

RECURSOS PEDAGÓGICOS PARA O ENSINO DE PROGRAMAÇÃO DE ESTUDANTES COM DEFICIÊNCIA VISUAL: UMA REVISÃO SISTEMÁTICA DA LITERATURA

Rafaela Robe, IFRS Campus Porto Alegre, rafaelavrobe@gmail.com

Bruna Poletto Salton, IFRS Campus Porto Alegre, bruna.salton@ifrs.edu.br

Silvia Bertagnolli, IFRS Campus Porto Alegre, silvia.bertagnolli@poa.ifrs.edu.br

Resumo. O ensino de programação é uma atividade complexa que muitas vezes é abordada em sala de aula utilizando-se de elementos visuais (imagens, vídeos e animações). Porém, o processo de ensino e de aprendizagem fica prejudicado quando um aluno com deficiência visual não consegue acompanhar as atividades realizadas em sala de aula. Desse modo, é essencial utilizar ou propor recursos pedagógicos que viabilizem a inclusão desses estudantes, possibilitando a compreensão dos conceitos apresentados em aula. Este artigo apresenta o percurso e os resultados de uma revisão sistemática da literatura com foco na identificação desse tipo de recurso pedagógico. Os principais resultados encontrados demonstram que os recursos para o ensino de programação relacionam-se às múltiplas possibilidades de percepção, em especial, a auditiva e a tátil.

Palavras-chaves: ensino de programação, estudantes com deficiência visual, recursos pedagógicos.

Pedagogical Resources for Teaching Programming to Students with Visual Disabilities: a systematic literature review

Abstract. *Teaching programming is a complex activity that is often approached in the classroom using visual elements (images, videos and animations). However, the teaching and learning process is compromised when a student with disabilities cannot follow the activities carried out in the classroom. Thus, it is essential to use or propose pedagogical resources that enable the inclusion of these students, to promote the understanding of the concepts presented in the class. This article describes the route and results of a systematic literature review focusing on identifying this type of pedagogical resource. The main results show that the resources for teaching programming are related to the multiple possibilities of perception, in particular, auditory and tactile.*

Keywords: *programming teaching, visually impaired students, pedagogical resources.*

1. Introdução

O Ensino de Programação é considerado na área de Computação uma das tarefas mais complexas que um docente precisa desempenhar, pois é necessário explicar aos estudantes como abstrair conceitos do mundo real na forma de um código fonte, o qual dará origem a um programa que resolve um determinado problema. Em cursos de nível Superior da área de Computação um dos elementos que influencia a evasão é a dificuldade que muitos estudantes apresentam para aprender a programar. Isso ocorre porque não é trivial descrever uma realidade utilizando uma linguagem de programação com uma determinada sintaxe (Hinterholz, 2009; Hoed, 2016; Souza; Batista; Barbosa, 2016).

Nos últimos anos com a expansão da Rede Federal de Educação Profissional, Científica e Tecnológica (RFEPCT) pode-se perceber a ampliação no número de cursos

técnicos integrados ao ensino-médio vinculados ao eixo “Informação e Comunicação” do Catálogo Nacional de Cursos Técnicos (CNCT) (Mec, 2016). A introdução a estratégias de programação de computadores encontra-se presente em quase todos os cursos previstos nesse documento. Isso significa que os desafios enfrentados pelos docentes, no ensino de programação, são ainda maiores, devido ao perfil dos estudantes neles matriculados.

Aliado a todo esse contexto, percebe-se um movimento da inclusão do Pensamento Computacional (PC) na educação básica (Menezes; Nunes; Livi, 2018). Conforme estabelecem as diretrizes para ensino de Computação na Educação Básica (Sbc, 2019, p.5) “O Pensamento Computacional se refere à capacidade de compreender, definir, modelar, comparar, solucionar, automatizar e analisar problemas (e soluções) de forma metódica e sistemática, através da construção de algoritmos.”. Corroborando a introdução do PC na educação básica é possível encontrar Referenciais de Formação em Computação: Educação Básica (Raabe et al., 2017, p. 4) propostos pela Sociedade Brasileira de Computação (SBC) que reforçam que na educação básica é necessário desenvolver nos estudantes “a capacidade de solução de problemas ao utilizar o pensamento computacional para **criar** processos e produtos” (grifo nosso). Sendo que os mesmos autores argumentam que os pilares do pensamento computacional possibilitam o desenvolvimento de várias competências, dentre as quais destaca-se: “ser capaz de descrever as soluções por meio de algoritmos de forma que máquinas possam executar partes ou todo o algoritmo proposto, bem como de construir modelos computacionais para sistemas complexos” (Raabe et al., 2017, p. 5).

Assim, com base em toda a contextualização prévia é possível perceber que o ensino de programação é uma atividade essencial para diversos cursos, tanto para os de nível Superior, como para a Educação Básica. Várias pesquisas têm sido conduzidas para minimizar os problemas enfrentados com o ensino de programação, por exemplo, programação visual com blocos (Amaral; Medina; Tarouco, 2015; Leôncio et al., 2017; Ribas et al., 2017) estratégias lúdicas baseadas em jogos e gamificação (Galdino; Neto; Costa, 2015; Raposo; Dantas, 2016; Scaico; Scaico, 2016) e robótica educacional (Rodrigues; Nogueira; Queiroga, 2017).

Embora existam várias estratégias para resolver esse problema poucas são as abordagens que se preocupam com o desenvolvimento de soluções para alunos com deficiência visual. Dados divulgados no Censo de 2010 (Ibge, 2010) apontam que 23,9% da população brasileira possui algum tipo de deficiência (física, visual, auditiva ou intelectual), e que destes 18,6% possuem algum tipo de deficiência visual. Visando assegurar o direito “à vida, à saúde, à sexualidade, à paternidade e à maternidade, à alimentação, à habitação, à educação, à profissionalização, ao trabalho” (grifo nosso) foi aprovada a Lei nº 13.146 (Brasil, 2015), que institui a Lei Brasileira de Inclusão da Pessoa com Deficiência, chamada de Lei Brasileira de Inclusão (LBI). Mesmo com a existência dessa Lei pode-se perceber que nas instituições de ensino é possível encontrar poucos recursos pedagógicos que favoreçam o processo de ensino e aprendizagem de estudantes com deficiência visual, em especial na área de Computação.

Partindo-se dessa contextualização prévia, o presente trabalho tem como objetivo sintetizar os principais recursos pedagógicos acessíveis direcionados para o ensino de programação que contemplem estudantes com deficiência visual. Para tanto, foi conduzida uma Revisão Sistemática de Literatura (RSL) (Kitchenham; Charters, 2007), tomando como período de recorte dessa análise os últimos cinco anos, ou seja, de 2015 a 2020. Toda sistematização da pesquisa encontra-se descrita na seção 2, na seção 3 é apresentada a discussão dos resultados alcançados, e, por fim, na seção 4 são descritas algumas conclusões obtidas com a presente RSL.

2. Método de Pesquisa

Conforme já mencionado, o presente trabalho optou pela utilização de uma RSL, a qual tem o propósito de mapear estudos que possuam relação com um determinado tópico de busca. Para conduzir essa revisão um conjunto de passos sistematizados por Kitchenham (2004) foi adotado no presente trabalho: o **planejamento**, que compreende a definição das questões de pesquisa, a *string* de busca, o estabelecimento de critérios de inclusão e exclusão e as bases onde a investigação será realizada; a **condução da pesquisa**, a partir da chave, a qual é aplicada nas bases selecionadas, uma lista de estudos é relacionada, e usando esse conjunto de estudos primários¹, os critérios de inclusão e exclusão são aplicados; a **análise sistematizada dos resultados**, consiste em elaborar esquemas de classificação em categorias e analisar os resultados de modo a responder à questão de pesquisa e auxiliar o pesquisador a decidir o futuro do tema investigado.

2.1 Planejamento da Pesquisa

Essa etapa iniciou-se pela definição do objetivo da pesquisa, o qual consiste em identificar estudos científicos que apresentem quais são as categorias de recursos pedagógicos adotados no ensino de programação para estudantes com deficiência visual. Para alcançar esse objetivo um conjunto de questões foi definido (Quadro 1).

Quadro 1 – Questões de pesquisa que guiam a investigação

Q1 – Quais são os principais recursos pedagógicos utilizados para auxiliar o processo de ensino e aprendizagem de programação para estudantes com deficiência visual?
Q2 – Quais são os conteúdos mais explorados com os recursos identificados?
Q3 – Os recursos pedagógicos identificados são baseados em software ou em hardware?
Q4 – O estudo foi aplicado com qual público alvo?
Q5 – Quais os principais desafios encontrados no uso dos recursos identificados?

Fonte: Elaborado pelos autores, 2020.

Visando delimitar um espaço temporal de investigação, optou-se por realizar a busca por trabalhos nos últimos cinco anos (2015 a 2020). Para a pesquisa foram realizadas buscas em bases de dados internacionais e nacionais (Quadro 2), em periódicos da área de informática na educação reconhecidos nacionalmente (RBIE - Revista Brasileira de Informática na Educação, e RENOTE - Revista Novas Tecnologias na Educação), nas publicações do Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg), pois possui relação direta com o ensino de programação, e no portal de publicações da CEIE (Comissão Especial de Informática na Educação) que abrange os principais eventos da área.

Quadro 2 – Base de dados Nacionais e Internacionais consultadas

Bases Nacionais	Bases Internacionais
(1) Portal de teses e dissertações da Capes	(8) ACM Digital Library
(2) BDTD ²	(9) IEEE Xplore
(3) Portal de publicações da CEIE	(10) Scopus
(4) RBIE	(11) SpringerLink
(5) RENOTE	
(6) WAlgProg	
(7) Portal de periódicos da Capes	

Fonte: Elaborado pelos autores, 2020.

¹ Um estudo primário é aquele que fornece uma evidência direta sobre a(s) questão(ões) de pesquisa definida(s) (Kitchenham; Charters, 2007).

² BDTD – Biblioteca Digital Brasileira de Teses e Dissertações, disponível em: <http://bdt.d.ibict.br/vufind/>

A composição da *string* de busca foi gerada a partir da combinação das palavras-chaves que estão relacionadas com o objetivo do presente trabalho, pela tradução dos termos para a língua inglesa (visando a pesquisa nas bases internacionais), pela identificação dos sinônimos dos termos e pela sua composição através de operadores lógicos (*OR* e *AND*). Assim, foi gerada a seguinte chave em português: “(*recurso OR estratégias OR prática OR metodologia OR método OR experimentação*) *AND* (*ensino de programação*) *AND* (*deficiência visual*)”. Considerando o idioma inglês a chave gerada correspondente é: “(*resource OR strategies OR practice OR methodology OR method OR experimentation*) *AND* (*teaching programming*) *AND* (*visual impairment OR visual disabilities*)”.

Os critérios de inclusão e exclusão foram definidos com o intuito de levar em consideração nesta investigação somente artigos que fossem contribuir efetivamente para a ajudar a responder as questões de pesquisa descritas no Quadro 1. Os critérios de inclusão foram usados de modo que apenas estudos relacionados com o objetivo de pesquisa desta RSL integrassem a análise dos resultados. Já os de exclusão serviram para retirar da investigação artigos duplicados, inacessíveis na íntegra, fora do contexto dessa RSL, ou estudos secundários. No Quadro 3 encontram-se uma síntese dos critérios que foram definidos para a presente pesquisa.

Quadro 3 – Critérios de Inclusão e Exclusão de trabalhos

Critérios de inclusão	Critérios de exclusão
CII – Artigos que apresentam recurso ou estratégias ou prática ou metodologia ou método ou experimentação vinculada ao ensino de programação para alunos com deficiência visual alinhados ao objetivo de pesquisa desta RSL CI2 – Artigos publicados entre 2015 e 2020	CE1 – Artigo já localizado em outra base de dados CE2 – Artigo sem acesso gratuito ao texto completo CE3 – Artigo que não possui como foco o objetivo de pesquisa deste trabalho CE4 – Artigo não é estudo primário

Fonte: Elaborado pelos autores, 2020.

2.2 Condução da Pesquisa

Essa etapa da RSL foi realizada a procura por estudos nas 11 bases de dados indicadas no Quadro 2. Utilizando-se a *string* de busca, foram identificados 40 trabalhos, os quais foram analisados os títulos, resumos e palavras-chave, e submetidos aos critérios de exclusão CE1, CE2, CE3 e CE4. Isso possibilitou separar os artigos que possuíam alguma relação com o objeto de pesquisa deste trabalho, daqueles que não atendiam aos critérios estabelecidos ao objetivo de pesquisa. O Quadro 4 apresenta o quantitativo dos trabalhos identificados em cada base de dados, bem como os critérios de exclusão aplicados, resultando em 13 estudos que integrarão a próxima etapa da RSL.

Quadro 4 – Quantitativo dos estudos científicos identificados³

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Estudos identificados	2	9	1	0	0	0	7	18	9	0	2
Critério CE1	-	1	-	-	-	-	7	-	1	-	-
Critério CE2	-	-	-	-	-	-	-	1	7	-	1
Critério CE3	-	8	-	-	-	-	-	6	-	-	-
Critério CE4	-	-	1	-	-	-	-	1	-	-	1
Estudos pré-selecionados	2	0	0	0	0	0	0	10	1	0	0
Total estudos para analisar:	13										

Fonte: Elaborado pelos autores, 2020.

³ Os números usados na primeira linha do Quadro 4 correspondem aos números das bases de dados listadas no Quadro 2. Por exemplo, a RENOTE corresponde à base de dados (5) como identificado no Quadro 2.

Após a classificação descrita pelo Quadro 4 foi realizada a leitura do título, resumo, palavras-chave, introdução e conclusão dos 13 estudos selecionados. O Quadro 5 apresenta os artigos selecionados para análise.

Quadro 5 – Quantitativo dos estudos científicos identificados

ID	Referências do Estudo
A01	OLIVEIRA, J. D. GoDonnie: definição e avaliação de uma linguagem de programação para comandar robô por programadores iniciantes com deficiência visual. 190 p. Dissertação (Mestrado – Mestrado em Ciência da Computação) – PUCRS, 2017.
A02	REIS, R. P. B. CardBot: Tecnologia Educacional Assistiva para Inclusão de Deficientes Visuais na Robótica Educacional. 76p. Tese (Doutorado). Programa de Pós-graduação em Engenharia Elétrica e de Computação da Universidade Federal do Rio Grande do Norte, 2017.
A03	THIEME, A.; MORISSON, C.; VILLAR, N.; GRAYSON, M.; LINDLEY, S. Enabling Collaboration in Learning Computer Programing Inclusive of Children with Vision Impairments. Conference on Designing Interactive Systems, p. 739–752, 2017. ACM Press.
A04	KANE, K. S.; KOUSHIK, V.; MUEHLBRADT, A. Bonk: Accessible Programming for Accessible Audio Games. ACM Conference on Interaction Design and Children, p. 132-142, 2018. ACM Press.
A05	WANG, Z.; WAGNER, A. Evaluating a Tactile Approach to Programming Scratch. ACM Southeast Regional Conference, Short Papers – Kennesaw, GA, April, p. 226-229, 2019.
A06	KEARNEY-VOLPE, C.; HURST, A.; FITZGERALD, S. Blind Web Development Training at Oysters and Pearls Technology Camp in Uganda. Proceedings... Web For All 2019 Personalization – Personalizing the Web, May, p. 1-10, 2019.
A07	LUDI, S.; BERNSTEIN, D.; MUTCH-JONES, K. Enhanced Robotics! Improving Building and Programming Learning Experiences for Students with Visual Impairments. Special Interest Group on Computer Science Education (SIGCSE), p. 372-377, 2018.
A08	ONG, J. S. Y.; AMOAH, N. A. O.; GARRETT-ENGELE, A. E.; PAGE, M. I.; MCCARTHY, K.; MILNE, L. T. Demo: Expanding Blocks4All with Variables and Functions. Conference on Computers and Accessibility, ASSETS, p. 645-647, 2019.
A09	MILNE, L.; LADNER, R. E. Blocks4All: Overcoming Accessibility Barriers to Blocks Programming for Children with Visual Impairments. Conference on Human Factors in Computing Systems, Apr., p. 1-10, 2018.
A10	VILLAR, N.; MORRISON, C.; CLETHEROE, D.; REGAN, T.; THIEME, A.; SAUL, G. Physical Programming for Blind and Low Vision Children at Scale. Extended Abstracts of the CHI Conference on Human Factors in Computing Systems. May, p. 1-4, 2019.
A11	SCHANZER, E.; BAHRAM, S.; KRISHNAMURTHI, S. Accessible AST-Based Programming for Visually-Impaired Programmers. Special Interest Group on Computer Science Education (SIGCSE), p. 773-779, 2019.
A12	BAKER, C. M; MILNE, L. R.; LADNER, R. E. StructJumper: A Tool to Help Blind Programmers Navigate and Understand the Structure of Code. Conference on Human Factors in Computing Systems, Apr., p. 3043-3052, 2015.
A13	PAPAZAFIROPULOS, N.; FANUCCI, L.; LEPORINI, B.; PELAGATTI, S.; RONCELLA, R. Haptic Models of Arrays Through 3D Printing for Computer Science Education. International Conference on Computers Helping People with Special Needs, p. 491–498, 2016. Springer.

Fonte: Elaborado pelos autores, 2020.

2.3 Análise dos Resultados

Nessa etapa da RSL as questões de pesquisa são examinadas na análise dos artigos, e verifica-se como cada estudo pode ser associado a elas. Todo o processo de extração de dados de cada trabalho, e o mapeamento das suas características encontram-se descritos na próxima seção.

3. Resultados e Discussões

Ao concluir a leitura dos 13 trabalhos selecionados eles foram agrupados por categorias com a finalidade de responder: as questões de pesquisa (Quadro 1) e o objetivo de pesquisa dessa RSL. Cada uma das categorias foi estabelecida com base em afirmações encontradas nos trabalhos analisados. Os dados desses estudos foram sumarizados de forma qualitativa/descriptiva, fazendo a correlação com outros trabalhos vinculados ao tema.

A primeira categoria identificada foi a dos **dispositivos programáveis combinados a outros recursos**, onde os estudos A01, A02 e A07 se enquadram. O **estudo A01** define e avalia a linguagem de programação GoDonnie, a qual tem como foco estimular as habilidades de orientação e mobilidade de pessoas com deficiência visual. O trabalho explorou as questões de movimentação (deslocamento e rotação) através da robótica educacional (RE), o que possibilita ao docente abordar os conteúdos de tipos de dados (somente inteiro), estruturas de início e fim de bloco, estruturas de controle condicionais (*if*, *if/else*) e de repetição (*for*, *while*). Além disso, o trabalho define uma linguagem de programação de fins específicos e utiliza robôs para aplicar os comandos criados com a linguagem. A **pesquisa A07** utiliza a RE para incentivar o ensino de programação. Ele usa a plataforma LEGO Mindstorms e o JBrick⁴ que oferece suporte ao aluno, incluindo avisos auditivos e ferramentas de numeração de linhas para auxiliar os estudantes a localizarem-se dentro do programa. A sua abordagem possibilita ensinar conceitos básicos da programação (variáveis, estruturas de controle condicionais e de repetição), usando uma sintaxe específica da plataforma. O trabalho integra *hardware* e *software* para possibilitar o ensino de programação, unindo a abstração do virtual (código) com a mobilidade do robô no mundo real. O **estudo A02** define o CardBot que é um modelo de ensino-aprendizagem que combina *software* e *hardware*, e onde um robô é programado através de cartões e há um aplicativo que atua como um mediador, reconhecendo a linguagem e repassando ao robô o(s) movimento(s) que deve(m) ser realizado(s). Cada cartão possui uma marcação que determina qual a ação que o robô deverá executar. Este trabalho apresenta como diferencial a possibilidade dos professores criarem “linguagens de programação tangíveis assistivas”, ou seja, cabe ao docente criar o alfabeto da linguagem e depois pode aplicá-la com os estudantes.

A segunda categoria consiste na **linguagem de programação física** (PPL - *Physical Programming Languages*) (Thieme et al., 2017), onde os comandos são representados por objetos físicos que ao serem unidos podem criar programas. Esses trabalhos também poderiam se encaixar na Categoria 4 (recursos táteis), mas optou-se por definir essa categoria, visando agrupar trabalhos fundamentados em PPL. Nessa categoria foram identificados os **trabalhos A03 e A10**, sendo que ambos abordam a linguagem Torino⁵, que tem como foco a aprendizagem computacional para crianças de 7 a 11 anos. Para criar o código os estudantes conectam e manipulam objetos táteis para criar música, histórias em áudio ou poesia. Nesta linguagem cada entidade física (chamada “*pod*”) foi projetada para ser taticamente e visualmente distinta. Cada um possui inclinações e texturas diferentes, e, além disso, ao clicar nos seus botões há uma resposta de áudio imediata, que indica qual é o elemento que está sendo manipulado. Com esse tipo de recurso pedagógico o estudante é encorajado a acompanhar o programa em execução, tocando ou apontando com precisão para cada *pod* à medida que o programa avança. Isso pode auxiliar o estudante a compreender melhor a depuração já que a sua mão já está na posição para corrigir o erro quando ele é

⁴ JBrick é compatível com leitores de tela comuns e displays em braile, permite que o código seja facilmente localizado pelo número da linha e fornece feedback visual e de áudio (Ludi; Ellis; Jordan 2014).

⁵ Mais informações disponíveis em: <https://news.microsoft.com/pt-br/tag/project-torino/>

detectado. Uma característica interessante é que essa linguagem pode ser usada por alunos com deficiência visual, baixa visão, assim como os estudantes videntes. Por fim, essa PPL permite explorar a ideia de como funciona um programa, sua estrutura geral, variáveis, observar como funcionam os laços e conceitos complexos, como laços aninhados.

A terceira categoria estabelecida diz respeito aos **ambientes de programação baseados em blocos** (*Blocks-Based Programming Environments* - BBPEs) (Ong et al., 2019), nela enquadram-se os estudos A05, A08 e A09. Ela foi criada, porque é bem comum, hoje em dia, que escolas e instituições utilizem BBPEs para o ensino de estudantes na Educação Básica. O CodeBox64 (**trabalho A05**) compreende uma solução para o ensino de programação usando Scratch e blocos. Ele oferece uma abordagem tátil para a entrada de dados, e é semelhante a um dispositivo periférico composto por seis botões, onde quatro deles permitem que o usuário controle o mouse e os outros dois botões permitem que o aluno selecione diferentes menus e arraste os blocos no script que está na tela. À medida que os alunos navegam pelo Scratch, o CodeBox64 gera pistas auditivas para indicar qual bloco o aluno está tentando arrastar/soltar. Ele possibilita explorar os recursos básicos da programação através dos blocos do Scratch: variáveis, estruturas de controle condicionais (*if, if/else*) e de repetição (*for, while*). Já os **estudos A08 e A09** dizem respeito a mesma ferramenta - Blocks4All – com ela os blocos podem ser acessados usando o VoiceOver, que fornece o nome, a localização e o tipo de bloco que está sendo manipulado. A interface gráfica dessa ferramenta é semelhante a do Scratch, mas o grande diferencial é que todas as movimentações são narradas. O ambiente possui uma estrutura que facilita o seu uso por alunos com baixa visão, sendo que cada bloco possui um áudio associado o que permite ao aluno localizar os blocos de sua programação. Essa ferramenta disponibiliza como recursos de programação variáveis, laços e estruturas condicionais, blocos para começo e fim de instrução, bem como instruções aninhadas colocadas verticalmente.

A quarta categoria definida é composta por trabalhos relacionados a **ambientes de programação acessíveis** (A04, A11, A12). No caso do **estudo A04** é apresentado o Bonk, que compreende um kit de ferramentas de programação que suporta programação de mídia acessível, utilizando uma linguagem específica e fornecendo aos desenvolvedores conversão de texto em fala, ou seja, controle sobre os parâmetros de voz, incluindo, tom, velocidade de reprodução e volume. Ele permite a definição de funções com simplificações de sintaxe reduzindo o uso de elementos *case sensitive* (por exemplo, *on_key_press*). Possibilita ao programador explorar variáveis, estruturas condicionais e de repetição, e a estrutura ramificada permite ao desenvolvedor separar um jogo em pedaços menores e gerenciáveis, para posteriormente combiná-los (através de funções) formando um único jogo. O **estudo A11** apresenta o CodeMirror-Blocks (CMB) que corresponde a um kit de ferramentas para criar sistemas totalmente acessíveis e baseados em navegação. O CMB cria uma árvore de sintaxe abstrata (*Abstract Syntax Tree* - AST) totalmente acessível que é renderizada no formato de blocos, estrutura mais familiar aos alunos videntes. Além disso, ele comunica a organização do programa usando descrições narradas e permite navegação usando teclas de atalho. Os elementos de programação variáveis, condicionais, laços e funções são manipulados com estruturas chamadas ASTNode, que podem ser renderizados como texto ou como uma árvore DOM (*Document Object Model*) no navegador. A **pesquisa A12** aborda o StructJumper, que é um plug-in do Eclipse que cria uma árvore hierárquica com base na estrutura de aninhamento de uma classe Java. O programador utiliza uma estrutura de árvore (semelhante ao estudo A11) para obter uma visão geral da organização do código da classe (incluindo todos os métodos e fluxos de controle), sendo que ele pode alternar rapidamente entre a árvore e o editor de texto para ter uma ideia de onde cada elemento está dentro da estrutura aninhada, e pode percorrer a árvore gerada usando teclas de atalho.

Dentre os trabalhos analisados esse foi o único que permitiu explorar além de recursos básicos de programação, conceitos de orientação a objetos.

A quinta categoria é estabelecida com base em **recursos hápticos** e é composta pelos trabalhos A06 e A13. O **estudo A06** parte do desenvolvimento de diagramas táteis, com o intuito de ensinar a posição dos elementos em uma página Web. Esses recursos são aplicados ao ensino de operadores do Javascript (<, >, +, |, =, entre outros) e conceitos de HTML (*HyperText Markup Language*) tais como: aninhamento, estrutura de TAGs, definições, sintaxe HTML, tipos de elementos, atributos e conceitos de hierarquia. A **pesquisa A13** utiliza modelos hápticos impressos em uma impressora 3D para ensinar conceitos como estruturas de dados e matrizes. O modelo criado utiliza-se de cilindros de várias alturas, onde a altura corresponde ao valor do elemento. Esses elementos são, então, encaixados em uma bandeja que representa a matriz, sendo possível aplicar algoritmos de classificação e pesquisa com essas representações 3D.

Com relação ao público-alvo dos estudos investigados percebe-se que a maioria deles foi conduzida com participantes da Educação Básica (A02, A03, A04, A05, A07, A09 e A10, A13). Já os trabalhos A01 e A06 foram realizados com estudantes da faixa etária de 18 a 34 anos, no caso do primeiro eram alunos de cursos de graduação e de pós-graduação, e no segundo estudantes com diversos níveis de conhecimento sobre programação. O trabalho 11 foi realizado com 13 participantes, sendo que 12 deles eram pessoas com deficiência visual, separados em três faixas de experiência com programação. A pesquisa do trabalho A12 foi conduzida com 7 programadores com deficiência visual.

Considerando-se os estudos analisados, pode-se perceber que o grande desafio está em integrar a todas as soluções outras possibilidades de percepção, em especial, a auditiva e a tátil (Perovano; Melo, 2019). Isso ocorre porque a programação é composta por vários elementos específicos, como: código *case sensitive*, delimitadores de bloco, símbolos especiais (“; () {}”) que quando lidos por leitores de tela podem confundir os alunos com deficiência visual. Os leitores de tela não têm acesso à estrutura de um programa, pois os tokens são lidos apenas uma vez, e o programa é dividido em uma série de linhas, o que prejudica a sua compreensão. Além disso, a navegação por códigos complexos usando um leitor de tela também é uma tarefa complexa, visto que os ambientes de programação possuem inúmeras janelas que auxiliam na depuração, correção e testes. Outro desafio identificado é a inexistência de ferramentas para apoiar o compartilhamento de código e o trabalho colaborativo entre pessoas com deficiência visual.

4. Conclusões

Ao concluir a RSL foi possível sistematizar os trabalhos em 5 categorias principais, que apresentam recursos pedagógicos que podem ser utilizados para o ensino de programação para estudantes com deficiência visual. Através dessa pesquisa percebe-se que existem diferentes possibilidades de recursos e que cabe ao docente selecionar o mais adequado, considerando o conteúdo a ser abordado, a faixa etária e características individuais do estudante (se possui resíduo visual ou não; se tem memória visual ou não; se conhece Braille; se tem boa habilidade com leitor de tela).

A partir de todos os resultados obtidos percebe-se que todo e qualquer recurso deve propiciar a autonomia dos estudantes com deficiência visual, fornecendo informações de forma tátil ou sonora, inclusive fornecendo compatibilidade com os leitores de tela. O uso de etiquetas em Braille deve ser utilizado para identificar os elementos táteis, quando for o caso. O uso de leitores de tela para a programação em ambientes como Eclipse e Visual Studio pode não ser suficiente devido estrutura complexa do código, pois a navegação

prejudica a legibilidade dos códigos por parte desses estudantes. Já uma pessoa com leitor de tela, ao utilizar um software que oferece *feedback* sonoro, mas com o qual o leitor não consegue interagir, poderá sentir-se confusa, uma vez que não compreender o que está acontecendo, por não ter retorno do leitor de tela e, dessa forma, não conseguir ter o controle das ações.

Considerando-se os trabalhos analisados poucos exploraram as teorias da aprendizagem na proposição dos recursos, bem como estabelecem guias ou orientações para o docente adotar ou replicar as práticas pedagógicas realizadas, ou ainda que facilitem uma progressão dos estudos por parte dos estudantes. A presença de estudantes com deficiência visual em salas de aula tem aumentado muito nos últimos anos, portanto, é imprescindível a proposição de recursos que ofereçam a estes alunos o acesso ao conhecimento, priorizando a sua autonomia, e na medida do possível que os recursos desenvolvidos possam ser usados tanto pelos alunos videntes quanto pelos alunos com deficiência visual, de modo que a aquisição do conhecimento possa ocorrer também com os pares.

Referências

AMARAL, E.; MEDINA, R.; TAROUCO, L. M. R. **Processo de Ensino e Aprendizagem de Algoritmos Integrando Ambientes Imersivos e o Paradigma de Blocos de Programação Visual**. Tese (Doutorado em Educação) UFRGS, 2015, 2015. p. 255. Disponível em: <<https://lume.ufrgs.br/handle/10183/131053>> Acesso em: 3 ago. 2018.

BRASIL. Lei n. 13.146, de 6 de julho de 2015. **Institui a Lei Brasileira de Inclusão da Pessoa com Deficiência**, v.152, n. 127, p. 2-11, 07 jul. 2015. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm> Acesso em: 03 nov. 2018.

GALDINO, C.; NETO, S.; COSTA, E. KidCoder: Uma Proposta de Ensino de Programação de forma Lúdica. **Anais... SBIE**, 2015, 2015. p. 687-691. Disponível em: <<http://br-ie.org/pub/index.php/sbie/article/view/5340>>.

HINTERHOLZ, O. J. Tepequém: uma nova Ferramenta para o Ensino de Algoritmos nos Cursos Superiores em Computação. **Anais... WEI**. p. 485-488, 2009. Disponível em: <http://csbc2009.inf.ufrgs.br/anais/pdf/wei/st02_04.pdf> Acesso em: abr. 2018.

HOED, R. M. **Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de Computação**. 188 p. Dissertação (Mestrado). Universidade de Brasília, 2016.

IBGE. Instituto Brasileiro de Geografia e Estatística. Censo 2010. Disponível em: <<https://censo2010.ibge.gov.br/>> Acesso em: 15 maio 2019.

KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. **Technical Report EBSE 2007-001**. Disponível em: <https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf> Acesso em: mar. 2020.

KITCHENHAM, B. **Procedures for performing systematic reviews**. 33, p. 1-26, 2004. UK: Keele University.

LEÔNICIO, N. N. et al. Programação em blocos com o Mit App Inventor: Um relato de experiência com alunos do ensino médio. **Anais... WEI**, p. 1159-1163, 2017. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/7325>> Acesso em: 10 mar. 2020.

- LUDI, S.; ELLIS, L.; JORDAN, S. An accessible robotics programming environment for visually impaired users. **Proceedings...** Conference on Computers & accessibility – ASSETS, p. 237–238, 2014. New York: ACM Press.
- MEC. **Catálogo Nacional de Cursos Técnicos**. Brasília/DF. Disponível em: <<http://portal.mec.gov.br/component/content/article?id=52031>> Acesso em: 5 ago. 2018.
- MENEZES, C. S.; NUNES, D.; LIVI, M. A. C. **Pensamento Computacional: Revisão bibliográfica**. Porto Alegre: UFRGS, 2018. Disponível em: <https://lume.ufrgs.br/handle/10183/197566>. Acesso em: 03 out. 2019.
- ONG, J. S. Y.; AMOAH, N. A. O.; GARRETT-ENGELE, A. E.; PAGE, M. I.; MCCARTHY, K.; MILNE, L. T. Demo: Expanding Blocks4All with Variables and Functions. **Proceedings...** Conference on Computers and Accessibility, p. 645-647, 2019.
- PEROVANO, L. P.; MELO, D. C. F. **Práticas inclusivas**. Campos dos Goytacazes: Multicultural, 2019. 176 p.
- RAABE, A. L. A.; ZORZO, A. F.; FRANGO, I.; RIBEIRO, L.; GRANVILLE, L. Z.; SALGADO, L.; CRUZ, M. J. K.; BIGOLIN, N.; CAVALHEIRO, S. A. C.; FORTES, S. Referenciais de Formação em Computação: Educação Básica. **Anais...** Congresso Sociedade Brasileira de Computação, 2017. Disponível em: <<https://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf>> Acesso em: 25 nov. 2019.
- RAPOSO, E. H. S.; DANTAS, V. O Desafio da Serpente - Usando gamification para motivar alunos em uma disciplina introdutória de programação. **Anais...** SBIE, p. 577-586, 2016. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/6739>> Acesso em: 10 ago. 2019.
- RIBAS, E.; et al. Programação visual para introdução ao ensino de programação na Educação Superior: uma análise prática. **RENOTE**, v. 14, n. 2, 17 jan. 2017. Disponível em: <<http://seer.ufrgs.br/index.php/renote/article/view/70671>> Acesso em: 26 ago. 2018.
- RODRIGUES, L. C.; NOGUEIRA, G. C.; QUEIROGA, A. Greenfoot e Jogos de Tabuleiro no Ensino Superior. 2. Experiências no ensino de Programação Orientada a Objetos: RoboCode. **Anais...** Workshop de Informática na Escola, 2017. p. 598-607. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/7278>> Acesso em: 12 ago. 2019.
- SBC. **Diretrizes para ensino de Computação na Educação Básica Ensino de Computação na Educação Básica**. 2019. Disponível em: <<https://www.sbc.org.br/educacao/diretrizes-para-ensino-de-computacao-na-educacao-basica>> Acesso em: 12 dez. 2019.
- SCAICO, A.; SCAICO, P. Uso de Jogos em Cursos Introdutórios de Programação - Uma Revisão Sistemática. **Anais...** SBIE. p. 549-558, 2016. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/6736>> Acesso em: 12 ago. 2019.
- SOUZA, D. M.; BATISTA, M. H. S.; BARBOSA, E. F. Problemas e Dificuldades no Ensino e na Aprendizagem de Programação: Um Mapeamento Sistemático. **RBIE**, v. 24, n. 1, p. 39-52, 2016.
- THIEME, A.; MORISSON, C.; VILLAR, N.; GRAYSON, M.; LINDLEY, S. Enabling Collaboration in Learning Computer Programming Inclusive of Children with Vision Impairments. **Proceedings...** Conference on Designing Interactive Systems, p. 739–752, 2017. New York: ACM Press.