

Análise de ferramentas para o ensino de Computação na Educação Básica

Eraylson Galdino da Silva¹, Aline F. Barbosa¹, Sebastião R. S. Neto¹, Renato H. O. Lopes¹, Ariane N. Rodrigues²

¹Universidade de Pernambuco, Campus Garanhuns (UPE)
CEP 55.294-902– Garanhuns – PE – Brasil

²Centro de Informática – Universidade Federal de Pernambuco (UFPE)
CEP 50.740-560– Recife – PE – Brasil

{eraylson,
alineferreirabarbosa, sebast.rogers, renato.hill, ariane.upe}@gmail.com

Abstract. *The integration of concepts of logic programming in Basic Education has been emerging since the processes of teaching is current technological changes and to apply computing in Basic Education is a way to motivate students for the IT market. In this sense, this paper presents an analysis of software that assist in teaching and learning these concepts. To this end, five tools (RoboMind, Scratch, SuperLogo, Alice and Kodu Lab) were selected through an empirical analysis of interfaces, usability and fidelity to the concepts of computing, highlighting the strengths and weaknesses from the perspective of licensees in computing.*

Resumo. *A inserção de conceitos de lógica de programação na Educação Básica tem sido emergente, visto que os processos de ensino atuais visam as mudanças tecnológicas e que aplicar a computação na Educação Básica é uma forma de motivar os alunos para o mercado de TI. Neste sentido, o presente trabalho apresenta uma análise de softwares que auxiliam no ensino e aprendizagem destes conceitos. Para tal, foram selecionadas cinco ferramentas (RoboMind, Scratch, SuperLogo, Kodu Lab e Alice), através de uma análise empírica sobre as interfaces, usabilidade e a fidelidade com os conceitos de computação, destacando os pontos positivos e negativos sob a ótica de licenciandos em Computação.*

1. Introdução

Atualmente o mercado brasileiro na área de TI apresenta um vasto déficit de profissionais. De acordo com um levantamento realizado pela Associação Brasileira de Empresas de Tecnologia da Informação e Comunicação (Brasscom) em 2014 cerca de 78 mil vagas, das quais apenas 33 mil serão preenchidas por profissionais formados em cursos superiores. Segundo Nelson Wortsman, diretor de infraestrutura e convergência digital da Brasscom, as causas para a falta de profissionais referem-se desde o baixo interesse dos estudantes brasileiros por ciências exatas até a alta evasão dos cursos ligados a tecnologia (BRASSCOM, 2013).

Um dos fatores que geram os elevados índices de evasão dos cursos da área de computação, refere-se ao fato dos alunos apresentar dificuldades nas disciplinas que abrangem os conteúdos de raciocínio lógico-abstrato, algoritmos e programação (MCGETTRICK et al. 2004). Segundo (RAABE et al. 2005) a disciplina de programação exige o desenvolvimento de estratégias de solução de problemas com exigência lógico-matemático, o que permite gerar para muitos problemas de apreensão dos conceitos abstratos.

Nesta perspectiva, (DALTRO, 2011) afirma que uma das alternativas para aumentar o número de interessados pelos cursos da área de computação nas universidades, está relacionado a inserção introdutória dos conceitos de ciência da computação na educação básica. No entanto, o ensino desses conceitos ainda se restringe àqueles que optam por cursos de graduação na área.

Dessa forma, faz-se necessário introduzir noções sobre Ciência da Computação na Educação Básica como uma forma de despertar o interesse dos alunos quanto a essa ciência e estimulá-los a seguirem carreira nas áreas de TI. O ensino da computação fornece insumos para os alunos no desenvolvimento do raciocínio lógico e na capacidade de lidar e resolver problemas cotidianos, contribuindo na formação de profissionais criativos e preparados para a sociedade tecnológica moderna.

Ao considerar a problemática da evasão de cursos de computação associado às carências na base educacional, o presente artigo se propõe em investigar ferramentas para o ensino de computação no ensino básico. Apresenta-se uma análise sobre os pontos positivos e negativos de softwares construídos para auxiliar o ensino de conceitos computacionais.

O presente artigo está organizado da seguinte maneira: a seção 2 apresenta conceitos e reflexões sobre o ensino da Computação. A seção 3 apresenta as ferramentas e suas perspectivas análises de forma detalhada. E por fim, a seção 4 apresenta algumas conclusões acerca do tema.

2. Ensino da Computação

O ponto de partida no que se refere ao ensino de computação na educação básica é a sua conceitualização. Devido aos termos “Informática” e “Computação” serem constantemente confundidos, faz-se necessário sua diferenciação. Segundo (DALTRO, 2011) a informática (TIC) é o uso de aplicativos como, por exemplo, softwares de escritório (Word, Excel, Power Point) e computação é a ciência do desenvolvimento e implementação de softwares.

A inserção da informática no ambiente escolar, segundo (VALENTE, 1997) se alterna entre atividades tradicionais de ensino e aprendizagem com o uso do computador. Esta prática auxilia o processo de ensino-aprendizagem de algum conteúdo didático.

Segundo a Sociedade Brasileira de Computação (SBC) a computação é tão importante que “deve ser ensinada desde o ensino fundamental, assim como as ciências Física e Matemática”. O ensino de computação se restringe a cursos superiores e técnicos da área, projetos e pequenas ações são realizados por universidades na educação básica. De acordo com (FRANÇA et al. 2013) a computação “pode contribuir, de maneira interdisciplinar, na busca de soluções de problemas diversos, através da disseminação do chamado pensamento computacional”.

Segundo (DALTRO, 2011) o ensino de computação está relacionado com o conceito de pensamento computacional. Adicionalmente o autor defini o pensamento computacional como o processo cognitivo utilizado pelos seres humanos para encontrar algoritmos para resolver problemas. É uma forma de instigar a “capacidade de dedução e de resolução de problemas, exige a utilização do raciocínio lógico e matemático, durante a formação básica dos estudantes” defende (ALENCAR, 2012). Considera-se importante ressaltar que o currículo das escolas brasileiras não contempla a ciência da computação como componente fixo. Na prática, projetos e trabalhos extracurriculares são executados para favorecerem o ensino de computação.

Em países como Estados Unidos, o modelo *Model Curriculum for K–12 Computer Science* (CSTA K-12) é adotado para o ensino de computação. Acredita-se que este currículo pode servir de referência e embasamento para o ensino da ciência da computação na educação básica no Brasil. O K-12 tem como objetivo preparar os alunos para compreender a natureza da ciência da computação e seu lugar no mundo moderno.

Acredita-se que possuir referências como o currículo *K–12* auxilia ao processo de ensino da computação estimulando à disseminação desta ideia. A computação enquanto ciência que estimula a lógica e a matemática deve possuir seu espaço na educação básica como disciplina, já que ainda não existe no Brasil um currículo para o ensino desta ciência.

3. Ferramentas para Ensino da Computação

Este trabalho avalia cinco ferramentas de forma empírica, analisando a sua interface, a usabilidade e a fidelidade com os conceitos de computação.

3.1 RoboMind

RoboMind (RoboMind, 2014) é um ambiente de desenvolvimento criado por Arvid Halma (Universidade de Amsterdam), com o intuito de facilitar no ensino de conceitos iniciais de programação de computadores. Através deste ambiente é possível realizar um ensino de conceitos de lógica para programação e ter uma noção inicial das áreas de Inteligência Artificial e Robótica, através de um robô virtual, que precisa ser programado para realizar algumas tarefas dentro do mapa. O RoboMind utiliza uma linguagem de programação chamada Robo/Roo, através dela é possível programar um robô no ambiente para realizar movimentos dentro de um mapa bidimensional.

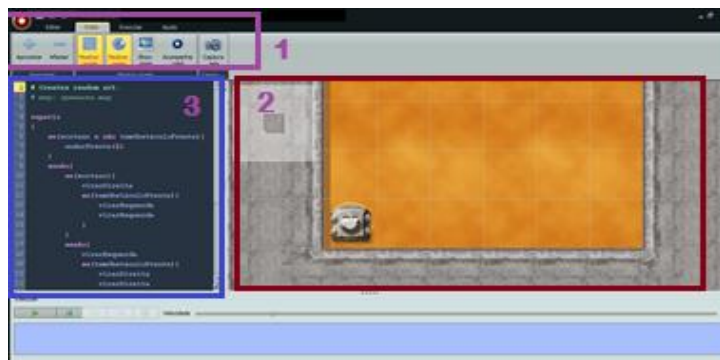


Figura 1. Interface ferramenta RoboMind

O destaque 1 na figura 1 compõe o menu da ferramenta. O destaque 3 é o painel de script, nessa área o aluno pode escrever o código do programa informando o que o robô deve fazer. O destaque 2, é o painel mundo, onde se visualiza o mundo ou mapa com o robô. Na parte inferior ao painel de execução e o painel de mensagens, onde é possível respectivamente, mandar o robô executar os comandos e ver as mensagens de erros ou de avisos geradas pelo robô.

A linguagem Robo/Roo apresenta um conjunto de instruções para movimentação do robô, como apresenta a Tabela 1:

Tabela 1. Instruções para movimentação do robô

Instrução	Descrição
<i>andarFrente(n)</i>	faz o robô se movimentar para frente
<i>andarTrás(n)</i>	faz o robô se movimentar para trás
<i>andarNorte(n)</i>	faz o robô se movimentar para o norte
<i>andarSul(n)</i>	faz o robô se movimentar para o sul
<i>andarLeste(n)</i>	faz o robô se movimentar para o leste
<i>andarOeste(n)</i>	faz o robô se movimentar para o oeste
<i>virarEsquerda()</i>	move o robô 90° para a esquerda
<i>virarDireita()</i>	move o robô 90° para a direita

A linguagem ainda contempla com comandos para estrutura de seleção: (*se-senão*), e repetição contendo três versões: (*repetir(n)*), repete a instrução *n* vezes; (*repetir()*), repete continuamente a instrução; e (*repetirEnquanto(condição)*), repete enquanto a condição for satisfeita. Através desses principais comandos da linguagem, torna-se possível adotar o RoboMind para o ensino de conceitos de lógica para programação associado aos conceitos de expressões lógicas, estruturas de decisão e estruturas de repetição.

O trabalho de (VAHLICK et al. 2009) apresenta um relato de experiência do ensino de algoritmos e programação de computadores utilizando o RoboMind como ferramenta para facilitar o ensino e aprendizagem e defende que com o RoboMind os alunos têm contato de uma maneira mais simples com o estudo de algoritmos, que podem experimentá-los executando por inteiro, ou passo-a-passo, e assim criando uma compreensão melhor das estruturas de programação. A respeito da linguagem (VAHLICK et al., 2009) defende que é mais simples e direta que programar em Java, melhorando no entendimento dos conceitos de programação.

Em relação as limitações, identificou-se que, não é possível abordar todos os conteúdos sobre a lógica de programação, por mais que a linguagem consiga trabalhar conceitos importantes como estruturas de decisão e repetição. Ressalta-se que ainda não se tem no Robomind explicitamente os conceitos de variáveis e constantes, que são importantes para um melhor entendimento da lógica de Computação.

O feedback que a ferramenta possibilita e a facilidade da linguagem, são pontos positivos da ferramenta, além de ser possível transformar o algoritmo feito na ferramenta para a linguagem Java e executá-lo em um robô da Lego, como apresentado do trabalho de (VAHLICK et al., 2009).

3.2 Scratch

O Scratch é um ambiente de desenvolvimento visual criado por Lifelong Kindergarten Group (LLK), grupo de pesquisa do MIT Media Lab. O ambiente tem entre seus objetivos o propósito de introduzir a programação de maneira fácil e rápida para aqueles que não possuem nenhum tipo de experiência no assunto (MALONEY et al. 2010). O Scratch utiliza uma linguagem visual, onde é possível criar jogos, animações e histórias interativas.

A Figura 2 ilustra o ambiente do Scratch, onde é dividida por quatro partes. No destaque 1, está o menu, com opções para editar o arquivo, compartilhar e obter ajuda. No destaque 2 tem os comandos visuais que são utilizados para determinar o que o objeto deve fazer. O destaque 3, é composta por abas, na primeira é inserido pelos alunos os comandos, na segunda para fazer alterações nos trajés e na terceira para definir sons. O destaque 4 é usada para visualizar os resultados dos comandos e para criar ou editar os sprites que são os objetos ou a imagem de background chamada de palco.

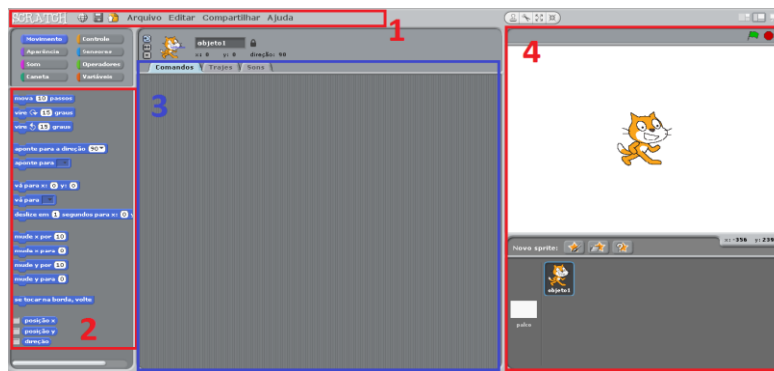


Figura 2. Tela inicial ambiente Scratch

A linguagem do Scratch é bastante interativa, os blocos de comando são arrastados e soltos na coluna do meio em seguida encaixadas como peças de quebra-cabeça. Através dos comandos da linguagem é possível ensinar/aprender os seguintes conceitos de programação: entrada e saída de dados, variáveis, operadores aritméticos e relacionais, estruturas de repetição e estruturas condicionais. Ainda, segundo (FRANÇA et al, 2013) através do Scratch é possível ensinar/aprender conceitos de computação como sequência, eventos, paralelismo, loop e condicionais.

O trabalho de (AURELIANO et al, 2012) tem como objetivo avaliar o uso do Scratch como uma abordagem alternativa para o processo de ensino e aprendizagem de programação. Segundo relatos dos autores, o Scratch teve uma influência positiva como ambiente de programação para um curso introdutório em programação e destaca a interface gráfica da ferramenta por permitir que o programa seja visualizado enquanto é executado.

A principal limitação que o Scratch apresenta é a falta de um feedback passo a passo sobre o que o aluno produz na ferramenta. Esse feedback poderá ajudar o aluno a entender qual parte do algoritmo está sendo executado e qual apresenta um erro, por exemplo. A diversidade de objetos lúdicos: jogos, animações, e histórias interativas, que se pode construir na ferramenta, e a linguagem em blocos são os pontos positivos que se destacam na ferramenta.

3.3 Linguagem Logo

A LOGO foi desenvolvida pelo instituto de Tecnologia de Massachusetts (MIT) em 1967, baseada na ideia de Piaget e chefiada por Seymour Papert (MOTTA et al, 2010). Fundamenta-se no LISP, uma linguagem de programação interpretativa, utilizada para o desenvolvimento de inteligência e, também, como uma calculadora que, ao contrário de calcular expressões aritméticas, avalia expressões simbólicas.

Por meio da LOGO, em cada uma dessas áreas, é possível trabalhar conceitos como a construção numérica, a construção da linguagem escrita (letras, palavras), essencial para o processo de letramento; implementação de jogos desenvolvidos para as áreas já mencionadas. Além de permitir a construção de desenhos, pinturas, como uma espécie de caderno eletrônico. A LOGO disponibiliza uma série de comandos e ferramentas, que tem por objetivo trabalhar a lógica de forma que a criança é a principal atuante na construção reflexiva do próprio raciocínio.

Na Figura 3 é apresentado o ambiente da linguagem LOGO. O destaque 1 compõe a barra de menus. O destaque 2 representa a tartaruga. No destaque 3 compõe a Janela de Comandos onde o usuário pode escrever o seu código. O destaque 4 compõe algumas funções como, executar, pausar entre outras que poderão auxiliar o usuário na escrita do código.

A linguagem Logo ainda conta com alguns comandos básicos para movimentação conforme apresenta a tabela 2.:

Tabela 2. Comandos para movimentação da Tartaruga

(pf) para frente	(pt) para trás	(pe) para esquerda	(pd) para direita
------------------	----------------	--------------------	-------------------

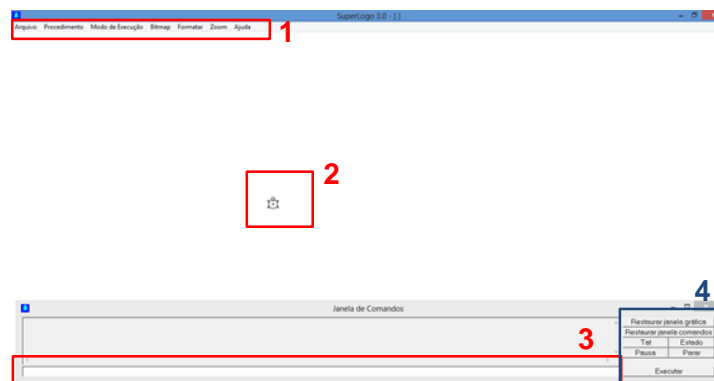


Figura 3. Tela inicial SuperLogo

O artigo (SILVA, R. M. 2008) expõe a utilização da linguagem de programação Logo na educação de crianças, bem como as vantagens deste uso para formação do seu intelecto. Torna-se evidente que a linguagem de programação Logo é ótima para o desenvolvimento das habilidades intelectuais da criança, que ao interagir com o computador através da “tartaruga” constrói as estruturas mentais de forma privilegiada, fazendo com que o raciocínio lógico se desenvolva, favorecendo a aprendizagem das matérias curriculares principalmente da matemática.

Dentre as limitações do ambiente da linguagem Logo destaca-se a maneira em que é apresentado seus conceitos de lógica aos usuários, há pouca interação e o ambiente deixa a desejar no aspecto lúdico porque a interface é pouco atraente e interativa. Geralmente, os alunos preferem ambientes que possibilitem uma interação lúdica pela apresentação de certos conceitos. Como pontos positivos destacam-se a flexibilidade, rigidez e a simplicidade encontrada quando a mesma é utilizada para auxílio de alguma atividade, além da linguagem apresentar uma enorme quantidade de comandos, possibilitando o desenvolvimento de vários objetos.

3.4 Kodu Game Labs

O Kodu Game Lab foi desenvolvido pelo laboratório de pesquisas FUSE (Future Social Experiences) Labs, mantido pela Microsoft. Segundo (SOUZA et al. 2013) este software é utilizado para criar jogos, em num ambiente tridimensional e multimídia cujo principal objetivo é estimular em sala de aula a aprendizagem lúdica da programação de computadores moldada nos paradigmas da orientação à objetos.

Na figura 4, de acordo (SOUZA et al. 2013) o aluno por intermédio do professor é instruído a escolher um Mundo, o usuário conta com uma grande quantidade de opções divididas que são: Downloads, Amostras, Lições, e por fim Meus Mundos onde o mesmo tem acesso ao seu mundo construído, como apresenta o destaque 1.



Figura 4. Tela opções do Kodu

Após a apresentação do ambiente Kodu o aluno tem acesso aos personagens como pode ser observado na figura 5. O aluno poderá programá-los e o ambiente oferece algumas ações básicas como (quando... faça), ou seja, o personagem poderá realizar uma determinada ação após a ocorrência do evento. Como exemplo, pode-se citar: “Quando: a tecla direcional (para esquerda) do teclado for pressionada > Faça: ande para a esquerda” (SOUZA et al. 2013).



Figura 5. Tela de blocos de programação Kodu

No artigo (SOUZA et al. 2013) os autores apresentam o potencial pedagógico do Kodu, descrevendo que os alunos podem aprender conceitos de raciocínio lógico, jogando

com objetos e vários elementos em um ambiente tridimensional e multimídia, como alternativa ao método tradicional instrucionista.

Dentre as limitações do Kodu podem ser observadas a quantidade de personagens disponíveis no ambiente, só à disposição (robôs, discos voadores, naves e outros “figurantes” semelhantes), como também na quantidade de objetos que podem ser inseridos dentro do mundo construído. Em relação à programação, apesar de existir uma opção mediante a qual o usuário pode inserir os personagens já pré-programados para executarem determinadas tarefas, há opções em que o mesmo pode programar seu personagem. Seu principal ponto positivo é a maneira em que é apresentada os conceitos de lógica de maneira lúdica e interativa.

3.5 Alice

Alice é um ambiente de programação tridimensional desenvolvido na Universidade de Virginia e Universidade de Carnegie Mellon, por um grupo de pesquisa liderado por Randy Pausch. Tem como propósito apoiar estudantes de programação através do desenvolvimento de animações jogos em ambientes 3D. (Hebert, 2011).

O programa dispõe de um ambiente virtual denominado World (Mundo) onde as interações acontecem. A partir de elementos pré-programados que acompanham o ambiente é possível criar estórias seguindo uma construção lógica. A Figura 6 apresenta a tela inicial do Alice.

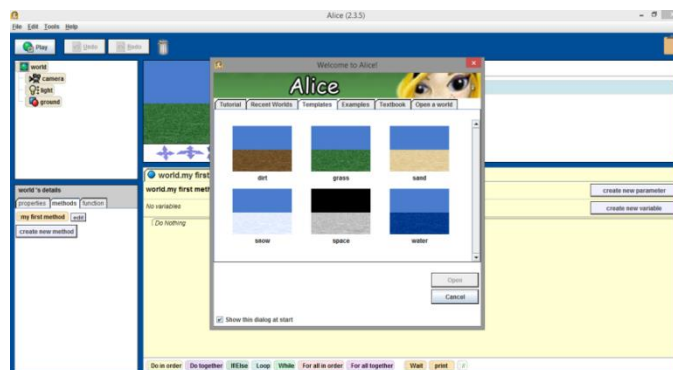


Figura 6. Tela inicial Alice

Na interface, estudantes podem arrastar gráficos, onde existem instruções correspondentes a comandos padrões na produção de programação orientada a objetos, tal como ocorre com Java, C++ ou Builder (BARROS et al, 2012).

Dentre as limitações do Alice se destacam segundo (VALASKI et al, 2012) o seu aspecto lúdico, pouca dinamicidade, diversos bugs, travamentos, problemas de salvamento, inserção de sons. Entre seus pontos positivos, permite a aprendizagem de programação de maneira divertida, ajuda na materialização de conceitos abstratos, introduz conceitos de orientação a objetos como (objeto, classe e métodos) e por fim contribui no desenvolvimento do raciocínio lógico.

4. Conclusões

O processo de ensino e aprendizagem de conceitos da computação, como a própria lógica para programação traz várias dificuldades, desde a abstração de problemas até a resolução algorítmica dos mesmos. No entanto, as ferramentas analisadas nesse trabalho, apresentam grande potencial para facilitar este processo, de forma lúdica e dinâmica, como demonstrado nos relatos do uso das ferramentas pelos artigos.

A utilização dessas ferramentas no âmbito escolar, propõem mais que o ensino dos conceitos de computação, através delas os alunos podem ser desafiados, os tornando mais criativos, com mais facilidade na resolução de problemas habituais, construindo assim o pensamento computacional, como defende (FRANÇA et.al 2013).

Espera-se que, através das limitações identificadas na análise dos softwares, possa ser consideradas em trabalhos futuros no sentido de desenvolver de uma nova ferramenta ou melhorias nas próprias ferramentas existentes apresentadas nesse trabalho. E que através dos pontos positivos, os professores possam escolher qual ou quais ferramentas utilizar em suas aulas.

Não faz parte do escopo desse trabalho, apresentar todas as funcionalidades das ferramentas, ou comparar e determinar qual a melhor ferramenta, deixando assim uma lacuna para possíveis trabalhos futuros.

Contudo, acreditamos, que o ensino da computação na educação básica, tem um grande potencial, e que pode colaborar na inserção de conhecimentos básicos de computação. Através do uso dessas ferramentas é possível ensinar e aprender os conceitos de lógica para programação de forma mais lúdica, e, através da aprendizagem desses conceitos os alunos possam desenvolver o seu pensamento computacional. Favorecendo a preparação dos alunos para a sociedade da informação em que vivem atualmente. Considera-se importante também o fator de interesse que pode ser despertado pelas alunos ao considerar cursos de computação, desde cedo mesmo na Educação Básica.

Referências

- ALENCAR, Yugo Manguiera de ; SCAICO, Pasqueline Dantas; DA SILVA, Jarbele Cássia. Jogando com Números Binários: uma Possibilidade para Estimular o Raciocínio Lógico e o uso da Matemática. Conferencias LACLO, v. 3, n. 1, 2012.
- ALICE. (2014). Disponível em: <http://www.alice.org/index.php>. Acesso em 20/03/2014.
- BRASSCOM. O Mercado de Profissionais de TI no Brasil. Disponível em <http://www.brasscom.org.br> Acesso em 2014.
- FRANÇA, Rozelma Soares; SILVA, Waldir Cosmo; AMARAL, Haroldo José Costa. Despertando o interesse pela ciência da computação: Práticas na educação básica. In: Proceedings of International Conference on Engineering and Computer Education. 2013. p. 282-286.
- FRANÇA, S. R.; SILVA, WC da; AMARAL, HJC do. Ensino de Ciência da Computação na Educação Básica: Experiências, Desafios e Possibilidades. In: XX Workshop de Educação em Computação (WEI'2012). Curitiba, PR, Brasil. 2012. <http://www.jornaldaciencia.org.br/Detail.jsp?id=69969> Acesso em: 08 de Abril 2011.

- KODU. (2014). Disponível em: <http://www.kodugamelab.com>. Acesso em 20/03/2014.
- MCGETTRICK, AD, Boyle, R, Ibbett, R, Lloyd, J, Lovegrove, G & Mander, K 2005, 'Grand challenges in computing: Education - a summary' *Computer Journal*, vol 48, no. 1, pp. 42-48.,10.1093
- MOTTA, Marcelo Souza; SILVEIRA, Ismar Frango. Contribuições do SuperLogo ao ensino de geometria. In: *Informática na Educação: teoria e prática*. 2010
- NUNES, Daltro José. (2010). Computação ou informática? Disponível em <<http://www.jornaldaciencia.org.br/Detalhe.jsp?id=69969>> Acesso em: 29 de mar. 2014.
- NUNES, Daltro José. (2011). Ciência da Computação na Educação Básica. Disponível em <<http://www.jornaldaciencia.org.br/Detalhe.jsp?id=79207>> Acesso em: 29 de mar. 2014.
- RAABE, A. L. A., Silva, J. M. C., (2005) “Um Ambiente para Atendimento as Dificuldades de Aprendizagem de Algoritmos”. In: XXV Congresso da Sociedade Brasileira de Computação. São Leopoldo/RS.
- ROBOMIND. (2014). Disponível em: <http://www.robomind.net/pt/>. Acesso em 20/03/2014.
- SBC, Sociedade Brasileira de Computação (SBC). Disponível em:http://www.sbc.org.br/index.php?option=com_content&view=category&layout=blog&id=215&Itemid=82. Acesso em: 23 Mar. 2014
- SCRATCH. (2014). Disponível em: <http://scratch.mit.edu>. Acesso em 20/03/2014.
- SILVA, Ronaldo Machado. O uso da linguagem Logo na educação infantil. Disponível em <<http://www.sucesumt.org.br/mtdigital/anais/files/OUsodaLinguagemdeProgramacaoLogo.pdf>> Acesso em: 23 Mar. 2014.
- SOUZA, Paulo Roberto de Azevedo; DIAS, Lucimeri Ricas. Kodu Game Labs: Apresentação e reflexão sobre os jogos criados e publicados na comunidade Kodu BR. In: *Simpósio Brasileiro de Jogos e Entreterimento Digital*. 2013.
- SOUZA, Paulo Roberto de Azevedo; DIAS, Lucimeri Ricas. Kodu Game Labs: Estimulando o Raciocínio Lógico através de Jogos. In: *Simpósio Brasileiro de Informática na Educação*. 2012.
- SUPERLOGO. (2014). Disponível em: <http://projetologo.webs.com/slogo.html>. Acesso em
- TUKER, A. (2006) A Model Curriculum for K–12 Computer Science. Final Report of the ACM K–12 Task Force Curriculum Committee.
- VALASKI, J., PARAISO, E. C., (2012) “Limitações da Utilização do Alice no Ensino de Programação para Alunos de Graduação”. In: *Simpósio Brasileiro de Informática na Educação*.
- VALENTE, José Armando. (1997). *Informática na Educação: instrucionismo x construcionismo*.