

Ensino de programação com apoio de robótica para pessoas com deficiência visual: uma revisão sistemática da literatura

Juliana Damasio Oliveira
Pontifícia Universidade Católica do Rio Grande do Sul PUCRS
Programa de pós-graduação em Ciência da Computação
Porto Alegre, RS
Email: juliana.damasio@acad.pucrs.br

Resumo—Este artigo tem como objetivo apresentar uma revisão sistemática da literatura dos estudos referentes ao uso de robótica para o ensino de programação para pessoas com deficiência visual (DV). Como resultado foram identificadas as metodologias de ensino, os kits de robótica e programação utilizados, as boas práticas e dificuldades enfrentadas no ensino de programação com o apoio da robótica.

Keywords—Ensino de programação, Robótica, Deficiência visual, Revisão sistemática.

I. INTRODUÇÃO

Os primeiros conceitos de robótica educacional surgiram com Papert [1], com a criação da linguagem LOGO onde através de comandos era possível movimentar uma tartaruga gráfica. A principal ideia da linguagem LOGO era incentivar crianças a aprender programação de uma forma motivadora e lúdica [1]. A partir dessa ideia, surgiram kits de robótica que possuem suas próprias linguagens e ambientes de programação, entretanto muitos destes, utilizam-se de recursos visuais que acabam por se tornar inacessíveis para pessoas com deficiência visual (DF).

Este trabalho faz parte de um projeto maior em que está sendo criada uma linguagem de programação para comandar um robô. Este ambiente está sendo desenvolvido com acessibilidade para permitir a inclusão de crianças com DF. O objetivo dessa revisão sistemática verificar os trabalhos existentes de ensino de programação com robótica para pessoas com DF. Esta caracterização do estado da arte poderá auxiliar no direcionamento de passos futuros do projeto, pois, algumas novas direções podem ser tomadas a partir desta revisão.

Após essa seção introdutória, o restante deste artigo foi organizado da seguinte forma: a seção II apresenta a revisão sistemática realizada, junto com o protocolo seguido e os resultados obtidos. Na seção III a conclusão deste trabalho é exibida.

II. REVISÃO SISTEMÁTICA DA LITERATURA

Conforme Kitchenham [2], uma revisão sistemática da literatura é um processo metódico para identificar, avaliar e interpretar as evidências científicas disponíveis e relevantes sobre um tema específico de interesse. O protocolo utilizado neste trabalho foi baseado no trabalho de [2].

Para este trabalho, a revisão sistemática tem como finalidade auxiliar no entendimento de como a programação está sendo ensinada com o apoio de robótica para pessoas que são cegas. Deste modo, a aplicação desta técnica ocorreu nas seguintes etapas:

A. Planejamento

A fase de Planejamento, aborda a elaboração do objetivo das questões de pesquisa relativas ao tema, as estratégias de busca até os critérios de seleção.

Objetivo: Tem-se definido o seguinte objetivo para esta revisão:

Identificar e analisar procedimentos metodológicos que façam uso de robótica como apoio ao ensino de programação para pessoas que são cegas.

Questões de pesquisa: A partir do objetivo definido anteriormente foi definida a seguinte questão de pesquisa primária (Q1) e as secundárias (a, b, c, d):

Q1- Quais procedimentos metodológicos estão sendo utilizados no ensino de programação com robô para pessoas que são cegas?

- a. Que metodologias de ensino vem sendo utilizadas para ensinar programação de robôs para pessoas que são cegas?
- b. Quais as características dos ambientes de programação que tem sido utilizado por usuários que são cegos?
- c. Quais são exemplos de boas práticas no ensino de programação de robô para pessoas que são cegas?
- d. Quais foram as dificuldades/limitações no uso de robótica como apoio ao ensino de programação para pessoas que são cegas?

Estratégias de busca: Este item descreve onde e como foram realizadas as buscas, bem como as palavras-chave que geraram a String de busca e as línguas aceitas para os resultados selecionados.

- 1) **Fontes:** Foram definidas como fontes de pesquisa deste trabalho as seguintes bases de dados:

- ACM Digital Library ¹: escolhida por possuir uma base de dados focados exclusivamente na área de computação.
- ScienceDirect²: selecionada por possuir artigos da área de Ciência da computação e Engenharia.
- IEEEExplore ³: escolhida por ter conteúdo científico e técnico publicado pela IEEE. Contém artigos de várias áreas entre elas Ciência da Computação e Engenharia.
- Scopus ⁴: selecionada por indexar diversas bases de pesquisa como IEEE, ACM, ScienceDirect dentre outras.

- 2) **Palavras-chave:** baseado no objetivo e na questão de pesquisa levantados anteriormente, foram definidas as seguintes palavras-chave:

- Pessoa com deficiência visual:
Sinônimos em português: cego, deficiente visual, deficiência visual, cegueira.
Sinônimos em inglês: blind, visually impaired, visual disability, blindness, student disability, visual impairments.
- Programação
Sinônimos em português: programador, programação.
Sinônimos em inglês: programming, programmer.
- Robótica
Sinônimos em português: robótica, robô, robôs, interação homem-robô, robô assistivo, robótica, robô programável, robótica educativa/pedagógica, robótica pedagógica.
Sinônimos em inglês: robotic, robot, robotics.

- 3) **Definição dos artigos de controle:** Para este trabalho foi definido um artigo de controle que havia sido identificado previamente em buscas não sistemáticas na base Scopus. Este artigo foi selecionado para futura validação da string de busca. O artigo é *The Use of Robotics to Promote Computing to Pre-College Students*, 2011.

- 4) **String de Busca:** as strings de busca utilizadas em cada base de dados estão descritas na Tabela I. As buscas foram realizadas nos campos *abstract*, *title* e *key-words*.

- 5) **Idiomas:**
Inglês: por ser o idioma padrão para publicação internacional.
Português: para que se pudesse verificar as pesquisas nacionais.

Critérios de seleção: Os critérios definidos para seleção de estudos foram:

- Critérios de inclusão
II- O resultado deve estar nos idiomas definidos.

¹<http://dl.acm.org/>

²<http://www.sciencedirect.com/>

³<http://ieeexplore.ieee.org/Xplore/home.jsp>

⁴<https://www.scopus.com/>

Tabela I: String de dados

Base	String
Scopus	TITLE-ABS-KEY (("blind" OR "visually impaired" OR "visual disability" OR "blindness" OR "student disability" OR "unsighted pupils" OR {visual impairments})) AND programm* AND Robot*)
ScienceDirect	TITLE-ABSTR-KEY ("blind" OR "visually impaired" OR "visual disability" OR "blindness" OR "student disability" OR "unsighted pupils" OR "visual impairments") AND program AND Robot [All Sources(Computer Science,Engineering)].
IEEEExplore	((("Abstract": "blind" OR "Abstract": "visually impaired" OR "Abstract": "visual disability" OR "Abstract": "blindness" OR "Abstract": "student disability" OR "Abstract": "student disability" OR "Abstract": "visual impairments" OR "Abstract": "unsighted pupils") AND (p_Abstract: "programm" OR "Abstract": "programming" OR "Abstract": "develop") AND ("Abstract": "robot" OR "Abstract": "robotic" OR "Abstract": "robotics"))
ACM	recordAbstract:((impairments OR blind OR blindness OR disability) AND (programming OR program) AND (robotics OR robotic OR robot)) acmdlTitle:((impairments OR blind OR blindness OR disability) AND (programming OR program) AND (robotics OR robotic OR robot)) keywords.author.keyword:((impairments OR blind OR blindness OR disability) AND (programming OR program) AND (robotics OR robotic OR robot))

I2- O resultado deve estar disponível integralmente e ser disponibilizado pela instituição de ensino.

I3- O resultado deve conter no título, nas palavras-chave, ou no resumo alguma relação com o tema deste trabalho (usuário cego, programação, robótica).

- Critérios de exclusão

E1- O resultado não está relacionado ao tema do trabalho.

E2- Artigos com idiomas diferentes do Inglês e do Português.

E3- No caso de estudos similares ou duplicados, somente o mais recente foi considerado.

E4- Trabalhos que abordem questões de ensino de programação de forma geral, e não com pessoas com deficiência visual.

E5- Trabalhos que abordem questões de uso de robótica de forma geral, e não com pessoas com deficiência visual.

E6- O resultado não ser da área de Ciência da Computação ou Engenharia.

B. Execução

Nesta subseção será mostrado como foi realizada a seleção dos artigos e como foi o procedimento de seleção dos trabalhos e a extração dos resultados.

Procedimento de seleção: A seleção dos artigos foi realizada da seguinte forma:

- 1) Execução da String em cada base de busca.
- 2) Aplicação de filtro nas bases para selecionar apenas trabalhos da área de Ciência da computação e Engenharia.
- 3) Exportação da lista de trabalhos no formato

Tabela II: Artigos retornados na busca

Base	Número de estudos retornados	Número de artigos selecionados
Scopus	66	8
ACM	19	0
IEEEExplore	7	1
ScienceDirect	33	0
Total	125	9

- bibtex.
- 4) Importação dos arquivos bibtex na ferramenta Start, que é uma ferramenta que apoia na organização de revisões sistemáticas.
 - 5) Foi realizado um primeiro filtro pela autora deste trabalho, onde foram aplicados os critérios de seleção com base na leitura do resumo, palavras-chave e título, para os 125 trabalhos retornados das diferentes bases. Foram aceitos 9 trabalhos. Na Tabela II é apresentado para cada base a quantidade de artigos encontrados e aceitos.
 - 6) Após, foi realizado um segundo filtro por uma especialista da área, que confirmou os dados anteriores, ou seja, verificou os trabalhos rejeitados e aceitos. Após esse filtro persistiram 9 trabalhos aceitos.
 - 7) Por último, os artigos aceitos na ferramenta Start foram exportados para um arquivo no formato Excel. Este arquivo continha as seguintes colunas: *ID Paper, Title, Authors, Abstract, Status/Selection, Year, Journal, Url e Inclusion and Exclusion Criteria*. Além das colunas geradas a partir da exportação foram incluídas as seguintes colunas: Q1, a, b, c, d, referentes a questão de pesquisa primária e secundárias deste trabalho.

Extração dos resultados Os artigos exportados para Excel foram lidos na íntegra afim de verificar se respondem as questões de pesquisa e como respondem.

C. Resultado da Revisão

Nesta seção serão respondidas as questões de pesquisa identificadas neste trabalho. No Gráfico 1 são apresentados a distribuição dos artigos por anos.

Q1- Quais procedimentos metodológicos estão sendo utilizados no ensino de programação com robô para pessoas que são cegas?

a. Que metodologias de ensino vem sendo utilizadas para ensinar programação de robôs para pessoas que são cegas?

Os estudos selecionados apresentam como metodologias de ensino o uso de oficinas de robótica [3], [4], [5], [6], [7], [8], os estudos [9], [10], [11] não fizeram atividades com pessoas cegas. A duração das oficinas variou entre os trabalhos, assim como os métodos aplicados e quantidade de participantes:

[3], [4], [5], [6] Ofereceram tutoriais com conceitos de programação e a sintaxe da linguagem. Os tutoriais foram fornecidos aos estudantes como um modelo de programação

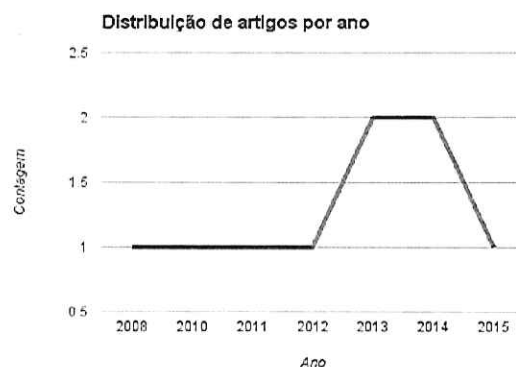


Figura 1: Distribuição de artigos por ano

que tanto ensinou os alunos, bem como dirigiu-os para executar várias tarefas. [5] As tarefas foram projetadas de tal forma que cada tarefa se baseava no conhecimento alcançado a partir das tarefas anteriores, aumentando sistematicamente o nível de habilidade necessário para conclusão. Ainda, no [5] durante todas as tarefas, o instrutor fazia perguntas que requeriam ao aluno dar uma resposta verbal.

Nos estudos [3], [4], [5], [6] foram aplicados pré e pós questionários. No [8] realizaram entrevista semi-estruturada após as atividades. Em [7] coletaram comentários dos usuários sobre a experiência na oficina.

➔ [7] Realizaram uma aula experimental de programação. Todos os participantes foram incumbidos de criar um programa sequencial e um programa de desvio condicional para executar ao longo de um percurso traçado em material de EVA, com uma linha preta mostrando o caminho a traçar Figura 2c.

Entre a quantidade e as idades dos participantes, temos: [3], [4] tiveram a participação de 14 alunos com deficiência visual. Sendo [3] com idade de 12 à 18 anos e [4] não informaram a idade. [3], [4] Não informaram sobre o conhecimento dos alunos sobre programação. No estudo [5] tiveram 9 participantes deficientes visuais, não informaram a idade apenas que cursavam *middle schools*. Em [6] participaram 32 alunos, não informaram a faixa etária dos alunos somente que cursavam *middle e High schools*. Alguns tinham experiência em programação. No trabalho [7] participaram 7 alunos de *high school e junior high school*, não informaram as idades exatas, nenhum participante tinha experiência em programação. No [8] houve 10 pessoas com deficiência visual, não dizem a faixa etária, os participantes tinham níveis diferentes de conhecimento sobre programação (3- não tinham experiência e 7- tinham alguma experiência).

A duração e quantidade de oficinas: [3] A oficina durou 4 dias, não foi informado a duração das oficinas. [4] Foram realizadas 3 oficinas, não foi informado a duração das oficinas. [5] Duas semanas, não foi informado a duração das oficinas. [6] Os participantes tinham 2 horas para terminar os tutoriais e 2 horas para criar estratégias para resolver o desafio de jogo. Não foi informado a quantidade de oficinas. [7] 90 minutos para realizar as atividades. Não foi informado a quantidade de oficinas. [8] Os participantes usavam o JBrick de 3-4 horas por dia durante 4 dias.

Alguns trabalhos descrevem o ambiente de deslocamento do robô: [3] criaram um labirinto para o robô se deslocar que ficava sobre uma mesa Figura 2a. [6] o ambiente era modificado de acordo com a atividade. A Figura 2b apresenta o espaço da atividade desafio que era como um jogo. [7] criaram um percurso com EVA no chão para o robô se deslocar Figura 2c.

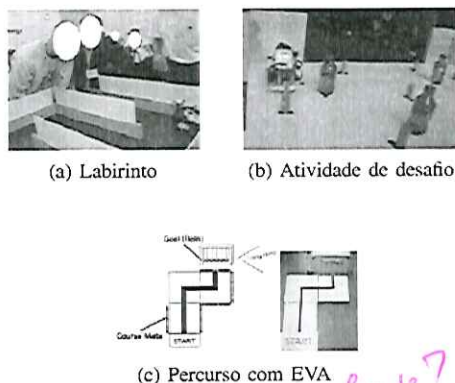


Figura 2: Ambientes de deslocamento do Robô

Forma de realização das atividades: [3], [4], [6], [8] As atividades foram realizadas em equipes, [3], [6] não dizem exatamente quantos alunos cada equipe continha, [4] equipes de 3 alunos e [8] de 2-3 alunos que foram de formados de forma aleatória. Enquanto que [5], [7] realizaram as atividades individualmente.

b. Quais as características dos ambientes de programação que tem sido utilizado por usuários que são cegos?

Dos nove trabalhos analisados, seis utilizam o kit de robótica LEGO mindstorm NXT [3], [9], [4], [5], [6], [8] conforme (Ver Figura 3a). Os três restantes [10], [7], [11] criaram um robô próprio utilizando placa Arduino (Ver Figura 3b). Maiores detalhes dos kits e da linguagem de programação estão abaixo:

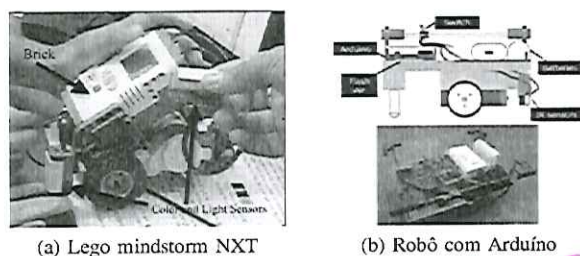


Figura 3: Tipos de Robôs

[3], [4] Os estudos são do mesmo grupo de pesquisa, o kit de robótica LEGO mindstorm NXT era composto por três motores (cada um já montado em uma estrutura com caixa de redução e sensor de giro/velocidade), os sensores utilizados nos robôs da oficina são de toque, luz, ultrasonic e som. O ambiente de desenvolvimento escolhido foi o BricxCC que possui código aberto. A linguagem utilizada foi NXC que é

parecida com a linguagem C. Essa foi escolhida por ter uma sintaxe simples, ser fácil de aprender e facilmente legível com o software de leitor de tela.

[9], [8] Estes estudos pertencem ao mesmo grupo de pesquisa [Ludi:2008,Ludi:2011], utilizaram o mesmo kit de robótica, mas migraram o ambiente BricxCC para um novo sistema chamado JBrick. O BrickCC era em Delphi e o JBrick foi desenvolvido em Java. O menu do JBrick foi simplificado para 5 opções, para minimizar o que é falado pelo leitor de telas, no artigo não falam quais as opções, o BrickCC tinha 8 opções.

[5], [6] Pertencentes ao mesmo grupo de pesquisa, utilizaram o kit de robótica LEGO mindstorm NXT composto por dois motores com rodas e codificadores internos para cálculo de odometria, dois sensores de toque para detectar a entrada do usuário e incidente de colisão, um sensor de luz para detectar um marco no chão, e um sensor de ultrasonic para detectar um objeto em frente do robô. Escolheram o BricxCC como ambiente de desenvolvimento. Para fornecer feedback háptico, utilizaram um controle remoto Wii (Wiimote). No trabalho [5], incluíram um feedback resumido que foi "encarnado" em um agente inteligente chamado Robbie, que fornece feedback de áudio para o aluno depois que seu programa é executado. Robbie foi escrito em em linguagem C ++.

[10], [7], [11] Possuem os mesmos pesquisadores, o robô criado pelo grupo é composto de uma placa microcontroladora Arduino UNO, um cartão microSD sem fio, uma campainha, dois motores, uma caixa de velocidades, e as baterias. A programação era realizada no que chamaram de P-CUBE através de Blocos de programação feitos de cedro japonês, equipados com leitor de RFID (ver Figura 4). Cada bloco tinham comandos associados para movimentar o robô, os comandos eram: Movimento (forward, Right, Left, backward), Timer (1st, 2nd, 3rd, 4th), IF (IF START IRsensor(L) END, IF START IRsensor(R) END) e LOOP (robô móvel repete os movimentos). A execução é feita da seguinte forma: a informação de tipo de bloco é obtida a partir da etiqueta RFID do bloco de programação. A etiqueta de RFID é transmitida para o PC. Após, a informação é transmitida para o robô móvel, usando um cartão microSD, ou seja, o robô não é executado diretamente é necessário conectar o cartão microSD no robô. No estudo [7], depois de fazerem testes com usuários, remodelaram o robô para que as informações do RFID fossem enviadas através de um botão para o robô, assim não seria necessário o uso do cartão microSD.

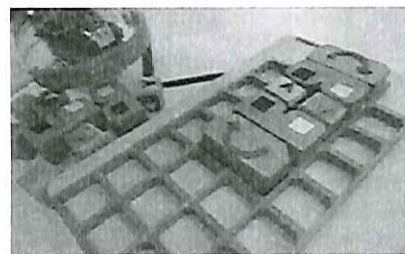


Figura 4: Blocos de programação feitos de cedro japonês

c. Quais são exemplos de boas práticas no ensino de programação de robô para pessoas que são cegas?

Os artigos apresentaram práticas para o ensino de programação de robô para pessoas cegas:

- Fornecer uma sessão de treinamento as pessoas que auxiliaram na oficina, para que os mesmos saibam as estratégias necessárias para trabalhar com pessoas com deficiência visual [3].
- É importante orientar os alunos sobre a sala, sua área de trabalho e o equipamento [3].
- As mesas, cadeiras, etc. devem ter espaço suficiente para os alunos com as bengalas atravessarem a área de trabalho com segurança [3].
- Uso de etiquetas em Braille para os principais componentes do Brick [3].
- As oficinas serem realizadas em uma sala grande ou várias salas pequenas para controlar o nível de ruído. Além dos alunos, os leitores de tela também falam constantemente [3].
- JAWS e ZoomText, estavam disponíveis para os alunos usarem no computador. Durante o período de orientação do workshop, o software foi personalizado para os estudantes com a sua assistência [3].
- Cada equipe deve ter um assistente para auxiliar os alunos. Isso permite a facilitação da atividade e garante que todos os alunos estejam ativos em suas respectivas equipes [3].
- Fornecer tutoriais com código comentado [4], [5].
- Previamente perguntar como cada participante gostaria de receber o tutorial, por exemplo, material impresso em braille ou com fontes grandes [3], [4].
- Criar uma folha de referência rápida para os símbolos ou comandos. Isto é especialmente útil para alunos com deficiência visual que têm de aprender muitos novos comandos e sintaxes [4].
- Colocar comentários no código com explicação e cuidar para ser conciso e limitado em largura de modo que mais do conteúdo seja legível para os participantes, minimizando a necessidade de percorrer o tela [4].
- Quebrar as linhas para não prejudicar as pessoas com baixa visão que precisam aumentar a tela [4].
- Coloque o labirinto ou outra área para executar o robô a uma altura acessível para todos os participantes [4].
- Peça que os participantes se revezem na programação do robô. O mesmo vale para iniciar e parar o programa no próprio robô [4].
- As atividades devem ser acessíveis a todos os participantes que representam uma diversidade de deficiências visuais. Independentemente de saber se um participante é totalmente cego ou com visão subnormal, todos devem ser capazes de realizar todas as atividades [4].
- As ferramentas de softwares a serem utilizados devem ser compatíveis os softwares leitores de tela e um

ampliadores de tela, tais como JAWS e ZoomText. A interface e o código-fonte devem ser capazes de serem ouvidos / vistos pelos participantes [4], [6].

- Os participantes devem ser capazes de interagir com seu robô em um nível básico, a fim para transformá-lo em on / off, ligar motores e sensores, e executar o programa desejado [4].
- Os tutoriais e desafios de projeto devem facilitar uma progressão na programação e construção de habilidades de engenharia de software e exploração [4].
- As atividades devem ser adequadas à idade e diversão [4].
- Manter todos os participantes ativos dando funções para cada participante. Isso diminui o impacto de uma personalidade dominadora. Isto é especialmente interessante em grupos mistos, nota-se como muitos alunos com deficiência visual são muitas vezes ignorados [4].
- Apresentar informações usando vários meios: como oralmente e escrito no quadro/apostila [4].
- Fornecer folhetos e cartões de referência é útil, mas sempre os participantes a tomar suas próprias notas [4].
- Encoraje os participantes a relacionar com o seu próprio mundo as habilidades e desafios do robô. Muitos estudantes deficientes visuais e cegos podem relacionar os sensores com os seus próprios paradigmas de navegação [4].
- Como acontece com qualquer classe, ter suporte para os participantes é importante. Um único instrutor com uma classe cheia é assustadora. Se os pais não estiverem disponíveis, alunos de computação ou engenharia de sua universidade podem ser úteis para fornecer orientação aos participantes [4].
- Passe uns minutos no início da atividade orientando os participantes sobre as partes do robô para garantir que cada participante se torne familiarizado com o robô e com o download de programas, etc [4].
- Ao usar um leitor de tela, fornecer fones de ouvido para cada usuário [4].
- Fornecer informações táteis e de áudio. As informações táteis foram colocadas nos blocos de programação para que os usuários pudessem identificar os tipos de blocos, e o robô móvel fornecia sons diferentes quando virava para a direita ou para a esquerda a fim de apresentar dados do estado de operação do robô [10], [11].
- Fornecer um conjunto de comando simples (biblioteca) que podem ser construídos em etapas para que os alunos possam progredir para aprender mais facilmente codificações mais complicadas [6].
- Fornecer sinais de feedback multi-modais para que os alunos podem facilmente testar seu robô programado e corrigir / atualizar seus códigos [6].

- O local onde o robô se deslocará, é interessante ter o percurso com material EVA e informações táteis para os alunos conseguirem identificar o percurso a fazer com o robô [7].
- Projetar as atividades para que os participantes aprendam com os erros e redesenhem conforme necessário o programa [8].

d. Quais foram as dificuldades/limitações no uso de robótica como apoio ao ensino de programação para pessoas que são cegas?

As limitações enfrentadas nos trabalhos [4], [10], [11] foi relacionado ao processo de transferência do programa, pois, é necessário pegar o programa em um cartão microSD e conectar no robô para que ele execute. Durante as atividades uma pessoa com visão auxiliava na hora da transferência. Os alunos comentaram que “Eu quero executar operações robô móvel sozinho” [7]. Outro problema foi relacionado as pessoas com deficiência visual distinguem o início de blocos (IF, LOOP) e finais de blocos (IF, LOOP). As diferenças não foram suficientemente claras para os usuários [7].

No trabalho [3], [4] o software BricxCC, desenvolvido em Delphi, não era totalmente compatível com JAWS e, portanto, requeria a ajuda de uma pessoa com visão às vezes. Por exemplo, a ajuda às vezes é necessária para ajudar os alunos a encontrar a linha certa de código com base no erro do compilador.

Sinais de feedback que distinguem entre as curvas esquerda e direita, foram mal compreendidos pelos usuários [5].

Algumas dificuldades foram relatadas por [8]. Houve problemas com a navegação no código e como trabalho futuro, irão realizar um estudo mais aprofundado com uso de diferentes áudios (Por exemplo, *pitch*, *earcons*), para auxiliar na orientação do código e navegação. Participantes cegos, que aceleraram a leitura do texto, por vezes, se perderam durante a orientação do código em termos de construção como *if / then* e repetir blocos especialmente quando se tentava corrigir os erros. Em particular questões de declarações aninhadas *If / Then*. O problema foi impactante pois alguns participantes não estavam familiarizados com o uso de pontuação, tais como chaves e colchetes, incluindo a sua localização no teclado. Para a maioria dos participantes a possibilidade de trabalhar com um robô foi completamente nova. E a experiência por ser novidade pode ter influenciado os resultados.

III. CONCLUSÃO

Neste trabalho, foram apresentados os resultados de uma revisão sistemática sobre o ensino de programação com o apoio de robótica para deficientes visuais que foram publicados em relevantes eventos da área de Ciência da Computação e Engenharia.

Os resultados obtidos neste estudo indicam como metodologia de ensino o uso de oficinas de robótica onde é os alunos são ensinados através da aplicação de tutoriais que possuem código comentado. Os kits de robótica mais utilizados são os da LEGO mindstorm NXT, junto com o ambiente de programação BricKCC. Várias boas práticas no ensino de programação foram descobertas com a revisão, assim como as

dificuldades enfrentadas nos projetos. Todos esses dados nos nortearam durante o desenvolvimento do projeto que inclui a nova linguagem de programação e ambiente de robótica.

REFERÊNCIAS

- [1] C. J. Solomon and S. Papert, “A case study of a young child doing turtle graphics in logo,” in *Proceedings of the June 7-10, 1976, National Computer Conference and Exposition*, ser. AFIPS '76. New York, NY, USA: ACM, 1976, pp. 1049–1056. [Online]. Available: <http://doi.acm.org/10.1145/1499799.1499945>
- [2] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, “Lessons from applying the systematic literature review process within the software engineering domain,” *J. Syst. Softw.*, vol. 80, no. 4, pp. 571–583, 2007.
- [3] S. A. Ludi and T. Reichlmayr, “Developing inclusive outreach activities for students with visual impairments,” in *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*. ACM, 2008, pp. 439–443.
- [4] S. Ludi and T. Reichlmayr, “The use of robotics to promote computing to pre-college students with visual impairments,” *Trans. Comput. Educ.*, vol. 11, no. 3, pp. 20:1–20:20, 2011.
- [5] A. M. Howard, C. H. Park, and S. Remy, “Using haptic and auditory interaction tools to engage students with visual impairments in robot programming activities,” *IEEE Transactions on Learning Technologies*, vol. 5, no. 1, pp. 87–95, 2012.
- [6] C. Park and A. Howard, “Engaging students with visual impairments in engineering and computer science through robotic game programming (research-to-practice),” 2013. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84884340801partnerID=40md5=7f41c0b777d665890c4fb65835324f02>
- [7] S. Kakehashi, T. Motoyoshi, K. Koyanagi, T. Oshima, H. Masuta, and H. Kawakami, “Improvement of p-cube: Algorithm education tool for visually impaired persons,” 2015. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84922603228partnerID=40md5=5344c414263d682f4782c8d1c8916f04>
- [8] S. Ludi, L. Ellis, and S. Jordan, “An accessible robotics programming environment for visually impaired users,” 2014, pp. 237–238. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84911376346partnerID=40md5=54075766fb7327216a9ee812d33506d4>
- [9] S. Ludi, M. Abadi, Y. Fujiki, P. Sankaran, and S. Herzberg, “Jbrick: Accessible lego mindstorm programming tool for users who are visually impaired,” in *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 2010, pp. 271–272.
- [10] S. Kakehashi, T. Motoyoshi, K. Koyanagi, T. Oshima, and H. Kawakami, “P-cube: Block type programming tool for visual impairments,” *2013 Conference on Technologies and Applications of Artificial Intelligence*, vol. 0, pp. 294–299, 2013.
- [11] T. Motoyoshi, S. Kakehashi, H. Masuta, K. Koyanagi, T. Oshima, and H. Kawakami, “The usefulness of p-cube as a programming education tool for programming beginners,” vol. 2015-November, 2015, pp. 297–300. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84954045119partnerID=40md5=b47ca41674390f25a5455b1d69d190ab>