

Um Simulador Visual de Leitor de Telas para Auxílio à Interpretação de Questões de Acessibilidade por Avaliadores Videntes

Felipe Tassario Gomes, André de Lima Salgado, Lianna Mara Castro Duarte,
Flávia de Souza Santos, Renata Pontin de Mattos Fortes

Instituto de Ciências Matemáticas e de Computação
São Carlos, SP - Brasil

(felipe.tassario.gomes,alsalgado,liannaduarte,flaviasantos)@usp.br, renata@icmc.usp.br

Abstract. *Several mechanisms support diagnostic evaluation of Web accessibility issues. Automatized tools evaluate accessibility based on conformance verification of codes in regard to recognized accessibility guidelines. However, even after accessibility evaluation with such tools, it is necessary to count on human evaluators to analyse the issues indicated by the tool. Therefore, the process of evaluating accessibility with automatized tools is still biased by the expertise effect. This study described and qualitatively evaluated a simulation tool that aims to support the perception and interpretation of automatized tools' reports. The findings show that the tool influenced the evaluator's interpretation and perception about accessibility issues faced by visually impaired users. This indicated an approach that support the perception and interpretation of automatized tool outcomes.*

Resumo. *Diversos mecanismos auxiliam na avaliação diagnóstica de questões de acessibilidade na Web. Ferramentas automatizadas avaliam a acessibilidade a partir da verificação de conformidade de códigos com diretrizes reconhecidas para acessibilidade. Entretanto, após a o uso de tais ferramentas, é necessário a participação de avaliadores humanos para averiguação das questões indicadas pela ferramenta. Nesse contexto, o processo de avaliação automática de acessibilidade na Web ainda está sujeito ao viés interpretativo de tais avaliadores humanos (efeito expertise). Este estudo descreveu e avaliou qualitativamente um simulador de leitor de tela para auxílio à percepção e interpretação de resultados obtidos de ferramentas de avaliação automática. Os resultados mostram que a ferramenta influenciou a interpretação e percepção do avaliador humano sobre problemas de acessibilidade enfrentados por usuários com deficiência visual. Assim, contribui com uma abordagem própria para auxílio à interpretação de resultados de ferramentas automatizadas.*

1. Introdução

Segundo recente relatório do *Digital in 2017 Global Overview*¹, a quantidade de residências brasileiras que possuem acesso à Internet no ano de 2017 ultrapassou 66%. Dentre as principais causas para o considerável crescimento do acesso à Internet nas residências brasileiras, está a popularização de tecnologias *Web* em *smartphones*, *tablets* e outros dispositivos móveis. Tais dispositivos móveis são utilizados por mais de 50% dos brasileiros, para acesso a conteúdos *Web*.

Desde o início da era da computação, as pessoas têm sonhado com computadores que pudessem interagir de maneira natural usando fala, gestos, inteligência e interface com humanos da mesma maneira natural que nos comunicamos uns com os outros [1, p. 77].

As novas maneiras de interação com computadores, por meio de fala e gestos por exemplo, são bem-vindas ao arsenal de possibilidades para projeto de interfaces. Entretanto, apesar do grande apelo pelo desenvolvimento de interfaces naturais, a utilização de novos mecanismos de interação não é garantia de melhoria na interação. Todos os novos meios de interação possuem potencial de melhorá-la, mas trazem novos desafios e problemas ao campo de Interação Humano-Computador [2].

Para alcançar os níveis sonhados para interfaces naturais, é preciso galgar com qualidade as possibilidades precedentes. Em outras palavras, é preciso projetar interfaces de qualidade no presente para que seja plausível projetar interfaces mais naturais no futuro. Neste contexto, destacam-se dois aspectos de qualidade de interfaces computacionais: usabilidade e acessibilidade [3].

Usabilidade é compreendida como um dos principais aspectos de qualidade de interfaces computacionais. A ISO 25066 [3] define usabilidade como “*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*”.

De maneira similar, acessibilidade também é compreendida como importante aspecto de qualidade de interfaces computacionais. Ela é definida também pela ISO 25066 [3], a partir dos mesmos conceitos que compõem usabilidade, porém voltados para usuários de populações com maior gama de características e capacidades. O escopo deste projeto exerce a vertente formativa de usabilidade e acessibilidade, pois atenta aos conhecimentos sobre o desenvolvimento de interfaces mais usáveis e mais acessíveis [4].

Existem mecanismos que auxiliam na avaliação diagnóstica de questões de acessibilidade na *Web* [5]. Dentre eles, destacamos as ferramentas de avaliação automatizada. Tais ferramentas auxiliam na avaliação diagnóstica de acessibilidade a partir da verificação de conformidade de códigos com diretrizes reconhecidas para acessibilidade [5], [6]. Além disso, ferramentas de avaliação automatizada de acessibilidade frequentemente indicam problemas potenciais de acessibilidade,

¹ <https://wearesocial.com/special-reports/digital-in-2017-global-overview>

requisitando que avaliadores humanos (especialistas em acessibilidade de software) verifiquem o real impacto das eventuais contradições indicadas pelas ferramentas [5].

Desta forma, o processo de avaliação de acessibilidade ainda é sujeito ao viés investigativo de avaliadores humanos. Uma das questões discutidas na literatura é a influência da experiência (efeito *expertise*) destes avaliadores na qualidade dos resultados de avaliações de usabilidade e acessibilidade [7], [8]. Por isso, entendemos que o desenvolvimento de soluções que auxiliem avaliadores humanos na interpretação de potenciais problemas de acessibilidade na *Web* seja requisito para avanços na área e suporte da qualidade dos resultados diagnosticados por meio de ferramentas automáticas. Assim, podemos sintetizar esta lacuna na seguinte questão de pesquisa:

Questão de Pesquisa: Como auxiliar avaliadores humanos na interpretação de resultados diagnosticados como problemas de acessibilidade gerados por ferramentas automáticas de avaliação?

Cientes de que a cegueira está entre as principais deficiências abordadas por recursos de Tecnologia Assistiva (em especial, os leitores de tela) em sistemas computacionais, este artigo descreve um método para demonstração e simulação do estado de acessibilidade encontrado por usuários cegos durante a navegação em sítios na *Web*. Nossa abordagem visou simplificar a compreensão do uso de leitor de telas em páginas *Web* para avaliadores humanos videntes. O termo vidente é usualmente adotado para qualificar as pessoas que não estão com deficiência visual [9]. Nossa hipótese é de que com o uso de um simulador que demonstre visualmente o conteúdo lido por leitores de tela em uma página *Web* auxilie avaliadores humanos na interpretação de resultados diagnósticos de problemas de acessibilidade, identificados por ferramentas automáticas. Nesse contexto, visamos traduzir os resultados auditivos de um leitor de tela (que usuários cegos usualmente percebem ao navegarem nas páginas *Web*) em resultados visuais (que usuários videntes estão habituados a perceber). Este trabalho descreve a criação de uma ferramenta e uma avaliação empírica inicial para essa abordagem.

O restante deste artigo apresenta uma revisão da literatura relacionada (Seção 2), metodologia de desenvolvimento (Seção 3), resultados alcançados e discussões (Seção 4) e conclusões do estudo (Seção 5).

2. Revisão de Literatura

Esta seção apresenta a revisão de literatura na área, em especial sobre os trabalhos que fundamentam nossa abordagem. Portanto, foram estudados os seguintes tópicos: “*Acessibilidade na Web para pessoas com deficiência visual*”, “*Avaliação de Acessibilidade*” e “*Ferramentas Automáticas de Avaliação de Acessibilidade*”.

2.1. Acessibilidade na Web para pessoas com deficiência visual

Acessibilidade na *Web* refere-se a possibilitar a percepção, o entendimento, a navegação e a interação com a *Web*, por todas as pessoas, que livres de barreiras (ou com barreiras reduzidas), ainda que contando com uso de recursos de Tecnologia Assistiva, também possam utilizar ou contribuir com a geração de conteúdo (*Web* 2.0) e desenvolvimento

de sistemas *Web* [10]–[12]. Segundo Petrie et al. [12, p. 3], acessibilidade em tecnologias da *Web* também pode ser compreendida como:

Todas as pessoas, especialmente as pessoas com deficiência e idosos, podem usar sites em uma variedade de contextos de uso, incluindo os mais usuais, e os que contam com apoio de Tecnologia Assistiva. Para alcançar isso, os sites precisam ser concebidos e desenvolvidos para apoiar a usabilidade em todos os contextos.

De acordo com a WAI², organização criada pelo W3C³, que tem como missão definir princípios e regras de *design* e desenvolvimento de sites que sejam acessíveis a pessoas com deficiências [13], a acessibilidade na *Web* engloba as questões relacionadas com o modo como todas as deficiências, seja visual, auditiva, cognitiva, da fala, física ou neurológica, afetam a *Web* [14].

Devido à relevância e necessidade de criação de conteúdo acessível na *Web*, a WAI desenvolveu um conjunto de recomendações com o objetivo de tornar o conteúdo na *Web* mais acessível. Dentre as recomendações existentes, o documento WCAG⁴ consiste em uma referência de suma importância, quanto às questões de acessibilidade na *Web* [15], [16], e tornou-se base importante, inclusive para vários países que buscam regulamentar a adoção de regras de acessibilidade na concepção da informação disponível na *Web* [17]–[20].

Esses documentos com recomendações, juntos, fornecem soluções complementares para o desenvolvimento da *Web* acessível. Segundo Tangarife [19], a adoção das diretrizes de acessibilidade elaboradas pelo W3C, por meio da WAI, é vantajosa para qualquer usuário. Desta forma, a sua adoção pela comunidade de desenvolvimento, de forma universal, representa um passo importante na busca por tornar a *Web* sempre mais acessível.

Em termos gerais, a definição de acessibilidade na *Web* visa garantir seu uso, principalmente, por pessoas com deficiência, direta ou indiretamente, com apoio de recurso de Tecnologia Assistiva. No Brasil, a definição de Tecnologia Assistiva é dada pela Secretaria Especial dos Direitos da Pessoa com Deficiência [21]:

Tecnologia Assistiva é uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social.

As dificuldades encontradas por um usuário durante a utilização de uma página *Web* variam de pessoa para pessoa. Qualquer problema que ocorra durante a interação do usuário com sistemas *Web* pode prejudicar sua satisfação e a eficácia do uso [3].

² WAI (*Web Accessibility Initiative*)

³ W3C (*World Wide Web Consortium*)

⁴ WCAG (*Web Content Accessibility Guidelines*)

Dessa forma, é importante entender as dificuldades encontradas por cada grupo de usuários, a fim de corrigi-las e melhorar o uso. Isso poderá melhorar, conseqüentemente, a usabilidade e acessibilidade do conteúdo *Web*.

Assim, o grupo de usuários com deficiência visual é importante ser considerado nesse contexto, por constituírem um grande número de pessoas que possuem tal deficiência proveniente de diversas situações: usuários idosos, usuários com cegueira parcial ou completa, de nascença ou adquirida, etc.

A navegação na página é um dos obstáculos mais marcantes [22], mas existem inúmeros outros, de categorias diferentes. Geraldo e Fortes [22] categorizam e detalham as potenciais barreiras para pessoas cegas diante do acesso à páginas *Web* em três categorias:

- Problemas técnicos: referem-se aos aspectos relacionados com questões tecnológicas adotadas na implementação das páginas. Exemplos de problemas técnicos são: atualização automática da página, formulários com redirecionamento, formulários sem rótulos adequados e abertura inesperada de novas janelas.
- Problemas semânticos: fatores que sobrecarregam aspectos cognitivos dos usuários cegos, necessárias para compreensão do conteúdo *Web*. São exemplos de problemas semânticos: conteúdos alterados dinamicamente, falta de contexto, dependência de cor e conteúdos em tabela.
- Problemas na navegação: aspectos relacionados com as formas disponibilizadas para localização no conteúdo disponibilizado e questões de navegação no conteúdo. São exemplos de problema de navegação: menus muito complexos, conteúdo inacessível pelo teclado, falta de *feedback* e ajuda, e falta de recuperação de erro.

Existe uma diversidade de produtos de Tecnologia Assistiva destinados a usuários com deficiência visual, parcial ou completa. Alguns exemplos destes são *Displays* de Braille, Lupas Virtuais, Temas de Alto-Contraste, Leitores de Tela, entre outros. Para a finalidade deste trabalho, discute-se em detalhes o Leitor de Tela.

Os leitores de tela são ferramentas que auxiliam usuários com deficiência visual a utilizarem as interfaces gráficas de um software, incorporando essencialmente um sistema de narração de texto. Porém vai além, sendo capaz de interpretar e interagir com a interface gráfica do programa, descrevendo-a para o usuário e ajudando-o na sua utilização⁵. De forma geral, descreve os elementos fundamentais de uma interface gráfica de um software, percorre e coloca foco em cada elemento, e indica ao usuário cego qual ação que ele pode realizar no elemento atual.

Apesar das capacidades técnicas dos leitores de tela, a liberdade de forma presente na criação de uma página *Web* gera desafios significativos para a questão de sua acessibilidade. Observa-se frequentemente falta de padronização de conceitos da

⁵<http://www.afb.org/info/living-with-vision-loss/using-technology/assistive-technology-videos/screen-reading-technology/1235>

interface de usuário, como, por exemplo, um site que se utiliza de menus para a navegação entre áreas, enquanto outro que faz uso de *sidebars* para o mesmo propósito.

Esta falta de padronização faz cada página ser um desafio individual para um usuário com deficiência visual, que precisa conseguir contextualizar, entender e aprender a navegá-la. Em um trabalho intitulado “*Como é a Web se você não pode vê-la*”, Asakawa [23] discute essa questão, mostrando as dificuldades reportadas por usuários com deficiência visual, e explicando quais erros (técnicos ou não) de criação daquela página levam aos problemas identificados.

2.2. Avaliação de Acessibilidade

A literatura apresenta diferentes abordagens para a avaliação de acessibilidade. Tais abordagens são, comumente, chamadas de métodos de avaliação de acessibilidade. Em geral, esses métodos sistematizam o processo de avaliação de acessibilidade e podem ser classificados como teste, questionamento, inspeção, modelagem analítica e de simulação.

Os **Métodos de participação de usuários em testes** envolvem a participação do usuário (leigo ou experiente) como objeto principal da metodologia. Observação e etnografia são exemplos de metodologias que se enquadram nesta classe [5].

Os **Métodos de participação de usuários em questionamentos** requerem que os usuários respondam a questionamentos especificados por organizadores, como questionário e entrevistas [5].

Os **Métodos de participação de avaliadores em inspeção** requerem a inspeção de conformidade da interface com diretrizes existentes na literatura. Tais diretrizes podem ser específicas ou amplas (como heurísticas). Todas essas abordagens podem ser conduzidas por avaliadores humanos. Abordagens baseadas em diretrizes específicas podem ser inspecionadas também por ferramentas automáticas [5].

Os **Métodos de modelagem analítica** se propõem a modelar o uso do software em avaliação para a predição de tempo e complexidade que um usuário levaria para utilizá-lo, *design reviews* de complexidade de tarefas; análise de carga cognitiva; e também modelos programáveis que estabelecem passos ou heurísticas para um programa automatizado agir como se fosse um usuário real [24].

Os **Métodos de simulação** visam imitar as condições de uso ou simular o uso da ferramenta por um usuário típico (ou um usuário específico a ser estudado). Isto inclui simular a experiência que um usuário iniciante terá ao utilizar o objeto de estudo, seus passos de navegação e de descoberta, etc [24].

2.3. Ferramentas Automáticas de Avaliação

Na literatura são relatados vários mecanismos que auxiliam na avaliação diagnóstica de questões de acessibilidade na *Web* [4], [5]. Entre estes mecanismos destacamos as ferramentas de avaliação automatizada, que auxiliam a avaliação diagnóstica de

acessibilidade a partir da conformidade de códigos e diretrizes reconhecidos para acessibilidade [5], [6].

Para exemplificar este domínio, apresentamos duas ferramentas automáticas de avaliação populares na área: TAW e Cynthia Says⁶. Ambas as ferramentas trabalham sobre um endereço de página fornecido para realizar uma validação de conformidade com o padrão WCAG 2.0.

O documento WCAG é voltado para a criação de conteúdo *Web*, visto que orienta como tornar o conteúdo de sites mais acessível às pessoas com deficiência. Atualmente, o documento possui duas versões [25]. WCAG 1.0 é composto de recomendações gerais e pontos de verificação específicos que são divididos em 3 níveis de prioridade. Alguns exemplos de exigências são: fornecer conteúdos equivalentes relativos a imagens, fornecer informações do contexto e fornecer mecanismos de navegação. Ao passo que WCAG 2.0 é uma atualização de WCAG 1.0 para abranger tecnologias mais avançadas da *Web* 2.0.

A ferramenta TAW, segundo seu site oficial⁷, foi criada como referência técnica para a WCAG 2.0. Ela visa comprovar o nível de acessibilidade atingido por conteúdos *Web*. Após a comprovação, é possível utilizar o logotipo da TAW indicando o referido nível de acessibilidade comprovado pela ferramenta. Para executar o teste automático, é necessário fornecer à ferramenta o endereço da página que se deseja avaliar.

A ferramenta Cynthia Says pertence à University of St Andrews, Escócia. Ela também foi criada para checar a conformidade de conteúdo *Web* com as diretrizes WCAG 2.0. Assim como a TAW, ela também funciona via página *Web* e requer apenas a indicação do endereço da página que se deseja avaliar.

Apesar dos benefícios do uso de ferramentas de avaliação automática, elas frequentemente demonstram problemas potenciais de acessibilidade, de forma que é necessário que a verificação do real impacto das contradições indicadas pela ferramenta seja realizada por especialistas em acessibilidade de software [5]. Desta forma, o processo de avaliação de acessibilidade ainda é sujeito ao viés investigativo de tais avaliadores humanos. Este estudo explorou a hipótese de que uma ferramenta específica de simulação possa melhorar a percepção de potenciais problemas de acessibilidade.

Ferramentas de simulação já estão disponíveis no mercado. Neste contexto, a Fangs Screen Reader Emulator⁸ (Emulador de Leitor de Tela) é uma ferramenta que oferece uma visualização do conteúdo que seria lido por um software de leitor de telas. Ela funciona como extensão para o navegador Firefox⁹. A Fangs extrai toda informação que seria utilizada pelos leitores, incluindo as *tags* do HTML. Essas informações são exibidas em uma janela separada em forma de texto sequencial. A ferramenta também

⁶ https://www.st-andrews.ac.uk/itsold/web/accessibility/cynthia_validator.shtml

⁷ <https://www.tawdis.net/#>

⁸ <https://addons.mozilla.org/en-US/firefox/addon/fangs-screen-reader-emulator/>

⁹ <http://www.mozilla.org/firefox>

oferece forma de visualização da lista de cabeçalhos (*headings*) e *links* identificados na página. Ela oferece suporte à percepção de potenciais problemas de acessibilidade, porém em forma textual, explorando pouco a visão dos usuários videntes. Assim, acreditamos que a ideia envolvida na Fangs possa ser melhorada a fim de ilustrar a leitura feita por leitores de tela mantendo a posição visual original dos elementos. Isso tornaria mais rápida a comparação entre lógica visual da disposição dos elementos da página e lógica acessível dos mesmos, uma vez que será utilizada por usuários videntes.

O objetivo deste estudo foi criar uma ferramenta de simulação que auxilie avaliadores humanos na interpretação de potenciais problemas de acessibilidade na *Web*, e sirvam de requisito para avanço na área e suporte da qualidade dos resultados diagnósticos de ferramentas de avaliação automáticas.

3. Metodologia

Este trabalho reporta um estudo, com prova de conceito, para um simulador do estado de acessibilidade encontrado por usuários cegos durante a navegação em sítios na *Web*. Nomeamos a ferramenta como *SRS - Screen Reader Simulator*, que em Português pode ser traduzido como Simulador de Leitor de Tela. A metodologia adotada para o desenvolvimento da ferramenta de simulação foi adaptada do modelo incremental de processo de software, como apresentado por Sommerville [26, p. 22] (vide Figura 1). Utilizamos das características do modelo incremental descritas por Sommerville como justificativa para a escolha do mesmo. As principais razões para a escolha do modelo de processo incremental para o desenvolvimento de nossa ferramenta foram:

- Facilidade e baixo custo para realizar mudanças durante o desenvolvimento. Isso foi necessário visto o caráter exploratório de nossa pesquisa.
- Disponibilidade de clientes (pesquisadores do próprio grupo) em participar do processo de desenvolvimento para fornecimento de *feedback*.

A Figura 1 apresenta o modelo de processo incremental adaptado por nós para o desenvolvimento do simulador. Na Figura 1, indicamos sete (7) atividades de desenvolvimento. O processo é iniciado com a descrição da ideia, a qual é apresentada na Seção de Introdução deste artigo. As demais atividades são: especificação, construção da versão inicial, desenvolvimento, construção da versão para prova de conceito, estudo de caso e construção da versão para divulgação.

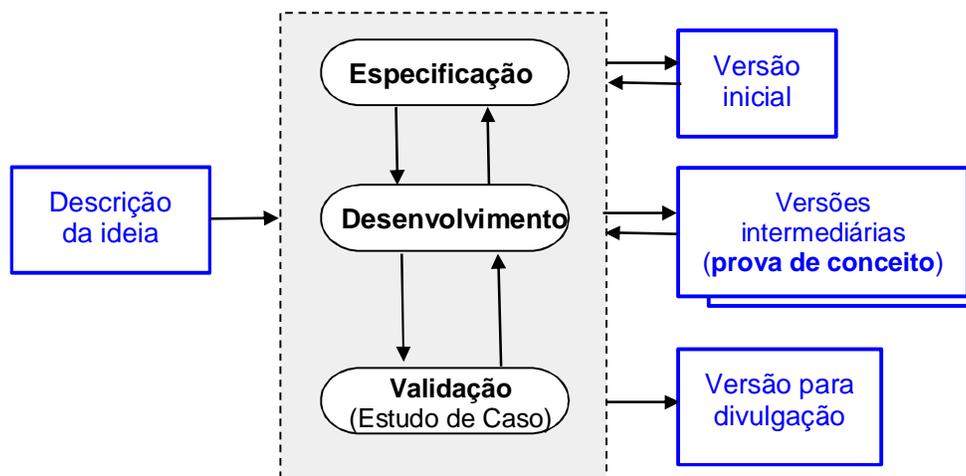


Figura 1. Modelo incremental adaptado de Sommerville [26, p. 22]

A atividade de especificação compreende a especificação inicial do simulador *SRS* e a consequente análise de requisitos realizada pelos agentes do projeto. Ao final dessa atividade, temos documentado os requisitos referentes à cada versão do simulador.

A atividade de desenvolvimento compreende o conjunto de técnicas utilizadas para codificação do simulador *SRS*. A atividade de construção da versão para prova de conceito representa o projeto e implementação de uma versão considerada apropriada para compor a prova de conceito, cuja validação ocorrerá durante a atividade de estudo de caso. Na atividade de estudo de caso, realizamos experimentos com avaliadores de acessibilidade a fim de provar os benefícios do uso de nosso simulador para atingir o objetivo deste estudo. Finalmente, a atividade de construção da versão de divulgação é automaticamente processada quando assumimos a versão mais desenvolvida do simulador e divulgamos neste artigo.

As seções seguintes descrevem e discutem os resultados obtidos com o processo de desenvolvimento do simulador, conforme descrito e ilustrado nesta seção.

4. Resultados e Discussões

Neste trabalho, desenvolvemos e descrevemos uma ferramenta para demonstração e simulação do estado de acessibilidade encontrado por usuários cegos durante a navegação em sítios na *Web*. Nossa abordagem visou simplificar a compreensão do uso de leitor de telas em páginas *Web* para avaliadores humanos videntes. Portanto, o intuito foi de traduzir os resultados auditivos de um leitor de telas em resultados visuais aos desenvolvedores videntes. Esta seção apresenta o resultado das atividades de desenvolvimento conduzidas.

4.1. Especificação

Esta seção apresenta os resultados de especificação de software. Está dividida entre as duas versões desenvolvidas para a ferramenta *SRS* de simulação. Para cada versão, apresentamos e justificamos os requisitos elicitados.

4.1.1. Versão inicial e análise de requisitos

A ferramenta *SRS* tem como requisito principal simular a experiência de usuários com deficiência visual, durante a interação com leitores de tela. Assim, a interação com *SRS* objetivou proporcionar ao máximo a sensação de aspectos característicos do cenário que usuários cegos percebem, durante o uso de leitores de tela, os elementos das páginas *Web* em que navegam. Nesse contexto, consideramos os seguintes requisitos (R) para o desenvolvimento do *SRS*:

- R1: o sistema deve ter a interação com o sítio *Web* prioritariamente pelo uso de teclado.
- R2: o sistema deve impossibilitar a visualização de imagens da página.
- R3: o sistema deve oferecer visualização de apenas um bloco de texto por vez, simulando a leitura de cada bloco feita pelos leitores de tela.
- R4: o sistema deve simular a percepção pontual somente do elemento em foco pelo leitor de tela;
- R5: o sistema deve atender a necessidade de navegação na página para todos os seus elementos. Isso deve ocorrer na ordem em que os elementos estão dispostos na página *Web*.

Para tanto, a ferramenta *SRS* deve executar procedimentos de inicialização com a página avaliada. Isso ocorre para ativar e desativar funcionalidades específicas da página. Em primeiro momento, o simulador desativa a interação com a página via *mouse*, permitindo apenas que a navegação seja realizada por teclado. Ressaltamos que isso não impossibilita o uso do *mouse* para demais tarefas no computador. Desse modo, o requisito “R1 - somente teclado” foi atendido. Com essa medida, conseguimos expor as seguintes situações de acessibilidade prejudicada nas páginas avaliadas:

- botões ou links não acessíveis por teclado,
- rolagem da página customizada ou difícil de ser controlada sem auxílio de *mouse*.

Para atender ao requisito R4, o simulador *SRS* esconde todos os elementos da página que não estejam sendo lidos pelo leitor de tela. Assim, *SRS* exhibe apenas o bloco de texto que contém o foco do cursor atual. Dessa forma, o contexto global da página é escondido, restando apenas a informação pontual que pode ser lida para o usuário. Consequentemente, ao usar o *SRS*, avaliadores necessitam percorrer o conteúdo completo da página, em ordem, até atingir o bloco de texto com a informação desejada. Esse percurso sequencial para acesso ao conteúdo de cada bloco de texto na página atende aos requisitos R3 e R5.

Outros obstáculos para usuários cegos, identificados na literatura como relacionados à acessibilidade na *Web*, também podem ser expostos pela ferramenta para atender aos requisitos R3, R4 e R5. Um exemplo disso é o caso das atualizações

dinâmicas da página, que podem ser imperceptíveis para o usuário e fazer com que ele fique confuso quanto às informações importantes. Também, pode fazer com que o usuário não seja capaz de contextualizar corretamente a informação sendo exibida.

Em sua primeira versão, a ferramenta *SRS* apresenta as funcionalidades básicas visando atender os requisitos estabelecidos. Ao ser acionada, a primeira versão da *SRS* desativa o uso do *mouse* na página e a obscurece toda a página, exceto o elemento em foco (vide Figura 2).

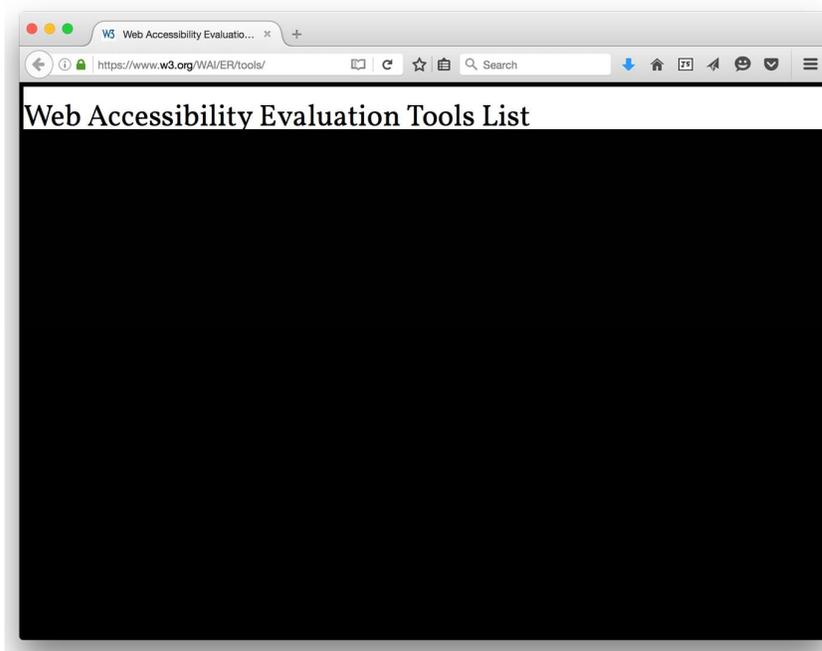


Figura 2. *SRS* Versão 1: após o simulador ser ativado, ilustração da página com todo o seu conteúdo escurecido, exceto pelo bloco de texto que contém foco.

Para a navegação na página usando a primeira versão da *SRS*, deve-se utilizar a tecla TAB no teclado. Ao pressioná-la, o foco da página (e do simulador) é modificado para o próximo elemento textual da página. A Figura 3 ilustra esse contexto de navegação na página, enquanto o simulador está ativado. Para ler o conteúdo presente na página, o usuário percorre os blocos de texto utilizando a tecla TAB.

O conjunto de características descritos nesta seção define a primeira versão do simulador *SRS*. Com essa versão, o grupo de pesquisadores se reuniu a fim de identificar melhorias e propor incremento para o desenvolvimento de uma versão apropriada à prova de conceito planejada. A próxima seção apresenta as características incrementais que compuseram a segunda versão do simulador *SRS*, chamada versão para prova de conceito.

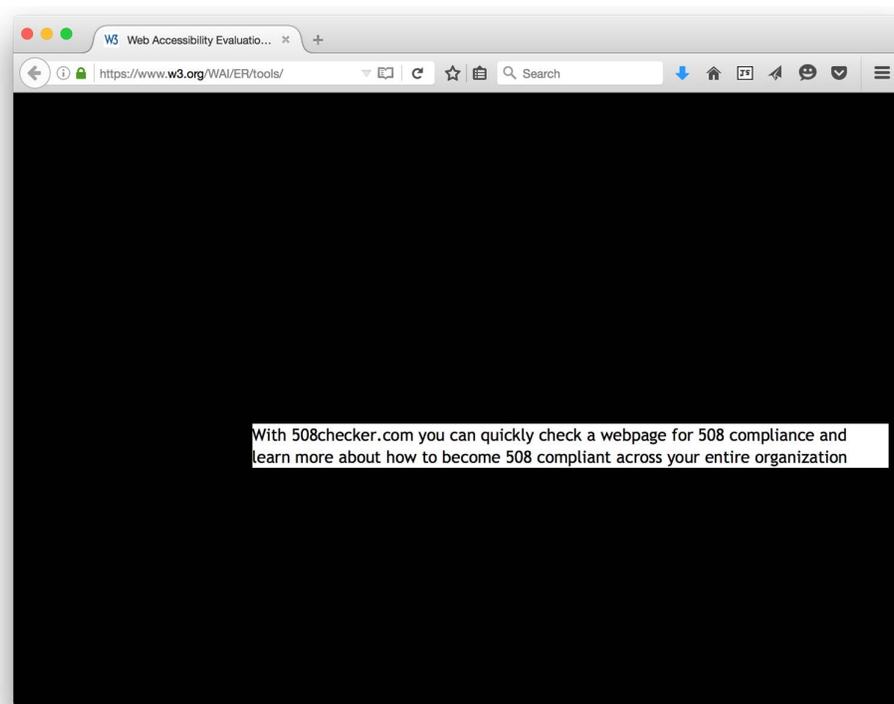


Figura 3. Uso da SRS - Versão 1: ilustração da página ao ser utilizada com o simulador ativado, em que o foco de texto é percorrido com a tecla TAB até atingir outro conteúdo.

4.1.2. Versão para prova de conceito

A primeira percepção que nos fez desenvolver incrementos à primeira versão do *SRS* foi de que leitores de tela populares entre os anos de 2016 e 2017 (época de desenvolvimento desta pesquisa) não se limitam a ler os textos visíveis em uma página: existem informações invisíveis a um vidente que são expostas quando usa-se um leitor de tela. Por exemplo, destacamos as descrições alternativas para *links*, legendas para imagens, *links* especiais destinados à utilização do páginas com teclado.

Os padrões de comportamento durante a utilização dos leitores de tela, pelos seus usuários, são bastante diversos. Borodin *et al.* [27] relatam que existem inúmeras estratégias para uso das teclas de navegação em uma página *Web*, durante o uso de leitores de tela. Ainda, destacam que cada usuário de leitores de tela desenvolve seus próprios costumes e estratégias. Essas diferenças ocorrem por diferentes motivos, como:

- Diferentes níveis de experiência dos usuários com uso de tecnologias de computação.
- Uso de diferentes ferramentas, impactando nas preferências dos usuários.
- Preferências pessoais diversas.

Dentre os exemplos de estratégias de navegação descritos por Borodin *et al.* [27], destacamos: a navegação por elementos *headings* na página; a busca por palavras-chave utilizando o mecanismo de busca do navegador; a busca por proximidade, quando

o usuário tenta encontrar um link ou algum texto que acredita estar próximo, para então continuar uma busca sequencial. Entendemos que as características destacadas por Borodin *et al.* [27] permanecem válidas e importantes para a literatura e profissionais na área de acessibilidade, pois tratam-se de características próprias do comportamento humano. Por isso, apesar de temporalmente distante do momento de execução desta pesquisa, consideramos as contribuições de Borodin *et al.* [27] como especialmente relevantes para o desenvolvimento de nosso simulador.

Neste ponto, destacamos dois leitores de tela, dada sua popularidade entre usuários:

- NVDA¹⁰ (*NonVisual Desktop Access*) - um leitor de tela gratuito e de código aberto, e
- JAWS¹¹ (*Job Access With Speech*) - uma opção comercial bem estabelecida no mercado.

Destacamos também que existe disponibilidade, consideravelmente alta, de conteúdo na *Web* para ensino sobre a utilização dessas ferramentas, tais como: documentação oficial, guias de utilização, e-books¹² e vídeo-tutoriais^{13,14}.

Todos os requisitos elicitados para o *SRS* foram implementados como um arquivo *javascript* que pode ser incluído em navegadores populares, como Chrome e Firefox. A seção seguinte descreve detalhes de desenvolvimento e implementação.

A partir das características recém descritas, adicionamos as seguintes funcionalidades à segunda versão do simulador *SRS*:

- R6: o sistema deve preservar a tecla TAB com seu comportamento padrão no navegador (percorre apenas elementos de interação, como *links*, botões e campos de texto);
- R7: o sistema deve oferecer parada em elementos de imagem para a visualização de sua legenda;
- R8: o sistema deve oferecer parada em elementos especiais somente visíveis para leitores de tela;
- R9: o sistema deve oferecer parada em elementos não-visíveis, porém acessíveis, como elementos de um menu que se encontra fechado;
- R10: o sistema deve oferecer painel de controle com botões de comando e exibição de textos alternativos, quando disponíveis;
- R11: a navegação do sistema deve ser capaz de percorrer para frente ou para trás através de todos elementos;
- R12: a navegação do sistema deve ser capaz de percorrer para frente ou para trás, exclusivamente por meio de elementos *headings*;
- R13: o sistema deve ter funcionalidade de semi-transparência para facilitar análise e compreensão por parte do desenvolvedor.

¹⁰ <https://www.nvaccess.org/>

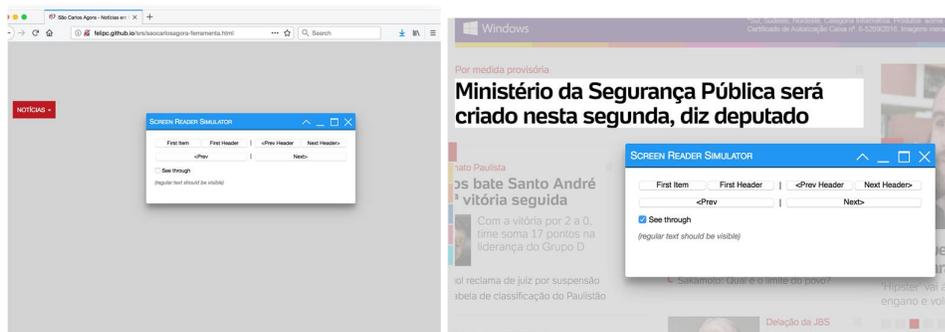
¹¹ <http://www.freedomscientific.com/Products/Blindness/JAWS>

¹² http://www.londrina.pr.gov.br/dados/images/stories/Storage/cmdpd/NVDA/manual_NVDA.pdf

¹³ <http://svrc.vic.edu.au/tutorials/windows/videos-jaws-basics/>

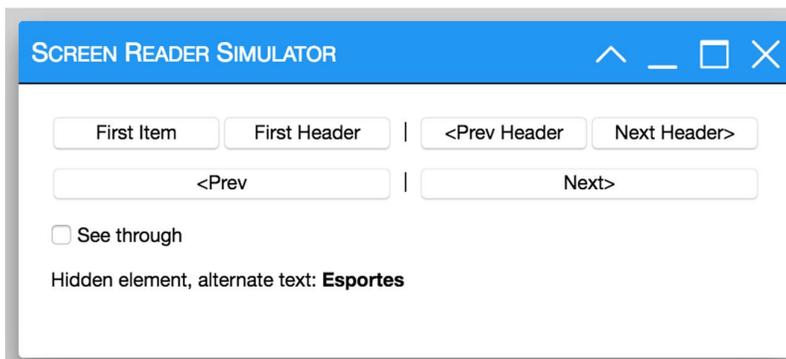
¹⁴ <http://www.cnib.ca/en/living/how-to-videos/tools-and-tech/Pages/The-free-NVDA-screen-reader.aspx>

Além desses oito novos requisitos, entendemos que o alto contraste entre fundo de tela preto com a janela de visualização branca poderia causar desconforto no uso do *SRS* pelos desenvolvedores videntes. Para melhorar esta questão de usabilidade, modificamos a cor para cinza em vez de preto, diminuindo o contraste. A Figura 4 apresenta a segunda versão do *SRS*.



(a) *SRS* ativado

(b) Modo semi-transparência



(c) Painel de Controle

Figura 4. Segunda versão do simulador *SRS*

A Figura 4(a) mostra o simulador ativado, enquanto a Figura 4(b) mostra o modo de semi-transparência do simulador. Essas figuras foram expostas aqui para uma visão da diferença entre os modos, que está no plano de fundo da extensão. Em ambas as figuras, apresentamos o painel de controle do simulador *SRS* no centro do navegador. A Figura 4(c) exibe o painel de controle em detalhes.

Na Figura 4(c), é possível constatar que os termos do painel de controle do simulador *SRS* estão em Inglês. Adotamos o Inglês para os termos visando aumentar a abrangência e potencial de uso do simulador. Assim, destacamos a função para primeiro item (*First Item*), primeiro cabeçalho (*First Header*), cabeçalho prévio (*Prev Header*), próximo cabeçalho (*Next Header*), elemento prévio (*Prev*), próximo elemento (*Next*), função de transparência (*See through*) e indicação de texto alternativo do elemento oculto (*Hidden element, alternative text*).

4.2. Desenvolvimento

O simulador *SRS* foi projetado como aplicação Web, criada predominantemente com linguagem *javascript*. Assim, por meio de um arquivo *javascript* codificado, *SRS* pode ser incluído nos principais navegadores correntes, tais como Chrome e Firefox. Ainda, esse arquivo pode ser empacotado pela API *WebExtensions*¹⁵ e ser utilizado como extensão para tais navegadores.

Os principais componentes lógicos da aplicação funcionam apenas com linguagem *javascript*, exceto a funcionalidade de esconder o fundo que utiliza, também, as linguagens CSS (*Cascading Style Sheets*) e SVG (*Scalable Vector Graphics*). As descrições de estilo em CSS foram utilizadas para detalhar o estilo do painel da ferramenta, enquanto as descrições SVG foram utilizadas para criação da máscara (transparente) que encobre o site e mostra apenas o elemento em foco.

Para as atividades de codificação, adotamos o navegador Firefox Quantum edição para desenvolvedores¹⁶, pois ele facilita o acesso à funcionalidades de desenvolvimento (e.g. inspeção da página carregada, análise de desempenho de processos e *scratchpad* para edição de *Javascript* em tempo real).

4.3. Estudo de caso

Após o desenvolvimento da segunda versão do simulador *SRS*, realizamos um estudo de caso para prova de conceito. O objetivo do estudo de caso foi explorar as características do simulador de maneira empírica, verificando a aplicação das abstrações envolvidas em sua engenharia de requisitos e comparando com ferramentas similares. Assim, buscamos avaliar em quais aspectos o simulador é capaz de superar ferramentas atuais e oferecer um novo benefício para as atividades de verificação de acessibilidade de uma página *Web*.

Nesta etapa de avaliação, contamos com a participação de dois avaliadores. Ambos avaliadores são especialistas em acessibilidade. Nesse sentido, destacamos algumas de suas experiências no tema: participação em pesquisas sobre acessibilidade por mais de dois (2) anos, experiência com observação e análise de testes de acessibilidade com usuários, participação prévia em inspeções de acessibilidade na *Web*, cursando pós-graduação cuja pesquisa principal abrange acessibilidade na *Web*. A quantidade de avaliadores envolvidos nesta etapa foi definida a partir da possibilidade do envolvimento de profissionais com tal experiência no estudo. Destacamos que tais profissionais são difíceis de se encontrar.

Para as avaliações, utilizamos dois *sites* de notícias. Ambos os *sites* de notícias são usualmente atualizados mais de uma vez ao dia. O *site Notícias Regionais* possui alcance regional, centrado na cidade de São Carlos (SP). O *site Notícias Nacionais* possui alcance nacional, é popular em todo o Brasil. Para ambos, a avaliação de acessibilidade foi feita na sua página inicial, salva estaticamente para garantir que as

¹⁵ <https://developer.mozilla.org/pt-BR/Add-ons/WebExtensions>

¹⁶ <https://www.mozilla.org/pt-BR/firefox/developer/>

análises ocorressem sobre o mesmo conteúdo. A quantidade de páginas *Web* avaliadas também foi limitada pela disponibilidade dos avaliadores, visto que o acréscimo de páginas iria superar o tempo disponível dos mesmos para participação na avaliação. Ambos os avaliadores foram convidados a avaliar apenas a primeira página desses portais, de forma exploratória.

O processo de avaliação adotado foi realizado em duas maneiras, segundo a participação de cada avaliador. Na primeira maneira de avaliação, um dos avaliadores foi encarregado de realizar uma avaliação de acessibilidade com ferramentas de validação já existentes no mercado. Assim, as ferramentas foram utilizadas: *TAW* e *Cynthia Says*.

Na segunda maneira de avaliação, o outro avaliador foi encarregado de realizar a avaliação sobre as mesmas duas páginas, usando como ferramenta computacional, apenas o simulador *SRS*. O simulador, por sua característica exploratória e interativa, não gera resultados quantitativos como as ferramentas de validação. Assim, a comparação entre os resultados se fez de forma qualitativa, observando as capacidades de cada ferramenta em auxiliar na detecção e relato de problemas sobre as barreiras de acessibilidade presentes nas páginas.

A Tabela 1 mostra um resumo dos resultados obtidos pelo primeiro avaliador com o uso das ferramentas *TAW* e *Cynthia Says*. Quantificamos os resultados de acordo com as diretrizes WCAG 2.0. Segundo cada ferramenta e para cada página *Web*, mostramos o número de diretrizes aprovadas, reprovadas, diretrizes que necessitaram de análise humana após uso da ferramenta e quantidade de diretrizes que não se aplicam na página (por inexistência de conteúdo para avaliação). Destacamos que o maior número de resultados de ambas as ferramentas está, justamente, na categoria que requer análise humana posterior. Esse fato reforça a motivação deste estudo.

Tabela 1. Avaliação das páginas com ferramentas automáticas

Diretrizes WCAG 2.0	<i>Notícias Regionais</i>		<i>Notícias Nacionais</i>	
	TAW	CynthiaSays	TAW	CynhiaSays
Aprovadas	2	1	1	1
Reprovadas	16	9	17	11
Dependentes de Análise Humana	26	39	26	38
Não se aplica	17	12	17	11

A Tabela 2 mostra exemplos de resultados encontrados pelo primeiro avaliador com o uso ferramentas de avaliação automática. Ele necessitou de, aproximadamente, cinco (5) horas para concluir as avaliações. Destacamos que os erros exemplificados podem ou não representar barreiras reais de acessibilidade nas páginas, dependendo de

qual elemento e conteúdo específico a que se referem. A descrição dos mesmos não é suficiente para concluir sobre a existência de barreiras.

Tabela 2. Erros relatados pelas ferramentas de validação

(a) Elementos <form> sem <label> correspondentes
(b) Texto e imagem consecutivas apresentam <i>link</i> para o mesmo lugar
(c) <i>Link</i> com texto muito longo
(d) Nenhum elemento <h1>
(e) <i>Links</i> com o mesmo texto mas com destinos diferentes

Durante a segunda parte a avaliação, quando o segundo avaliador utilizou o simulador *SRS* para inspecionar as páginas, o próprio avaliador foi responsável por descrever os resultados. O avaliador utilizou a ferramenta e compôs um relatório individual de diagnóstico das questões de acessibilidade encontradas na página. A Tabela 3 traz exemplos de potenciais problemas de acessibilidade relatados pelo avaliador com o uso da *SRS*.

Tabela 3. Erros relatados pela ferramenta de simulação

(a) Primeiro elemento de cabeçalho da página está após muito conteúdo
(b) Textos alternativos são pouco explicativos
(c) Menus apresentam opções em excesso
(d) Existem elementos sem texto alternativo
(e) Primeiro elemento da página não é útil
(f) Uso de jargão e siglas em conteúdo alternativo (exemplo, "SCA" no São Carlos Agora)
(g) Descrição alternativa da imagem não a descreve, apenas detalha <i>copyright</i>
(h) Conteúdo principal da página não está descrito por um elemento <i>heading</i>
(i) Existe muito conteúdo secundário antes de se chegar ao principal

Os resultados entregues pelo avaliador que utilizou o *SRS* indicam o potencial da ferramenta como modo de mitigar o efeito *expertise* [7] na interpretação de resultados de avaliação de acessibilidade em páginas da *Web*. Isso se justifica porque o

avaliador em questão utilizou dos resultados exibidos na interface da *SRS* para descrever os problemas de acessibilidade encontrados. De fato, compreendemos que o uso da *SRS* sugere uma percepção dos problemas de acessibilidade existentes (visto que ela altera a forma de exibição visual do conteúdo). Nesse sentido, o sucesso no uso da ferramenta, e das funcionalidades previstas pela mesma, implica no sucesso como forma de auxílio na percepção de potenciais problemas de acessibilidade. Assim, entendemos que a adoção da *SRS* fica condicionada às preferências do avaliador humano.

O segundo avaliador levou cerca de duas (2) horas para finalizar a avaliação da página. Esse tempo inclui identificação e listagem de potenciais barreiras de acessibilidade. Para diminuir esse tempo e somar os benefícios do simulador à ferramentas automáticas, sugerimos pesquisas futuras sobre a possibilidade de junção de ambos. Assim, o simulador *SRS* poderá atuar em relatórios de ferramentas automáticas para facilitar a compreensão dos resultados que necessitem de análise humana.

Ainda considerando os problemas de acessibilidade destacados pelo avaliador com a ferramenta *SRS*, podemos agrupá-los entre problemas em comum (aqueles também encontrados pelo uso de validadores) e problemas inéditos (encontrados apenas com o uso do *SRS*), como segue:

- (a) **Erros em comum com validadores** - Alguns dos erros encontrados também foram apontados na análise com os validadores. Entretanto, mesmo neste caso, a simulação é capaz de apontar com maior objetividade os erros que representam obstáculos ou frustrações reais do uso daquele *website*, ao invés de fazer parte de uma lista extensa de erros agrupados juntos pela ferramenta de validação.
- (b) **Erros inéditos** - Outros erros só foram detectados através do simulador de leitor de tela *SRS*. Por exemplo, o erro em que a descrição textual de uma imagem está implementada, porém o texto não é útil para sua compreensão. Também, podemos ressaltar a descrição do erro que aponta que o conteúdo principal do site não corresponde ao conteúdo em que o elemento `<h1>` foi utilizado.

Desta forma, a ferramenta *SRS* se mostrou capaz de detectar tipos de problemas que não estavam abrangidos pelas ferramentas de validação tradicionais. Isso evidencia características qualitativas do simulador *SRS* que superaram as demais ferramentas utilizadas. Ainda, o uso da ferramenta permitiu a geração de relatório que descrevesse problemas com indicação de seus impactos potenciais, enquanto os validadores entregam relatórios mistos de problemas e potenciais problemas de acessibilidade. Isso indicou que o simulador *SRS* possibilita ao avaliador compreender melhor a perspectiva de usuários com deficiência visual. Tal descrição ainda é muito limitada nas demais ferramentas.

Finalmente, não obtivemos comentários sobre as funcionalidades da ferramenta ou possíveis melhorias. Como o segundo avaliador realizou suas tarefas de inspeção sem empecilhos, consideramos que a segunda versão do simulador *SRS* é válida para utilização por avaliadores. Por isso, consideramos a segunda versão como suficiente para divulgação (como previsto no modelo de desenvolvimento).

5. Conclusões

Este estudo descreveu e avaliou uma ferramenta computacional para demonstração e simulação do estado de acessibilidade encontrado por usuários cegos durante a navegação em sítios na *Web*. Nossa abordagem visou simplificar a compreensão do uso de leitor de telas em páginas *Web* para avaliadores humanos videntes. Neste sentido, visamos traduzir os resultados auditivos de um leitor de telas em resultados visuais. Neste sentido, buscamos responder à seguinte questão de pesquisa:

Questão de Pesquisa: Como auxiliar avaliadores humanos na interpretação de resultados diagnosticados de problemas de acessibilidade gerados por ferramentas automáticas de avaliação?

Entendemos que a criação de ferramentas de simulação, como a *SRS*, ajude avaliadores humanos na interpretação de resultados diagnósticos de problemas de acessibilidade gerados por ferramentas automáticas de avaliação. A prova de conceito conduzida neste estudo mostrou que a *SRS* é capaz de ajudar tais avaliadores na compreensão de questões de acessibilidade para usuários cegos, ou com outra deficiência visual, envolvidas na interação dos mesmos com páginas da *Web*. Observamos que a ferramenta auxiliou o avaliador vidente a compreender melhor a perspectiva de usuários do respectivo perfil a partir de dicas visuais, traduzindo os aspectos lidos por leitores de tela em elementos visuais gerados pela ferramenta. Portanto, suporta a hipótese deste estudo.

Os resultados deste estudo indicam que o *SRS* possui potencial de mitigar o efeito *expertise* [7] na interpretação de resultados de avaliação de acessibilidade em páginas da *Web*. Isso se justifica pois o avaliador em questão utilizou dos resultados exibidos na interface da *SRS* para descrever os problemas de acessibilidade encontrados. Por isso, compreendemos que, de fato, a interface do *SRS* foi utilizada durante o processo cognitivo do avaliador de percepção dos problemas.

O presente estudo objetivou explorar as características qualitativas do uso do *SRS*, com destaque aos benefícios identificados pelo uso do mesmo. Por isso, nosso estudo abordou uma amostra mínima de avaliadores de acessibilidade. Mesmo assim, entendemos que as conclusões dispostas neste estudo permanecem, pois possuem caráter exploratório e qualitativo. Ainda, nossas conclusões estão relacionadas às características de uma ferramenta inovadora para a literatura. Trabalhos futuros podem explorar o caráter quantitativo da eficiência do *SRS*, considerando avaliação empírica da mesma com amostras de avaliadores. Isto poderá indicar à literatura se o uso da *SRS* é benéfico para a maioria dos avaliadores, ou apenas para determinado perfil de avaliadores. Ainda, sugerimos que estudos futuros investiguem a possibilidade de junção do simulador *SRS* com relatórios de ferramentas automáticas, a fim de facilitar o reconhecimento dos resultados que necessitem de análise humana.

6. Agradecimentos

Agradecimentos aos processos nº 2017/15239-0 e nº 2015/24525-0, da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), que financiaram esta pesquisa.

Referências

- [1] H. Alkhatib et al., “IEEE CS 2022 report,” IEEE Computer Society, pp. 25–27, 2014.
- [2] D. A. Norman, “Natural user interfaces are not natural,” *Interactions*, vol. 17, no. 3, pp. 6–10, May 2010.
- [3] ISO/IEC 25066, “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,” no. ISO/IEC 25066:2016(en). 2016.
- [4] J. R. Lewis, “Usability: Lessons Learned ... and Yet to Be Learned,” *International Journal of Human–Computer Interaction*, vol. 30, no. 9, pp. 663–684, Sep. 2014.
- [5] A. de Lima Salgado, L. A. do Amaral, P. C. Castro, and R. P. de Mattos Fortes, “Designing for Parental Control: Enriching Usability and Accessibility in the Context of Smart Toys,” in *Computing in Smart Toys*, J. K. T. Tang and P. C. K. Hung, Eds. Cham: Springer International Publishing, 2017, pp. 103–125.
- [6] L. C. Serra, L. P. Carvalho, L. P. Ferreira, J. B. S. Vaz, and A. P. Freire, “Accessibility Evaluation of E-Government Mobile Applications in Brazil,” *Procedia Comput. Sci.*, vol. 67, pp. 348–357, Jan. 2015.
- [7] G. Brajnik, M. Vigo, Y. Yesilada, and S. Harper, “Group vs Individual Web Accessibility Evaluations: Effects with Novice Evaluators,” *Interact. Comput.*, vol. 28, no. 6, pp. 843–861, Nov. 2016.
- [8] G. Brajnik, Y. Yesilada, and S. Harper, “The Expertise Effect on Web Accessibility Evaluation Methods,” *Human–Computer Interaction*, vol. 26, no. 3, pp. 246–283, Aug. 2011.
- [9] M. L. P. França-Freitas and M. S. C. A. Gil, “O desenvolvimento de crianças cegas e de crianças videntes,” *Revista Brasileira de Educação Especial*, vol. 18, pp. 507–526, Set. 2012.
- [10] R. Rutter et al., “Web Accessibility: Web Standards and Regulatory Compliance,” Apress, 2006.
- [11] F. D. N. Grillo and R. P. M. Fortes, “Tests with Blind Programmers Using AWMo: An Accessible Web Modeling Tool,” in *Universal Access in Human-Computer Interaction. Design and Development Methods for Universal Access*, Heraklion, Crete, Greece: Springer, 2014, pp. 104–113.
- [12] H. Petrie, A. Savva, and C. Power, “Towards a Unified Definition of Web Accessibility,” in *Proceedings of the 12th Web for All Conference, 2015*, pp. 35:1–35:13.
- [13] G. A. D. Lucca, A. R. Fasolino, and P. Tramontana, “Web Site Accessibility: Identifying and Fixing Accessibility Problems in Client Page Code,” in *Seventh IEEE International Symposium on Web Site Evolution, 2005*, pp. 71–78.
- [14] World Wide Web Consortium (W3C), “Essential Components of Web Accessibility,” 2005. [Online]. Available: <http://www.w3.org/WAI/intro/components.php>.
- [15] B. Kelly, D. Sloan, L. Phipps, H. Petrie, and F. Hamilton, “Forcing Standardization or Accommodating Diversity?: A Framework for Applying the WCAG in the Real World,” in *Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A), Chiba, Japan, 2005*, pp. 46–54.

- [16] A. P. Freire, R. Goularte, and R. P. de Mattos Fortes, “Techniques for developing more accessible web applications: a survey towards a process classification,” in Proceedings of the 25th annual ACM international conference on Design of communication, 2007, pp. 162–169.
- [17] T. M. Tangarife and C. Mont’Alvão, “Acessibilidade Web: Um Estudo Exploratório Do Conhecimento Do Desenvolvedor Web Brasileiro.” ATIID, 2005.
- [18] F. Godinho, C. Santos, A. F. Coutinho, and P. Trigueiros, “Tecnologias de Informação sem Barreiras no Local de Trabalho,” Instituto de Desenvolvimento e Inspeção das Condições de Trabalho, Vila Real, 2004.
- [19] T. Moreira Tangarife and T. M. Tangarife, “A Acessibilidade Nos Websites Governamentais: Um Estudo De Caso No Site Da Eletrobrás.” .
- [20] J. M. Nicácio, Técnicas de acessibilidade: criando uma Web para todos. Maceió - AL: Edufal - Editora da Universidade Federal de Alagoas, 2010, p. 100.
- [21] SDH/PR, Secretaria de Direitos Humanos da Presidência da República, Tecnologia Assistiva, 1st ed. Comitê de Ajudas Técnicas, 2009, p. 138.
- [22] R. J. Geraldo and R. P. Fortes, “Dificuldades De Usuários Cegos Na Interação Com A Web: Uma Análise Sobre As Pesquisas,” Revista de Sistemas e Computação - RSC, vol. 3, no. 2, Dec. 2013.
- [23] C. Asakawa, “What’s the web like if you can’t see it?,” in Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A), 2005, pp. 1–8.
- [24] H. Takagi, C. Asakawa, K. Fukuda, and J. Maeda, “Accessibility Designer: Visualizing Usability for the Blind,” SIGACCESS Access. Comput., no. 77–78, pp. 177–184, Sep. 2003.
- [25] World Wide Web Consortium (W3C), “A vocabulary and associated APIs for HTML and XHTML,” 2012. [Online]. Available: <http://www.w3.org/TR/2012/CR-html5-20121217/>.
- [26] I. Sommerville, “Engenharia de Software,” São Paulo: Pearson Prentice Hall, 2011.
- [27] Y. Borodin, J. P. Bigham, G. Dausch, and I. V. Ramakrishnan, “More Than Meets the Eye: A Survey of Screen-reader Browsing Strategies,” in Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A), Raleigh, North Carolina, 2010, pp. 13:1–13:10.