

Um modelo de acessibilidade para cegos em sistemas de programação visual

Marcos Devaner do Nascimento
(Aluno de doutorado)¹,
Anarosa Alves Franco Brandão
(Orientadora)¹

¹Escola Politécnica, Universidade de São Paulo
Caixa Postal 05508-010 – São Paulo – SP – Brazil

{marcos.devaner, anarosa.brandao}@usp.br

Resumo. Segundo a Organização Mundial de Saúde (OMS), 246 milhões pessoas apresentam perda moderada ou severa da visão. Pessoas com deficiência visual severa (cegos) utilizam-se de tecnologias assistivas, como leitores de tela, para interagir com ambientes virtuais. Para o uso efetivo de ambientes virtuais por cegos, são necessárias uma série de adequações relacionadas à acessibilidade. Essa pesquisa visa, por meio de estudos heurísticos e empíricos, propor um modelo de programação acessível para cegos.

1. Introdução

Segundo a Organização Mundial de Saúde (OMS) [world health organization 2011], 246 milhões de pessoas apresentam perda moderada ou severa da visão. Pessoas com deficiência visual severa (cegos), utilizam-se de tecnologias assistivas, como leitores de tela, para interagir com ambientes virtuais. Esses ambientes estão presentes no cotidiano, incorporando a computação às tarefas diárias por meio de diversos dispositivos como computadores, *smartphones*, *tablets* entre outros.

Para que cegos possam fazer uso efetivo de ambientes virtuais é necessária uma série de adequações relacionadas à percepção do ambiente, navegação e suporte a leitores de tela [Khurana et al. 2018, Kacorri et al. 2017]. Essas adequações devem ser aplicadas com base no tipo e grau de deficiência, levando em considerações as habilidades para interagir com o ambiente [Bowe 2000].

Atualmente, existem algumas diretrizes para tornar ambientes web acessíveis, como as Diretrizes de Acessibilidade para Conteúdo da Web (WCAG), estabelecidas pela instituição que define os padrões adotados pela rede mundial de computadores (W3C)[Consortium et al. 2008]. Essas diretrizes definem uma série de implementações, que devem ser feitas em páginas web, para que tenham amplo acesso, facilitando a navegação e a percepção para pessoas com deficiências.

Embora as diretrizes propostas pela WCAG sejam um caminho para acessibilidade em ambientes web, inclusive aqueles dedicados ao ensino e aprendizado, estudos mostram que alguns dos ambientes virtuais de aprendizagem (AVA) largamente utilizados não aplicam tais diretrizes. Como consequência, pessoas com deficiência ainda encontram uma série de barreiras ao utilizá-los [Buzzi et al. 2009, Calvo et al. 2014, Primo et al. 2017, Calvo et al. 2011].

A problemática de acessibilidade estende-se também para as ferramentas utilizadas integradas ou em conjunto a esses AVA como, por exemplo, ferramentas de apoio a programação visual (ProgV) [do Nascimento et al. 2019]. Essas ferramentas fornecem elementos gráficos para representar comandos e naturalmente evitam várias armadilhas de sintaxe presentes nas linguagens de programação de texto tradicionais. Elas são utilizadas para apoiar o ensino e aprendizado de programação de computadores.

Embora as ferramentas para ProgV têm se mostrado eficientes para ensino e aprendizado de programação de computadores [Chang et al. 2017, Dobesova 2014], indivíduos cegos não podem se beneficiar desse recurso, pois existem limitações relacionadas a manipulação dos componentes e interação com a ferramenta que podem se tornar uma barreira para eles.

Este trabalho propõe realizar uma pesquisa qualitativa com objetivo de descrever, decodificar e analisar problemas e soluções relacionados a aplicação de ProgV para cegos. A pesquisa será guiada pelas seguintes questões: (i) O modelo de programação visual dá suporte ao uso de leitores de tela?; (ii) Quais os problemas de acessibilidade, para cegos, em um modelo de programação visual?; (iii) Quais os impactos gerados pela aplicação de modelo de programação visual acessível para cegos?.

Para responder às questões de pesquisa, é proposta a aplicação análises heurísticas, além de métodos empíricos para avaliar a acessibilidade de ferramentas para ProgV para cegos. Assim, espera-se identificar os desafios de acessibilidade durante a interação e execução de uma tarefa utilizando um modelo de ProgV. Com isso, será feito um mapeamento das barreiras de acessibilidade em ferramentas de ProgV, para então, com base no método *Design Science Research (DSR)* [Pimentel et al. 2019], propor um modelo de programação visual acessível para cegos.

Como resultado, espera-se que tal modelo ajude a diminuir as barreiras de acessibilidade existentes em ferramentas de ProgV e proporcionar à cegos as mesmas oportunidades de aprendizado em relação ao seus homólogos videntes.

2. Motivação para o estudo

Dados os impactos positivos sobre o uso de ferramentas para ProgV no ensino e aprendizado de programação, realizou-se uma pesquisa exploratória, comparando-se as ferramentas Alice, iVProg e Scratch [do Nascimento et al. 2019]. Essa pesquisa teve como objetivo analisar aspectos técnicos, perfil de usuário, além da verificação de recursos de acessibilidade e adequação para cegos. Os resultados mostraram que essas ferramentas não possuem recursos de acessibilidade, além de não existir nenhum indicativo de que estejam adequadas para cegos. Além do estudo exploratório buscou-se, por meio de revisão bibliográfica, entender quais os principais desafios encontrados por cegos em AVA, buscando levantar e compreender as principais barreiras de acessibilidade para cegos nesses ambientes.

Os estudos realizados previamente, tiveram como objetivo levantar as principais barreiras de acessibilidade para cegos em diferentes tipos de ambientes virtuais, além de buscar na literatura possíveis soluções para essas barreiras. Com isso, foi possível entender o contexto de acessibilidade para cegos e construir o estado da arte para este trabalho.

3. Metodologia

Além de estudos bibliográficos para contextualização do problema abordado, realizados previamente, buscou-se a aplicação do método de pesquisa *Design Science Research (DSR)* [Pimentel et al. 2019]. O método DSR é aplicado a pesquisas na área de ciência e tecnologia, tendo como princípio a construção de artefatos tecnológicos para intervenção sobre determinados problemas. O DSR propõe três atividades:

1. **contextualização do problema:** para a contextualização, serão realizadas análises heurísticas de acessibilidade das ferramentas de programação visual Scratch, iVProg e Blockly. Essa análise será feita por um especialista com base nas diretrizes de usabilidade [Nielsen 2005] e acessibilidade [W3C 2008]. Essa avaliação tem como objetivo identificar e compreender os problemas de acessibilidade para cegos, em ambientes de programação visual. Será aplicado o Collab4All [Luque 2019], um método para identificar desafios e soluções de acessibilidade, com base nas diretrizes da WCAG, para incluir pessoas com deficiência em trabalho colaborativo. No contexto dessa pesquisa, entende-se por colaboração a interação do usuário com o ambiente e artefatos para execução de uma tarefa, por exemplo, a construção de um algoritmo, com a eventual intermediação do professor.
2. **Artefato:** Segundo Pimentel et al. [2019], um artefato pode ser um modelo criado a partir das relações de um vocabulário conceitual. Sendo assim, os problemas de acessibilidade encontrados nas análises heurísticas, além das diretrizes propostas pela WCAG, servirão de subsídios para levantamento e proposição de soluções e, conseqüentemente, a criação de um modelo de programação visual para cegos. Este modelo deve aplicar diretrizes de acessibilidade e tecnologias assistivas que possam ser integradas a qualquer ambiente de ProgV. A adoção deste modelo por ferramentas existentes proporcionará a diminuição (ou extinção) das barreiras de acessibilidade, proporcionando à cegos as mesmas oportunidades de aprendizado que indivíduos não cegos.
3. **Avaliação do artefato:** essa avaliação será realizada aplicando métodos empíricos, com usuários cegos. Para isso, serão realizados estudos comparativos entre o modelo acessível, proposta nesse trabalho, e os modelos subjacentes às ferramentas analisadas em estudos prévios. Assim, será possível avaliar o modelo proposto, medindo seu grau de utilidade e eficiência para cegos. Espera-se nessa etapa medir os impactos positivos do modelo sobre o ensino e aprendizado de programação para cegos, além de levantar eventuais melhorias a serem incorporadas pelo modelo.

4. Plano de trabalho e Cronograma

Para o desenvolvimento deste trabalho são propostas as seguintes atividades:

1. Pesquisa exploratória;
2. Publicação da pesquisa [do Nascimento et al.];
3. Análise heurística;
4. Escrita e defesa de exame de qualificação de doutorado;
5. Participação no StudX@CBIE2019;
6. Criação do artefato (modelo de programação visual para cegos);

7. Escrita e submissão de artigo descrevendo o modelo;
8. Análise empírica;
9. Escrita e submissão de artigo descrevendo os resultados da análise empírica;
10. Escrita, depósito e defesa da tese;

	2018				2019			
	Jan-Mar	Abr-Jun	Jul-Set	Out-Dez	Jan-Mar	Abr-Jun	Jul-Set	Out-Dez
At. 1				XXX	XXX			
At. 2						XXX		
At. 3							XXX	XXX
At. 4							XXX	XXX
At. 5								XXX

	2020				2021			
	Jan-Mar	Abr-Jun	Jul-Set	Out-Dez	Jan-Mar	Abr-Jun	Jul-Set	Out-Dez
At. 6	XXX	XXX	XXX					
At. 7			XXX	XXX				
At. 8				XXX	XXX			
At. 9					XXX	XXX		
At. 10			XXX	XXX	XXX	XXX	XXX	XXX

5. Considerações finais

A proposta de trabalho em desenvolvimento, apresentada neste plano, trata-se de uma pesquisa científica baseada no método *Design Science Research*. A pesquisa aborda a temática de ensino inclusivo, tratando como problema o contexto de ensino e aprendizado de introdução a programação e o uso de ferramentas para programação visual. Para esse contexto, espera-se responder as questões de pesquisa propostas neste trabalho. Para respondê-las, serão aplicados uma série de estudos heurísticos e empíricos, além de propor um modelo de programação visual acessível para cegos. Espera-se com essa proposta, diminuir as barreiras de acessibilidade encontradas para ferramentas de programação visual e, com isso, tornar o ensino e aprendizado de introdução a programação cada vez mais inclusivo, proporcionando as mesmas oportunidades de aprendizado para cegos e não-cegos.

Referências

- Bowe, F. (2000). *Universal design in education: Teaching nontraditional students*. Greenwood Publishing Group.
- Buzzi, M. C., Buzzi, M., and Leporini, B. (2009). Accessing e-learning systems via screen reader: an example. In *International Conference on Human-Computer Interaction*, pages 21–30. Springer.
- Calvo, R., Iglesias, A., and Moreno, L. (2011). Is moodle accessible for visually impaired people? In *International Conference on Web Information Systems and Technologies*, pages 207–220. Springer.

- Calvo, R., Iglesias, A., and Moreno, L. (2014). Accessibility barriers for users of screen readers in the moodle learning content management system. *Universal access in the information society*, 13(3):315–327.
- Chang, C.-K., Yang, Y.-F., and Tsai, Y.-T. (2017). Exploring the engagement effects of visual programming language for data structure courses. *Education for Information*, 33(3):187–200.
- Consortium, W. W. W. et al. (2008). Web content accessibility guidelines (wcag) 2.0.
- do Nascimento, M. D., de Souza, L. M., de Oliveira Brandão, A., Félix, I. M., Dantas, D. L., Ferreira, B. M., and de Oliveira Brandão, L. Which visual programming language best suits each school level? a look at alice, ivprog, and scratch.
- do Nascimento, M. D., de Souza, L. M., de Oliveira Brandão, A., Félix, I. M., Dantas, D. L., Ferreira, B. M., and de Oliveira Brandão, L. (2019). Which visual programming language best suits each school level? a look at alice, ivprog, and scratch.
- Dobesova, Z. (2014). E-learning for visual programming language. In *2014 IEEE 12th IEEE International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 103–108. IEEE.
- Kacorri, H., Kitani, K. M., Bigham, J. P., and Asakawa, C. (2017). People with visual impairment training personal object recognizers: Feasibility and challenges. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 5839–5849. ACM.
- Khurana, R., McIsaac, D., Lockerman, E., and Mankoff, J. (2018). Nonvisual interaction techniques at the keyboard surface. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 11. ACM.
- Luque, L. (2019). *Collab4All: a method to foster inclusion in computer-supported collaborative work*. PhD thesis, Universidade de São Paulo.
- Nielsen, J. (2005). Ten usability heuristics.
- Pimentel, M., Filippo, D., and Santoro, F. M. (2019). *Design Science Research: fazendo pesquisas científicas rigorosas atreladas ao desenvolvimento de artefatos computacionais projetados para a educação*, volume 1. Metodologia de Pesquisa Científica em Informática na Educação.
- Primo, L., Ulbricht, V., and Fadel, L. M. (2017). Accessibility in the virtual learning environment moodle identification of problems' class. In *World Conference on Information Systems and Technologies*, pages 571–580. Springer.
- W3C (2008). Web content accessibility guidelines (wcag) 2.0. *W3C Recommendation*, 11.
- world health organization (2011). World report on disability 2011.