuário, em vez de termos próprios do sistema. Seguir convenções do mundo real, fazendo a informação aparecer em uma ordem natural e lógica.
- 3. Controle e liberdade do usuário** - Os usuários frequentemente escolhem funções por erro e irão precisar de uma saída de emergência bem sinalizada para sair do estado indesejado sem ter que passar por um diálogo extenso. Suportar undo e redo.
- 4. Consistência e padrões** - Os usuários não deveriam ter que imaginar se palavras, situações, ou ações diferentes significam a mesma coisa. Seguir plataforma de convenções.
- 5. Prevenção de erro** - Ainda melhor do que boas mensagens de erro é um design cuidadoso que previne o acontecimento de um problema em primeiro lugar. Tanto eliminar condições de propensão ao erro quanto checar por tais condições e apresentar uma opção de confirmação ao usuário antes que o mesmo cometa à ação.
- 6. Reconhecimento em vez de lembrança** - Minimizar a carga de memória do usuário fazendo objetos, ações, e opções visíveis. O usuário não deveria ter que se lembrar da informação de uma parte do diálogo para outra. Instruções para o uso do sistema deveriam estar visíveis ou facilmente resgatáveis sempre que apropriado.
- 7. Flexibilidade e eficiência de uso** - Aceleradores – não vistos por usuários novatos – devem frequentemente acelerar a interação para o usuário experiente de forma que o sistema possa abranger ambos os usuários inexperientes e experientes. Permitir que os usuários adaptassem ações frequentes.
- 8. Estética e design minimalista** - Diálogos não deveriam conter informação que é irrelevante ou raramente necessária. Toda unidade extra de informação em um diálogo compete com as unidades relevantes de informação e diminui sua visibilidade relativa.
- 9. Ajudar os usuários a reconhecer, diagnosticar, e recuperar de erros** - Mensagens de erro deveriam ser elaboradas em linguagem plena (não em códigos), precisamente indicar o problema, e fornecer a solução de maneira construtiva.

- 10. Ajuda e documentação** - Apesar de que o sistema possa ser usado sem documentação, pode ser necessário prover ajuda e documentação. Qualquer informação do tipo deveria ser fácil de buscar, ser focada nas tarefas dos usuários, listar passos concretos para serem executados, e não ser muito extensa.¹⁷

¹⁷Retirado e traduzido do Portal Nielsen & Norman Group, acessado em 03 de outubro de 2016 pelo endereço: <https://www.nngroup.com/articles/ten-usability-heuristics/>

Biografia Resumida dos Autores

Renata Pontin M. Fortes é professora do Departamento de Ciências de Computação do ICMC (Instituto de Ciências Matemáticas e de Computação) na Universidade de São Paulo, campus de São Carlos. Possui doutorado na área de Engenharia Web pelo Instituto de Física de São Carlos, da USP, e mestrado e graduação pelo ICMC/USP em São Carlos. Atualmente é professora associada II do ICMC/USP. Tem experiência na área de Ciência da Computação, atuando principalmente nos seguintes temas: engenharia de web, projetos de software livre, acessibilidade na web, interação homem-computador e processos de software. Tem diversas publicações em periódicos, conferências e capítulos de livros na área de Acessibilidade e Engenharia Web.



Humberto Lidio Antonelli é doutorando no programa de Pós-Graduação em Ciências da Computação e Matemática Computacional no ICMC/USP. Possui mestrado em Ciências da Computação pelo ICMC/USP e graduação em Ciência da Computação pela Universidade Federal de Goiás. Pesquisador na área de acessibilidade e usabilidade de interfaces Web para diferentes contextos. Tem experiência na área de Interação Humano-Computador e Linguagens de Programação Web. Atualmente, realiza pesquisas com avaliação de acessibilidade em *Rich Internet Applications* (RIA). Atua principalmente nos seguintes temas: acessibilidade, gerenciamento de decisões, Interface Humano-Computador e Engenharia Web.



André de Lima Salgado é mestrando em Ciências da Computação e Matemática Computacional no ICMC/USP. Possui Bacharelado em Sistemas de Informação pela Universidade Federal de Lavras (UFLA). Realiza pesquisas na área de Interação Humano-Computador (IHC), com foco em inspeção de Usabilidade com Avaliação Heurística conduzidas por avaliadores com pouca experiência na área, também chamados novatos. Adicionalmente, o autor também realiza pesquisas relacionadas a Experiência de Usuário, Acessibilidade, Arquitetura da Informação, Brinquedos Inteligentes e *Design* Centrado no Usuário. O autor é membro dos grupos de pesquisa ALCANCE (UFLA) e INTERMÍDIA (ICMC/USP).

