

ARTIGO

ROBÔ DE BAIXO CUSTO PROGRAMÁVEL POR VOZ PARA PORTADORES DE NECESSIDADES ESPECIAIS APRENDEREM PROGRAMAÇÃO: PROJETO E ALGORITMOS¹Ângelo Magno de Jesus²**RESUMO**

Este artigo apresenta o projeto e a implementação de algoritmos para desenvolvimento de um robô móvel educativo programável por voz. O ensino de programação, através da robótica, pode cultivar nos estudantes o interesse por disciplinas como a matemática e a física e ainda auxiliar no desenvolvimento de capacidades relacionadas ao Pensamento Computacional, que envolvem raciocínio lógico e sistemático e resolução de problemas. No entanto, crianças e jovens que são portadoras de algum tipo de deficiência física ou problemas motores relacionados as mãos podem não ter tanta facilidade de realizar a programação de um robô educativo, por não terem possibilidades de digitar os comandos via o teclado do computador. Este artigo apresenta o projeto de um robô móvel de baixo custo programável por voz, assim como os algoritmos implementados para o sistema de reconhecimento de voz. O robô segue comandos básicos, como se movimentar, abrir e fechar uma garra para pegar objetos. O objetivo principal é auxiliar alguns portadores de certas deficiências a entender princípios de programação através de programação por voz.

Palavras-chave: Robótica educativa. Ensino de Programação. Inclusão.

1 INTRODUÇÃO

Muitos são os benefícios educacionais que podem ser concretizados pelo ensino através da Robótica Educacional aliada à lógica de programação. Através da robótica, estudantes podem testar conceitos e desenvolver experiências práticas relacionados a física, matemática, lógica de programação, entre outras. Em uma sociedade em que as descobertas científicas e tecnológicas são constantes, é importante que as instituições de ensino forneçam um ensino no qual os estudantes possam aprender a “como aprender novos conceitos”, principalmente com o apoio de tecnologias. Desta forma, kits de robótica como o Lego Mindstorms e outros podem despertar o interesse de estudantes para aprender diversos conceitos relacionados à tecnologia de forma ativa e divertida. No entanto, muitos portadores

¹ **Como citar este artigo:** JESUS, Ângelo Magno de. Robô de baixo custo programável por voz para portadores de necessidades especiais aprenderem programação. **ForScience**: revista científica do IFMG, Formiga, v. 6, n. 1, e00293, 2018.

² **Autor para correspondência:** Ângelo Magno de Jesus, IFMG, e-mail: angelo.jesus@ifmg.edu.br.

de deficiência nas mãos/braços, ou mesmo pessoas que apresentam dificuldades motoras, podem não conseguir realizar a programação de robôs por não terem a possibilidade de interagir adequadamente com os dispositivos básicos comumente utilizados para programação de robôs que são o mouse e o teclado. Mesmo robôs educativos programados através de alta abstração requerem a manipulação de blocos visuais via mouse. Para se desenvolver uma educação de qualidade também é preciso desenvolver um acesso igualitário para o conhecimento, sem excluir portadores de necessidades especiais. Neste contexto, apresentamos neste artigo o desenvolvimento de um robô de baixo custo para portadores de deficiência nos braços ou mãos aprenderem conceitos básicos de programação através de comandos de voz. Em Deus et al. (2014) foi apresentada somente uma visão geral do sistema com o objetivo de explicar suas aplicações práticas. Neste artigo, descrevemos detalhadamente os arcabouços teóricos, métodos e algoritmos utilizados para o desenvolvimento do robô móvel. O robô móvel foi desenvolvido com tecnologias livres e materiais de baixo custo para se tornar uma tecnologia acessível economicamente. O desenvolvimento do projeto foi fundamentado na teoria do Construcionismo (PAPERT, 1993) e foi motivado pela importância de se estimular o Pensamento Computacional (WING, 2006) em estudantes.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Pensamento Computacional

O ensino da lógica de programação nos dias de hoje tem uma importância muito além do ensino técnico para formar profissionais de Tecnologia da Informação. Wing (2006) descreve como o Pensamento Computacional pode auxiliar no desenvolvimento de habilidades relacionadas a resolução de problemas. Segundo Wing (2006), trata-se de uma habilidade fundamental para todos, que pode inclusive auxiliar na escrita e na aritmética. Conforme CSTA e ISTE (2011) o Pensamento Computacional é um processo de resolução de problemas que inclui (mas não está limitado) as seguintes características: formular problemas de forma que se possa utilizar um computador e outras ferramentas para resolvê-los; organizar e analisar dados logicamente; representar dados através de abstrações, como modelos e simulações; automatizar soluções através do pensamento algorítmico (uma série de passos ordenados); identificar, analisar e implementar possíveis soluções com o objetivo de alcançar

a combinação mais eficiente e efetiva de etapas e recursos; e generalizar e transferir este processo de resolução de problemas para uma grande variedade de problemas. Barr e Stephenson (2011) relatam como as ferramentas mentais do Pensamento Computacional (como utilizar abstrações, automação, criação de algoritmos, analisar e coletar dados) pode trazer benefícios para o ensino fundamental, incluindo benefícios para questões interdisciplinares. Apesar do Pensamento Computacional não se referir apenas a programar computadores, nota-se, obviamente, uma grande interseção entre as habilidades exigidas na programação e as utilizadas no Pensamento Computacional.

2.2 Construcionismo

Como relatado anteriormente, o desenvolvimento do robô de baixo custo programável por voz teve como embasamento a teoria do Construcionismo. Papert (1993) apresenta o Construcionismo como sua versão pessoal do Construtivismo de Piaget. O autor relata que sua teoria se caracteriza por examinar mais de perto a construção mental. O autor também destaca que esta teoria possui uma concepção menos mentalista, uma vez que atribui uma importância especial às construções que ocorrem no mundo como um apoio para o que ocorre na cabeça. Papert também caracteriza o Construcionismo por atribuir uma significativa importância à ideia de construir na cabeça reconhecendo mais de um tipo de construção e formulando perguntas a respeito dos métodos e materiais utilizados. Nas palavras de Papert (1993) o Construcionismo possui:

[...] a conotação de 'conjunto de peças para construção', iniciando com conjuntos no sentido literal, como o Lego, e ampliando-se para incluir linguagens de programação (consideradas como "conjuntos" a partir dos quais programas podem ser feitos) até cozinhas como 'conjuntos' com os quais são construídas não apenas tortas, mas receitas e formas de matemática-em-uso (PAPERT, 1993, p. 137).

Papert também argumenta que, geralmente, a construção do conhecimento que ocorre na mente se torna mais prazerosa quando é apoiada por um tipo de "construção pública" da vida real. Entende-se a construção pública como a criação de algum tipo de artefato concreto ou abstrato, como por exemplo, uma casa de lego, um castelo de areia, uma torta, um programa de computador, um poema, uma empresa, uma teoria sobre o universo etc.

2.3 Relações

O robô móvel programado por voz descrito neste trabalho está fundamentado no Construcionismo por permitir que o aluno construa modelos de algoritmos em sua mente e implemente estas construções no mundo real através da execução dos algoritmos através do robô. O trabalho também se relaciona com o Pensamento Computacional uma vez que o estudante pode formular estratégias e algoritmos para solucionar problemas relacionados a navegação do robô em ambientes com diversos obstáculos.

3 TRABALHOS RELACIONADOS

Muitos projetos têm sido realizados para melhorar o ensino de programação e Pensamento Computacional através da utilização da robótica de baixo custo. Costella et al. (2016) apresentam o EduPi, um minicomputador de baixo custo para auxiliar na aprendizagem de programação de computadores e de robótica. O EduPi é baseado na plataforma Raspberry Pi e possui 11 ambientes de aprendizagem de programação e robótica, incluindo material de estudo. O DuinoBlocks, proposto por Alves et al. (2014), é um ambiente de programação visual para robótica educacional utilizando a plataforma Arduino, que foi desenvolvido para facilitar a iniciação na programação de professores e alunos de escolas públicas brasileiras parceiros do Programa Um Computador por Aluno do Governo Federal. Miranda et al. (2010) apresentam o desenvolvimento de um Kit de Robótica Educacional adequado para a realidade brasileira. Os autores propõem o kit RoboFácil, o qual é composto por uma implementação de software e hardware. A programação se dá através da manipulação e interconexão de objetos representados visualmente. Ferreira et al. (2016) descrevem o Se-Robô como um aplicativo voltado para dispositivos móveis que pode ser utilizado para o ensino de programação de robôs de baixo custo construídos na plataforma Arduino. O trabalho de Ferreira et al. (2016) foca no ensino de programação e robótica voltado para o público mais infantil, especialmente para crianças entre 6 a 12 anos de idade.

4 METODOLOGIA E DESENVOLVIMENTO

Esta seção apresenta os métodos, projetos e algoritmos utilizados para a concepção do robô móvel. Desta forma, para construção do robô foram considerados materiais e tecnologias

que fossem baratos e de fácil acesso. Para desenvolvimento do hardware foi utilizada a tecnologia de prototipação eletrônica Arduino, por ser de baixo custo e altamente difundida. Para desenvolvimento do software, foram utilizadas as linguagens de programação Java, por possuir muitos recursos para desenvolvimento de sistemas deste tipo, e a linguagem Arduino (baseada em C/C++). O ciclo de desenvolvimento seguiu as etapas: (A) Análise do problema, (B) Desenvolvimento de um protótipo, (C) Implementação em hardware e software e (D) Realização de testes. Estas etapas foram realizadas em ciclo, ou seja, a partir da etapa (D) poderia se retornar a etapa (A) para desenvolvimento de novas funcionalidades ou aprimoramento de outras.

A Figura 1, extraída Deus et al. (2014), ilustra um protótipo geral de funcionamento do sistema. O usuário deverá interagir com o software informando, por língua falada, os comandos básicos de programação do robô. Ao dizer a palavra "pronto" os comandos serão enviados automaticamente via rede sem fio (bluetooth) para que o robô os execute em sequência.



Figura 1- Protótipo de Funcionamento Geral do Sistema
 Fonte: Deus et al. (2014).

A Figura 2 mostra a arquitetura geral de funcionamento do sistema. Nesta arquitetura, pode-se notar que a solução é distribuída em dois componentes principais, o computador que deverá rodar o software de programação e o robô que deverá receber os comandos enviados via bluetooth do computador e os executar. O Ambiente de programação é dividido em três camadas: (1) A Interface gráfica com usuário, responsável por capturar o áudio do microfone e mostrar para o usuário o algoritmo com os comandos a serem executados pelo robô; (2) O Processo de reconhecimento de voz, responsável por receber o áudio gravado do microfone e o transformar em um vetor de caracteres, este processo é feito remotamente através da API externa; (3) O componente Estrutura de Programação responsável por adicionar novos comandos e verificar se a estrutura da algoritmo está correta; (4) Conector, que é responsável por estabelecer a conexão remota, via bluetooth, com o robô, e enviar os comandos através de flags.

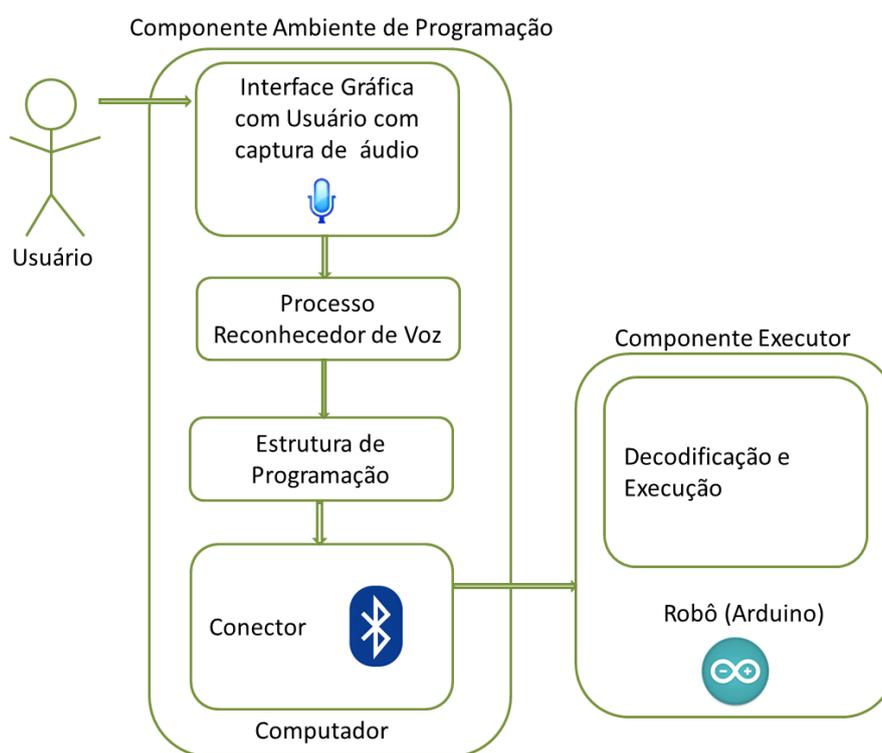


Figura 2 - Arquitetura Geral do Sistema
 Fonte: Deus et al. (2014).

4.1 Desenvolvimento do Hardware

O hardware é composto por: placa de prototipação eletrônica Arduino; dois motores DC simples de 6 Volts; garra de acrílico; servo motor com 1.6 Kg*cm (0.16 Nm) de torque; placa de bluetooth do tipo RN-42; sensor infravermelho do tipo breakout sensor de linha analógico - QRE1113; sensor ultrassônico HCSR04, com ângulo de efeito de 15°, alcance de 2cm ~ 4m e precisão 3mm. Estes componentes foram adquiridos ou coletados separadamente, sendo que nenhum Kit de Robótica predefinido foi utilizado na montagem do robô.

4.2 Linguagem de Programação

A linguagem de programação proposta, apesar de ser simples, trabalha com os principais conceitos iniciais de programação como sequenciamento, condicional e repetição. Os comandos de programação são divididos em: (A) comandos direcionais para movimentação do robô; (B) Comandos de controle da garra (abrir/fechar), (C) Comando condicional, que utiliza o sensor ultrassônico para verificar a existência de um obstáculo e decidir se deve executar o próximo comando ou não; (D) Repetição, capaz de executar um comando em loop. A Gramática Regular que representa a linguagem de programação é ilustrada pelo Quadro 1.

$\langle lista_comandos \rangle \rightarrow \langle comando \rangle \langle lista_comandos \rangle / \lambda$
 $\langle comando \rangle \rightarrow frente / trás / direita / esquerda / abrir /$
 $fechar /$
 $repita \langle num \rangle / barreira \langle comando \rangle$
 $\langle num \rangle \rightarrow 0 \langle num \rangle / 1 \langle num \rangle / 2 \langle num \rangle / 3 \langle num \rangle / 4 \langle num \rangle / 5 \langle num \rangle /$
 $6 \langle num \rangle / 7 \langle num \rangle / 8 \langle num \rangle / 9 \langle num \rangle / \lambda$

Quadro 1 - Gramática Regular da Linguagem de Programação
 Fonte: Próprio autor.

4.3 Algoritmos

O sistema funciona através de dois algoritmos principais, um que é executado no computador e outro que é executado no robô. O Algoritmo 1 conforme Quadro 2, denominado Processo de Reconhecimento de Voz, tem por objetivo: (1) realizar a leitura dos comandos de

voz do usuário, conforme um dicionário pré-estabelecido conforme a sintaxe da linguagem de programação; (2) enviar estes comandos para reconhecimento e conversão para comandos textuais e (3) enviar os comandos reconhecidos para o robô executar. O Algoritmo 2 conforme Quadro 3, denominado Decodificação e Execução, funciona dentro do robô móvel e tem por objetivo receber os comandos e os executar sequencialmente.

```

=====
Algoritmo 1 - Processo Reconhecimento de Voz
=====
1 lista_comandos ← ∅
2 comando_reconhecido ← ""
3 dicionário ← {"frente", "trás", "direita", "esquerda"...}
4 enquanto comando_reconhecido ≠ "pronto" faça
5     áudio ← ler_entrada_microfone()
6     comando_reconhecido ← reconhecedor_audio_remoto(áudio, dicionário)
7     se comando_reconhecido = "apaga" então
8         lista_comandos.remove_ultimo( )
9     se não
10        lista_comandos.adicionar(comando_reconhecido)
11    fim se
12 fim enquanto
13 para cada comando ∈ lista_comandos faça
14     flag ← obter_flag(comando)
15     se comando = "repita" então
16         n ← lista_comandos.obeter_próximo(comando)
17         para cont de 0 até n faça
18             enviar_para_robô(flag)
19     fim para
20 se não
21     enviar_para_robô(flag)
22 fim se
23 fim para

```

Quadro 2 - Processo de Reconhecimento de Voz
 Fonte: Próprio autor.

=====
Algoritmo 2 - Decodificação e Execução
=====

```
=====
1 barreira ← 0
2 enquanto verdadeiro faça
3   flag ← leia_entrada()
4   se barreira = 1 então
5     barreira ← 0
6     distância ← sensor_ultra_sonico( )
7     se distância > 0.20 m então
8       continue
9     fim se
10  fim se
11  escolha flag
12    caso FLAG_FRENTE:
13      frente(0.5)
14    caso FLAG_TRAS:
15      tras(0.5)
16    caso FLAG_DIREITA:
17      direita(0.5)
18    caso FLAG_ESQUERDA:
19      esquerda(0.5)
20    caso FLAG_ABRIR:
21      abrir_garra( )
22    caso FLAG_FECHAR:
23      fechar_garra( )
24    caso FLAG_BARREIRA:
25      barreira ← 0
26  fim escolha
27 fim enquanto
```

Quadro 3 - Processo de Reconhecimento de Voz
Fonte: Próprio autor.

5 RESULTADOS

Como resultados foram obtidos o ambiente de programação (Figura 3) e o robô móvel (Figura 4). Nos experimentos, os algoritmos demonstraram funcionar corretamente, sendo que os comandos reconhecidos por voz eram enviados e executados pelo robô sem ocorrência de

falhas. Não foram realizados testes sistemáticos com o reconhecimento dos comandos de voz, uma vez que o reconhecimento do sinal de áudio se dá através de APIs de terceiros, além do fato do resultado do reconhecimento depender da linguagem padrão utilizada. As implementações utilizaram inicialmente a API da Microsoft (MICROSOFT, 2017), e posteriormente a API Google Speech (GOOGLE, 2017) pelo fato desta última aceitar comandos em português enquanto que a primeira aceita apenas comandos em inglês. É importante ressaltar que a implementação do Processo de Reconhecimento de Voz é independente de uma API específica, sendo que a modularização do sistema permite que diferentes APIs de reconhecimento de voz possam ser incorporadas ao sistema.

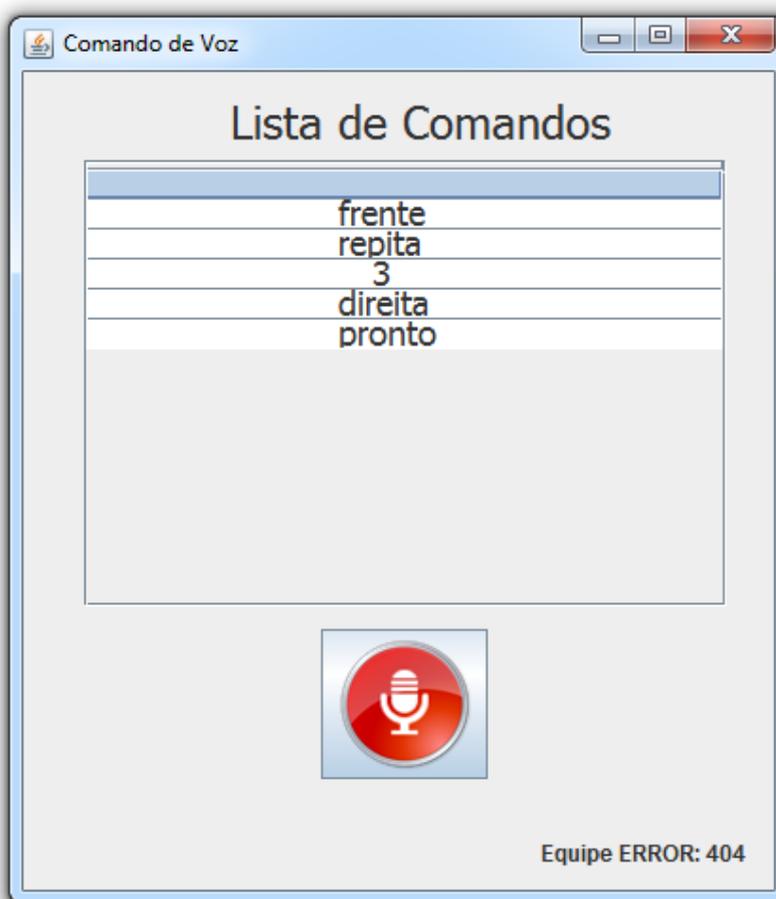


Figura 3 - Ambiente para programação por voz
Fonte: Deus et al. (2014).

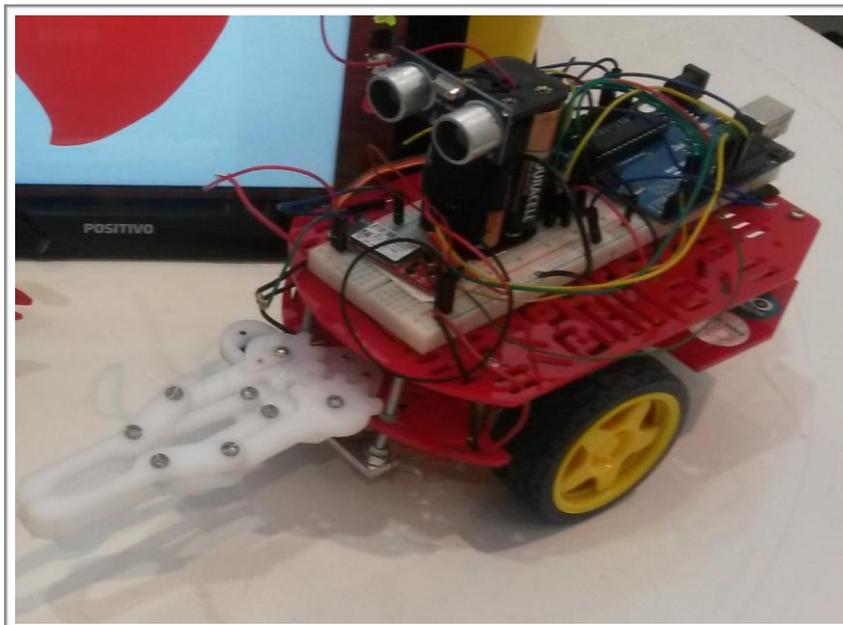


Figura 4 - Robô Móvel
Fonte: Próprio autor.

6 CONSIDERAÇÕES FINAIS

Este artigo apresentou o projeto de um robô de baixo custo programável por voz. O robô e seu ambiente de programação foram desenvolvidos com o objetivo de tornar o mundo da robótica e programação acessível para portadores de deficiências nos braços ou mãos, que geralmente não conseguem utilizar o teclado ou mouse para realizar a programação “convencional”. Para tornar isto possível, também foi desenvolvida uma linguagem, que apesar de ser bastante simples, utiliza conceitos básicos de programação como sequenciamento de comandos, estruturas condicionais e estruturas de repetição. Nos experimentos, o robô e todo o sistema que o envolve demonstraram funcionar corretamente. Entre as limitações encontradas, está o fato da necessidade do uso de um computador para reconhecimento dos comandos de voz. Este problema pode ser resolvido através da utilização de plataformas móveis, especialmente celulares, que são mais acessíveis financeiramente. A transferência do ambiente de programação para plataformas móveis está sendo estudada pelos autores.

LOW COST PROGRAMMABLE VOICE ROBOT FOR SPECIAL NEEDS PEOPLE LEARN PROGRAMMING - PROJECT AND ALGORITHMS

ABSTRACT

This article presents the design and implementation of algorithms for the development of a mobile educational programmable voice robot. The teaching of programming, through robotics, can cultivate in students the interest in disciplines such as mathematics and physics and assists in the development of capacities related to Computational Thinking, which involve logical and systematic reasoning and problem solving. However, children and youth who have some kind of physical disability or hand-related motor problems may not have ease to perform an educational robot, as they are not able to type the commands via the computer keyboard. This paper presents the design of a low cost mobile robot programmable by voice, as well as the algorithms implemented for the voice recognition system. The robot follows basic commands such as moving, opening and closing a claw to grab objects. The main purpose is to help some people with certain disabilities understand programming principles through voice programming.

Keywords: Educational robotics. Teaching programming. Inclusion.

REFERÊNCIAS

ALVES, R. M.; SAMPAIO, F. F.; ELIA, M. da F. Duinoblocks: desenho e implementação de um ambiente de programação visual para robótica educacional. **Revista Brasileira de Informática na Educação**, v. 22, n. 3. 2013. Disponível em: <<http://br-ie.org/pub/index.php/wcbie/article/view/3167/2735>>. Acesso em: 15 fev. 2017.

BARR, V.; STEPHENSON, C. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. **Acm Inroads**, v. 2, n. 1, p. 48-54, 2011. Disponível em: <[https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/BarrStephensonInroads Article.pdf](https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/BarrStephensonInroads%20Article.pdf)>. Acesso em: 14 fev. 2017.

COSTELLA, L.; LICKS, G. P.; TEIXEIRA, A. Uma solução livre e de baixo custo para prática e aprendizagem de programação e robótica. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE). 27., 2016, Uberlândia, MG. **Anais...Uberlândia, MG., UFU, 2016**. p. 846.

COMPUTER SCIENCE TEACHERS ASSOCIATION-CSTA; STANDARTS FOR COMPUTER SCIENCE EDUCATOR-ISTE. **Computational thinking teacher resources**. 2011. Disponível em: <<http://csta.acm.org/curriculum/sub/compthinking.html>>. Acesso em: 15 fev. 2017.

DEUS, C. F. de et al. Desenvolvimento de um Robô de Baixo Custo Programável por Voz para Portadores de Necessidades Especiais. In: MOSTRA NACIONAL DE ROBÓTICA. 4., 2014, São Carlos, SP. **Anais... São Carlos, SP., UNESP, 2014**. p. 328-331.

FERREIRA, L. A. C.; JESUS, Â. M. de; RUFO, M. C.; SANTOS, F. M. **Se-Robô:** Aplicativo para Robótica Educacional de Baixo Custo. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE). 27., 2016, Uberlândia, MG. **Anais...**Uberlândia, MG., UFU, 2016. p. 1285.

GOOGLE. **Cloud Speech API**. Disponível em: <<https://cloud.google.com/speech/?hl=pt-br>>. Acesso em: 15 jul. 2017.

MICROSOFT. **Microsoft Speech API**. Disponível em: <[https://msdn.microsoft.com/en-us/library/ee125663\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/ee125663(v=vs.85).aspx)>. Acesso em: 15 jul. 2017.

MIRANDA, L. C. de; SAMPAIO, F. F.; BORGES, J. A. dos S. Robofácil: especificação e implementação de um kit de robótica para a realidade educacional brasileira. **Brazilian Journal of Computers in Education**, v. 18, n. 03, p. 46, 2011. Disponível em: <<http://www.br-ie.org/pub/index.php/rbie/article/view/1275/1126>>. Acesso em: 15 fev. 2017.

PAPERT, S. **A máquina das crianças:** repensando a escola na era da informática. Porto Alegre: Artmed, 1993. p. 220.

WING, J. M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, mar. 2006. Disponível em: <<https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>>. Acesso em: 14 fev. 2017.

DADOS DO AUTOR

Ângelo Magno de Jesus

Lattes: <http://lattes.cnpq.br/8650963652700290>

E-mail: angelo.jesus@ifmg.edu.br

Doutorando em Ensino de Ciências e Matemática pela Universidade Cruzeiro do Sul. Mestre em Informática pela Pontifícia Universidade Católica de Minas Gerais (PUC-Minas).

Atualmente é professor de Informática do IFMG - *Campus* Ouro Branco.