

Auxiliando usuários leigos na produção de conteúdo acessível

Guilherme Gonçalves¹, Daniela Trevisan¹, Luciana Salgado¹, José Viterbo¹

¹Instituto de Computação – Universidade Federal Fluminense
24210-310 – Boa Viagem, Niterói – RJ – Brasil

{galves,daniela,luciana,viterbo}@ic.uff.br

Abstract. *One of the biggest difficulties related to Web 2.0 content production today is how to create these contents and support them in the long-term for visually impaired users. In this sense, this work presents a comparative analysis of three main online WYSIWYG HTML editors useful for non-experts users to produce accessible content. We consider in our method, relevant characteristics from the editors and data collected in user tests. Results lead us to understand that the communication of accessible content conformity is not yet completely clear for non-experts users and new strategies are needed to improve and support the production quality.*

Resumo. *Uma das maiores dificuldades atuais relacionadas à produção de conteúdos da Web 2.0 está em como criá-los e mantê-los acessíveis a longo prazo para usuários deficientes visuais. Neste sentido, este trabalho apresenta uma análise comparativa dos três principais editores HTML WYSIWYG, úteis na produção de conteúdo acessível. Consideramos em nossa análise, características relevantes dos editores e dados coletados em testes com usuários. Os resultados nos levam a entender que a comunicação sobre a conformidade dos conteúdos acessíveis ainda não é completamente clara para usuários que não sejam especialistas e são necessários novas estratégias para melhorar e suportar a qualidade da produção.*

1. Introdução

Em uma época em que a geração de conteúdo acessível tem sido alvo de estudos em vários domínios, como internet [Minin et al. 2015b, do Sacramento et al. 2014, Viganico and Bagatini 2014], TV digital interativa [Piccolo et al. 2007, Médola 2009, Rodrigues et al. 2010] e até mesmo cinema digital [Domingues et al. 2016], os usuários deixam o papel exclusivo de consumidores e têm em suas mãos a capacidade de produzir conteúdos na Web [Moreno et al. 2008]. Desta forma, enfrentamos um grande problema a longo prazo: a manutenção da acessibilidade para pessoas com deficiência.

A principal vantagem oferecida pela Web 2.0 para esses usuários é a variedade de ferramentas de criação, tal como *Content Management Systems* (CMS) e editores visuais de conteúdo HTML. Essas ferramentas permitem que os usuários criem seu conteúdo web de maneira fácil, sem conhecimentos técnicos de linguagens de programação (JavaScript, PHP), de marcação (HTML) ou até mesmo diretrizes de acessibilidade [Minin et al. 2015b].

Nesse contexto, podemos citar algumas diretrizes voltadas para conteúdos Web: WCAG 2.0 (*Web Content Accessibility Guidelines*) [W3C 2008], WAI-ARIA 1.0 (Ac-

cessible Rich Internet Applications Suite) [W3C 2014], *Section 508* [USA 2001] e IBM-WAC (*IBM Web Accessibility Checklist*) [IBM 2009], eMAG (Modelo de Acessibilidade em Governo Eletrônico) [Brasil 2014].

Elas guiam os produtores e desenvolvedores, fornecendo regras e ajudando a manter o conteúdo produzido sempre acessível de acordo com as especificações [Moreno et al. 2008]. No entanto, é mais difícil manter um website no mesmo padrão de acessibilidade ao longo do tempo do que construí-lo [do Sacramento et al. 2014] e no geral, o tempo e recursos necessários para o treinamento de pessoas são consideráveis altos [Antonelli and Fortes 2015].

Mesmo com o apoio das diretrizes e a manutenção da acessibilidade, a conformidade na internet ainda permanece baixa [Power et al. 2012]. Isso pode ser encarado como problemas na comunicação e entendimento das definições das regras, levando a dificuldades na execução da implementação e à falta de ferramentas que auxiliem de uma maneira mais intuitiva a produção e manutenção desse conteúdo [Moreno et al. 2008, Minin et al. 2015b, Swallow et al. 2016]. Porém, produzir conteúdo acessível não basta, precisa-se de conteúdos que além de acessíveis, sejam úteis. Ou seja, não é suficiente atender às regras de conformidades, é preciso que esta informação seja também útil para o usuário final [Power et al. 2012].

Editores HTML WYSIWYG (*What you see is what you get*), caracterizados por aplicações com a capacidade de visualizar o conteúdo da forma que ele será entregue [Viganico and Bagatini 2014], têm se mostrado ferramentas com potencial neste tema pela possibilidade de terem *plugins* de verificação da conformidade do conteúdo produzido durante a sua escrita.

Desta forma, o objetivo deste trabalho é fazer uma análise da situação atual de como usuários, não especialistas em acessibilidade, produzem conteúdo acessível e com isso, evidenciar possíveis problemas e recomendações quanto ao uso desses editores. Uma metodologia foi adotada para inspecionar os editores HTML WYSIWYG que possuam recursos de acessibilidade e avaliá-los em testes de usabilidade com usuários, a fim de descobrir o desempenho, a satisfação e a qualidade (utilidade da informação).

A próxima seção destaca algumas pesquisas importantes sobre acessibilidade e a produção de conteúdo acessível feitas na última década. Depois, na seção 3, é abordada a pesquisa, levando em consideração a inspeção dos editores HTML WYSIWYG e o procedimento dos testes com usuários para a avaliação das ferramentas. Na seção 4, é aberto um espaço para a apresentação dos resultados obtidos e a discussão, onde algumas recomendações também são feitas para definir em qual contexto de uso cada editor se encaixa melhor. Finalmente, na última seção, são expostas algumas considerações, limitações e planos para trabalhos futuros.

2. Acessibilidade, a produção de conteúdo e a situação atual da Web

As pesquisas com foco em acessibilidade na última década têm buscado entender como a ligação entre websites e conteúdos publicados são vistos e compreendidos pelos internautas, inclusive os deficientes visuais. Por outro lado, a investigação da adoção, eficiência e importância das diretrizes também ocorre em paralelo.

Algumas pesquisas exploram o tema, propondo estudos para o entendimento

da situação atual: impactos na inclusão de pessoas deficientes à diversidade de informações, aos serviços disponíveis e a dificuldade da manutenibilidade ao longo prazo de conteúdos acessíveis [Power et al. 2012, do Sacramento et al. 2014, Swallow et al. 2016]. Outras abordam a concepção de estratégias para se lidar com as diretrizes de acessibilidade por meio do desenvolvimento de ferramentas ou técnicas aplicadas aos recursos tecnológicos, visando auxiliar os produtores de conteúdo [Moreno et al. 2008, Viganico and Bagatini 2014, Minin et al. 2015a, Minin et al. 2015b].

Para os desenvolvedores do Portal da Casa de Oswaldo Cruz¹ e vencedores do Prêmio Nacional de Acessibilidade², por meio da experiência relatada, é notável o desafio de se manter o conteúdo do site acessível, especialmente quando o produtor de conteúdo tem pouco ou nenhum conhecimento de linguagens e padrões da Web, incluindo acessibilidade [do Sacramento et al. 2014].

Power et al. registraram em seus achados que a conformidade de *websites* às regras da WCAG 2.0 não garantiria que os usuários teriam menos problemas ou que tornaria o conteúdo mais acessível para um número maior de pessoas com deficiência, visto que nem todos os casos são cobertos. Esses fatos, segundo eles, estão ligados a três grandes problemas enfrentados pelos usuários: a inexistência de regras suficientes, a não implementação das regras especificadas e a ineficácia da cobertura delas. Em adição, os resultados do estudo mostraram baixa conformidade de acessibilidade nos *websites* escolhidos, indicando que os profissionais ainda tem que fazer um grande esforço para criar e mantê-los. Segundo eles, isto pode acontecer devido à falta de entendimento dos padrões, ferramentas de apoio ou por algum dos três problemas destacados [Power et al. 2012].

Ainda neste ponto, Swallow et al. investigam as possíveis causas desses problemas durante o desenvolvimento de *websites*. Segundo eles, a forma como as diretrizes expõem as definições das regras pode ser confusa e isto acaba tornando mais árdua a tarefa destes profissionais, que não tem uma orientação prática de como proceder [Swallow et al. 2016].

Moreno et al. abordam, em um método mais técnico, uma maneira para tentar garantir a acessibilidade em conteúdos produzidos por usuários não experientes. Fazendo uso de modelos acessíveis e elaborando estruturas e anotações semânticas, eles desenvolveram toda uma arquitetura voltada para garantir a manutenção da acessibilidade e auxiliar os usuários durante o processo de edição [Moreno et al. 2008].

Minin et al. investigam a adequação de editores WYSIWYG em relação às diretrizes internacionais WCAG 2.0 e ATAG (*Authoring Tool Accessibility Guidelines*) 1.0. No estudo, eles identificam as características importantes (ex: descrição de figuras, legenda de tabelas, etc) e implementam um editor WYSIWYG protótipo com a verificação de problemas de acessibilidade. Entretanto, com os resultados, os autores notam que a forma de comunicar os erros ainda precisa ser melhorada, já que os testes com usuários menos experientes mostraram que a maioria dos participantes não entenderam as dicas fornecidas [Minin et al. 2015b].

Em outro trabalho, Minin et al. investigam técnicas apropriadas para o desenvol-

¹Portal da Casa de Oswaldo Cruz. <http://www.coc.fiocruz.br/> Acesso em 28/10/2017.

²Prêmios organizados pelo governo brasileiro e pelo escritório local do W3C para incentivar o desenvolvimento de sites acessíveis.

vimento de uma ferramenta que produz um exame ENEM (Exame Nacional do Ensino Médio) por meio digital. A pesquisa contou com o uso das diretrizes para suportar a acessibilidade do conteúdo da prova, levando em conta necessidades específicas de candidatos com deficiência visual total e parcial. A contribuição pode ser vista como a aplicação prática da produção de conteúdo acessível em um caso real, onde os produtores trabalhavam nas questões do exame, sendo guiados por ferramentas orientadas pelas regras [Minin et al. 2015a].

Ainda no contexto da educação, Viganico e Bagatini realizam um estudo acerca da acessibilidade, usabilidade e deficiência visual para a definição de critérios na escolha de editores WYSIWYG acessíveis que possam ser aplicados em ambientes virtuais de aprendizagem. Os editores indicados, TinyMCE [Ephox 2017b] e CKEditor [CKSource 2017], foram os que melhor se encaixaram na avaliação. Segundo os autores, eles possuem confiabilidade em suas comunidades de desenvolvimento, recursos de acessibilidade, bom desempenho quando carregado em uma página, qualidade e documentação do código fonte que viabilizam o desenvolvimento de melhorias e, também, a compatibilidade com os principais navegadores de Internet. Em adição, conclui-se que essas ferramentas são importantes para a inclusão de pessoas deficientes nesses ambientes [Viganico and Bagatini 2014].

Todo esse cenário reforça a relevância das pesquisas sobre ferramentas de suporte à construção de conteúdos acessíveis, especialmente considerando a complexidade das diretrizes e regras, em contraste com a forte e crescente presença de usuários leigos atuando nesse domínio.

3. Metodologia

O principal objetivo deste estudo consiste em responder duas questões: (1) Editores HTML WYSIWYG com *plugins* de verificação de conformidade realmente auxiliam usuários não experientes na produção de conteúdo acessível? (2) Essas ferramentas melhoram a qualidade do que é produzido, evidenciando ou alertando os usuários sobre possíveis problemas de usabilidade do conteúdo? Para responder tais questões o estudo foi organizado em três etapas.

A primeira etapa foca na inspeção comparativa dos principais editores HTML WYSIWYG disponíveis com características relevantes neste cenário de produção e validação da conformidade dos conteúdos. Em seguida, avalia-se por meio de teste de usabilidade com usuários leigos, onde dados de desempenho, satisfação e qualidade do conteúdo são coletados. Por fim, o conteúdo proveniente dos testes de usabilidade são submetidos a uma validação automática de acessibilidade usando a ferramenta daSilva.org³

3.1. Inspeção dos editores com suporte a edição de conteúdo acessível

Levando em consideração os principais editores HTML WYSIWYG existentes como: Froala editor, ContentTools, Summernote, Redactor, NicEdit, YUI Rich Text Editor, OpenWYSIWYG, jHtmlArea, Mercury Editor, CKEditor, TinyMCE, Textbox.io [1stwebdesigner.com 2015, sitepoint.com 2016], selecionamos apenas aqueles que tem suporte à produção de conteúdo acessível, sendo eles: CKEditor [CKSource 2017], TinyMCE [Ephox 2017b] e Textbox.io [Ephox 2017a].

³<http://www.dasilva.org.br/> Acesso em 28/10/2017.

A inspeção teve como intuito verificar como esses editores HTML WYSIWYG selecionados atendem às características desejáveis para a produção e validação da conformidade dos conteúdos (Veja a Tabela 1).

Tabela 1. Características avaliadas nos editores, onde (✓) contempla, (-) não contempla e (?) não foi encontrado

Características avaliadas	CKEditor	TinyMCE	Textbox.io
Disponível nas línguas mais faladas do mundo	✓	✓	✓
Suporte oficial dos navegadores de internet atuais	✓	✓	✓
Fácil de ser embutido em uma página Web	✓	✓	✓
Formatação de texto	✓	✓	✓
Inserção de imagens	✓	✓	✓
Inserção de tabelas	✓	✓	✓
Inserção de fórmulas matemáticas	✓	✓	?
Fórmulas matemáticas estão no padrão da WCAG	✓	✓	?
Fórmulas são interpretadas por leitores de tela	✓	✓	?
Conformidade às diretrizes da WCAG	✓	✓	✓
Conformidade às diretrizes da 508 Section	✓	✓	✓
Conformidade às diretrizes da IBM WAC	✓	?	?
Conformidade às diretrizes da WAI-ARIA	✓	?	✓
Plugin verificador de acessibilidade	✓	✓	✓
Plugin verificador de acessibilidade gratuito	✓	-	✓

3.2. Procedimento dos testes com participantes

O procedimento contou com a intercalação da ordem dos editores entre os participantes, a fim de amenizar o efeito de aprendizagem [Rubin. 1994]. O objetivo da avaliação é verificar a qualidade da produção de conteúdo acessível nos editores mencionados em uma questão de vestibular. Além disso, foram coletadas informações de satisfação, desempenho, complexidade das dicas fornecidas pelo recurso de verificação de acessibilidade e a preferência geral de cada participante entre as três ferramentas, utilizando uma estrutura de escala Likert variando de 1 a 5.

Dez pessoas participaram do experimento: 6 respondentes do sexo masculino e 4 respondentes do sexo feminino com idades variando de 19 a 32 anos (M: 24 anos e DP: 4,37). Primeiramente, foi solicitado aos participantes o preenchimento do termo de consentimento livre e esclarecido, concedendo a permissão para coleta de dados durante o estudo. Em seguida, eles forneceram informações para o mapeamento do perfil dos voluntários no experimento. No geral, todos possuíam conhecimentos de inglês para leitura de manuais e documentações, mas nenhum deles conheciam sobre regras e padrões de acessibilidade. Todos os participantes informaram utilizar regularmente computadores e editores de texto no dia a dia.

O teste consistiu na realização de cinco atividades, em que o participante deveria inserir um conteúdo pré-fornecido no editor, abrangendo: (1) formatação de texto, (2) inserção de figuras, (3) escrita de fórmula matemática, (4) composição de tabela de dados e (5) verificação se o conteúdo está adequado aos padrões através de um recurso do editor.

Esta última atividade foi desconsiderada no editor TinyMCE, por ser um recurso pago. Esperava-se que os participantes usassem as dicas de acessibilidade fornecidas pelo editor para criar conteúdo acessível, então nenhuma dica foi incluída no material fornecido pelo avaliador.

Após o término das atividades com cada editor, o participante preencheu um formulário de satisfação com duas perguntas, a primeira aborda a facilidade no uso dos recursos do editor, a fim de verificar se o usuário ficou satisfeito com a utilização dos recursos; a segunda aborda a complexidade das dicas fornecidas pelo recurso de verificação de problemas de acessibilidade. Ao final do teste, o usuário também informou qual o editor preferido, baseado na sua experiência obtida durante o teste.

4. Resultados e Discussão

Os resultados alcançados mostram um pouco sobre a situação atual da acessibilidade em conteúdos, permitindo que os dados de usabilidade, qualidade da acessibilidade e os questionamentos da pesquisa destaquem pontos de discussão sobre o tema, além de mostrar recomendações sobre o contexto de uso dos editores.

4.1. Resultados da avaliação de usabilidade

Os critérios de usabilidade verificados foram o desempenho dos participantes ao realizar as tarefas e a satisfação. Avaliação de desempenho revela que as médias (M) e desvios padrões (DP) da duração das atividades, em minutos, entre o CKEditor (M: 10,54; DP: 5,07), TinyMCE (M: 8,64; DP: 4,37) e Textbox.io (M: 6,73; DP: 2,05) foram bem diferentes. Para a atividade 5, em que o editor TinyMCE foi desconsiderado, o CKEditor e o Textbox.io mostraram, respectivamente, tempos similares: M: 15,47; DP: 6,97 e M: 13,54; DP: 7,07. Em vista desses dados e pelo o que foi observado durante os testes, a duração das atividades não assegurou que um editor fosse melhor ou pior, já que os usuários poderiam demorar mais em situações de descrição dos atributos acessíveis, entendimento das dicas fornecidas ou até mesmo por estarem com dificuldade.

Para a satisfação, levando em consideração a facilidade de uso dos recursos em uma escala de 1 (muito fácil) a 5 (muito difícil), todas as ferramentas ficaram posicionadas entre o intervalo 2 (fácil) e 3 (neutro), e registraram as seguintes médias: 2,6 no CKEditor, 2,2 no TinyMCE e 2,5 no Textbox.io. Os dados também mostram que os editores foram avaliados mais frequentemente nestes intervalos e observa-se também que o tempo de duração geral das atividades para os participantes não afetou suas opiniões. Além disso, as avaliações da complexidade das dicas fornecidas pelo recurso de verificação, variando de 1 (muito simples) a 5 (muito complexo), mostram que ambas as ferramentas ficaram posicionadas entre as complexidades 2 (simples) e 3 (neutro) com médias de 2,4 para o CKEditor e 2,8 para o Textbox.io. Por fim, no geral, a quantidade média de pedidos de ajuda entre os participantes em relação ao CKEditor (M: 1,0), TinyMCE (M: 1,1) e Textbox.io (M: 2,2) permaneceu baixa.

A preferência informada esclarece que o CKEditor e o TinyMCE foram eleitos os editores preferidos, por respectivamente 50% e 40% dos participantes. Apesar disso, o CKEditor teve a maior média de tempo para a realização das tarefas e a da escala da facilidade de uso, indicando uma maior dificuldade. Esse fato pode ter relação com a forma em que as fórmulas eram escritas no *plugin* oficial, pois deveriam ser em Tex, tornando a

tarefa mais demorada para os participantes, que utilizaram uma boa parte do tempo para o entendimento da linguagem. Por outro lado, o TinyMCE explora esse recurso de outra forma, utilizando um *plugin* de terceiros com atalhos de equações predefinidas para o preenchimento dos valores e fornecendo o resultado em Tex e o Textbox.io o faz utilizando caracteres especiais.

4.2. Resultados da avaliação da qualidade da acessibilidade

Os critérios foram: nível de atendimento às diretrizes de acessibilidade e a usabilidade do conteúdo. Este último critério está relacionado com o fator de utilidade do conteúdo.

O experimento realizado mostrou que a grande maioria dos conteúdos foram concluídos parcialmente, ou seja, ainda continham problemas de acessibilidade (falta de descrição nas figuras, falta de legenda e resumo na tabela). Isto leva a crer que o recurso de verificação ainda não é suficiente para que usuários leigos produzam conteúdos acessíveis.

Outro ponto observado foi que os usuários inseriam os atributos de acessibilidade mais frequentemente durante a atividade de verificação (conforme a Figura 1). Isso já era esperado, pois sendo usuários menos experientes a tendência seria que o recurso de verificação fornecesse as dicas apropriadas para a correção. No que se refere aos três atributos avaliados, a presença da descrição nas figuras foi bem similar entre os editores, já as legendas e resumos de tabelas foram bem mais presentes no Textbox.io do que no CKEditor. O que mostra uma relação com a observação levantada por [Swallow et al. 2016] sobre uma linguagem não tão clara presente nas diretrizes e consequentemente nas dicas.

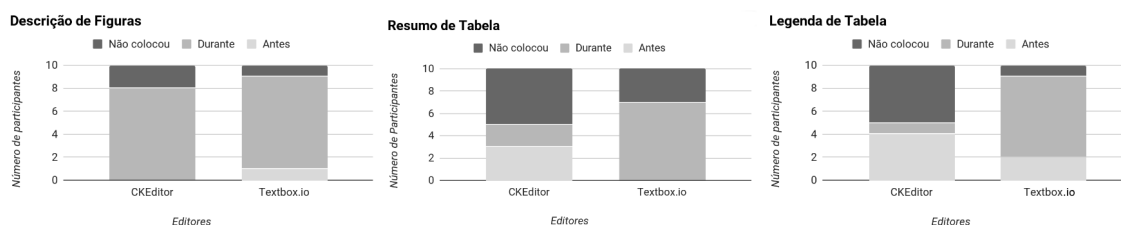


Figura 1. Análise do uso dos atributos acessíveis antes e durante a atividade de verificação

Sobre as fórmulas matemáticas, não foram colocadas na avaliação da acessibilidade pois nenhum dos recursos de verificação ofereceu dicas sobre tal item. Acredita-se que a falta de dicas pode ser ocasionada pela não implementação de regras específicas, um dos problemas apontados por [Power et al. 2012], ou pelas fórmulas serem tratadas como texto. Outro fato relacionado está no modo como as fórmulas podem ser implementadas nos editores, permitindo que plugins, como MathJax⁴, utilizem Tex ou MathML como entrada ou até mesmo a transformação delas em imagens, colocando o texto das fórmulas nas descrições, como faz o plugin WIRIS⁵.

Além disso, a avaliação automática do conteúdo através do site daSilva.org mostrou que a quantidade de erros e avisos da produção dos participantes foi relativamente baixa. A média dos erros reportados para os editores foi de: 5,8 (CKEditor); 9,2

⁴<https://www.mathjax.org/> Acesso em 28/10/2017.

⁵<http://www.wiris.com/> Acesso em 28/10/2017

(TinyMCE) e 4,9 (Textbox.io), enquanto a média de avisos, respectivamente, foi de: 4,1 (CKEditor); 6,0 (TinyMCE) e 4,0 (Textbox.io). Percebe-se que o CKEditor e o Textbox.io foram bem similares e o TinyMCE apresentou uma maior quantidade de erros e avisos, devido a falta do recurso de verificação.

De posse desses resultados pode-se, então, responder o questionamento (1). Observa-se a grande dificuldade para produzir conteúdos acessíveis, mesmo utilizando ferramentas com um recurso, cujo foco é auxiliar os usuários. A existência de erros, mesmo sendo em baixa quantidade, indica que as opções atuais ainda não fornecem um produto perfeitamente acessível.

Outro fato é que mesmo quando alguns usuários entendiam os problemas indicados pelo recurso de verificação, eles não inseriam descrições de figuras, legendas ou mesmo resumos de tabela com informações úteis para deficientes visuais (veja a Figura 2). Esses dados respondem ao questionamento (2), mostrando que nenhuma das ferramentas analisadas apresenta suporte ou dicas de como os produtores leigos poderiam fornecer conteúdo, além de acessível, útil para os deficientes visuais.

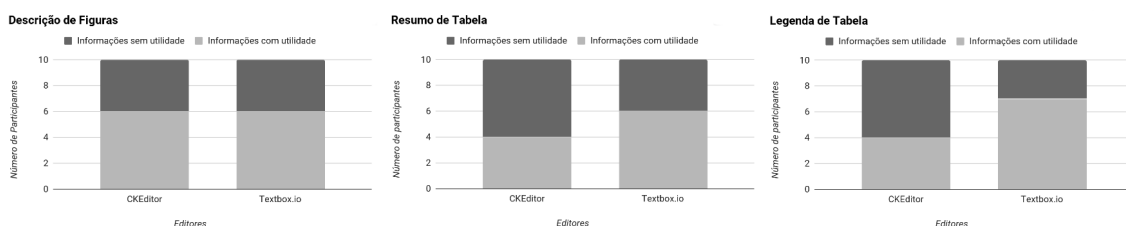


Figura 2. Quantidade de participantes que adicionaram informações úteis para usuários deficientes visuais nos campos esperados

5. Considerações finais e recomendações aos produtores de conteúdo

Este estudo avaliou três editores online HTML WYSIWYG, de modo a verificar a situação atual de conteúdo acessível criado por usuários leigos e responder aos questionamentos.

Os resultados mostram que tanto o CKEditor, quanto o TinyMCE fornecem a grande maioria de recursos necessários, além de terem sido os mais preferidos pelos participantes. Sem deixar de lado o Textbox.io, cujo foco é a utilização em dispositivos móveis, este também possui boa parte dos recursos necessários, incluindo a verificação de problemas de acessibilidade. Entretanto, o Textbox.io por ser um *software* recente, ainda lhe faltam recursos e *plugins* de extensão para concorrer com os outros dois, já maduros neste mercado.

Produtores em geral podem se beneficiar com o uso desses editores onde o principal diferencial está na funcionalidade de verificação de problemas de acessibilidade, cujo recurso o CKEditor e o Textbox.io contemplam de forma gratuita, enquanto o TinyMCE por meio de uma assinatura mensal.

Com as respostas aos questionamentos, observa-se que os editores HTML WYSIWYG investigados auxiliam na produção de conteúdo acessível, mas que ainda não existe uma solução que forneça um conteúdo livre de problemas. Acredita-se que o processo de avaliação desses problemas ainda está em andamento, uma vez que as ferra-

mentas de criação mais populares estão adotando cada vez mais as recomendações das diretrizes de acessibilidade e da usabilidade do conteúdo.

Deste modo, conclui-se que seria necessário mediar através de ferramentas com recursos guiados por regras, tal como editores HTML WYSIWYG, a fim de que usuários leigos produzam conteúdos acessíveis, como por exemplo exames educacionais digitais [Minin et al. 2015a] e ambientes virtuais de aprendizagem [Viganico and Bagatini 2014], e mantenham estes acessíveis ao longo do tempo.

Esta pesquisa apresentou a limitação de não ter incluído o recurso de verificação para o editor TinyMCE, por ser um item pago. Assim, tanto a avaliação e a comparação acabou sendo prejudicada.

Com o conhecimento adquirido neste estudo espera-se, em trabalhos futuros, poder avançar em soluções para melhor orientar e guiar os produtores na geração de conteúdo acessível em editores HTML WYSIWYG no contexto de exames educacionais.

6. Agradecimentos

Agradecemos aos participantes que contribuíram para o desenvolvimento deste estudo.

Referências

- 1stwebdesigner.com (2015). 10 best wysiwyg html editor collection from 2016. <http://1stwebdesigner.com/best-wysiwyg-html-editor/>. Acesso em: 10/06/2017.
- Antonelli, H. L. and Fortes, R. P. d. M. (2015). A support for developers implement the accessibility guidelines regarding to web menus. In *Proceedings of the 21st Brazilian Symposium on Multimedia and the Web*, WebMedia '15, pages 33–40, New York, NY, USA. ACM.
- Brasil (2014). Recomendações de acessibilidade para construção e adaptação de conteúdos do governo brasileiro na internet. <https://www.governoeletronico.gov.br/documentos-e-arquivos/eMAGv31.pdf>. Acesso em: 23/06/2017.
- CKSource (2017). Ckeditor. <http://ckeditor.com/>. Acesso em: 23/06/2017.
- do Sacramento, C. C., e Silva, C. S., Conceição, F. I., de Moraes, F. L. D., de Castro, I. M., Zorzanelli, L. P., and Barros, Y. S. (2014). Portal da casa de osvaldo cruz e o prêmio nacional de acessibilidade na web. In *Companion Proceedings of the 13th Brazilian Symposium on Human Factors in Computing Systems*, IHC '14, pages 5–8, Porto Alegre, Brazil, Brazil. Sociedade Brasileira de Computação.
- Domingues, L. A., Campos, V. P., Araújo, T. M., and S. Filho, G. L. d. (2016). Accessibility in digital cinema: A proposal for generation and distribution of audio description. In *Proceedings of the 22Nd Brazilian Symposium on Multimedia and the Web*, Web-media '16, pages 119–126, New York, NY, USA. ACM.
- Ephox (2017a). Textbox.io. <https://textbox.io/>. Acesso em: 23/06/2017.
- Ephox (2017b). TinyMce. <https://www.tinymce.com/>. Acesso em: 23/06/2017.
- IBM (2009). Ibm web accessibility checklist. <http://www-03.ibm.com/able/guidelines/web/accesswebtest.html>. Acesso em: 16/06/2017.

- Médola, A. (2009). Televisão digital brasileira e os novos processos de produção de conteúdos: os desafios para o comunicador. *Revista da Associação Nacional dos Programas de Pós-Graduação em Comunicação, E-Compós, Brasília-DF*, 12(3):1–12.
- Minin, H., Trevisan, D. G., and Viterbo, J. (2015a). Enem inclusivo: identificando técnicas para a realização assistida pelo computador. In *Proceedings of the XXVI Brazilian Symposium on Computers in Education, SBIE '15*, pages 1–10, Maceió, Alagoas, Brazil. CBIE.
- Minin, H. C., Alemán, J. J., do Sacramento, C., and Trevisan, D. G. (2015b). A wysiwyg editor to support accessible web content production. In *9th International Conference, UAHCI 2015, Held as Part of HCI International 2015*, pages 221–230. Springer International Publishing.
- Moreno, L., Martínez, P., and Ruiz, B. (2008). Guiding accessibility issues in the design of websites. In *Proceedings of the 26th Annual ACM International Conference on Design of Communication, SIGDOC '08*, pages 65–72, New York, NY, USA. ACM.
- Piccolo, L. S. G., Melo, A. M., and Baranauskas, M. C. C. (2007). *Accessibility and Interactive TV: Design Recommendations for the Brazilian Scenario*, pages 361–374. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Power, C., Freire, A., Petrie, H., and Swallow, D. (2012). Guidelines are only half of the story: Accessibility problems encountered by blind users on the web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, pages 433–442, New York, NY, USA. ACM.
- Rodrigues, K. R. H., Melo, E. L., Nakagawa, P. I., and Teixeira, C. A. C. (2010). Interação com conteúdo complementar para apoio ao entendimento de programas televisivos. In *Proceedings of the IX Symposium on Human Factors in Computing Systems, IHC '10*, pages 91–100, Porto Alegre, Brazil, Brazil. Brazilian Computer Society.
- Rubin., J. (1994). *Handbook of usability testing – How to plan, design and conduct effective tests*. New York: Wiley.
- sitepoint.com (2016). 10 best jquery and html5 wysiwyg plugins. <https://www.sitepoint.com/10-best-html-wysiwyg-plugins/>. Acesso em: 10/06/2017.
- Swallow, D., Petrie, H., and Power, C. (2016). Understanding and supporting web developers: Design and evaluation of a web accessibility information resource (webair). *Studies in health technology and informatics*, 229:482–491.
- USA (2001). Section 508. <https://www.section508.gov/content/about-us>. Acesso em: 16/06/2017.
- Viganico, T. E. and Bagatini, D. (2014). Teste de acessibilidade em um editor de texto web. *3º Seminário Nacional de Inclusão Digital (SENID)*, pages 1–10.
- W3C (2008). Web content accessibility guidelines (wcag) 2.0. <https://www.w3.org/TR/WCAG20/>. Acesso em: 25/06/2017.
- W3C (2014). Accessible rich internet applications (wai-aria) 1.0. <https://www.w3.org/TR/wai-aria/>. Acesso em: 25/06/2017.