

Uma arquitetura pedagógica auxiliada por tecnologias para ensino e aprendizagem de programação

Orivaldo de Lira Tavares¹, Crediné Silva de Menezes^{1,3}, Rosane Aragón²,
Lucinéia Barbosa da Costa¹

¹Programa de Pós-graduação em Informática – Universidade Federal do Espírito Santo (UFES)

²Programa de Pós-graduação em Educação – Universidade Federal do Rio Grande do Sul (UFRGS)

³Faculdade de Educação – Universidade Federal do Rio Grande do Sul (UFRGS)

tavares@inf.ufes.br, credine@gmail.com, rosane.aragon@ufrgs.br, cliklucineia@gmail.com

Abstract. *Each year several technologies are developed as an aid to the process of teaching and learning programming. However, promote learning is still a big challenge. There is a shortfall in the use of pedagogical architectures that are actually effective to aid learning. To meet this demand, we present a pedagogical architecture aided by technologies, aiming to achieve real progress in learning. We also present results of applying this pedagogical architecture in a real classroom with students of undergraduate courses in Computer Science and Computer Engineering from a university.*

Resumo. *A cada ano diversas tecnologias são desenvolvidas como auxílio ao processo de ensino-aprendizagem de programação. No entanto, promover de fato a aprendizagem ainda é um grande desafio. Há uma carência no uso de arquiteturas pedagógicas que sejam de fato eficazes para auxiliar a aprendizagem. Para atender a essa demanda, apresentamos uma arquitetura pedagógica auxiliada por tecnologias, objetivando alcançar progressos reais de aprendizagem. Apresentamos ainda resultados da aplicação dessa arquitetura pedagógica em uma turma de alunos dos cursos de graduação em Ciência da Computação e Engenharia de Computação.*

1. Introdução

A aprendizagem de programação de computadores é considerada um desafio, isso porque requer do aluno uma combinação de habilidades, não relacionadas à sua realidade. As deficiências nessas habilidades somadas a hábitos de estudos pouco disciplinados e centrados na memorização, além de conhecimentos prévios desestruturados, ampliam ainda mais as dificuldades de aprendizagem.

Essas dificuldades podem ser atenuadas com o uso de abordagens pedagógicas apoiadas por tecnologias que possibilitem acompanhar o percurso do estudante pelas etapas do processo de ensino-aprendizagem.

O professor como mediador do processo de ensino-aprendizagem precisa possuir a percepção dos pontos de dificuldades dos alunos e neles intervir formativamente. São necessárias mudanças em suas estratégias e aumento da eficiência de sua mediação com o uso de tecnologias, para que possa atingir o objetivo maior, que apoiar o aluno na construção de conhecimento [COSTA, 2011].

Caminhando nessa direção, apresentamos uma arquitetura pedagógica apoiada por tecnologias para o ensino-aprendizagem de programação. Com a aplicação dessa

arquitetura, o professor acompanha o desenvolvimento das habilidades de programação dos estudantes.

Entendemos, portanto, que a aprendizagem de programação observada pela ótica cognitiva não é um dom, mas sim um conjunto de conhecimentos que qualquer um pode conseguir construir, desde que use estratégias que promovam o desenvolvimento das habilidades envolvidas no processo de programar [Anderson, 2000].

Este trabalho está organizado conforme segue: na seção 2, apresentamos uma revisão da literatura; na seção 3, mostramos o uso inovador das tecnologias digitais; na seção 4, apresentamos a arquitetura pedagógica usada; na seção 5, mostramos os experimentos e resultados obtidos e, por fim, na seção 6, concluimos com as considerações finais seguidas das referências bibliográficas.

2. Referencial teórico

A cada ano o número de turmas de introdução à programação é maior e muitos desses alunos não desenvolveram as habilidades necessárias ao aprendizado de programação [Tavares, 2012]. Essas habilidades envolvem: abstração, saber resolver problemas, validar soluções, buscar várias soluções para cada problema, analisar criticamente soluções de outros autores etc.

Em geral, as ferramentas de apoio ao processo ensino-aprendizagem de programação existentes, se propõem a propiciar uma aprendizagem fácil e intuitiva através de sofisticadas tecnologias. Entretanto, não existe uma unanimidade com relação ao uso eficaz dessas ferramentas para atingir os objetivos pretendidos. Dessa forma, destacamos que as tecnologias não bastam, faz-se necessário a adoção de arquiteturas pedagógicas que de fato contribuam como o processo de construção de conhecimento dos alunos.

Autores que se interessam pelo aspecto metodológico da aprendizagem de programação enfatizam duas abordagens: o uso de heurísticas, como Wirth (1971) e Polya (2004) e o desenvolvimento da habilidade de resolver problemas, como Falker e Palmer (2009). No entanto, avaliar e diagnosticar os problemas de aprendizagem e dar *feedback* são tarefas difíceis para os professores, a começar pelo esforço de corrigir exercícios manualmente das muitas atividades de turmas com muitos alunos.

Gomes e Mendes (2000) afirmam que os principais erros apresentados pelos aprendizes durante o treinamento de programação são: erros de sintaxe e semântica, dificuldades na compreensão do enunciado dos problemas e na concepção de algoritmos, e a incapacidade de detectar erros de lógica de programação.

Ribeiro (2012) relata que entre as dificuldades apresentadas pelos estudantes estão aquelas relacionadas à capacidade de abstração e as dificuldades impostas pela sintaxe e estruturas abstratas da linguagem de programação. Para Souza (2012) a dificuldade dos alunos está em compreender os conceitos de programação e/ou possuir visões erradas sobre a atividade de programar.

Todos esses fatores levam o aluno que tem pouco ou nenhum conhecimento prévio sobre o assunto a abandonar o curso, pois ele sente a necessidade de um acompanhamento personalizado que o professor nem sempre pode dar e que os métodos de ensino tradicionais nem sempre preveem.

3. Uso inovador das Tecnologias Digitais

As tecnologias da informação e comunicação têm sido usadas como suporte para potencializar a aprendizagem, facilitando o acesso às fontes de informação assim como a interação entre os participantes do processo de aprendizagem.

Mais recentemente, tem-se buscado novas ideias sobre como conceber e viabilizar um novo modelo de uso das tecnologias na educação que considerem com mais ênfase as teorias da aprendizagem.

Neste sentido, Carvalho et al (2007) propõem o conceito de Arquiteturas Pedagógicas, sustentadas em três componentes: 1) Concepção pedagógica forte, que considere a “Pedagogia da Incerteza”, que sintetiza as ideias de Paulo Freire e Jean Piaget sobre a importância de criar condições para o aluno construa suas próprias certezas; 2) Sistematização metodológica, onde está o fundamento das APs, que busca proporcionar aos estudantes atividades intelectuais interessantes, as quais se desenvolvem por meio da interação, e 3) Suporte telemático, que auxilia no processo.

Nesta nova concepção do uso de TICs na Educação, a ênfase recai sobre a abordagem pedagógica, que deve ser tomada como ponto de partida. Quanto aos recursos digitais, devemos colocá-los a serviço da proposta, constituindo-se portanto em um instrumentos para sua implementação.

4. Arquitetura Pedagógica Utilizada

Usamos como referência a teoria de *Piaget* (1972). Nela, o indivíduo passa por um processo de desenvolvimento cognitivo que se inicia com a manipulação de objetos concretos, passando pelo uso de operações mentais, até desenvolver a habilidade de formalização, permitindo abstrair detalhes sem importância, generalizar características de objetos, dentre outros processos cognitivos.

Real (2011) afirma que as arquiteturas pedagógicas em sua proposta rompem com a pedagogia tradicional. Isso porque o uso de recursos digitais fomenta uma aprendizagem interativa, na qual o discente torna-se o sujeito de sua aprendizagem.

O professor tem um papel fundamental nesse processo, pois deve propiciar espaços e orientações por meio de arquiteturas pedagógicas que apoiem a construção da autonomia e do conhecimento dos aprendizes.

Realizamos estudos do tipo quantitativo descritivo, com a aplicação de questionários, análise de relatórios dos alunos sobre a construção de programas e acompanhamento da turma na resolução dos problemas. A abordagem pedagógica teve como base:

- resolução de problemas de forma individual;
- resolução de problemas cooperativamente;
- busca de informações esclarecedoras de dúvidas;
- socialização das descobertas;
- revisão do trabalho dos pares e

- aprendizagem com a observação e a comparação de várias soluções.

No período do desenvolvimento da pesquisa, foi sugerido à turma o uso do Método de Construção de Programas, que neste artigo será referenciado como *MCP*. O *MCP* é um método criado pelos autores desta pesquisa. É estruturado em seis etapas, nas quais os estudantes passam por cada uma delas para a construção de algoritmos.

Após a conclusão das etapas, os estudantes são incentivados à publicação do relatório da construção do programa em um *wiki*. Isso permite que cada estudante possa aprender com a solução publicada e contribuir com sugestões e críticas para a formação de soluções ainda melhores. Essa é uma estratégia que potencializa a aprendizagem. A Tabela 01 apresenta as etapas do *MCP* - Método de Construção de Programas.

Tabela 01. Etapas do MCP

Etapas	Objetivo
Compreensão do problema	Compreender as principais dificuldades apresentadas pelo aluno em relação à compreensão do problema proposto
Planejamento do teste	Definir mapeamentos entre dados de entrada relevantes e resultados esperados
Especificação da solução	Especificar uma solução a partir da composição dos métodos e/ou funções de mapeamento usados no plano de teste.
Codificação do programa	Codificar a solução em uma linguagem de programação
Teste da solução	Testar a correção da solução codificada
Avaliação do processo e seus resultados	Analisar cada etapa da construção da solução, consolidando os pontos fortes e corrigindo os pontos fracos.

Etapa 1 - Compreensão do problema: nessa etapa o estudante relata a sua compreensão do problema proposto, levando em consideração a relação entre os dados de entrada e saída, o domínio de entrada e a resposta esperada;

Etapa 2 - Planejamento do teste: nessa etapa o estudante define os mapeamentos entre os dados de entrada, representativos dos subdomínios relevantes do problema, e os resultados esperados;

Etapa 3 - Especificação da solução: nessa etapa o estudante identifica os métodos e/ou funções que compõem a solução do problema e constrói uma ou mais opções de solução. Aqui ele explicita as funções usadas para mapear os dados de entrada nos resultados esperados, usados implicitamente na etapa dois, e compõe essas funções para construir uma solução;

Etapa 4 - Codificação do programa: nessa etapa, é codificada a solução em uma linguagem de programação. Para esta pesquisa, optamos por codificar as soluções em *Haskell*, que é uma linguagem de programação funcional e de fácil compreensão para os graduandos de computação, por usar a linguagem matemática conhecida por eles;

Etapa 5 - Teste da solução: nessa etapa, tendo como base o plano de teste, é feito o teste da correção da solução codificada. O teste da solução consiste em usar os dados de

entrada, do plano de teste, para avaliar o funcionamento da solução codificada, comparando os resultados esperados do plano de teste com os resultados obtidos na execução do código da solução;

Etapa 6 - Avaliação do processo e seus resultados: nessa etapa, o estudante analisa cada etapa da construção da solução, consolidando os pontos fortes e corrigindo os pontos fracos. Nessa análise é sugerida uma quantificação de vários critérios, tais como, legibilidade, manutenibilidade, reusabilidade, eficiência e recursos necessários.

4.1. Recursos digitais

O educador, autor de uma arquitetura pedagógica, usa recursos tecnológicos que possam viabilizar e/ou tornar mais eficiente a implementação da abordagem pedagógica escolhida por ele, de modo a potencializar a aprendizagem do estudante, com o objetivo de alcançar a excelência na forma de mediar a aprendizagem e fazer com que o conhecimento do estudante seja construído de modo interativo e agradável. Seguem os recursos tecnológicos usados na arquitetura pedagógica usada nesta pesquisa:

Wiki – ferramenta de comunicação assíncrona, muito usada em cursos a distância para a elaboração de hipertextos. Nesta pesquisa fizemos uso dessa ferramenta para suporte as aulas de programação de computadores. Com ela os estudantes publicavam, visualizavam e modificavam suas soluções e podiam visualizar e comentar as soluções dos seus pares, criando assim um grande hipertexto sobre o percurso de aprendizagem de toda a turma;

Fórum – ferramenta de comunicação assíncrona, muito usada em cursos a distância para a criação de discussões e debates. Nesta pesquisa fizemos uso dessa ferramenta para criação de debates sobre assuntos relacionados à disciplina de programação de computadores. Para isso, o professor criava cada fórum, inseria mensagens, fazia modificações, excluía e visualiza as mensagens, enquanto o aluno visualizava, inseria e comentava mensagens;

Sala de interação (sala de chat) – ferramenta de comunicação síncrona, usada para discussões com a participação simultânea dos interessados. Com ela o professor criava salas para interagir com os alunos, trocava mensagens sobre o conteúdo abordado em sala de aula ou nas listas de exercícios e orientava o estudante sobre como resolver dúvidas;

Resolução de problemas - o termo “resolução de problemas” refere-se à solução de problemas propostos pelo professor. Com essa ferramenta, criada especialmente para a arquitetura pedagógica desta pesquisa, o professor tinha autonomia em criar, modificar, excluir e visualizar uma lista de exercícios. O aluno podia selecionar um exercício e construir o programa para resolver o exercício, enquanto percorria as etapas do método MCP. Nesse percurso de construção do programa, o estudante usava um editor especialmente desenvolvido para apoiar a documentação da construção do programa, desde a compreensão do problema até a análise da solução desenvolvida;

Questionário – ferramenta usada pelo professor para exercitar os alunos. Nos questionários o professor podia criar, modificar, excluir e visualizar as questões dos questionários, enquanto os estudantes podiam responder aos questionários criados;

Tarefa – ferramenta de comunicação assíncrona, com ela o professor criava, modificava e visualizava uma tarefa (atividade proposta pelo professor), enquanto o estudante podia visualizar cada tarefa e submeter um arquivo, com a solução da tarefa especificada.

5. Experimentos e resultados

Os experimentos de avaliação da aprendizagem foram realizados em uma turma real de 12 alunos dos cursos de Ciência da Computação e Engenharia da Computação de uma IES, durante o primeiro semestre de 2012.

A avaliação de aprendizagem desses alunos levou em conta as soluções desenvolvidas para 18 problemas. Para dar suporte à realização das atividades, o professor disponibilizou vídeo-aulas e apostilas digitalizadas, como apoio ao processo de aprendizagem.

Ao todo foram aplicadas sete listas de exercícios. Cada uma possuía uma quantidade específica de tarefas, como mostra a tabela 02. Todas as listas possuíam os mesmos objetivos, o que as diferenciava era o nível de dificuldade de cada uma, como segue:

Tabela 02- Exercícios

Lista de exercícios	Qtd de exercícios	Objetivos
Lista1	9	✓ Analisar o envolvimento dos alunos nas atividades;
Lista 2	2	
Lista 3	2	✓ Identificar as habilidades do aluno para resolução de problemas;
Lista 4	2	
Lista 5	1	✓ Analisar as várias formas de resolução de problemas desenvolvidas pelos alunos;
Lista 6	1	
Lista 7	1	✓ Verificar as principais dificuldades detectadas pelos alunos.

Lista 1 – nessa lista foram apresentados nove tipos de problemas simples. Os alunos deviam buscar a solução de cada problema e, após isso, escrever um relatório detalhado de como desenvolveram o processo de resolução.

Lista 2 – nessa lista foram apresentados dois tipos de problemas. Em um deles, o aluno devia desenvolver funções para calcular o salário líquido de um funcionário, levando em consideração o salário base, as faltas, as horas-extras simples e as horas-extras especiais. No outro, o aluno deveria determinar a área de cada uma das subfiguras de um Tangram.

Lista 3 – nessa lista de exercícios foram apresentados dois problemas. Em um deles, o aluno devia descrever todas as pedras de um dominó numérico em *Haskell* com o uso de *list comprehension* (um dos recursos da linguagem *Haskell* para descrição de

listas). Já no segundo, o aluno era desafiado a ser proativo e representar outros elementos/cenários do domínio que poderiam ser descritos com *list comprehension*.

Lista 4 – nessa lista foram apresentados dois problemas. No primeiro, o aluno devia adicionar no seu *wiki* descrições em *Haskell* para os números primos, incluindo os testes no *Hugs* (interpretador *Haskell*) para essas descrições. No segundo, o aluno devia descrever os números primos usando *list comprehension* ou outros recursos da linguagem *Haskell*.

Lista 5 – essa lista apresentava apenas um problema e o aluno devia construir o máximo de soluções possíveis. Essas soluções codificadas na linguagem funcional *Haskell*, recebiam uma configuração do dominó na mesa e da mão do jogador e deviam produzir a lista de peças da mão do jogador que pudessem ser usadas na próxima jogada do dominó em andamento.

Lista 6 – nessa lista foi aplicado um problema em que o aluno devia construir uma função em *Haskell* que identificasse as pedras da mão do jogador que pudessem ser jogadas na mesa, para resultarem em pontuação adicional. (havia um critério do jogo para pontuação adicional de certas configurações da mesa).

Lista 7 – nessa lista foi aplicado um jogo contendo dois oponentes, no qual um deles devia inventar uma senha e o outro devia adivinhá-la. Os estudantes deviam jogar, fazendo anotações em cada tentativa de acerto da senha e da resposta do jogador adversário, criador da senha.

5.1. Principais dificuldades percebidas

Durante o período em que foi realizada a pesquisa, fizemos um levantamento das principais dificuldades encontradas pelos alunos na realização das atividades. O gráfico 01 mostra quais foram estas dificuldades e a porcentagem de cada uma delas.

No gráfico 01, apresentamos os cinco tipos de dificuldades encontradas pelos alunos durante a realização desta pesquisa. Percebeu-se que a concentração maior dessas dificuldades está na compreensão do problema (25%), etapa fundamental para desenvolvimento das soluções dos exercícios.

Outro ponto importante mostrado no gráfico 01 é que apenas 14% dos alunos encontraram dificuldades para resolver os problemas individualmente e esta dificuldade caiu para 12% quando as soluções foram desenvolvidas colaborativamente.

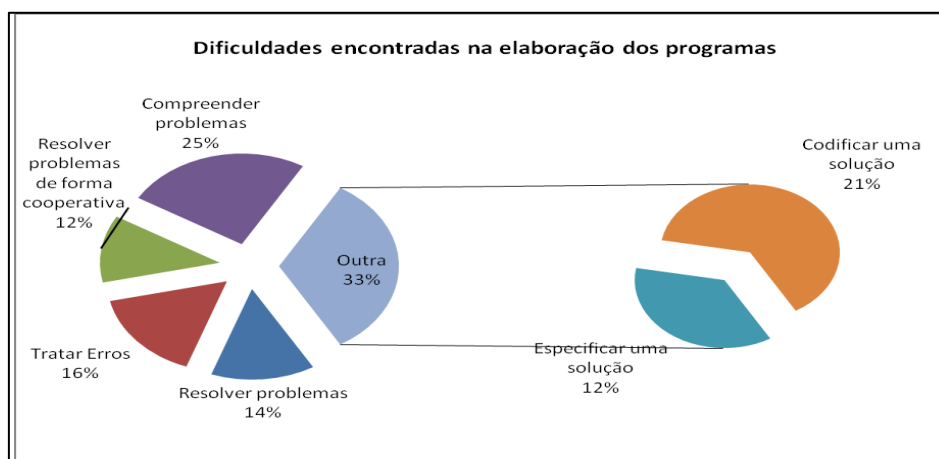


Gráfico 01: Dificuldades encontradas na elaboração dos programas

Notamos também que 21% dos estudantes encontraram dificuldades na codificação do problema e que apenas 16% deles encontraram dificuldades para testar de erros.

5.2. Principais aprendizagens

Durante a realização desta pesquisa, buscamos observar as principais aprendizagens dos alunos decorrentes no desenvolvimento dos programas. Abaixo listamos quais foram essas aprendizagens:

- Construção de soluções;
- Independência na aprendizagem;
- Busca de meios para tornar o código do programa mais legível, de modo a facilitar a apresentação da solução aos colegas da turma;
- Reflexão para corrigir funções incorretas;
- Importância de se pensar em várias soluções, como forma de ganhar familiaridade com o problema;
- Compreensão do problema antes de resolvê-lo;
- Vantagens de trabalhar de forma colaborativa;
- Motivação em continuar a desenvolver projetos de programação, mesmo após o término da disciplina;
- Reflexão sobre a própria solução, de modo a acelerar a compreensão da solução elaborada.

5.3. Avaliação da aprendizagem

A arquitetura pedagógica usada, propiciou um canal permanentemente aberto de trocas, de interações, diálogos, construções entre professor-aluno e aluno-aluno.

Com o uso dos recursos digitais, todos os integrantes tiveram acesso às soluções publicadas de toda a turma, conseguindo desta forma analisar as soluções dos outros e compará-las com as próprias soluções. Esse procedimento usa o potencial da turma de produzir várias possíveis soluções e aumenta a tomada de consciência e a aprendizagem sobre o processo de construção de soluções para cada problema.

Apresentamos no gráfico 02 uma comparação entre as atividades desenvolvidas por cada estudante e suas respectivas médias.

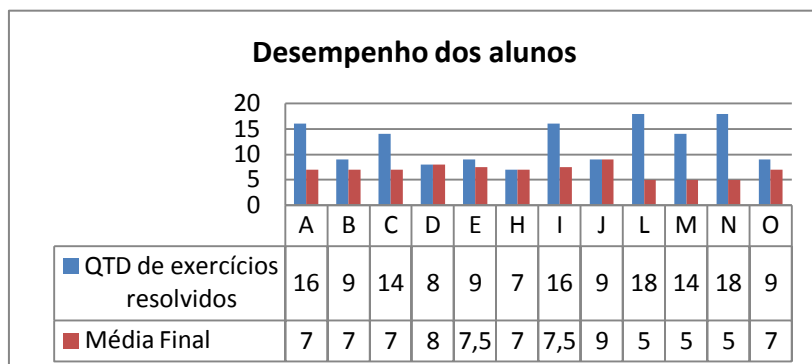


Gráfico 02: Desempenho dos alunos

Aparentemente não há relação entre a quantidade de exercícios resolvidos pelos estudantes e a média obtida. Acontece que alguns alunos, reprovados na disciplina em uma tentativa anterior, apresentavam dificuldade maior durante as etapas do desenvolvimento do programa. Esses alunos foram incentivados a se aplicarem mais na resolução das listas de exercícios e, seguindo essa orientação, conseguiram aprender e serem aprovados.

6. Considerações finais

Esse trabalho apresentou uma arquitetura pedagógica dotada de recursos digitais para apoiar e acompanhar o estudante no processo de aprendizagem de programação. Essa arquitetura pedagógica tem como objetivo ampliar as interações entre todos os participantes da turma e promover a pró-atividade, a autoria e o espírito crítico no estudante.

Constatamos que com o uso de arquiteturas pedagógicas podemos favorecer a aprendizagem, por meio do acompanhamento melhor da trajetória de aprendizagem de cada estudante e pelo uso da força de trabalho do coletivo para favorecer a aprendizagem de cada indivíduo.

Durante o acompanhamento da aprendizagem do estudante, houve um mapeamento entre as habilidades desenvolvidas e os conteúdos aprendidos. Assim, o professor pode ter um diagnóstico das lacunas de aprendizagem e pode intervir diretamente nas dificuldades identificadas em cada estudante.

Para a continuação das pesquisas a partir deste trabalho, entre outras atividades, buscaremos construir arquiteturas pedagógicas para apoiar a aprendizagem de programação em cursos realizados na modalidade a distância.

7. Referências

Anderson, J.; Cognitive psychology and its implications. New York and Basingstoke: Worth Publishers, (2000). 181-183 p.

- Carvalho, M. J. S.; Nevado, R. A. de; Menezes, C. S. de. (2007) “Arquiteturas pedagógicas para a educação a distância”. In: Nevado, R. A. de; Carvalho, M. J. S.; Menezes, C. S. de. (Orgs.). *Aprendizagem em rede na educação a distância: estudos e recursos para formação de professores*. 1 ed. Porto Alegre: Ricardo Lenz, p. 36-52.
- Carvalho, M.J.S.; Nevado, R. A.; Menezes, C. S.; “Arquiteturas Pedagógicas para Educação a Distância: Concepções e Suporte Telemático”. In: XXI Simpósio Brasileiro de Informática na Educação. João Pessoa, 2010.
- Costa, L. B. ; Oliveira, M. G.; Metodologia ANEA para avaliação online de Lógica de Programação. In: XXII Simpósio Brasileiro de Informática na Educação/ XVII Workshop de Informática na escola, 2011, Aracaju. Anais do XXII Simpósio Brasileiro de Informática na Educação/VII Workshop de Informática na escola, 2011.
- Falkner, K., Palmer, M.; Developing authentic problem solving skills in introductory computing classes, In: SIGCSE'09, Chattanooga, Tennessee, USA, (2009).
- Gomes, A. e Mendes, A. (2000). Suporte à Aprendizagem da Programação com o Ambiente SICAS. Actas do V Congresso Ibero-americano de Informática Educativa, Viña del Mar, Chile.
- Marcon, K., Machado, J.B.; Carvalho, M. J.S.: Arquiteturas Pedagógicas e Redes Sociais: Uma experiência no Facebook. In: Congresso brasileiro de Informática na Educação, 2012.
- Piaget, J.; Evolução intelectual da adolescência à vida adulta (Tradução de Tania Beatriz Iwaszko Marques e Fernando Becker do artigo publicado em inglês sob o título: Intellectual Evolution from Adolescence to Adulthood, pela Human Development, 15:1-12, 1972).
- Polya, George. How to solve it: a new aspect of mathematical method. Princeton University Press, (2004).
- Real, L. M. C. ; Corbellini, Silvana . Proposta de uso de WIKI como Arquitetura Pedagógica: cooperação. In: 22º Simpósio Brasileiro de Informática na Educação - 17º Workshop de Informática na Escola, 2011, Aracaju. Anais do XXII SBIE - XVII WIE, 2011. v. Wapsed. p. 1-9.
- Ribeiro, R.S; Brandão L.O; Brandão; A. A. F.: Uma visão do cenário nacional do ensino de algoritmos e programação: uma proposta baseada no paradigma de programação visual. I Congresso de brasileiro de Informática na educação, 2012, Rio de Janeiro.
- Souza, D.M; Maldonado, J.C.; Barbosa, E. F.: Aspectos de Desenvolvimento e Evolução de um Ambiente de Apoio ao Ensino de Programação e Teste de Software. I Congresso de brasileiro de Informática na educação, 2012, Rio de Janeiro.
- Tavares, O. L.; Menezes, C.S.; Nevado, R.A.: Pedagogical architectures to support the process of teaching and learning of computer programming: In FIE2012- Frontiers in education conference, 2012.
- Wirth, Niklaus. Program Development by Stepwise Refinement, Communications of the ACM, vol. 14, pp. 221-227, (1971).