
MOTIVAÇÃO DISCENTE NO ENSINO-APRENDIZAGEM DE PROGRAMAÇÃO DE COMPUTADORES

Márcio Roberto de Lima¹

Murilo Cruz Leal²

Artigo publicado na Revista Educação e Tecnologia – v. 1 – n. 17 - ISSN 1414-5057. Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG).
<http://seer.dppg.cefetmg.br/index.php/revista-et>

Resumo

Este artigo relata um estudo realizado com 21 alunos formandos de um curso de Sistemas de Informação, com os quais se buscou identificar fatores relacionados à motivação discente frente ao aprendizado de programação de computadores. O trabalho envolveu a aplicação de um questionário diagnóstico com fins de se determinar a relação pregressa e envolvimento com as unidades curriculares de programação. Posteriormente, cinco alunos concordaram em participar de entrevista semiestruturada. As entrevistas foram gravadas, transcritas e submetidas à análise de conteúdo. Os fragmentos de interesse foram identificados e categorizados com base nas unidades de significado relacionadas ao foco da pesquisa. Em adição aos depoimentos colhidos, o resultado da análise apontou a complexidade do processo de ensino-aprendizagem de programação de computadores, sendo que do ponto de vista dos discentes percebeu-se que o principal fator motivacional de suas aprendizagens está ligado à atuação do professor das unidades curriculares de programação. O professor precisa ter domínio de conteúdo e competência pedagógica para propiciar a satisfatória didatização dos conteúdos.

Palavras-chave: programação de computadores, ensino-aprendizagem, motivação discente, ensino superior

Abstract

This article reports an experiment conducted with 21 senior students of a course in Information Systems, with which it sought whose objective was identify factors related to student motivation forward to learning

¹ Doutorando em Educação pela Universidade Federal de Minas Gerais (UFMG). Mestre em Educação, Universidade Federal de São João del-Rei (UFSJ), 2009. Professor do Departamento de Ciências da Educação da Universidade Federal de João del-Rei (DECED/UFSJ). E-mail: marcinholima@ufsj.edu.br

² Membro do Departamento de Ciências Naturais (DCNAT/UFSJ) – Professor do Mestrado em Educação da Universidade Federal de São João del-Rei – E-mail: mcleal@ufsj.edu.br

computer programming. The work involved the application of a diagnostic questionnaire for purposes of determining the relationship and previous involvement with the programming courses. Subsequently, five students agreed to participate in a semi-structured interview. The interviews were taped, transcribed and subjected to content analysis. The fragments of interest were identified and categorized based on units of meaning related to the research focus. In addition to the testimonies gathered, the analysis results indicated the complexity of teaching and learning computer programming, and from the standpoint of the students realized that the main motivating factor for their learning is linked to teacher performance of programming courses. Teacher must have mastery of content and pedagogical competence to provide a satisfactory didactization of content.

Keywords: *programming, computers, teaching and learning, student motivation*

INTRODUÇÃO

Em âmbito geral, a adoção das tecnologias digitais possibilita modificações nas formas de trabalho das organizações, proporcionando novas maneiras de “fazer” e, primordialmente, de se pensar o “fazer” (MORAES, 1997). Um exemplo dessa realidade são os empreendimentos tradicionais que incorporaram as transações virtuais a seu ramo de atuação, culminando com o surgimento das chamadas “empresas pontocom”, que têm na internet a plataforma de negociação direta com clientes (lojas virtuais, internet banking, sítios de leilões etc.).

Essa perspectiva de mudança e inovação também diz respeito à Educação, que gradativamente vem incorporando o computador nos processos de ensino-aprendizagem em todos os níveis e modalidades praticadas. Além disso, a constante presença das tecnologias nas atividades praticadas suscitou a criação de cursos para preparar profissionais para exercerem atividades específicas, tais como: programadores de computadores, técnicos em hardware, analistas de sistemas, administradores de banco de dados, engenheiros de *software*, *web designers* e tecnólogos diversos. Especificamente, no Ensino Superior, destaca-se o surgimento dos cursos de Ciência da Computação, Engenharia da Computação, Licenciatura em Computação, Sistemas de Informação, Tecnologia de Redes e Banco de Dados, entre outros. Entretanto, para além do domínio específico da área da computação, os processos educacionais do Ensino Superior das demais áreas de conhecimento vêm se adequando e incorporando unidades curriculares relacionadas às TIC em suas estruturas curriculares.

Conforme é apontado por Lévy (1999, p.26): “enquanto discutimos sobre os possíveis usos de uma dada tecnologia, algumas formas de usar já se impuseram”. Um dos reflexos desse movimento, diz respeito à atual agenda de pesquisa de cibercultura e educação, que enfoca o uso do computador e as tecnologias de rede com vistas aos processos de aprendizagem *online*. Perrenoud (2000) também traz colaborações ao problematizar o tema, isso no sentido de propor um deslocamento do foco de discussão da utilização ou não das TIC para a forma de sua

utilização. Assim, a relevância dos aspectos relacionados à qualidade da aprendizagem resultante do uso das TIC assume um papel fundamental, principalmente nos ambientes educacionais onde o computador é um elemento indispensável – como é o caso dos cursos específicos da área da computação.

A investigação aqui descrita teve como cenário um curso de Sistemas de Informação de uma universidade do interior de Minas Gerais e concentrou-se no núcleo de unidades curriculares (UCs) relacionado à programação de computadores. Conforme apresentado por Martins e Correia (2003), Rodrigues Júnior (2002), Schultz (2003), Chaves de Castro *et al* (2003), Delgado *et al* (2004), Maltempi e Valente (2000) e Petry (2005), os resultados obtidos nas UCs que envolvem programação de computadores são insatisfatórios, constatando-se: baixo nível de aprendizagem, desestímulo, evasão e reprovação. Mediante a essas constatações, a motivação do estudo foi a busca de uma melhor compreensão do processo de ensino-aprendizagem de programação de computadores, visando assim fornecer subsídios para seu aprimoramento. Especificamente, o estudo empreendido teve como objetivo investigar os fatores que levam a motivação/desmotivação dos alunos no processo de construção do conhecimento pelo computador via linguagem de programação.

Para isso, contou-se com a colaboração de 21 alunos formandos. Inicialmente, foi identificado o perfil da turma e sua relação pregressa com as unidades curriculares de programação (UCPCs), com o auxílio de um questionário. Posteriormente, cinco alunos do grupo se dispuseram a contribuir com depoimentos seguindo uma entrevista semiestruturada. Suas falas foram gravadas, transcritas e submetidas à análise de conteúdo, de acordo com Bardin (1977).

A PROGRAMAÇÃO DE COMPUTADORES E O SEU ENSINO

A afirmativa de que os ambientes educacionais são pobres em recursos que estimulem o pensamento e a expressão de ideias é habitual. Parte desse pensamento encontra força no próprio sistema educacional, que preserva sua base de ensino apoiada na transmissão e reprodução de conhecimento, na repetição e memorização de informações. Papert (1993/2008) lembra que o educador brasileiro Paulo Freire criticava essas práticas, e recorda a metáfora sugerida por ele, onde a escola seguia um “modelo bancário” no qual pequenas porções de informações seriam depositadas na mente dos educandos, em conformidade ao que acontece com dinheiro em uma conta.

Em Papert (1980/1985), percebe-se que o computador é uma ferramenta alternativa à demanda de um fazer educativo com bases mais sólidas, pois para ele o uso do computador pode

oferecer a seus usuários a possibilidade de pensar, refletir, expandir-se e, o principal: implementar suas ideias, por exemplo via linguagem de programação de computadores, LPC.

Existem inúmeras linguagens de programação, cada qual com características peculiares e propósitos diferentes, sendo que o seu uso em educação está presente em diferentes níveis e domínios do conhecimento. Dessa forma, é muito comum vermos cursos como os de matemática ou engenharias, entre outros, possuírem unidades curriculares de introdução à programação de computadores, embora, logicamente, sejam os cursos ligados à área da computação os que possuem uma maior carga horária dedicada à exploração de tais ferramentas. Geralmente, essas unidades curriculares exploram a atividade de resolução de problemas por intermédio das LPC. Valente (1993) esclarece que:

As linguagens para representação da solução do problema podem, em princípio, ser qualquer linguagem [...] No entanto, deve ser notado que o objetivo não é ensinar programação de computadores e sim como representar a solução de um problema segundo uma linguagem computacional. [...] Assim, como meio de representação, o processo de aquisição da linguagem de computação deve ser a mais transparente e a menos problemática possível. Ela é um veículo para expressão de uma idéia e não o objeto de estudo. (VALENTE, 1993, p.14)

Santos e Costa (2006) afirmam o seguinte:

As linguagens de programação constituem-se em uma ferramenta de concretização de produto de *software*, que representa o resultado da aplicação de uma série de conhecimentos que transformam a especificação da solução de um problema em um programa de computador que efetivamente resolve aquele problema. (p. 41)

Percebe-se, portanto, que o objetivo do uso das LPC é propiciar a constituição de um ambiente simbólico-operacional para a construção de programas.

A atividade cognitiva de construção de programas é dita programação de computadores. Ela exige o domínio de uma LPC por parte do programador, o que irá possibilitar a codificação do programa e seu processamento pelo computador; também requer o conhecimento da situação-problema abordada e criatividade, uma vez que uma solução pode ser expressa de diferentes maneiras. A construção de programas difere, portanto, da reprodução e memorização de informações e requer a formalização de raciocínio lógico, reflexão, dedicação à atividade e pesquisa em relação ao problema a ser modelado e também quanto à LPC adotada (MALTEMPI & VALENTE, 2000).

O processo DERD

Dada a relevância da interação estabelecida na atividade cognitiva de programação de computadores, vários autores – Valente (1993, 1999, 2005), Almeida (1999), Maltempi e Valente

(2000), Altoé e Penati (2005), Freire e Prado (1995) – descrevem-na em quatro etapas, a saber: **descrição** da resolução do problema nos termos da linguagem de programação, **execução** dessa descrição pelo computador, **reflexão** sobre o que foi produzido pelo computador e **depuração** dos conhecimentos por intermédio da busca de novas informações e da atividade cognitiva-reflexiva. Este processo será tratado pelo acrônimo “DERD”.

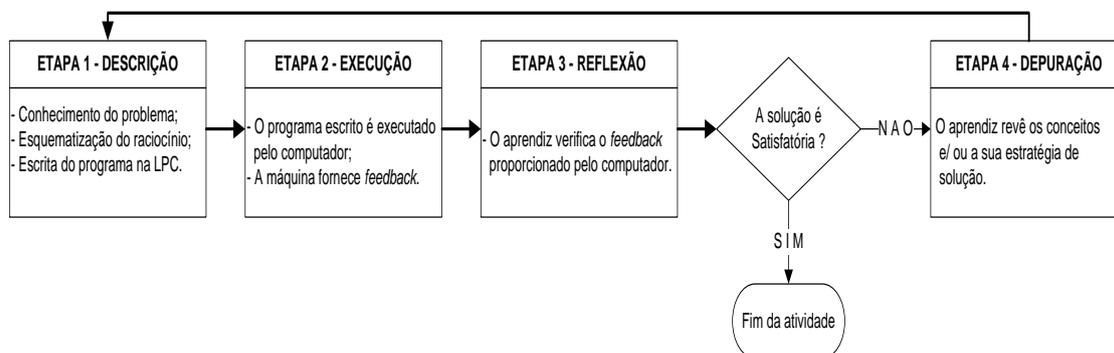


Figura 1 – Esquema do processo de DERD de ideias via LPC.

A Figura 1 sintetiza e esquematiza o conjunto de quatro etapas do processo de DERD. Esses estágios são interdependentes. Vale lembrar, entretanto, que são os estágios de reflexão (etapa 3) e depuração (etapa 4) que, quando bem explorados, possibilitam a concretização do conhecimento por parte do aluno. Isso acontece, essencialmente, quando o discente encontra uma situação não satisfatória relacionada a seu fazer. Papert (1980/1985, p. 39) afirma que “quando se aprende a programar um computador, dificilmente se acerta na primeira tentativa”. Almeida (1999, p.23) destaca que “o erro passa a ser então um revisor de ideias e não mais um objeto de punição, intimidação e frustração”. Da mesma forma, Valente (1999, p. 75) diz que “o processo de achar e corrigir um erro constitui uma oportunidade única para o aprendiz aprender sobre um determinado conceito envolvido na solução do problema ou sobre estratégias de resolução de problemas”.

Nota-se, portanto, que após o aluno receber o *feedback* do computador sobre a execução de seu programa, ele passa a tentar identificar a origem de um erro e saná-lo, ou ainda, empenhar-se na construção de melhorias em seu programa. Esse processo de depuração é, para o aluno, um momento de “pensar sobre o pensar” (TURKLE, 1984). A atividade de depuração na programação de computadores é valiosíssima do ponto de vista da aprendizagem e é exatamente por isso que deve ser estimulada. Assim, o processo de DERD, como um todo, configura-se como um exercício que demanda esforço, dedicação, concentração e motivação por parte dos alunos. Nesse ponto, torna-se imprescindível a atuação de um profissional da educação, que favoreça a aprendizagem dos alunos.

[...] o professor precisa compreender a representação da solução do problema adotada pelo aluno; acompanhar a depuração e tentar identificar as hipóteses, os conceitos e os possíveis equívocos envolvidos no programa; e assumir o erro como uma defasagem ou discrepância entre o obtido e o pretendido. Assim, o professor intervém no processo de representação do aluno, ajuda-o a tomar consciência de suas dificuldades e a superá-las; a compreender os conceitos envolvidos; a buscar informações pertinentes; a construir novos conhecimentos; e a formalizar esses conhecimentos. (ALMEIDA, 1999, p. 23)

Nessa perspectiva, constitui-se como uma ideia errônea, o pensamento de que, nessa dinâmica, o discente aprenderia sozinho; que basta apresentar-lhe um problema, colocá-lo na frente de um computador para que o processo de DERD se consolide. Muito pelo contrário. O aluno pode, por exemplo, incorrer numa situação em que ele não sabe um conceito – o que representaria uma estagnação do processo. Daí a importância do suporte a ser suprido pelo professor. Destaca-se que, além de elucidar e/ou sanar eventuais questionamentos do discente, o professor deve ser um estimulador do processo de DERD, mostrando-se disposto a cooperar e aprender em conjunto com seus alunos.

O ensino-aprendizagem de programação de computadores em questão

O processo de ensino-aprendizagem de programação de computadores é, portanto, complexo para docentes e discentes, sendo que os resultados alcançados têm se mostrado insatisfatórios. Dentre outros trabalhos que se dedicaram ao estudo da dinâmica educacional que envolve a aprendizagem de programação de computadores, comentamos, a seguir, os resultados de quatro pesquisas, com o objetivo de ampliar a percepção que temos de unidades curriculares de programação em cursos do Ensino Superior.

Martins e Correia (2003) apontam alguns motivos para a ineficácia do ensino de programação, destacando-se entre eles: as deficiências de formação observadas no Ensino Médio e Fundamental; o desconhecimento por parte dos ingressantes dos cursos de computação a respeito do perfil do aluno e das exigências inerentes à área; além do também ineficiente formato dos processos seletivos que não conseguem diagnosticar habilidades mínimas nos candidatos aos cursos superiores em questão.

Rodrigues Júnior (2002) aponta, fundamentalmente, a imprestabilidade do modelo de funcionamento tradicional das unidades curriculares de programação, ou seja: a apresentação teórica de conceitos e exemplos de aplicação seguida de listas de exercícios para solução. Essa abordagem não seria suficiente para motivar o envolvimento discente. Unido a isso, o autor reitera que existe o paradigma de que as unidades curriculares de programação de computadores são obstáculos acadêmicos dificilmente vencidos, e o pior: que alguns docentes não cooperam para diminuir tal impressão, contribuindo para aumentar tal negatividade.

Já Maltempi e Valente (2000) destacam que o ensino-aprendizagem de programação de computadores é extremamente significativo como processo educativo e de construção de conhecimentos aplicáveis. Entretanto, os autores relatam que a maioria dos alunos que chegam às universidades apresenta dificuldades de raciocínio lógico e criatividade – o que é refletido em seus desempenhos. Nesse sentido, o trabalho elucida a necessária postura ativa do aluno no processo e também a importância do professor enquanto mediador da aprendizagem. Apontam um modelo de construção de projetos contextualizados como mais indicado didaticamente para promover a aprendizagem da programação.

Finalmente, Pereira Júnior e Rapkiewicz (2004) relatam a importância das unidades curriculares de programação de computadores na formação acadêmica dos estudantes de computação. Também constata as dificuldades do processo de aprendizagem e indicam a motivação discente como fator indispensável a um melhor rendimento. A esse quesito associam o bom relacionamento professor-aluno, que ajuda na superação de desafios e acompanhamento de dificuldades particulares. Em seu trabalho, os autores mostram ainda divergências/convergências de diferentes metodologias utilizadas no processo educacional em questão, colocando a adequada mediação pedagógica como elemento capaz de amenizar as inúmeras dificuldades existentes.

INVESTIGANDO FATORES DE MOTIVAÇÃO/DESMOTIVAÇÃO DOS ALUNOS

Conforme estabelecido anteriormente, a experiência empreendida contou com a participação de 21 alunos formandos de um curso de Sistemas de Informação de uma universidade particular do interior do Estado de Minas Gerais. A pesquisa com os alunos foi iniciada com a aplicação de um questionário diagnóstico que enfocou, primordialmente, suas relações com as unidades curriculares de programação de computadores (UCPCs).

A síntese do perfil dos alunos envolvidos nas referidas UCPCs destacou um grupo que, ao longo de sua formação acadêmica, não aproveitou satisfatoriamente os conteúdos relacionados à programação de computadores. A relação com tais UC remontou a um sentimento de desmotivação, frustração, descontentamento e de pouca interação entre professor, aluno e ferramentas de *software*.

De posse desse diagnóstico, cinco alunos – aqui identificados pelos números 1, 2, 3, 4 e 5 – foram convidados para participar de uma entrevista individual, que buscou elucidar fatores que levariam à sua motivação/desmotivação no processo de construção de conhecimento via linguagem de programação. Destacam-se a seguir as percepções e concepções dos alunos, principais objetos da pesquisa empreendida.

A visão dos alunos sobre ensino-aprendizagem de programação

A aprendizagem de programação extrapola a transmissão, a repetição, a cópia de conteúdos e vem ao encontro de uma participação ativa, cooperativa e que exige dedicação dos praticantes. Ocorre que o ensino de programação de computadores enfrenta sérios problemas em relação ao rendimento alcançado por parte dos aprendizes. Esse fato coloca os sujeitos envolvidos na dinâmica educacional em uma situação de conflito entre o que é esperado em termos de construção de conhecimento e o que é realmente efetivado. Dois aspectos particulares da questão pedagógica foram apontados pelos Alunos 1, 5 e 2: a falta de introduções didatizadas³ aos temas e a fragmentação do conteúdo, como numa linha de produção fordista:

Aluno 1: *O primeiro momento que a gente teve contato com a programação, foi muito brutal, assim... Porque como você nunca teve contato... é difícil. É... sem uma preparação primeiro, falar mais ou menos: você precisa programar para quê? [...] E eu acho que as matérias ficaram faltando mesmo, pelo fato disso: de uma introdução menos agressiva e mais produtiva.*

Aluna 5: *Programação é lógica. E eu acho que muita gente não está acostumada a raciocinar com lógica. [...] eu acho que a primeira dificuldade foi isso: a forma pela qual foi passado [...] eu acho que teve uma deficiência de didática [do professor].*

Aluna 2: *[...] igual a uma charge que eu tinha visto: tinha um cara que fabricava parafuso, e aí ele passava esse parafuso para uma outra coisa, que seria feita com o parafuso. E aí um dia ele falou: “que será que fazem com esse parafuso?” E aí lá no final, tinha lá, um equipamento. Então a pessoa está fazendo aquilo ali, mas não sabe para quê que serve. Como que é? O quê que vai ser? O quê que o parafuso vai fazer? Vai fazer parte de quê? É de alguma coisa... mas o quê que é?*

Em suas falas, os alunos explicitaram seu descontentamento com a abordagem pedagógica utilizada no processo de ensino-aprendizagem de programação de computadores. Compreendeu-se que as referidas UCs exigiriam uma nova forma de se proceder com vistas à aprendizagem, constituindo-se uma abordagem que se introduz/desenvolve com aumento paulatino de complexidade, sempre articulada aos contextos de aplicação.

Os alunos e os fatores motivacionais ligados à aprendizagem de programação

A motivação de um aluno, em qualquer processo de aprendizagem, é fundamental para o seu sucesso. Guimarães e Boruchovitch (2004) afirmam que

³ A idéia de didatização aqui utilizada corresponde a “transposição didática”, no sentido proposto por Chevallard (1991), a qual compreende uma dinâmica de transformações/adaptações dos saberes de referência, com o objetivo de torná-los mais acessíveis e apropriados ao processo educacional. Tal transposição implica, em termos operacionais, dentre outras coisas, a complementação entre textos e figuras e entre conceituações e exemplos, a alternância de vozes entre professor e alunos, a proposição e discussão de exercícios.

um estudante motivado mostra-se ativamente envolvido no processo de aprendizagem, engajando-se e persistindo em tarefas desafiadoras, despendendo esforços, [...] buscando desenvolver novas habilidades de compreensão e de domínio (p. 143).

Os autores ampliam esse argumento elucidando que alunos motivados mostram-se entusiasmados durante a execução das tarefas propostas e que, ao concluí-las, revelam-se orgulhosos acerca dos seus desempenhos. No contexto do trabalho empreendido, o referido sentimento encontrava-se vinculado à atividade de programação de computadores e dizia respeito ao *empowerment*⁴ (sentimento de eficácia) manifestado – ou não – diante dos desafios de conceber e estruturar com sucesso um programa de computador.

Em um processo educacional é comum identificar alunos que variam entre dois pólos: aqueles que se aprofundam no conhecimento em busca de um embasamento que possa lhes valer no futuro – motivação intrínseca – e os que apenas se dedicam a cumprir o “regulamento”, ou seja, fazer provas e trabalhos visando à conclusão da unidade curricular com um resultado minimamente satisfatório – motivação extrínseca (Guimarães *et al*, 2002). Neste sentido, nas entrevistas empreendidas com os discentes, buscou-se determinar o que levou cada sujeito a escolher o curso de Sistemas de Informação. Destacam-se os trechos abaixo:

Aluno1: *[...] eu não queria fazer o curso [...] a partir do momento que eu fui e vi, que na minha turma tinha os meus amigos, que eu poderia dar certo me interessando mais pelo curso, eu decidi fazer o curso de Sistemas de Informação. [...]Eu não queria fazer nenhum curso de tecnologia. Eu queria fazer na área de farmácia mesmo, mas como não tinha aqui [...] eu fiquei para fazer o curso de tecnologia mesmo.*

Aluna2: *E quando eu fui tentar o vestibular, na verdade, foi uma coisa assim... Não foi planejada: O ano que vem eu vou tentar Sistemas de Informação. Não! Foi uma coisa meio no estalo.*

Aluna3: *No começo eu não sabia nem o que era o curso. Ai, depois, eu comecei a pegar gosto pelo negócio [...] Não foi fácil. Muitas matérias eu tive dificuldade: programação e banco de dados [...]*

Ficou evidente a falta de clareza e afinidade na escolha dos educandos pela área da computação; os alunos não apresentavam um conhecimento mais claro a respeito do que o Curso escolhido lhes exigiria. Os alunos não estavam certos do que viria ao longo de suas formações, ou ainda, não conheciam a estrutura e o perfil do egresso de um curso de Sistema de Informação.

Compreendeu-se, portanto, que nos casos relatados, o prazer de manipular a tecnologia foi dificultado, uma vez que os sujeitos envolvidos não estavam em um processo que lhes era

⁴ O termo *Empowerment* foi introduzido por Valente (1999) e assume um sentido associado à sensação que é experimentada por um discente ao lograr êxito na elaboração/construção de um produto, com a compreensão das etapas diretamente associadas.

plenamente afim. Isto foi entendido como uma das prováveis origens do problema com a programação. O simples “gostar da área” precisaria evoluir para uma dimensão mais ampla: a da dedicação e pesquisa, a do prazer em usar a tecnologia e, finalmente, a do aprender a programar um computador. Em outras palavras: uma escolha mais consciente e intrinsecamente motivada levaria os sujeitos a terem maiores condições de alcançar melhores níveis de aprendizado, desempenho e interesse pela área.

Não se pode omitir que o processo educacional é norteado, predominantemente, pela abordagem tradicional de ensino (MIZUKAMI, 1986) e que este modelo acompanhou os sujeitos desde os primeiros momentos em que foram alfabetizados. Dessa forma, ao longo de suas formações, os alunos não foram instigados a expressar suas ideias de forma lógica, ao trabalho cooperativo e à resolução de situações problema. Isto foi constatado quando os mesmos foram questionados a respeito da postura desejada de um aluno de programação:

Aluno 1: *O aluno ele tem que ter bastante interesse. Porque não é fácil mesmo não. Não é uma matéria que você vai ler e vai entender logo, logo. Não é igual a você ler um texto. Você tem que ler e procurar assimilar o que vai fazer e juntar com lógica, com matemática e outras coisas mais.*

Aluna 3: *Você tem que correr atrás, não pode ficar esperando só de quem está ensinando. Acho que faltou, até de mim mesmo, pelo fato de ter dado uma esfriada, uma desanimada, deixou um pouco de lado aquele negócio de todo dia sentar, entendeu?*

Aluna 2: *[...] acho que você também tem que fazer a sua parte. Não adianta. Achar que o professor vai te ensinar tudo. Não adianta.*

Aluna 5: *O aluno tem que ter no mínimo interesse em aprender. [...] se o aluno não quer aprender ele não vai aprender mesmo. Ele [aluno] tem que correr atrás e procurar estudar também.*

Nos moldes tradicionais de ensino, não existe estímulo à pesquisa, ao debate e ao posicionamento crítico. Os conteúdos trabalhados são retalhados e tratados desconexamente. Percebeu-se, portanto, nesses fatores parte da justificativa do choque sofrido pelos educandos envolvidos com a programação de computadores, que a tudo isto requer. Ficou, portanto, a cargo dos alunos (re)agirem e (re)direcionarem a sua forma de estudar, assumindo uma nova postura frente à aprendizagem.

Entretanto, o aluno de programação pode ter a oportunidade de experimentar uma sensação de *empowerment*, ou seja, deliciar-se com o prazer e com o orgulho de ter conseguido construir algo ligado ao seu interesse, a algum campo de aplicação e que compreende desde o planejamento inicial até a implementação final. O *empowerment* prevê o engajamento discente, o que se traduz no interesse pela prática de programação. Contudo, se um sujeito se envolve em um

meio que não lhe gera satisfação e interesse e que não o instrumentaliza para sua ação, ele não tem ali nem os instrumentos e nem a motivação para se realizar.

Nessa dinâmica de (re)adequação do papel do aluno para o sucesso com a programação, a dimensão motivadora se faz indispensável, no sentido de equilibrar o esforço empreendido, os resultados futuros, a efetivação da aprendizagem e a construção do conhecimento. Onde, na concepção do aluno, estaria(m) o(s) elemento(s) motivador(es) desse processo de ensino-aprendizagem?

Aluna3: *Eu acho que primeiro o professor teria de motivar mais o aluno. Procurar ver uma forma mais fácil, mais simples de atender cada pessoa. Porque cada um tem um grau de dificuldade diferente. E o aluno também correr atrás.*

Aluna5: *Eu acho que para motivar, o fundamental, seria ter um professor que tivesse uma boa didática. Entendeu? Desde o primário, eu nunca gostei de matemática, aí quando eu mudei de colégio, eu tive uma professora que explicava e eu fui entendendo tudo. Eu acabei sendo uma das melhores alunas de matemática no final do ginásio. Era a forma como ela explicava. [...] Não é que eu não gostasse de matemática, quem passava a matéria, talvez não soubesse transmitir direito.*

Aluna2: *Se o professor conseguir entender, que ele precisa mostrar para o aluno primeiro o que vai acontecer, pra depois ele ensinar o caminho que o aluno vai ter que percorrer pra chegar até lá, é uma forma de motivar o aluno. E aí o quê que acontece? Com isso, o grupo todo vai ficar motivado, porque você quer fazer, quer mostrar para o outro: “Olha eu consegui!”, mas como é que você fez? E você ensina pro outro... é mais uma forma de você aprender e de você ficar mais feliz. “- Nossa eu consegui e eu ainda consigo passar!”. Não adianta você aprender e depois não conseguir explicar nem o que você está fazendo.*

Os alunos colocaram o professor num lugar especial no processo de ensino-aprendizagem de programação. Além de ser o responsável por estruturar e criar as condições para a apresentação dos conteúdos programáticos, os alunos esperavam que a forma de conduzir o processo de aprendizagem extrapolasse a instrução direta, sendo necessária a incorporação de uma dimensão didática no seu fazer, objetivando a motivação e o envolvimento discente. Em nossa avaliação, isso implicaria, dentre outras coisas, a complementação entre textos e imagens e entre conceituações e exemplos, a alternância de vozes entre professor e alunos, caracterizando situações de diálogo, debate, ressignificação e *feedback* (PAPERT, 1985; PAPERT, 2008). Este indicativo encontra respaldo no trabalho de Guimarães e Boruchovitch (2004), o qual preconiza que a motivação intrínseca do aluno não é o resultado direto da instrução. Entretanto, a motivação pode ser influenciada pelas ações dos professores:

[...] alunos de professores com estilo motivacional promotor de autonomia demonstram maior percepção de competência acadêmica, maior compreensão conceitual, melhor desempenho, perseveram na escola, aumentam sua criatividade para as atividades escolares, buscam desafios, são emocionalmente mais positivos,

menos ansiosos, buscam o domínio e são mais intrinsecamente motivados [...].
(GUIMARÃES e BORUCHOVITCH, *op. cit.*, p.148.)

A ação docente apresenta-se, portanto, como um importante elemento motivacional inerente ao processo de ensino-aprendizagem de programação de computadores.

Também os Alunos 1 e 4 destacaram o professor como elemento motivador:

Aluno 1: Às vezes o fato de um **professor** ter uma didática interessante, às vezes poderia ser uma motivação. Ele próprio ser um motivado para explicar. E a gente, ia se tornar com isso. Porque um professor infeliz, ele tem alunos infelizes. Agora, um professor que é alegre e dá aula com felicidade, todo mundo ia chegar e falar: “A aula daquele cara é super produtiva. Ele conversa, ele explica e eu entendo. Tudo que ele faz é interessante para aula. O que ele traz ele estudou, e a gente sabe que ele está dando. E a gente vai em casa estudar para acrescentar na aula do professor”. Isso ia ser uma motivação interessante.

Aluno 4: É... Por mais que eu buscasse idéias, condições junto com os colegas, a faculdade nunca dava condições para a gente. Porque profissionais bons, **professores** da área de sistemas de informação existem por aí. Só que a maioria deles já tem o seu vínculo no mercado. E uma vez que precisando buscar profissionais, aqueles que foram encontrados, eles não tinham uma didática, assim... Não tinham uma experiência boa [...] eu não estou culpando os professores somente. Eu acho que faltou um pouco mais de ajuda da faculdade. Se tinha uma deficiência vamos bater em cima dela.

Os discursos dos alunos mostram o professor e sua forma de atuação como fatores motivacionais para a aprendizagem de programação. O Aluno 4 acrescentou ainda a necessidade de uma estratégia administrativa na seleção dos professores. Apesar da consciência de que deveriam fazer a sua parte, suas falas indicam que depositavam na figura do professor as esperanças de apoio na construção/efetivação do conhecimento.

UM PERFIL PARA O PROFESSOR DE PROGRAMAÇÃO

Diante da ênfase posta na figura do professor, buscou-se conhecer as características desejáveis para o profissional que trabalha com a docência de programação. Afinal, qual seria o perfil desse profissional?

Aluno 1: Uma pessoa que seja capaz de estudar e entender por completo aquilo que ele está querendo passar. Não que ele tenha que entender tudo, mas pelo menos aquela matéria que ele está dando no dia, procurar saber todas as saídas para uma explicação, para uma pergunta que um aluno fizer, ou uma dúvida... Ele dever saber exemplificar e fazer a pessoa entender naquele momento. Não deixar a pessoa pensar e depois ver o que vai acontecer com ela. [...] Eu acho que um professor [...] tem que ser uma pessoa, que assimila as dificuldades de todo mundo. Porque a partir do momento que tem algumas pessoas tem a facilidade com a programação, outros têm as dificuldades delas. Mas nem por isso elas estão desinteressadas. Elas ficam desinteressadas a partir do momento que aquilo vai acumulando e chega ao ponto dela já perder o fio da meada.

Aluno 4: Um bom professor na área de programação, a primeira coisa, é conhecer as ferramentas. Ele tem que ter domínio da sua ferramenta. [...] [o professor] pega uma ferramenta, estudou sobre ela, ouviu

dizer dela, quis aprendê-la primeiro, para depois jogar para os alunos. Os alunos vão trabalhar com ela. [...] Os professores [...] tinham um conhecimento técnico da ferramenta, mas eles não tiveram um conhecimento didático para passar para os alunos essa técnica deles.

Os Alunos 1 e 4 destacaram a necessidade do domínio dos assuntos tratados, bem como a capacidade de tornar os conteúdos inteligíveis a uma turma heterogênea, utilizando exemplos e atentando para as dificuldades que se estabelecem. Para eles, esse professor deveria ter afinidade com a área não apenas no sentido técnico, mas também no pedagógico. Em outras palavras, para dar aula de programação o professor precisaria dominar o conteúdo teórico, conhecer as linguagens e ambientes de programação (ferramentas) e, não menos, ter gosto e formação para o ofício docente. Pedagogicamente, a gerência das diferenças na aprendizagem, também é fator relevante. A Aluna2 e o Aluno4 ampliaram a discussão, introduzindo novos elementos:

Aluna2: [...] a primeira coisa é ter conhecimento. Tem realmente que ter embasamento daquilo que ele está falando. Não adianta falar para os cocos. Ele tem que estar falando aquilo que é a verdade. Segundo, acho que tem que ser um professor que tenha a vivência [...] Não adianta ele ensinar uma coisa que ele, realmente, não sabe fazer na prática. Porque é muito fácil... você falar...eu pego, dou uma lida numa revista, alguma coisa, comento na sala de aula ... faz aqui pra mim... não dá para fazer nesse sentido. [terceiro] aquele professor que tenta motivar o aluno, mostrar para o aluno que aquilo que vai ser ensinado para ele é legal, que vale a pena aprender, que produz alguma coisa, porque eu acho que não tem nenhum conhecimento, em área nenhuma, que não produz alguma coisa. Que dê um resultado.

Aluna3: Eu acho que o professor tem que estar mais preparado para estar ensinando ao aluno. [ou seja, que prepare uma] aula que incentive, que motive, que chame mais atenção do aluno, que seja uma aula mais clara, mais fácil de entender. Que ele procure mostrar de uma forma mais simplificada.

O aspecto “vivência prática de programação” é um elemento desejável ao professor. Compreendeu-se que esse atributo traria, para a sala de aula, a possibilidade de análise e contextualização do conteúdo por intermédio de situações práticas, advindas de fora do ambiente acadêmico. Mais uma vez, percebeu-se que os fatores motivacionais dos alunos estão associados à atuação do professor na condução das UCs, que deveria ser clara e acessível a todos.

Uma vez traçado o perfil para o professor de programação, foi levantado junto aos alunos qual seria o seu modo de ação. Como esse professor deveria atuar a fim de favorecer a aprendizagem de seu grupo de alunos?

Aluna3: Entrou um professor novo que achava que a turma estava no mesmo nível que ele, então, tacava um negócio que ninguém entendia, dava a aula dele e acabou. [...] Até, a gente ficava bobo de ver, a turma inteira, o “Fulano” chegava... Ele tinha uma cabeça... Não pegava uma folha... passava tudo no quadro... a gente achava aquilo ali fantástico. Né? Mas com relação a ele poder estar passando a aula ali para a gente... Não tinha como, porque ninguém conseguia entender nada que o homem estava falando. Alguma coisa a gente conseguia entender, porque ele falava mais claro.

Aluna5: o próximo que entrou [professor novo] era um cara que eu acho que entende muito, uma cabeça muito rápida de raciocínio... e ele sabia muito, mas eu acho que ele não tinha didática nenhuma.

[...] sabia muito para ele, porque ele chegava despachava no quadro aquele raciocínio todo, aquela programação toda e ia passando, ia passando e ele não queria saber se você estava pegando ou não. Ele queria cumprir o planejado, mas independente do pessoal estar pegando ou não.

Nas declarações acima, configurou-se a “aula espetáculo”, no sentido de uma demonstração de domínio técnico do conteúdo do ementário. Ou seja: o professor funcionava ao molde de um equipamento pré-programado, como um computador, que vai a uma unidade de armazenamento de dados, busca um conteúdo X, segue um roteiro estrito para cumprir sua demonstração, não introduz novos elementos ao contexto e dá por terminado o processo. Esse é um caminho que, na visão dos alunos entrevistados, não deveria ser seguido em um ambiente de aprendizagem, uma vez que eles expressam sua admiração pela capacidade de reproduzir informação, mas contestam a forma de apresentá-las aos outros. A Aluna³ foi veemente ao afirmar que ninguém entedia o que se tentava passar. Percebeu-se a inexistência de preocupação com a forma de se processar o conteúdo programático e também em como adequá-lo ao nível dos alunos. Aponta-se, nesse sentido, uma abordagem exclusivamente instrucionista do uso do computador e, não menos, da atuação do professor no processo.

O tecnicismo predominante na Ciência da Computação, área de conhecimento na qual se enquadra a programação de computadores, parece exercer uma influência muito forte no comportamento rígido dos professores da área. Entendeu-se, que seria necessário o desenvolvimento de uma cultura mais leve (*softer*), em conformidade a Papert (1980/1985, 1993/2008), para se tentar promover o processo educacional. Ou seja: a rigidez imposta pelo formalismo matemático da computação precisaria ser adequada a uma forma menos abstrata, mais inteligível e atraente para o aluno⁵.

CONSIDERAÇÕES FINAIS

As TIC são elementos centrais da sociedade moderna. Sua consolidação, enquanto instrumentos transformadores e possibilitadores das ações cotidianas, influenciou e continua a influenciar todas as áreas de conhecimento. O convívio com tais tecnologias se impôs, provocando transformações nas formas de se proceder em diversos campos de atividades humanas, incluindo a

⁵ O matemático Seymour Papert idealizou e construiu, durante a segunda metade dos 60 do século passado, um *hardware* – a tartaruga robô – que recebe e executa ordens de um usuário via *software* – a linguagem LOGO. A tartaruga desloca-se em uma superfície plana, deixando um rastro que forma uma figura. A programação com a linguagem LOGO se assemelha à construção de figuras geométricas seguindo os comandos nativos da ferramenta. Ao criarem programas em LOGO, os alunos assumem a postura ativa mediante à construção de suas aprendizagens. O professor abandona o status de transmissor de conceitos ao buscar compreender as ideias de seus alunos, intervindo no processo no sentido de provocar/cooperar com a construção da aprendizagem/conhecimentos. (LIMA, 2009 ; LIMA e LEAL, 2010).

educação. Este trabalho constatou a complexidade própria do processo de ensino-aprendizagem de programação e buscou sua melhor compreensão.

Com a ajuda dos discentes foi possível perceber que, mesmo com a introdução do computador no processo educacional, criando-se as condições para a instauração de uma nova perspectiva na aprendizagem (contextualizada e motivadora), a fragmentação dos assuntos e a falta de didatização continuam operando e dificultando a aprendizagem.

Conforme já estabelecido, o perfil do grupo de alunos que cooperaram com o presente trabalho indicou um baixo aproveitamento nas UCs de programação. Percebeu-se entre eles um sentimento de desânimo e mesmo de aversão frente às oportunidades do contato com a programação de computadores. Dessa forma, para eles, a efetivação da aprendizagem foi prejudicada. Dois indicativos colhidos relativos ao problema motivacional nas referidas UCs são a baixa afinidade dos educandos com a área da tecnologia e o desconhecimento dos discentes a respeito do currículo do curso de Sistemas de Informação, o qual tem na programação elemento central de estudos. Em outras palavras: uma escolha mais bem informada levaria os sujeitos a terem maiores condições de alcançar melhores níveis de aprendizado, desempenho e interesse pela área de programação.

A atividade de programação de computadores é complexa: exige dedicação e envolve raciocínio lógico, capacidade de abstração, concatenação de ideias e conhecimentos prévios. Esses pré-requisitos precisam ser explorados e incentivados desde a formação de Ensino Fundamental e Médio, consolidando-se ao longo do Ensino Superior. O desenvolvimento gradual dessas habilidades poderia ajudar a diminuir as dificuldades dos alunos frente às unidades curriculares de programação de computadores e suas exigências.

Da mesma forma, o processo de ensino-aprendizagem de programação exige uma contrapartida dos docentes. Compreendeu-se que um dos principais fatores motivacionais da aprendizagem é a forma de condução das UCs pelos professores. A didatização, o cuidado com as dificuldades de aprendizagem, a contextualização dos conteúdos, a adequação da linguagem utilizada nas exposições, o preparo técnico e pedagógico e a vivência prática na área de programação por parte dos docentes são características que foram entendidas como necessárias para uma melhor mediação da aprendizagem. Indiscutivelmente, a docência de programação é uma atividade que exige o conhecimento técnico e formal dos conteúdos. Ela demanda, igualmente, uma postura pedagógica que estruture e viabilize a aprendizagem dos envolvidos. Revela-se, portanto, um processo plural e que extrapola tecnicismos.

REFERÊNCIAS

ALMEIDA, Maria Elizabeth Bianconcini. *Informática e Formação de Professores*. Coleção Informática para mudança na Educação. MEC/ SEED/ ProInfo, 1999.

ALTOÉ, Anair; PENATI, Marisa Morales. O Construtivismo e o Construcionismo Fundamentando a Ação docente. In: ALTOÉ, Anair; COSTA, Maria Luiza Furlan; TERUYA, Teresa Kazuko. *Educação e Novas Tecnologias*. Maringá: Eduem, 2005, p 55-67.

BARDIN, Laurence. *Análise de conteúdo*. Lisboa: Edições 70, 1977.

CHAVES DE CASTRO, et al. Utilizando Programação Funcional em Disciplinas Introdutórias de Computação. *XI Workshop de Educação em Computação - WEI 2003*. Campinas, SP, Brasil.

CHEVALLARD, Yves. *La transposición didáctica: del saber sabio al saber enseñado*. Buenos Aires, Aique, 1991.

DELGADO, C. et al. Uma Abordagem Pedagógica para a Iniciação ao Estudo de Algoritmos. *XII Workshop de Educação em Computação (WEI'2004)*. Salvador, BA, Brasil.

FREIRE, Fernanda Maria Pereira; PRADO, Maria Elisabette Brisola Brito. Professores Construcionistas: A Formação em Serviço, in *Anais do VII Congresso Internacional Logo e I Congresso de Informática Educativa do Mercosul*. Porto Alegre, RS, LEC/UFRGS, 1995.

GUIMARAES, Sueli Édi Rufini; BORUCHOVITCH, Evely. O estilo motivacional do professor e a motivação intrínseca dos estudantes: uma perspectiva da Teoria da Autodeterminação. In *Psicologia Reflexão e Crítica*, Porto Alegre, v. 17, n. 2, 2004. p.143-150. Disponível em <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-79722004000200002&lng=en&nrm=iso>. Acessado em 17/06/2009.

GUIMARAES, Sueli Édi Rufini.; BZUNECK, José Alonso; SANCHES, Samuel Fabre. Psicologia educacional nos cursos de licenciatura: a motivação dos estudantes. In *Psicologia Escolar e Educacional*. Jun. 2002, vol.6, no.1, p.11-19. Disponível em: <http://pepsic.bvs-psi.org.br/scielo.php?script=sci_arttext&pid=S1413-85572002000100002&lng=pt&nrm=iso>. Acessado em 12/06/2009.

LIMA, Márcio Roberto de. *Construcionismo de Papert e ensino-aprendizagem de programação de computadores no ensino superior*. 141p. Dissertação (Mestrado em Educação). Universidade Federal de São João del-Rei, 2009.

LIMA, Márcio Roberto de. LEAL, Murilo Cruz. [Uso da linguagem logo no ensino superior de programação de computadores](#). *Revista Pindorama*: Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA), Eunápolis, BA, 2010. Disponível em: <<http://www.revistapindorama.ifba.edu.br/files/MARCIO%20ROBERTO%20DE%20LIMA%20UFVJM.pdf>>. Acesso em 7/12/2010.

LÉVY, Pierre. *Cibercultura*. São Paulo: Editora 34, 1999.

MALTEMPI, Marcus V., VALENTE, José Armando. Melhorando e Diversificando a Aprendizagem via Programação de Computadores. In: *International Conference on Engineering and Computer Education - ICECE 2000*. Proceedings (CDROM). São Paulo, Agosto/2000.

MARTINS, Sheila Wesley; CORREIA, Luiz Henrique de Andrade. O Logo como Ferramenta Auxiliar Desenvolvimento do Raciocínio Lógico – um Estudo de Caso. *International Conference on Engineering and Computer Education - ICECE 2003*, Santos. Acessado em 28-12-2004. <http://www.inf.ufsc.br/~scheila/icece2003.PDF>

MIZUKAMI, M. da G.N. *Ensino: as abordagens do processo*. São Paulo: EPU, 1986.

MORAES, Maria Cândida. *O paradigma educacional emergente*. 11. ed. Campinas: Papirus, 1997.

PAPERT, Seymour. *Logo: Computadores e Educação*. Brasiliense, São Paulo, 1985.

_____. *Constructionism: A New Opportunity for Elementary Science Education. A proposal to the National Science Foundation*. Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group, Cambridge, Massachusetts. 1986.

_____. *A Máquina das Crianças: Repensando a Escola na Era da Informática*. Ed. rev. Porto Alegre, Artes Médicas, 2008.

PEREIRA, J.; ROCHA, J. C.; RAPKIEWICZ, C. E. O Processo de Ensino e Aprendizagem de Algoritmos e Programação: Uma Visão Crítica da Literatura. In: *III Workshop de Educação em Computação e Informática do Estado de Minas Gerais*, 2004, Belo Horizonte, 2004.

PERRENOUD, Philippe. *10 Novas competências para ensinar*. Porto Alegre: ArtMed, 2000.

PETRY, Patrícia Gerent. *Um sistema para o ensino e aprendizagem de algoritmos utilizando um companheiro de aprendizado colaborativo*. 82p. Dissertação de mestrado (Ciência da Computação). Universidade Federal de Santa Catarina, 2005.

RODRIGUES JÚNIOR, Methanias Colaço. *Como Ensinar Programação?* Informática - Boletim Informativo Ano I nº 01, ULBRA. Canoas, RS, Brasil, 2002.

SANTOS, Rodrigo Pereira; COSTA, Heitor Augustus Xavier. Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática. *INFOCOMP – Journal of Computer Science*, Lavras/MG – Brasil, v. 5, n. 1, p. 41-50, 2006.

SCHULTZ, Max Rúbens de Oliveira. *Metodologias para Ensino de Lógica de Programação de Computadores*. Monografia de Especialização (Ciência da Computação). Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, Brasil. 2003, 69p.

TURKLEY, Sherry. *The Second Self: Computers and the Human Spirit*. Simon and Schuster, New York, 1984.

VALENTE, José Armando. Diferentes usos do computador na educação. In: *Computadores e conhecimento: repensando a educação*. 1ª ed. Campinas, NIED-Unicamp, 1993.

_____. Análise dos diferentes tipos de softwares usados na Educação. In: Valette, J.A. (org). *O computador na sociedade do conhecimento*. Ed. Campinas: UNICAMP/NIED, 1999.

_____. Aspectos críticos das tecnologias nos ambientes educacionais e nas escolas. *Educação e Cultura Contemporânea: Revista do Programa de Pós-Graduação em Educação da Universidade Estácio de Sá*. Rio de Janeiro, v.2, n.3, p.11-28, 2005.