

Desenvolvimento e Configuração de Cenários de Robótica para Fomentar a Aprendizagem de Programação aos Alunos do Ensino Fundamental

Thais Oliveira Almeida^{1,2}, José Francisco de Magalhães Netto², Tiago de Paula Custódio²

¹Departamento de Ciência da Computação – Universidade Federal de Roraima (UFRR) – 69.310-000 – Boa Vista – RR – Brasil

²Programa de Pós Graduação em Informática (PPGI) – Universidade Federal do Amazonas (UFAM) – Manaus – AM – Brasil.

thais.oliveira@ufrr.br, {jnetto, tpc}@icomp.ufam.edu.br

Abstract. *This paper presents the development and configuration of robotics scenarios to motivate students to learn early programming concepts. The contents presented here were tested with the Lego Mindstorms NXT robotic kit, and the Python programming language was integrated into the scenarios in view of its simplicity and similarity with object-oriented programming. The activities were made available in order of difficulty, and a Case Study with the work developed was carried out in a public school with Elementary School students with promising results.*

Resumo. *Este artigo apresenta o desenvolvimento e configuração de cenários de robótica para motivar alunos a aprenderem conceitos iniciais de programação. Os conteúdos aqui apresentados foram testados com o kit robótico Lego Mindstorms NXT, e a linguagem de programação Python foi integrada aos cenários tendo em vista sua simplicidade e similaridade com programação orientada a objetos. As atividades foram disponibilizadas por ordem de dificuldade, e um Estudo de Caso com o trabalho desenvolvido foi realizado em uma escola pública com alunos de Ensino Fundamental com resultados promissores.*

1. Introdução

A Robótica Educacional (RE) é um termo amplamente utilizado para indicar que os alunos devem programar as ações dos robôs, ou até mesmo projetá-los, criá-los e montá-los associando os conteúdos com fins educacionais. Em algumas escolas as atividades pedagógicas com robótica estão sendo utilizadas de forma interdisciplinar com o objetivo de promover o aprendizado de conceitos curriculares, e utilizando sistemas robóticos que integram o fazer como meio de aprender [D’Abreu et al. 2011].

A inserção de recursos tecnológicos no processo de ensino/aprendizagem ainda é um grande desafio para o Brasil, pois a maioria da população não tem acesso a recursos

computacionais ou jogos educativos [Santos et al. 2010]. [Trentin et al. 2013] afirmam que neste contexto de emergência da tecnologia na escola, a robótica apresenta-se como dispositivo potencializador da aprendizagem sendo considerada uma das dez áreas mais importantes de pesquisa no mundo, ainda pouco difundida no Brasil, principalmente em termos educacionais.

De acordo com [Azevedo et al. 2010] muitas pessoas conseguem identificar um dispositivo robótico, mas possuem alguma dificuldade em construir uma definição abrangente sobre o mesmo. Apesar de muitas pesquisas indicarem a RE como sendo uma ferramenta pedagogicamente rica, infelizmente ela não faz parte do cotidiano das escolas brasileiras [Medeiros Filho & Gonçalves 2008]. A explicação para tal fato, passa pela dificuldade na aquisição do equipamento, pois seu custo ainda é proibitivo.

Este artigo possui como objetivo desenvolver e configurar cenários de robótica para fomentar a aprendizagem de programação aos alunos do Ensino Fundamental em um laboratório remoto. Os passos adotados para alcançar este objetivo devem compreender as seguintes etapas: (1) O aluno acessa remotamente o laboratório e efetua seu cadastro; (2) O aluno seleciona a atividade a ser realizada; (3) O robô é controlado remotamente por um programa simples enviado por um aluno; (4) Um aluno pode começar a realizar as atividades sem entender conceitos abstratos de programação; (5) São disponibilizadas algumas funções para os alunos a fim de auxiliá-los na realização das tarefas.

Neste trabalho um cenário está associado a um desafio que um aluno pode realizar, associado a um grau de dificuldade. Os conteúdos aqui apresentados foram testados com o kit robótico *Legó Mindstorms NXT* e integrados a um laboratório remoto para realização dos experimentos. Nesta pesquisa, um programa é escrito na linguagem de programação NXT-Python (Python para *Legó NXT*). A linguagem de programação Python foi adotada tendo em vista sua simplicidade e similaridade com programação orientada a objetos. Os resultados mostraram que os cenários desenvolvidos foram capazes de despertar o interesse dos alunos, e motivá-los a aprender programação e robótica.

A metodologia desta pesquisa é de natureza qualitativa, pois a interpretação dos fenômenos e a atribuição de significados são básicas no processo. Ela não requer o uso de métodos e técnicas estatísticas [Gil 2010]. Do ponto de vista do seu objetivo, é de natureza exploratória, pois não requer a formulação de hipóteses para serem testadas, e se restringe por definir objetivos e buscar mais informações sobre determinado assunto de estudo [Gil 2010]. Foi realizado um Estudo de Caso em uma escola pública, com alunos do 7º e 8º ano do Ensino Fundamental, para realização dos experimentos. Para coleta dos dados, foi aplicado um questionário para que os alunos pudessem avaliar a metodologia e atividades propostas.

Para relatar o trabalho desenvolvido, o restante do artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados. A Seção 3 apresenta a descrição do trabalho, com o esboço dos cenários desenvolvidos e explanação sobre o processo de programação remota. O Estudo de Caso está descrito na Seção 4, e por fim a Seção 5 apresenta as conclusões obtidas.

2. Trabalhos Relacionados

Para avaliar o estado das pesquisas concernentes ao presente trabalho, esta seção apresenta alguns artigos que surgiram de uma investigação realizada no contexto de diferentes propostas para auxiliar no ensino de programação e robótica. Além disso, os trabalhos apresentados nesta seção foram úteis para identificar as contribuições e limitações, além de servir como elemento motivador.

A RE de Baixo Custo com Linguagem de Programação Visual no contexto do Programa Um Computador por Aluno do Governo Federal, é promovida no trabalho de [Alves et al. 2015]. Essa pesquisa trata especificamente do desenho e implementação de um ambiente de programação visual denominado DuinoBlocks, que estende os recursos de programação do Arduino para permitir ao iniciante, preferencialmente professores e alunos do ensino básico, programar um dispositivo robótico a fim de enriquecer o processo de ensino-aprendizagem.

Um relato de experiência da aplicação da plataforma Code.org, especificamente do jogo Labirinto Clássico, é descrito no trabalho de [Martins et al. 2016]. Os testes foram realizados com 168 crianças, do 8º e 9º ano do Ensino Fundamental de uma escola pública na cidade de Manaus – AM. O estudo realizado demonstrou que o jogo apoia o ensino de programação, além de fornecer uma ótima experiência aos usuários, alinhando motivação e aprendizagem.

O trabalho desenvolvido por [Monteiro et al. 2016] utiliza a RE como alternativa para o ensino de linguagem de programação, estimulando o Pensamento Computacional, em uma turma do Ensino Médio Integrado em Mecatrônica do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM). O hardware livre Arduino foi escolhido para realização do experimento, que consistiu de uma aplicação real do cotidiano dos alunos, a fim de fundamentar melhor os conceitos de programação estruturada.

A aplicação de uma atividade que introduz os conceitos básicos de algoritmos, com o uso de jogos de tabuleiros, para alunos do 5º ano de escolas públicas é apresentada no trabalho de [Pinho et al. 2016]. As atividades didáticas propostas visam aumentar o interesse dos estudantes ao lidar com conceitos como algoritmos e procedimentos, incluindo também abstração de dados e decomposição de problemas. Os resultados avaliados foram satisfatórios e os objetivos de desenvolver atividades e ações para a promoção de habilidades do pensamento computacional no âmbito da Educação Básica no Brasil, plenamente alcançados.

Todos os trabalhos apresentados nesta seção têm um reconhecimento significativo junto ao meio acadêmico e motivaram o desenvolvimento desta pesquisa. Como diferencial, neste artigo apresentamos o desenvolvimento e configuração de cenários de robótica, que fomentem a aprendizagem de programação aos alunos do Ensino Fundamental. Os materiais aqui apresentados podem ser utilizados em aulas presenciais (laboratórios físicos), sem a necessidade de estarem integrados a um laboratório remoto. Além disso, novas atividades podem ser incluídas permitindo a participação de alunos que possuam conhecimentos prévios em programação.

3. Descrição do Trabalho

Nesta pesquisa, um programa é escrito por um aluno na linguagem de programação Python. Em seguida, o programa é compilado e carregado no robô por meio da conexão *Bluetooth*. Depois que o código é enviado para o robô, o mesmo passa a ser controlado pelo programa, executando as ações que foram programadas. É possível utilizar os materiais aqui apresentados em laboratórios físicos, e carregar o código por meio da porta USB. A limitação dessa última alternativa relaciona-se aos programas que precisam ser modificados, consequentemente precisando ser novamente compilados, carregados e baixados o arquivo binário por meio da interface USB.

Os seguintes passos foram aplicados para a configuração dos cenários: (1) Os alunos não modificam o programa no robô. O programa é modificado no laboratório remoto; (2) Os comandos programados pelos alunos são enviados pelo robô através da comunicação sem fio *Bluetooth*; (3) O robô é controlado pelo programa que foi armazenado na base de dados, e executado diretamente no servidor. Esse programa descreve a combinação de comandos básicos. (4) Como linguagem de programação, Python é escolhida.

3.1. Configuração do Robô

O kit *Lego Mindstorms* é voltado para ensinar programação de robôs, e foi lançado comercialmente em 1998 [Liang et al. 2013]. O projeto foi originalmente inspirado por [Papert 1994] e desenvolvido em uma parceria entre o *Media Lab* do *Massachusetts Institute of Technology* (MIT) e o *Lego Group*. Trata-se de uma plataforma educacional que representa a evolução de seis décadas de técnicas de construção modular do Lego e métodos de ensino baseado em computador.

O kit é definido como um conjunto composto por uma variedade de peças físicas, que quando combinadas podem produzir uma variedade de robôs [Liang et al. 2013]. Cada componente tem suas funcionalidades específicas: os motores são os responsáveis por movimentar a estrutura da montagem; os sensores são os responsáveis pela coleta das informações junto ao meio externo. O controlador central, também conhecido por módulo RCX (*Robotic Command Explorer*) é o responsável pela parte inteligente, contendo o software que gerencia o sistema.

O robô utilizado nesta pesquisa encontra-se descrito na Figura 1 e possui 1 sensor de cor, 4 rodas e 2 motores responsáveis por direcionar as rodas esquerdas e direitas. Ele está equipado com um processador AM7 32-bits. Os alunos desenvolvem programas que podem controlar o robô remotamente, de maneira que seja realizada a tarefa que foi atribuída para eles (como por exemplo, acender a luz vermelha do sensor de cor).

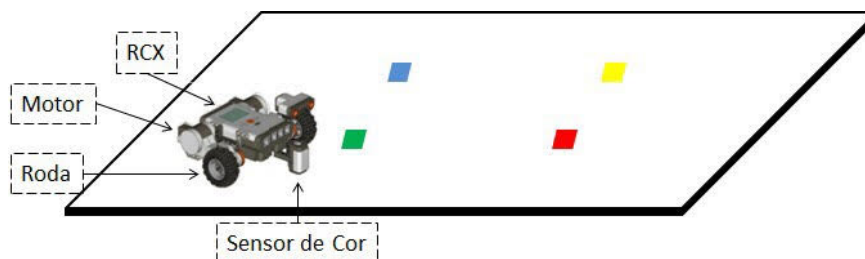


Figura 1. O robô e seus componentes

Todos os alunos usaram o mesmo robô nos experimentos que foram realizados. O robô pode percorrer o retângulo ao longo da linha preta desenhada sobre a área branca. Novos robôs podem ser configurados com o acréscimo de novos sensores e/ou motores, e novas atividades podem ser sugeridas aos alunos.

3.2. Adoção do Python

O Python¹ é uma linguagem de *script* orientada a objetos, disponível sem nenhum custo e suportada por várias plataformas, incluindo Windows, Linux e Mac. A adoção do Python justifica-se por ser uma linguagem utilizada globalmente, bem definida e fácil para usuários iniciantes em programação. O trabalho desenvolvido por [Shikama 2014] sustenta essa afirmação.

De maneira geral, um programa em Python pode ser lido facilmente. Ao contrário de outras linguagens de programação como C e C++, o Python utiliza a indentação (reco) como meio de identificar os blocos do programa. Caso um programa não esteja indentado corretamente, ele não funcionará como esperado. Acredita-se que este não é um problema sério para alunos iniciantes, tendo em vista que as definições de funções e variáveis não são complexas.

O NXT-Python² é um pacote utilizado para controlar um robô *Legó Mindstorms NXT* utilizando a linguagem de programação Python. A comunicação com o robô pode ser realizada via USB ou *Bluetooth*. Como nesta pesquisa utilizamos a tecnologia *Bluetooth* para programar o robô remotamente, foi necessário realizar também a instalação do *Pybluez*³ no servidor, um módulo em Python que permite acesso aos recursos *Bluetooth*. Além disso, o Python possui um problema de compatibilidade entre as versões 2 e 3. Para o correto funcionamento do NXT-Python, deve-se instalar a versão 2.6 do Python ou superior. O pacote utilizado nesta pesquisa não funciona com a versão 3 da linguagem.

3.3. Processo de Programação Remota

Nesta pesquisa, um programa é escrito na linguagem de programação NXT-Python. Após o programa ser enviado por um aluno para o servidor, o mesmo é armazenado

¹ <http://www.python.org>

² <http://pypi.python.org/pypi/nxt-python>

³ <https://github.com/karulis/pybluez>

como um arquivo de texto no Banco de Dados (BD) do sistema. Após isso, o arquivo executável é gerado. Caso o código não possua erros sintáticos, o programa, então, é executado no servidor e controla o robô remotamente. As ações são apresentadas para os alunos no navegador via câmera IP instalada no laboratório físico.

O robô irá mover-se para frente, ou fazer uma curva a direita ou esquerda de acordo com os comandos enviados pelo programa. A velocidade de rotação do robô também é controlada pelo programa. A Figura 2 apresenta a estrutura de camadas que foi desenvolvida para permitir a integração das tecnologias.

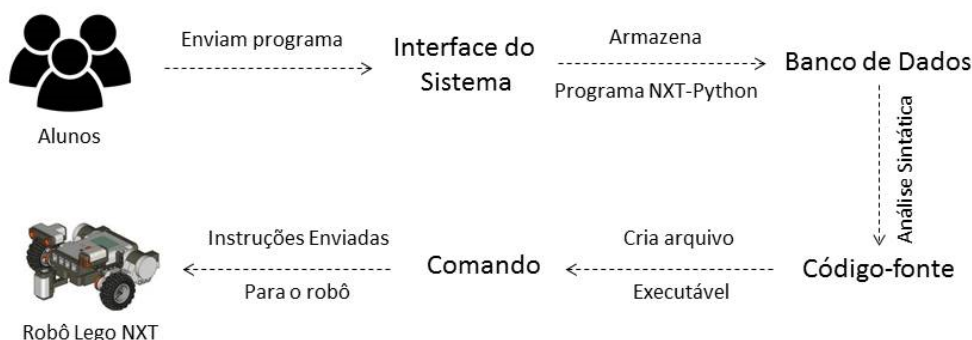


Figura 2. Processo de Programação Remota

O aluno digita o programa escrito na linguagem NXT-Python a partir da interface gráfica, que através de scripts desenvolvidos na linguagem PHP armazenam esses dados no servidor *Web*. O Quadro 1 apresenta as funções que os alunos podem usar em seus programas Python. Inicialmente foram criadas cinco funções para auxiliá-los na realização das tarefas. Novas funções podem ser desenvolvidas, de acordo com o conhecimento em programação de cada um.

Quadro 1. Funções e Significados

Função	Descrição
luz(brick, string_cor)	Acende o sensor de cor (vermelho, azul ou verde).
apaga_luz(brick)	Apaga luz do sensor de cor.
tempo_de_espera(tempo)	Essa função faz com que o robô espere por um tempo em segundos, até executar uma nova ação.
correr_por_tempo(brick, velocidade, tempo)	Permite que o robô percorra uma distância com base no tempo velocidade determinados.
rotaciona(brick, velocidade, numero_rotacao)	Rotaciona o robô de acordo com a velocidade e número de rotação.

A seguir temos um exemplo de código que pode ser utilizado para programar o robô (esse exemplo mantém acesa a luz azul do sensor de cor por cinco segundos e após esse tempo apaga). A classe *BlueSock* (utilizada na 1ª linha) possui as funções de comunicação via *Bluetooth*, processamento de dados do registro e sincronização entre o robô e o programa em execução no servidor. Como os detalhes de implementação dessa classe ficam ocultas, os alunos devem estar cientes apenas do endereço MAC do robô.

```

from      nxt.bluesock      import
BlueSock
from funcoes import *
ID = '00:16:53:11:25:73'
sock = BlueSock (ID)
brick = sock.connect()
luz(brick, "azul")
tempo_de_espera(5)
apaga_luz(brick)
sock.close()
    
```

A 1ª linha importa a classe *BlueSock* do módulo *nxt.bluesock*. Esta classe deve ser utilizada obrigatoriamente para que os comandos sejam enviados corretamente pela tecnologia *Bluetooth*, que emprega um endereço MAC de 48 bits. A 2ª linha importa as funções pré-definidas no sistema e estão detalhadas no Quadro 1. A 3ª linha gera o objeto "ID", em que o parâmetro '00:16:53:11:25:73' é um exemplo de endereço MAC do robô. A 4ª linha declara a variável "sock", que atrela o endereço MAC do robô à classe *BlueSock*. A 5ª linha realiza a conexão com o robô, e em seguida várias combinações de códigos e funções em Python podem ser inseridas. A última linha encerra a conexão com o robô.

3.4. Atividades Disponíveis

Na abordagem desenvolvida, foram propostas 5 atividades a serem realizadas pelos alunos, e a atribuição do perfil de cada um está relacionada ao seu desempenho ao programar o robô. É necessário que tenha um Supervisor/Professor acompanhando os alunos durante a realização das atividades, para auxiliar em possíveis dúvidas e "aprovar" ou "reprovar" ao final da realização de cada tarefa. Foi necessário tomar essa decisão, tendo em vista que podem ser enviados códigos que não possuam erros sintáticos, porém não atende logicamente ao que foi especificado no problema. É importante destacar que ao carregar uma atividade e concluí-la com êxito, é atribuída uma pontuação de acordo com o nível que ele se encontra, conforme está descrito no Quadro 2.

Quadro 2. Atividades Disponíveis

ID	Descrição	Pontuação	Nível
1	Manter acesa a luz verde do sensor de cor por 5 segundos.	10	Iniciante
2	Manter acesa a luz vermelha do sensor de cor por 5 segundos e desligá-la. Fazer o mesmo para a cor azul.	15	Básico
3	Mover o robô em linha reta por 10 segundos, com a luz azul ligada.	25	Intermediário
4	Fazer o robô caminhar ida e volta em linha reta. Na ida a luz verde deve estar ligada, na volta a luz vermelha.	30	Avançado
5	Programar o robô para percorrer todo o retângulo. Acender a luz azul caso esteja percorrendo o comprimento, e a luz verde caso seja a largura.	40	Experiente

4. Estudo de Caso

Os testes com os materiais desenvolvidos foram aplicados em duas turmas experimentais de ensino fundamental (7º e 8º ano respectivamente) do Colégio de Aplicação da Universidade Federal de Roraima, e tiveram a duração de duas semanas. Para participação na atividade, os alunos receberam um Termo de Consentimento Livre e Esclarecido (TCLE) para que pudessem solicitar a anuência dos seus responsáveis. No primeiro contato com os alunos foi realizada uma apresentação a cerca dos cenários de

robótica, e uma breve explanação sobre algoritmos e raciocínio lógico. Em seguida foi realizada a divisão de alunos por dia para realização dos testes, a ocorrer no laboratório de informática no horário oposto das aulas.

No total, 14 alunos com faixa etária entre 12 e 16 anos participaram dos testes. O laboratório físico estava em perfeito estado de funcionamento e apresentava conforto térmico. Para realização dos experimentos, havia a disponibilidade de uso de um computador por aluno, e todos afirmaram saber manusear o mesmo e navegar na Internet. A duração de cada experimento foi de 40 minutos, e somente 1 aluno era atendido por horário.

Os alunos foram convidados a escrever 5 programas simples (conforme detalhado no Quadro 2). Como resultados obtidos, 4 alunos concluíram todas as atividades no tempo estipulado, e 7 alunos realizaram 4 atividades. 2 alunos conseguiram concluir apenas 3 atividades (neste dia, a Internet estava muito lenta e interferiu no resultado). Somente 1 aluno concluiu 2 atividades, pois ele chegou atrasado para realização dos experimentos.

Ao final dos testes, foi aplicado um questionário para avaliar o desempenho dos alunos. As respostas foram positivas e motivadoras. Ao serem questionados sobre a metodologia utilizada, todos os alunos (100%) avaliaram como muito importante, e demonstraram-se interessados em continuar aprendendo programação e robótica. Em relação ao conhecimento adquirido em programação de computador, 43% afirmaram já ter algum conhecimento antes, e 57% apontaram que obtiveram muito conhecimento.

Embora alguns alunos tenham sentido dificuldade em entender a estruturação dos programas em Python, 71% dos alunos esboçaram uma impressão positiva no Estudo de Caso realizado, e demonstraram satisfação quando o robô movimentava-se corretamente. Por mais que a preocupação maior fosse com alunos que não tinham experiência com programação, após os testes realizados, a maioria afirmou que era mais fácil do que eles imaginavam.

Através deste Estudo de Caso, podemos identificar as seguintes vantagens sobre o trabalho desenvolvido: (1) A utilização de robôs pode atrair a atenção de estudantes que não tenham experiência em programação; (2) Como um programa em Python pode ser modificado rapidamente, os erros encontrados eram corrigidos realizando uma simples edição no código; (3) Os alunos ficaram impressionados quando o robô realizava o que estava proposto na tarefa; (4) Os alunos sugeriram que fossem disponibilizadas outras tarefas e tivessem mais tempo para praticar o que foi aprendido.

No entanto, encontramos algumas desvantagens: (1) Foi necessário muito esforço para orientar cada experimento; (2) O apoio de monitores era necessário para ajudar os alunos quando encontrassem problemas de programação; (3) O nível de dificuldades dos desafios que os alunos devem realizar deve ser reconsiderado, incluindo desde exemplos mais simples, a exemplos mais complexos. (4) O número de robôs é limitado (somente um robô foi utilizado).

5. Conclusões

Este artigo relatou o desenvolvimento e configuração de cenários de robótica para motivar alunos na aprendizagem de programação. Nosso objetivo por meio deste trabalho foi fornecer aos alunos a oportunidade de desenvolver programas em uma linguagem de programação que controlam um robô remotamente. Os alunos podem programar o robô para andar em linha reta, ou até mesmo para executar mais ações conforme solicitado. Com pequenas adaptações no robô, é possível incrementar o número de sensores e funções disponíveis para uso, assim como expandir a quantidade de atividades.

Sugere-se aos alunos que tenham interesse em utilizar os materiais aqui apresentados, possuir conhecimentos básicos em informática e ter experiência na utilização de computadores. Basicamente, sugere-se que saibam digitar texto e interagir com sistemas operacionais. Além disso, os alunos devem saber ligar e desligar um computador, assim como interagir com dispositivos de entrada e saída (teclado, mouse e monitor).

Deve-se encorajar a participação de alunos interessados em aprender programação e robótica, independente de seus conhecimentos prévios relacionados a lógica de programação. Uma vez que as características dos alunos variam de acordo com a idade, seus interesses, habilidade escolar e características da escola ou universidade, a capacidade de personalizar ou até mesmo expandir os materiais é considerada importante. Pretende-se melhorar os materiais aqui apresentados com base nos resultados dos experimentos realizados. Esperamos que esse estudo incentive iniciativas de inserção de programação no Ensino Fundamental.

Agradecimentos

Este trabalho recebeu o apoio do Programa de Apoio a Iniciação Científica (PAIC) da Fundação de Amparo e Pesquisa do Estado do Amazonas (FAPEAM).

Referências

- Alves, R. M., Sampaio, F. F., & Elia, M. D. F. (2015). “Duinoblocks: Desenho e implementação de um ambiente de programação visual para robótica educacional”. *Revista Brasileira de Informática na Educação*, 22(03), 126.
- Azevedo, S., Aglaé, A., & Pitta, R. (2010). “Minicurso: Introdução a Robótica Educacional”. 62^a Reunião Anual da SBPC. Disponível em: <<http://www.sbpnet.org.br/livro/62ra/minicursos/MC%20Samuel%20Azevedo.pdf>>, acesso em 30/05/2017.
- D'Abreu, J. V. V., Mirisola, L. G., & Ramos, J. J. (2011). “Ambiente de robótica pedagógica com br_gogo e computadores de baixo custo: Uma contribuição para o ensino médio”. In *Anais do XXII Simpósio Brasileiro de Informática na Educação* (Vol. 1, No. 1, p. 100–109).

- Gil, A. C. (2010). “Como elaborar projetos de pesquisa”. 5ª edição – São Paulo, Editora Atlas.
- Liang, H. N., Fleming, C., Man, K. L., & Tillo, T. (2013). “A first introduction to programming for first-year students at a chinese university using lego mindstorms”. In Teaching, Assessment and Learning for Engineering (TALE), IEEE International Conference on, (p. 233-238). IEEE.
- Martins, R., Reis, R., & Marques, A. B. (2016). “Inserção da programação no ensino fundamental: Uma análise do jogo Labirinto Clássico da Code.org através de um modelo de avaliação de jogos educacionais”. In Anais do XXII Workshop de Informática na Escola (Vol. 22, No. 1, p. 121).
- Medeiros Filho, D. A., & Gonçalves, P. C. (2008). “Robótica Educacional de Baixo Custo: Uma Realidade para as Escolas Brasileiras”. In Anais do XIV Workshop de Informática na Escola (Vol. 1, No. 1, p. 51).
- Monteiro, D., Bremgartner, V., Lima, H., & Salgado, N. (2016). “Uma Experiência do Uso Do Hardware Livre Arduino no Ensino De Programação De Computadores”. In Anais do XXII Workshop de Informática na Escola (Vol. 22, No. 1, p. 51).
- Papert, S. (1994). A máquina das crianças: repensando a escola na era da informática. Artes Médicas.
- Pinho, G., Weissshahn, Y., Cavalheiro, S., Reiser, R., Piana, C., Foss, L., Aguiar, M., Du Bois, A. (2016). “Pensamento Computacional no Ensino Fundamental: Relato de Atividade de Introdução a Algoritmos”. In Anais do XXII Workshop de Informática na Escola (Vol. 22, No. 1, p. 261).
- Santos, F. L., Nascimento, F. M. S., & Bezerra, R. M. (2010). “Reduc: A robótica educacional como abordagem de baixo custo para o ensino de computação em cursos técnicos e tecnológicos”. In Anais do XVI Workshop de Informática na Escola (Vol. 1, No. 1, p. 1304-1313).
- Shikama, T. (2014). “Development of Teaching Materials for Computer Programming using a Robot Remotely Controlled by a PC through Wireless Communication”. Informatics Society, (p. 29-36).
- Trentin, M. A., Pérez, C. A. S., & Teixeira, A. C. (2013). “A robótica livre no auxílio da aprendizagem do movimento retilíneo”. In Anais do XIX Workshop de Informática na Escola (Vol. 1, No. 1, p. 51-59).