UNISUL

III SIMFOP
Simpósio sobre Formação de Professores:
tecnologias e inovação na educação básica

ABORDAGENS LÚDICAS PARA O ENSINO E APRENDIZAGEM DE LÓGICA DE PROGRAMAÇÃO NA EDUCAÇÃO PROFISSIONAL

Ranieri Alves dos SANTOS[1]

ABSTRACT:

The teaching of programming logic provides mid-level curricula in computer science to theoretical and practical performance of students in programming languages, preparing to work in the development of computer programs. This teaching through representations in natural language construct logical and procedural aspects, approaching he student to complex algorithmic and mathematical functions in a standardized way. However, being a discipline sometimes unusual for some students, as it involves advanced calculations, logical and procedural representations. Such contents require an unusual dedication to students who once could not have developed the practice of reading and self-study got discouraged about the subject and culminating in a failing grade. Therefore, this article proposes a series of playful practices intended to speed the process of teaching and learning in teaching situations and apply the teaching-based didactic transposition of teaching logic programming.

KEYWORDS:

Teaching; Approach; Playful; Logic

1 Introdução

O ensino de lógica de programação é algo imprescindível na formação de um profissional para a área de desenvolvimento de software, área em constante expansão e carente de mão de obra. Esta disciplina, presente em todos os currículos de cursos de educação profissional é essencial para o aprendizado futuro de técnicas de programação mais avançadas.

A lógica de programação trata da na alfabetização do programador como profissional da área de desenvolvimento de software, servindo de disciplina base para todas as outras unidades curriculares que envolvam conteúdos ligados à programação, sendo o principal pré-requisito para que o aluno obtenha êxito em sua atividade profissional, fato este que acaba imprimindo a real importância da lógica de programação na formação dos profissionais da área de software.

Martins (2009) argumenta que uma das características do ser humano é buscar justificar o que acredita e compreender a forma com que os outros justificam o que acreditam,

[1]Pós-graduando em Ensino de Ciências pelo Instituto Federal de Santa Catarina (IFSC), cursando licenciatura em Educação Profissional pela Universidade do Sul de Santa Catarina (UNISUL), e bacharel em Sistemas de Informação pela UNISUL. Professor de programação no Serviço Nacional de Aprendizagem Industrial (SENAI) em Tubarão-SC.

sendo este um dos principais objetivos da lógica. O mesmo salienta que não é algo simples definir a lógica de programação, visto que esta aborda uma infinidade de problemas e não possui fronteiras definidas, sendo assim, os extremos são voltados à matemática e à filosofia.

O ensino de lógica de programação é algo delicado dentro das áreas da informática, pois exige atenção e raciocínio lógico apurado levando alunos que não tenham tais facilidades ou que não se dediquem a adquiri-las à eminente reprovação. Os conteúdos são sequenciais, dependentes um dos outros e de dificuldade crescente.

Alguns alunos ao iniciarem esta disciplina, por seu interesse em assuntos avançados relacionados à informática já conhecem alguma linguagem de programação, e se sentem autossuficientes no conteúdo, fato que atrapalha o próprio aprendizado do aluno. Por vezes esta disciplina não é encarada com bons olhos por parte do corpo discente, visto que é um conteúdo sofisticado e trabalhoso, que exige uma dedicação extra por parte de alunos que outrora não tenham a prática do estudo autônomo. Estes problemas resultam na falta de interesse dos alunos na disciplina, acarretando em baixas notas nas avaliações e culminando na reprovação.

Tendo em vista estas dificuldades, se faz necessária a inclusão de métodos diferenciados na ministração desta disciplina, colaborando com a realidade dos alunos e os motivando no estudo desta área. Este artigo apresenta um enfoque abrangente sobre o mercado de desenvolvimento de software, o ensino de lógica de programação e a sua importância dentro do mercado de tecnologia da informação, as abordagens propostas para auxiliar no ensino e aprendizagem do conteúdo em questão, e algumas considerações finais como conclusão, fruto de experiências anteriores na docência em turmas de nível médio, em cursos técnico na área de informática, com vistas à preparação de profissionais para atuação no mercado de desenvolvimento de software da região.

2 Mercado de Software

A programação, é compreendida pelo ofício da codificação de sistemas de softwares visando compor sistemas informatizados implementados em ambientes computacionais. Por décadas o ofício do programador esteve diretamente ligado ao meio científico e militar, onde profissionais técnicos desenvolviam softwares ligados aos departamentos de defesa e espaciais de suas nações. O desenvolvimento de software surgiu da necessidade da criação de programas para resolver cálculos complexos, para o controle de processos industriais e para aplicabilidades restritas a pequenas fatias de mercado. Com o passar do tempo, e com a evolução e democratização do acesso às tecnologias da informação a necessidade pelo desenvolvimento de softwares específicos se fez necessária.

Ao longo do tempo, a gestão de informações se tornou algo essencial no meio empresarial, onde uma organização que domina seus dados internos detém diferencial competitivo perante a concorrência (DAVENPORT, T.; PRUZAC, L., 1998). Devido a esta necessidade surgiu a carência do mercado por softwares específicos à realidade das empresas, envolvendo os seus processos e adequados ao seu ramo de atuação.

Com base nesta necessidade surgiram as fábricas de softwares, empresas especializadas no desenvolvimento de software sob medida, focadas na criação de sistemas de software customizadas para seus clientes, iniciando assim a busca por profissionais capacitados na área de programação, capazes de desenvolver softwares adequados às necessidades empresariais.

Apesar deste foco empresarial, o mercado de software não se restringe à este meio. De softwares são compostos os jogos de computador, os sistemas operacionais, os sistemas industriais, os celulares, dentre outros equipamentos domésticos, imprimindo aqui a abrangência da área de software.

O mercado de desenvolvimento de software no país e na região está em constante crescimento, as fábricas de software atendem à demandas locais e regionais, atuando também no mercado internacional, exportando softwares para outros países. Em 2010, no Brasil, o mercado de desenvolvimento de software demonstrou um crescimento de cerca de 23%, destoando do restante da sua cadeia produtiva perante a crise global em que o mercado passou, tendo estimado para 25% o crescimento para o ano de 2011 (BAGUETE, 2011).

Os papéis dos profissionais que atuam nos setores de desenvolvimento de sistemas de software são basicamente ligados aos ofícios de programador, analista, testador, e gerente. O programador, ou desenvolvedor, é o profissional que implementa as soluções utilizando de habilidades técnicas quanto às tecnologias ligadas ao desenvolvimento de software. O analista, por vezes denominado como analista de sistemas, é o profissional responsável pelo levantamento dos requisitos de cada sistema, desenvolvendo artefatos documentais visando fornecer ao programador subsídios funcionais para que este implemente o software. Testador é o profissional que garante a qualidade do software, buscando falhas e inconsistências nos sistemas. O gerente, ou gerente de projetos é o profissional dedicado à gestão dos projetos de software (RUP, 2011).

Os segmentos de mercado ligados ao desenvolvimento de software são compreendidas basicamente em fábricas de software e empresas de off-shore, ou de forma autônoma com consultoria e desenvolvimento freelance. As fábricas de software são empresas que possuem e desenvolvem produtos ou serviços ligados à sistemas de software padronizadas ou personalizadas de acordo com cada necessidade, atendendo diretamente as empresas que necessitam de sistemas informatizados. As empresas de off-shore atendem as próprias fábricas de software, atuando em desenvolvimento de soluções e módulos específicos dos sistemas que as fábricas de software não possuem tempo ou domínio tecnológico para o mesmo. Os profissionais autônomos que atuam no desenvolvimento de software podem ser consultores independentes ou desenvolvedores freelance, que trabalham diretamente em projetos dentro de empresas que necessitem de sistemas de software, em indústrias de software, e em empresas de off-shore, atendendo à demandas tecnológicas específicas (IT WEB, 2011).

3 Ensino de Lógica de Programação

Por se tratar de um assunto científico, a lógica também sofre os efeitos da transposição didática, um termo cunhado pelo sociólogo Michel Verret em 1975, e rediscutido em 1985 por Yves Chevallard. Esta teoria afirma que o saber não chega à sala de aula da mesma forma com que este foi constituído no meio científico, passando então por uma transformação: a transposição didática (CHEVALLARD, 1991). Esta teoria divide os níveis de transposição didática em três, o saber sábio, saber a ensinar e o saber ensinado.

A transposição didática do conhecimento científico, considerada como o conjunto de transformações adaptativas que tornam o conhecimento científico apto à ser ensinado (CHEVALLARD, 1985), tal como sua nomenclatura sugere, serve como método que visa facilitar o processo de ensino de conteúdos produzidos em meio acadêmico, utilizando as transformações do saber.

A base da transposição didática é o saber, segundo a teoria, são as transformações ocorridas no saber que compõem as transposições sofridas pelo conhecimento científico. Estas etapas são protagonizadas por agentes pertencentes ao ambiente denominado como noosfera, termo proposto por Chavallard (1991) para representar as instituições de transposição de saberes, compostas por conjuntos de instituições que determinam quais conteúdos científicos devem ser expostos no ambiente escolar. Na noosfera são realizadas as interações entre os ambientes didáticos e sociais, é nela que se pensa o funcionamento

didático, visando assegurar as relações entre o sistema de ensino e a sociedade global (ARSAC et al. 1989).

A transposição didática do conhecimento científico, considerada como o conjunto de transformações adaptativas que tornam o conhecimento científico apto a ser ensinado (CHEVALLARD, 1985), tal como sua nomenclatura sugere, serve como método que visa facilitar o processo de ensino de conteúdos produzidos em meio acadêmico, utilizando as transformações do saber.

De acordo com a transposição didática, o saber é subdividido em três dimensões, que representam cada uma das etapas sequenciais em que o conhecimento é transposto: o saber sábio, saber a ensinar, e o saber ensinado (CHEVALLARD, 1985).

O saber sábio, primeira etapa de transposição, pode ser entendido como o resultado científico bruto, recém criado e concebido em ambiente científico. A linguagem e a representação deste conhecimento é íntima e ligada aos membros deste ambiente e dos seus envolvidos, tornando inviável o uso do conhecimento nesta instância dentro de sala de aula. Mesmo quando o determinado conhecimento é compartilhado com a comunidade, este é feito por meio de publicações científicas, e mesmo assim, a linguagem continua muito próxima apenas da comunidade científica.

A segunda etapa da transposição, o saber a ensinar, parte do princípio de que o conhecimento científico, quando publicado, difundido e aceito pela comunidade deve ser transmitido à humanidade para que os interessados tenham domínio nesta área. Surge então a necessidade de que este conhecimento seja lapidado e estruturado de forma facilitada visando à compreensão de indivíduos que previamente não sejam envolvidos no ambiente científico.

Na terceira e última etapa da transposição didática, o saber ensinado, o conhecimento por sua vez é transmitido aos alunos, porém os materiais desenvolvidos no momento do saber a ensinar por si só não compõem um processo de ensino e aprendizagem, visto que são apenas recursos disponíveis no ambiente estudantil, ambiente este que forma o saber ensinado.

A lógica de programação é fruto da lógica filosófica, algo puramente científico, representando então o nível inicial da transposição didática, o saber sábio, onde foram estudados e definidos os comandos de programação que representam a realidade na forma algorítmica (FORBELLONE; EBERSPÄCHER, 2005).

Os comandos provenientes da lógica de programação são aplicados nas linguagens de programação de forma diferente umas das outras, diferindo certas vezes também nos comandos abordados. Portanto os materiais didáticos, jogos e objetos de aprendizagem voltados ao ensino de programação apresentam uma abordagem padronizada denominada de algoritmo. Esta heurística abordada pelos materiais didáticos para o ensino trata de comandos genéricos, próximos da linguagem natural humana, representando então o nível intermediário da transposição didática: o saber a ensinar.

Os materiais que abordam algoritmos por vezes apresentam abordagens distintas, tanto na sintaxe dos comandos, como nas técnicas de programação, portanto cabe ao professor definir qual adotar, voltando principalmente sua abordagem ao mercado local, já que se trata de uma disciplina profissionalizante voltada ao mercado de trabalho. As adaptações realizadas pelo professor ao proferir a disciplina representa último nível da transposição didática, o saber ensinado, ponto onde o professor ao ministrar a disciplina atua.

Os conteúdos envolvidos nas disciplinas de lógica de programação excetuando-se algumas distinções de taxonomia, são basicamente as expressões, as estruturas de condição, as estruturas de repetição, as variáveis indexadas e as funções. Expressões, são as representações lógicas de operações matemáticas, subdivididas em expressões aritméticas, que envolvem os comandos algébricos, e as expressões lógicas, que abordam os operadores lógicos e de comparação. As estruturas de condição, são formas de representação de decisões baseadas em expressões lógicas, que se obtém o resultado binário que retorna verdadeiro ou falso. As

estruturas de repetição são comandos lógicos que operam repetições pré-determinadas ou não em partes do código. Variáveis indexadas, são arranjos multidimensionais para o armazenamento de dados e as funções são procedimentos de código pré-definidos para a modularização e organização dos códigos em programas (MARTINS, 2009).

O contrato didático, um acordo que determina as regras entre os envolvidos no processo de ensino e aprendizagem (BROSSEAU, 1996), ao ser firmado entre o professor de lógica de programação e os alunos deve estar voltado aos objetivos da disciplina, definindo os limites do uso dos equipamentos. Em geral disciplinas deste gênero poderão utilizar simuladores computacionais nos laboratórios de informática para testar seus algoritmos. E o uso do laboratório é um detalhe sensível a ser firmado no contrato didático, já que o uso indevido dos computadores poderá comprometer consideravelmente o andamento das aulas.

Em geral, não há nada relativamente difícil quando se trata do conteúdo de lógica de programação, porém existem sim alunos que não possuem capacidades e aptidões necessárias para a compreensão de forma simples da lógica de programação, devido a obstáculos didáticos ou epistemológicos do indivíduo em questão (BYRNE; LYONS, 2001).

Durante o exercício docente, o professor se depara com situações adversas em seu fazer pedagógico, expressas por barreiras entre o professor e os alunos, e também entre o próprio conteúdo com os alunos. Estes obstáculos podem ser de natureza didática ou epistemológica, e são denominados de obstáculos didáticos e epistemológicos pelo filósofo francês Bachelard (1999). Segundo Bachelard (1999), os obstáculos didáticos ou epistemológicos são constituídos por dificuldades encontradas pelos discentes no momento da construção do conhecimento por meio dos processos educativos.

Um dos obstáculos encontrados durante a tarefa do ensino de lógica de programação deve-se ao conhecimento prévio de alguns alunos nas áreas de informática. Este obstáculo epistemológico ocorre quando o aluno já possui uma bagagem anterior sobre programação. Porém estes conhecimentos em geral são baseados em uma linguagem de programação específica, aprendida por meio de exemplos encontrados na internet.

Estes alunos com conhecimentos prévios adentram em disciplinas de lógica de programação já com um espírito apático às aulas, sendo necessária então a desconstrução do paradigma aprendido anteriormente para a construção do conhecimento do modo correto.

Brosseau (1996) afirma que a construção do conhecimento é baseada em dois momentos distintos baseados em duas situações diferenciadas, as situações didáticas, que envolvem o papel do educador como facilitador do processo de ensino, e as situações a-didáticas, onde não há o papel do professor.

Parte do aprendizado de lógica de programação pode ser comparada de forma análoga como o aprendizado de um instrumento musical. Para aprender o instrumento é necessário três ações do estudante: a primeira é o aprendizado mediado pelo professor, onde o aluno adquire os conhecimentos necessários para o domínio do instrumento; a segunda é a prática individual, visando adquirir maior habilidade e concentração; e a terceira é a prática em grupo, com outros instrumentistas, visando a concepção musical como um todo.

Desta forma deve ser o aprendizado de lógica de programação, onde no primeiro passo o aluno conta com o apoio do professor, já o segundo e o terceiro passos são baseados na sua própria capacidade de prática e em grupo, perfazendo então uma situação a-didática.

4 Abordagens Lúdicas Propostas

Visando alcançar os alunos em dificuldade, recuperando-os, se faz necessária a adoção de abordagens lúdicas para dinamizar as aulas e ensinar de uma forma interativa e ligada ao seu ambiente. Para tanto tais abordagens são baseadas em jogos, visto que este é um assunto em comum com os alunos deste nível, e de domínio de todos.

O pesquisador de tecnologias educacionais Papert (1994) afirma que os jogos computacionais abrangem experiências que o ambiente escolar não consegue criar, exigindo então do aluno esforços intelectuais e níveis de aprendizagem muito mais elevados que os utilizados nas atividades escolares, e por fim Piaget (1998) conclui que uma criança que joga desenvolve suas percepções, sua inteligência, suas tendências à experimentação, e seus sentimentos sociais.

Como abordagens válidas para o apoio aos processos de ensino e aprendizagem de lógica de programação no ensino profissionalizante podem ser baseadas nos jogos LightBot, Robocode, no brinquedo Lego Minstorms e na ferramenta Alice.

O LightBot (figura 1) é um jogo online, acessado pelo navegador de internet do portal Armor Games (2011), e é executado sobre uma plataforma Flash. Este jogo oferece ao aluno um robô que precisa transpor obstáculos para acender luzes nos locais indicados. Por meio dos comandos disponibilizados, o aluno programa as ações do robô para que o mesmo seja capaz de atender aos objetivos. Trata-se de um jogo iniciante, onde de forma gráfica o aluno exercita sua capacidade lógica, praticando os comandos sequenciais e o conceito de procedimento e função (ARMOR GAMES, 2011).

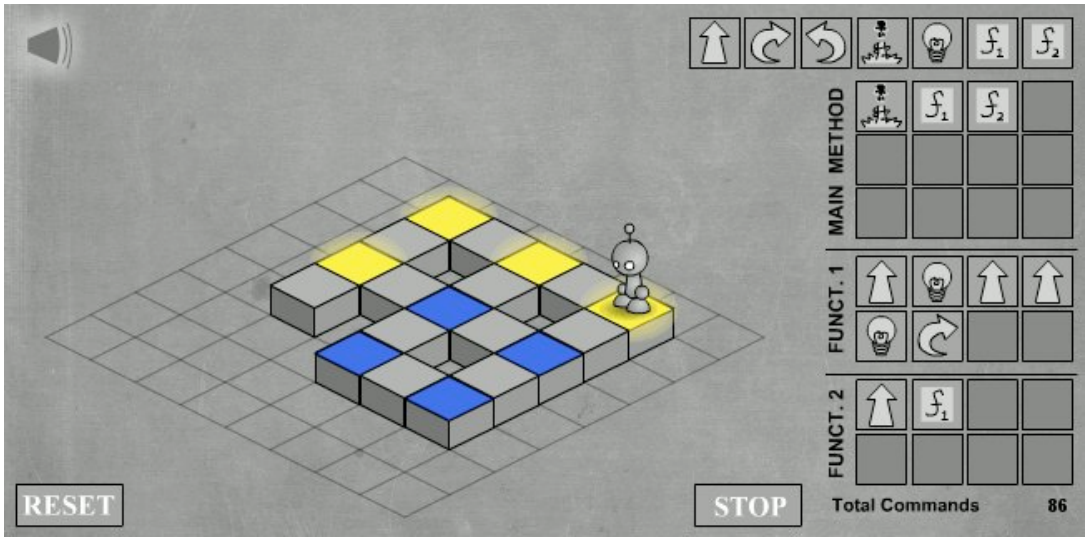

Figura 1: Jogo Light-Bot

O Lego Mindstorms (figura 2), baseado no brinquedo Lego, é um kit de robótica, onde utilizando as peças de montar em sensores, motores em engrenagens, o aluno é capaz de programar o comportamento do seu próprio robô e ver a execução de forma física. Os robôs são programados no computador, e os robôs são montados com as peças de Lego, a lógica programada no computador é transferida para o controlador do robô, que por sua vez executa o programa desenvolvido.

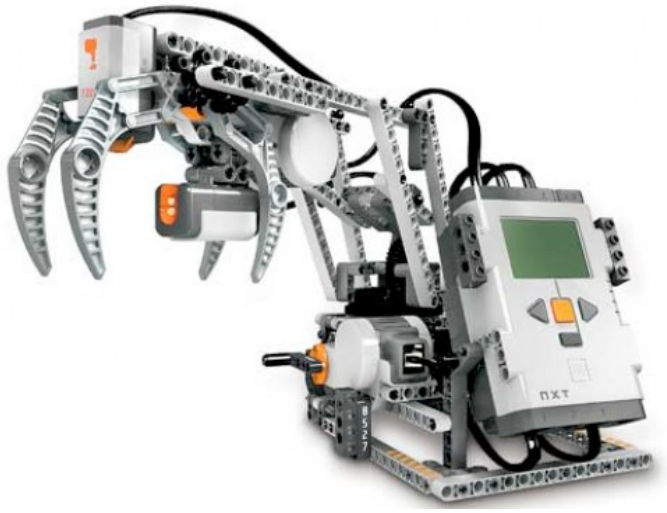

Figura 2: O kit Lego Mindstorms

O jogo Robocode (figura 3) é uma ferramenta desenvolvida pela IBM, e que exige do aluno conhecimentos mais avançados, onde cada aluno programa o comportamento do seu robô tanque de guerra visando destruir seus inimigos, outros tanques de guerra. Os robôs são programados já em uma linguagem de programação e aborda conceitos avançados de programação. Após os alunos programarem os seus robôs, estes são inseridos em um único computador que serve de arena para que os mesmos batalhem uns com os outros (ROBOCODE, 2011).

Figura 3: Batalha executada em Robocode

Alice é uma ferramenta gráfica criada pela universidade Carnegie Mellon, tendo como mentor o professor Randy Pausch (2008), autor do livro A Lição Final, que o desenvolveu com o objetivo de facilitar o aprendizado de programação de forma dinâmica e divertida, de forma que os alunos criem seus próprios mundos virtuais e em contrapartida estariam

aprendendo um ofício. O Alice foi implantado em diversas escolas e universidades americanas e está em plena ascensão em âmbito mundial. Com o software (figura 4), o aluno programa o comportamento de objetos utilizando o mouse da mesma forma que o programaria com comandos de linguagem de programação. Os comportamentos podem gerar animações tridimensionais e jogos de computador (ALICE, 2011).

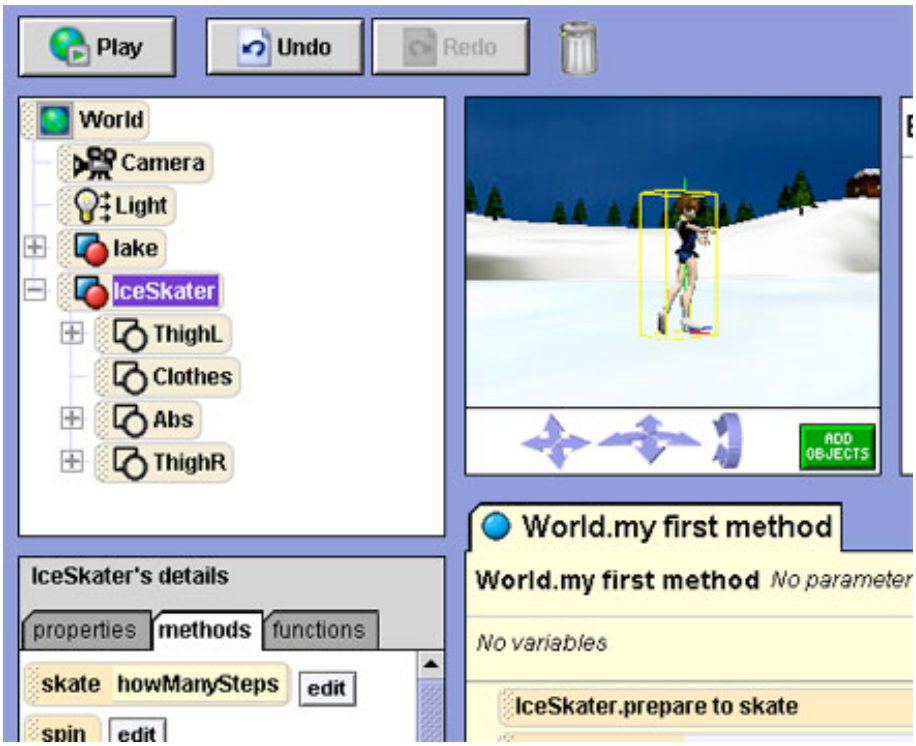

Figura 4: Tela de edição da ferramenta Alice

Estas ferramentas devem ser utilizadas como um meio para o processo de ensino e aprendizagem, e esta proposta deve ser deixada às claras aos alunos, desde o momento do contrato didático. A aplicação destas abordagens deve ser extremamente contextualizada, e voltada ao aprendizado do aluno, deixando claro o objetivo do uso da ferramenta, que pode ser facilmente desvirtuada, e confundida com uma simples brincadeira, ou com uma competição.

5 Conclusões

O presente artigo é fruto de ministrações da disciplina de lógica de programação, e da aplicação de abordagens lúdicas como estas. O fato dos alunos utilizarem de ferramentas dinâmicas e interativas para o ensino de uma atividade profissional tem se mostrado algo altamente eficaz dadas as experiências em turmas que usufruíram destes recursos em seu aprendizado de lógica de programação.

Estas abordagens por si só não compõem o fazer pedagógico de aulas de lógica de programação, ressaltando o papel do docente como facilitador do processo, que deverá intervir em situações onde o objetivo das abordagens esteja desvirtuado. Tais abordagens necessitam de supervisão constante, e do acompanhamento contextualizado com as linhas da disciplina visando cumprir com os objetivos propostos pelas abordagens.

Entre as abordagens propostas, o uso do Robocode, por exemplo, é uma constante nas turmas foco desta pesquisa, culminando em um evento anual, onde há um campeonato de programação, no qual os alunos são submetidos a desafios uns contra os outros, visando uma

competição sadia entre os alunos, e estimulado a prática e o aprendizado de programação de uma forma divertida.

A intervenção destas técnicas lúdicas como abordagem facilitadora do processo de ensino de programação deve também estar pautada junto ao objetivo da disciplina, visto que o processo de ensino e aprendizagem deve ter cunho motivacional visando com que este se torne divertido, porém por mais que a área de informática seja dinâmica, deve ser bem clara a visão de que os alunos estão sendo preparados para uma área profissional, ou seja, as atitudes envolvidas nas brincadeiras e competições envolvidas nestas abordagens não estarão presentes neste meio, e a menos que os alunos atuem no desenvolvimento destas soluções de entretenimento, o mesmo estará envolvido em um ambiente onde não haverão estas ferramentas.

Com base nos relatos dos próprios alunos é possível mensurar a eficácia destes métodos, visto que os mesmos afirmam que são modos dinâmicos e divertidos, modos fáceis de aprender algo difícil, utilizando brincadeiras.

Dados estes fatos, é notável a importância do uso destas e outras de abordagens buscando tornar algo simples e motivante o aprendizado de lógica de programação.

Referências

ALICE. Disponível em: <http://www.alice.org>. Acesso em 23 abr. 2011.

ARMOR GAMES. Disponível em: <http://armorgames.com/play/2205/light-bot>. Acessoem28 abr. 2011.

BACHELARD, G. **A formação do novo espírito científico**: contribuição para uma psicanálise do conhecimento. Rio de Janeiro: Contraponto, 1999.

BAGUETE. Disponível em: <http://www.baguete.com.br>. Acesso em 23 abr. 2011.

BROSSEAU, G. **Lês obstacles épistémologique set lês problèmes em mathématiques**. RDM, 1983.

BYRNE, P., LYONS, G. **The Effect of Student Attributes on Success in Programming**. In: Proceedings of the 6th Annual Conference on Innovation and Techmology in Computer Science Education, United Kingdom, 2001.

CHEVALLARD, Y. **La Transposition Didactique**: du savoir savant au savoir enseigné. La Pensee Sauvage Éditions. Genoble, 1991.

DAVENPORT, T.; H, PRUZAC, L. **Working Knowledge How Organizations Manage What They Know**. Harvard Business School Press, 1998.

FORBELLONE, André L. V. e EBERSPÄCHER, Henri F. **Lógica de Programação**: A construção de algoritmos e estruturas de dados. 3ª edição. São Paulo: Prentice Hall, 2005.

IT WEB. Disponível em: <http://www.itweb.com.br>. Acesso em 02 mai. 2011.

LEGO MINDSTORMS. Disponível em: <http://mindstorms.lego.com>. Acesso em 22 abr. 2011.

MARTINS, João Pavão. **Lógica de Programação**. Lisboa: Universidade Técnica de Lisboa, 2009.

PAPERT, Seymour. **A máquina das crianças**: repensando a escola na era da informática. Porto Alegre: Artes Médicas, 1994.

PAUSCH, Randy. **A Lição Final**. Rio de Janeiro: Ediouro, 2008.

PIAGET, Jean. **Psicologia e Pedagogia**. Rio de Janeiro: Forense Universitária,1998.

ROBOCODE. Disponível em: <http://robocode.sourceforge.net>. Acesso em 20 abr. 2011.

RUP. Rational Unified Process.Disponível em: <http://www.ibm.com/software/awdtools/rup>. Acesso em 21 abr. 2011.