

Complementando o Aprendizado em Programação: Revisitando Experiências no Curso de Sistemas de Informação da USP

Luciano A. Digiampietri, Sarajane M. Peres, Fábio Nakano, Norton T. Roman,
Priscilla K. Wagner, Bárbara B. C. Silva, Beatriz Teodoro, Douglas F. P. da Silva
Jr., Guilherme V. A. Pereira, Guilherme O. Borges, Gustavo R. Pereira, Marcelo
V. dos Santos, Maruschia Baklizky, Vitor A. Barros¹

¹Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)
Av. Arlindo Béttio, Ermelino Matarazzo – 03828-000 – São Paulo – SP – Brasil

{digiampietri, sarajane}@usp.br

***Abstract.** The undergraduate Information Systems course at the University of São Paulo constantly seeks to improve the education it offers to its students, something that requires constant innovation and enrichment of the teaching-learning process carried out by its lecturers and students. In the search of such enrichment, lecturers and students have been making some efforts, amongst which are the experiences presented in this paper: the Programming Challenges course and the Programming Contest for First Years. Both experiences are focused on complementing the learning process of programming logic, algorithms and data structures – didactically difficult subjects, yet indispensable to a high quality technical education. In this paper we revisit and extend some analysis in these experiences.*

***Resumo.** O curso de Bacharelado em Sistemas de Informação da Universidade de São Paulo trabalha pela constante melhoria na formação que oferece para seus alunos, o que requer um trabalho contínuo de inovação e aprimoramento do processo de ensino-aprendizagem executados por seus professores e alunos. Na busca desta melhoria, os professores e alunos vêm realizando algumas ações, dentre as quais estão as experiências apresentadas neste artigo: as disciplinas de Desafios de Programação e o Campeonato de Programação para Calouros. Ambas estão focadas na complementação do aprendizado de lógica de programação, algoritmos e estruturas de dados -- assuntos difíceis do ponto de vista didático, mas imprescindíveis na formação técnica de qualidade. O presente artigo revisita e estende análises sobre essas experiências.*

1. Introdução

Formar cidadãos com habilidades e competências para contribuir com a sociedade é o objetivo de qualquer instituição ou indivíduo que se proponha a trabalhar com o ensino. A formação de qualidade de um aluno no nível universitário tem consequências positivas tanto no âmbito individual quanto no coletivo, na medida em que uma boa formação pode melhorar a posição social do indivíduo e potencializar sua capacidade de atuação na sociedade. Contudo, o processo educacional pelo qual se dá a formação dos alunos é custoso e, embora o benefício seja condizente, a preocupação com a eficiência

do processo deve ser constante: quanto mais rápido e melhor o indivíduo aprende, maior é o benefício que o processo gera.

O aprendizado de um indivíduo depende em grande parte de sua motivação [Maehr e Midgley 1991, Lumsden 1994] e neste contexto há os que, movidos pela curiosidade, se dedicam a estudar – esta fonte de motivação existe desde a infância. Na medida em que o indivíduo se desenvolve, somam-se outras fontes, como possibilidades de ganhos (monetário, social, cultural), aprovação na disciplina ou reconhecimento dos colegas, dos professores e da família. A força da motivação gerada em cada fonte depende da experiência individual (por exemplo, uma experiência negativa com algum colega pode levar o indivíduo a dar menos valor a esse tipo de reconhecimento).

No contexto universitário, cabe à instituição de ensino, na figura de seus professores e também de seus alunos, enriquecer o ambiente acadêmico com a criação de fontes de motivação, considerando os recursos disponíveis (tempo, infraestrutura, pessoal) e, principalmente, atendendo aos interesses dos alunos. Um desafio imerso em um ambiente de competição – um campeonato com equipes ou mesmo um problema ou exercício contextualizado em um assunto interessante – estão entre as atividades motivadoras para os estudantes.

O Bacharelado em Sistemas de Informação da USP enfatiza fortemente o aprendizado em lógica de programação e em algoritmos e estruturas de dados, assuntos reconhecidamente difíceis. Disciplinas que trabalham com estes assuntos têm altas taxas de reprovação, como discutido por Bergin e Reilly (2005) e Mahmoud e Swayne (2004). A fim de contribuir para a minimização dos problemas advindos das dificuldades inerentes ao ensino nessas disciplinas, aumentar a assimilação desse conhecimento e conseqüentemente aumentar a qualidade da formação dos alunos, nós, autores do presente artigo, trabalhamos durante os anos de 2010 e 2011 em duas iniciativas extracurriculares, o BxComp – Primeiro Campeonato de Programação para Calouros, uma atividade que combina o trabalho colaborativo (dentro das equipes) com uma motivação de competitividade lúdica (entre as equipes) para resolução de problemas, e as Disciplinas de Desafios de Programação I e II, que fazem uso de dinâmicas de resolução de exercícios similares às dinâmicas usadas em competições de programação. A eficácia das atividades foi avaliada por meio de questionários respondidos pelos alunos participantes.

O presente artigo revisita e estende o trabalho apresentado por Digiampietri et al. (2012), de forma a aprofundar a descrição e as análises delineadas na versão original. Além disso, as lições aprendidas são aqui consolidadas junto ao planejamento para a continuidade do trabalho. As discussões apresentadas estão focadas no uso de desafios de programação e dinâmica de competições como formas de estimular o aprendizado de conceitos de programação. Outras estratégias que são aplicáveis ao objetivo de melhorar o processo de ensino e aprendizagem nesse contexto estão fora do escopo deste artigo.

O restante deste artigo está organizado da seguinte forma: na Seção 2 é definido o problema abordado na iniciativa aqui descrita e são apresentadas algumas iniciativas correlatas ao contexto de experiências com ensino de programação e atividades de competição e afins; na Seção 3, características do currículo do curso de Sistemas de Informação são apresentadas, considerando o contexto pedagógico da escola onde está inserido, o contexto específico de Projeto Pedagógico do curso, com destaques para sua adequação a currículos de referência na área e para a inserção das disciplinas de

Desafios de Programação; também é apresentado na Seção 3 o Programa de Educação Tutorial do Ministério da Educação implementado no curso através do Grupo PET-SI, principal responsável pela idealização e implementação do Campeonato de Programação BxComp; cada uma das experiências, o campeonato e as disciplinas, são descritas na Seção 4; na Seção 5, as avaliações e resultados previamente publicados são reapresentados, acompanhados de novas análises referentes ao relacionamento das iniciativas extracurriculares de complementação do ensino de programação e o desempenho de aprendizado apresentado pelos alunos; as lições aprendidas são colocadas na Seção 6 e os direcionamentos futuros na Seção 7.

2. Apresentação do Problema e Iniciativas Correlatas

Motivações para a realização das experiências aqui discutidas surgiram na identificação, por parte de professores e alunos, da existência de possibilidades interessantes de prover atividades extracurriculares de fortalecimento para o ensino de programação dentro do curso de Sistemas de Informação na USP.

A fim de reproduzir tais motivações neste texto, esta seção apresenta uma breve discussão sobre: o problema identificado no contexto do ensino de programação; a abordagem metodológica que inspirou a construção e implementação das atividades e que é inerente ao projeto pedagógico da escola onde o curso de Sistemas de Informação está implantando (veja Seção 3); iniciativas correlatas à proposição do uso de desafios de programação e competição de programação na área acadêmica e na indústria de software.

2.1. Apresentação do Problema

A busca pela melhoria nos processos de ensino e de aprendizagem na área de programação de computadores é uma evidente preocupação atual, mas não necessariamente nova. Já há algum tempo, professores-pesquisadores vêm analisando o seu trabalho de ensino de programação a fim de levantar hipóteses sobre como o processo pode ser melhorado.

Uma seção especial do periódico *Communications of the ACM*, chamada *The challenges of Teaching Computer Programming* [Sleeman, 1986] discutia, já nos anos 80, os anseios da comunidade acadêmica em relação ao ensino de programação. Um destaque interessante dos trabalhos publicados nesta seção é uma iniciativa de modelar, em um sistema de produção em LISP, o comportamento dos alunos de programação [Anderson e Skwarecki, 1986]. Os autores deste trabalho destacam que a experiência de levantar os requisitos desta modelagem e observar os resultados obtidos com a simulação do modelo, os ajudou a entender ainda mais como se dá o processo cognitivo relacionado ao aprendizado de programação. Esta ferramenta pode ser considerada, ainda, uma das primeiras a apoiar, em algum nível, o trabalho do educador da área mais básica da computação.

Ainda nos anos 80, Edsger W. Dijkstra apresentou os desafios do ensino da disciplina de programação destacando que, eventualmente, a prática do ensino nessa disciplina deveria transcender as metodologias utilizadas por disciplinas clássicas (como lógica e matemática) de forma a se adequar às novas características apresentadas por esta nova área de ensino [Dijkstra, 1989].

A dificuldade no aprendizado de lógica de programação está muitas vezes relacionada à dificuldade dos alunos em entenderem abstração; à complexidade inerente da disciplina; à formação insuficiente dos alunos em matemática e lógica; bem como na maneira como esta disciplina é ministrada [Gomes e Mendes, 2007; Jenkins, 2002].

Como rapidamente ilustrado nesta seção, dificuldades observadas pelos alunos, no aprendizado, e pelos professores, no ensino, dentro das disciplinas que estão intimamente ligadas à programação, são vivenciadas em várias instituições de ensino, em diferentes culturas. Especificamente, no caso da vivência relacionada ao contexto dos autores deste artigo (professores e alunos), percebe-se algumas situações que elucidam a existência de um problema que merece atenção:

- as disciplinas que se destinam a ensinar programação, possuem um índice de reprovação e/ou evasão alto que merece atenção especial, principalmente por serem disciplinas que fornecem o conhecimento básico para que o aluno se desenvolva bem em outras disciplinas mais avançadas do curso e, frequentemente, se constituem como pré-requisitos¹ de disciplinas avançadas;
- nas disciplinas mais avançadas, onde conceitos básicos de lógica de programação e estrutura de dados são necessários, alunos sentem alguma dificuldade em conectar os conceitos aprendidos nas disciplinas iniciais com as necessidades de resolução de problemas atuais;
- os alunos, principalmente os primeiros anos, possuem tempo livre durante a semana (já que ainda não estão envolvidos em trabalhos de iniciação científica e nem em estágios extracurriculares). Há, então, espaço para preenchimento deste tempo com atividades diretamente ligadas ao aprendizado, somado à possibilidade de aprendizado com convivência acadêmica.

Neste contexto, foi desenhado um cenário que propicia a melhoria do processo de ensino-aprendizado em programação por meio da realização das atividades extracurriculares discutidas aqui.

2.2. Aprendizado Baseado em Problemas

A metodologia de Aprendizado Baseado em Problemas (PBL - do inglês *Problem Based Learning*) destaca como meta no processo de ensino-aprendizagem dois fatores: a inclusão da prática do ensino em um ambiente educacional altamente propício à aproximação da teoria à sua aplicação prática [Savery, 2006]; o posicionamento do aluno em um papel protagonista no processo [Mergendoller, Maxwell e Bellisimo, 2006].

Um breve histórico sobre os conceitos filosóficos e pedagógicos que inspiram essa metodologia, bem como uma revisão sobre questões referentes à comparação desta metodologia com o ensino tradicional, podem ser encontrados em Savery (2006) e em Jolly e Jacob (2012).

Já Prince (2004) apresenta uma revisão crítica sobre metodologias ativas, incluindo PBL, em suas discussões. Esse autor contextualiza suas análises no ambiente de ensino da Engenharia, destacando que dada a natureza da prática de ensino já

¹ Reprovações nestes pré-requisitos predem os alunos nos anos iniciais do curso prejudicando tanto o próprio aluno como o fluxo de alunos pelas séries do curso.

existente na área, a qual usa extensivamente as práticas em laboratório, emergem algumas confusões sobre o que é ou não a prática do PBL. A fim de proporcionar uma forma de esclarecer esta questão, essa revisão faz referência a uma série de trabalhos onde definições e discussões teóricas elucidam o que se entende por metodologias ativas.

A discussão realizada no artigo de Prince (2004) pode ser mapeada para a área de ensino de programação, pois, assim como no ensino de Engenharia, as abordagens práticas em laboratório e, posteriormente na continuidade do aprendizado nos cursos da área de Computação, as abordagens de desenvolvimento de sistemas, tradicionalmente usadas pelos professores, se assemelham aos princípios de aplicação de PBL.

Classicamente, a aplicação de PBL no processo de ensino-aprendizagem envolve alguns passos básicos que incluem fases² de coleta de informação e de estudo de conceitos que devem durar alguns dias, e também a produção de um relatório final por parte dos alunos. No entanto, para a abordagem de aplicação de PBL discutida aqui, os problemas a serem resolvidos são de resolução rápida, em poucas horas, porém o conjunto de problemas apresentados aos alunos no decorrer de toda a atividade (durante o semestre) suportam a construção do aprendizado e sedimentação de conceitos como o desenvolvimento de uma habilidade (programar).

Jolly e Jacob (2012) reforçam, com base em suas leituras, que com o uso de PBL, problemas a serem resolvidos são oferecidos aos alunos como uma alternativa a aulas expositivas; que a metodologia incentiva que os alunos trabalhem em times para resolver o problema; e proporciona que eles retenham o conhecimento por eles produzido (ou descoberto ou adquirido) durante a experiência. Essas características vem ao encontro da natureza das atividades aqui discutidas: a resolução de desafios de programação nas aulas das disciplinas de Desafios de Programação e nas etapas do Campeonato de Programação contextualizam o uso dos conceitos de programação em situações rotineiras e práticas, como será detalhado na Seção 4.

O'Grady (2012) apresenta uma revisão sistemática contextualizada na aplicação de PBL em disciplinas ministradas em cursos de graduação e de pós-graduação da área de Computação. É interessante notar que em sua revisão o autor identifica que a disciplina onde mais se usa esta abordagem é a disciplina de Programação.

Pereira, Zebende e Moret (2010) relatam uma experiência de aplicação de PBL em programação de uma forma interdisciplinar. Os autores aplicaram PBL propondo aos grupos de alunos a resolução de um problema que envolvia três áreas do conhecimento: Lógica Computacional da Máquina de Turing; (b) conceitos de Teoria de Fractais; e (c) Autômatos Celulares Regulares. Os autores do artigo relatam que inicialmente a experiência não foi bem recebida pelos alunos, uma vez que estes manifestaram-se temerosos em relação a um potencial fracasso na atividade que exigia muitos conhecimentos ainda não adquiridos por eles. No entanto, no decorrer da experiência, a aquisição de novos conhecimentos que possibilitavam a resolução do problema motivou os alunos, os quais se engajaram e obtiveram sucesso em seus projetos e conseqüentemente no aprendizado dos conceitos.

² Os sete passos de “Maastricht”, pressupõe que o processo de aplicação de PBL, e consequente resolução do problema proposto no processo, será executado em algumas semanas ou meses.

2.3. Iniciativas correlatas - desafios de programação e competições

Atualmente, com a evolução dos recursos oferecidos pela própria área de computação e também motivadas pelo amadurecimento e disseminação da arte de programar, as iniciativas em relação à infraestrutura para apoiar abordagens de ensino perpassam pelos mais diversos tipos de recursos, desde o oferecimento de sistemas de tutoria, que ajudam professores a propor exercícios de programação e acompanhar o trabalho do aluno em seu processo de aprendizado [Queirós e Leal, 2012]; sistemas que assistem o aprendizado do aluno, analisando o seu ritmo e aprendizado e elevando automaticamente o nível de dificuldade dos exercícios a ele oferecidos [Nishimura, Kawasaki e Tominaga, 2011]; até o massivo uso da infraestrutura da tecnologia para maximizar o interesse do aluno pelo que está aprendendo, como em Wang e Feng (2010) onde os autores defendem uma prática de ensino de programação orientada aos desafios de programação disponíveis em repositórios na internet, suportados pelos ambientes de Julgamento Online.

Nesses sistemas estão armazenados milhares de problemas e suas respectivas soluções. Sempre que um usuário (geralmente um estudante em treinamento) envia sua própria solução, o sistema realiza o teste automático sobre ela, aprovando-a ou não. Estes sistemas estão disponíveis ininterruptamente. Talvez o mais antigo e maior deles seja o *UVa Online Judge*³, funcionando desde 1995 e atualmente com mais de 3.500 problemas. A análise dos dados coletados pelo sistema indica que estudantes em treinamento vão se tornando mais eficientes na solução de problemas, segundo estudo apresentado por Revilla, Manzoor e Liu (2008).

Competições de programação estão se tornando cada vez mais populares e têm sido realizadas não simplesmente como um “jogo”, mas como ferramentas importantes aplicadas em processos de ensino e aprendizagem, tanto na área acadêmica (para formação básica de profissionais e para motivação ao desenvolvimento de pesquisas), quanto na indústria (neste caso, como forma de treinamento e desenvolvimento de novas habilidades ou como seleção de funcionários).

Dentro da área acadêmica, as competições em áreas como linguagem de programação [Nishimura, Kawasaki e Tominaga, 2011; Boersen e Phillipps, 2006; Halim e Halim, 2010; Anton-Rodriguez, 2011], e projeto e desenvolvimento de software [Dagiene, 2006; Fantinato et al., 2011; Paulik e Krishnan 2001] estão consolidando-se como uma abordagem efetiva para o desenvolvimento de habilidades na área de computação. Em especial, a comunidade acadêmica de computação já realiza painéis de discussão [Hare, Cain e Cigas, 2011] e tutoriais de orientação [Baber e Hoelzeman, 2011] relacionados à organização de competições de programação, evidenciando que a preocupação com essa área está cada vez mais ganhando espaço.

Dentre as competições de programação mais populares e já sedimentadas estão a *International Olympiad in Informatics* (IOI) – voltada para alunos de ensino médio e ingressantes no nível superior; e a *ACM-International Collegiate Programming Contest* (ACM-ICPC), cujo público são alunos de graduação e ingressantes na pós-graduação. No Brasil, existem eliminatórias para essas competições: a Olimpíada Brasileira de Informática⁴ e a Maratona de Programação⁵, respectivamente.

³ <http://uva.onlinejudge.org/>

⁴ <http://olimpiada.ic.unicamp.br/>

⁵ <http://maratona.ime.usp.br/>

Do ponto de vista da indústria os principais destaques para o uso de competição de programação são: *Google Code Jam* (da Google), *Facebook Hackers Cup* (do Facebook) e *HP Code War* (da Hewlett Packard). Essas competições são promovidas pelas empresas com o objetivo de selecionar pessoas para compor seus quadros de recursos humanos. Vale ainda um destaque para a *TopCoder*⁶ que gerencia competições orientadas a negócios, fornecendo uma plataforma onde um projeto pode ser desenvolvido por competidores. Nas competições promovidas pela TopCoder, vence quem consegue resolver uma tarefa da melhor maneira.

Existem ainda as competições que visam a estimular pessoas em todo o mundo a resolver um problema real, ou seja, na existência de problemas realmente complexos, organizações e empresas os expõem a um grande número de pessoas e efetivamente criam uma maneira eficiente de quebrar barreiras da resolução dos problemas. Este é o caso discutido por Goldbloom (2010). Neste contexto, a plataforma Kaggle⁷ é uma das mais populares. Ela apresenta conjuntos de dados e problemas provenientes de empresas, instituições governamentais e de pesquisa, para que pessoas de todo o mundo produzam os melhores resultados. Premiações de alto valor monetário são comuns nas competições sediadas nesta plataforma.

3. O Curso de Sistemas de Informação

O curso de Bacharelado em Sistemas de Informação (SI) é oferecido pela Universidade de São Paulo na Escola de Artes, Ciências e Humanidades (EACH) desde 2005. A EACH foi formada, também em 2005, com o oferecimento de dez cursos de graduação (1020 vagas para novos graduandos a cada ano), de diferentes áreas e com a proposição de oferecer um ambiente de ensino, pesquisa e extensão interdisciplinar. A escola é gerida dentro de uma estrutura administrativa e pedagógica que permite alta interação entre profissionais e alunos das diversas áreas de conhecimento, proporcionando um ambiente propício para a interação e desenvolvimento de trabalhos interdisciplinares. Dentre as prerrogativas do projeto de formação da escola está o forte estímulo ao uso da metodologia PBL, no âmbito das mais diversas disciplinas dos cursos sediados nela. Como uma forma de iniciar esta prática nos cursos, a escola inclui, no currículo básico de todos os cursos, duas disciplinas chamadas Resolução de Problemas (I e II) onde a metodologia a ser aplicada deve, necessariamente, ser aderente aos princípios de PBL.

Dentro desta estrutura, o curso de SI está estruturado em três turmas, uma no período matutino com 60 vagas e duas no período noturno com 60 vagas cada, totalizando 180 ingressantes por ano. É um curso que possui uma grade curricular diferenciada⁸ por contar com um ciclo básico interdisciplinar que preenche 15,8% da carga horária do curso, com disciplinas de conteúdos transversais, cujo objetivo é proporcionar ao aluno condições para obter uma formação que transcende o conhecimento técnico inerente às necessidades de profissionalização. As disciplinas que compõem esse ciclo básico tratam de conceitos principalmente relacionados a metodologia científica, análise de dados, ciências da natureza, direito, psicologia, educação e arte. Os 84,2% restantes do curso compõem a formação específica do curso de SI e são compostos por disciplinas distribuídas nos núcleos de fundamentos e

⁶ <http://www.topcoder.com>

⁷ <http://www.kaggle.com/>

⁸ Veja a grade curricular do curso em <http://each.uspnet.usp.br/site/graduacao-cursos.php>.

tecnologia da computação, sistemas de informação, matemática, e contexto social e profissional, em conformidade ao disposto no Currículo de Referência da Sociedade Brasileira de Computação para Cursos de Graduação em Computação e Informática [SBC, 2003]⁹. Idealmente, o curso deve atender a 720 alunos anualmente e, de fato, tem mantido em média esse número de alunos matriculados.

O curso tem por objetivo oferecer uma formação ampla ao aluno, de forma a capacitá-lo a compreender o potencial da computação como atividade-meio. Com base neste objetivo, o trabalho desenvolvido pelos professores no curso de SI tem sido realizado com o intuito de alcançar metas importantes em várias vertentes, entre elas estão:

- manter uma grade curricular moderna, capaz de acompanhar as exigências da rápida evolução da área de computação e de administração em tecnologia da informação;
- fomentar iniciativas extracurriculares que contribuam para o incentivo da permanência do aluno na universidade, realizando atividades de ensino, pesquisa e extensão.

A fim de apresentar o curso de SI no que tange ao alcance das metas citadas e de relacioná-las às experiências apresentadas no presente artigo, o restante desta seção é dedicado ao detalhamento dos aspectos do projeto pedagógico do curso, que fundamentam as discussões aqui delineadas, e à apresentação do Grupo de Educação Tutorial de Sistemas de Informação, grupo de alunos que propôs e realizou, sob a supervisão de professores do curso, o BxComp.

3.1. Projeto Pedagógico do Curso de Sistemas de Informação

Como é esperado de qualquer curso oferecido em uma instituição de ensino superior de qualidade, o curso de SI objetiva formar profissionais de alto nível técnico e científico. Para isso, o Projeto Pedagógico deste curso [Masiero, 2010] prevê mecanismos para a construção da grade curricular e de atividades extracurriculares (complementares) que ofereçam ao aluno uma sólida formação e oportunidades de melhorar sua capacidade de assimilação e produção de conhecimento.

O perfil de formação esperado para um egresso do curso de Sistemas de Informação está altamente relacionado à necessidade de oferecimento de recurso humano profissional, e altamente capacitado, para atuar na automação de sistemas de informações para os mais diferentes tipos de processos. No documento que embasa o trabalho desenvolvido no curso de SI, esses tipos de processos são destacados de forma a mostrar a amplitude de atuação de um profissional egresso do curso: processos de produção, comuns na indústria química, siderúrgica, etc; processos de serviço, nos quais o produto final oferecido pela área de sistemas de informação é necessário para suportar outras atividades, como é o caso na indústria financeira, na engenharia, nas ciências ou na educação.

Assim, a importância e criticidade dos sistemas de informação exige a formação de profissionais que conheçam profundamente os “... *princípios, teorias e técnicas das áreas de computação e correlatas...*” [Masiero 2010, p.6] e adquiram competências e

⁹O núcleo de Ciências da Natureza, contemplado neste currículo de referência, é atendido por disciplinas do Ciclo Básico.

habilidades para lidar com as diferentes “... opções profissionais e das tendências tecnológicas do mercado, podendo atuar em diversas áreas” [Masiero 2010, p.6].

Ainda no Projeto Pedagógico do curso afirma-se que

“os egressos do curso de Bacharelado em Sistemas de Informação são capacitados a aplicar seus conhecimentos de forma independente, inovadora e empreendedora, acompanhando a evolução do setor de Tecnologia de Informação e **contribuindo para a busca de soluções nas diferentes áreas aplicadas...**” [Masiero 2010, p.7],

sendo enfatizado que os egressos devem ter desenvolvido, durante o curso de graduação, atividades que colaborem com:

“a obtenção de um perfil com forte embasamento conceitual em áreas que desenvolvam o raciocínio, senso crítico e habilidades intelectuais (formação básica)” [Masiero 2010, p.11],

“a capacidade de solução de problemas matemáticos” [Masiero 2010, p.11],

“o domínio do processo de projeto e implementação de sistemas computacionais, envolvendo principalmente software...” [Masiero 2010, p.11],

“a programação de sistemas computacionais utilizando diferentes paradigmas” [Masiero 2010, p.11].

Além disso o Projeto Pedagógico do curso aconselha “a forte ênfase no uso de laboratórios para capacitar os egressos no uso eficiente das tecnologias nas organizações” [Masiero 2010, p.2]. Isto posto, confirma-se aqui a forte relação entre a formação de um profissional da área de sistemas de informação, o conhecimento pleno de conteúdos da área de computação e a necessidade de desenvolvimento de habilidades e competências referentes ao raciocínio lógico e à criatividade na resolução de problemas - quesitos que podem ser potencializados com as atividades propostas nas experiências discutidas neste artigo.

Para alcançar os preceitos referentes ao desenvolvimento de raciocínio e habilidade de resolver problemas por meio de programação, o Projeto Pedagógico do curso de SI estabelece cinco disciplinas eletivas relacionadas ao aprendizado de programação, ministradas durante os dois primeiros anos do curso:

- Introdução à Ciência da Computação I (ICC I): aborda conceitos básicos de computação, lógica de programação e desenvolvimento de algoritmos com o objetivo principal de ensinar programação estruturada e orientada a objetos;
- Introdução à Ciência da Computação II (ICC II): aprofunda técnicas de desenvolvimento de algoritmos como indução, divisão e conquista, tentativa e erro, heurísticas; apresenta análise de algoritmos como teoria geral e aplica sobre os algoritmos desenvolvidos durante a disciplina;
- Algoritmos e Estrutura de Dados I (AED I): detalha o desenvolvimento e utilização de estruturas de listas, matrizes e árvores, bem como alocação dinâmica e eficiente de memória;

- Algoritmos e Estrutura de Dados II (AED II): aborda algoritmos para classificação, organização de arquivos, técnicas de indexação e grafos;
- Computação Orientada a Objetos (COO): relacionada principalmente ao ensino de conceitos avançados de orientação a objetos e padrões de projeto.

3.2. Programa de Educação Tutorial - Sistemas de Informação

O Programa de Educação Tutorial (PET) tem como objetivo motivar o envolvimento do aluno de graduação com atividades de ensino, pesquisa e extensão (a tríade universitária), de forma não dissociada, promovendo melhorias para os cursos de graduação do país e, principalmente, contribuindo para a modernização do ensino superior. É gerenciado nacionalmente pela Secretaria de Educação Superior/Ministério da Educação (SESu/MEC), estando, nas universidades, sob a tutela das Pró-Reitorias de Graduação (em sua forma clássica) ou das Pró-Reitorias de Extensão (no novo formato Conexões de Saberes).

O trabalho dentro deste programa segue o modelo de educação tutorial no qual alunos de graduação trabalham de forma cooperativa, sendo responsáveis pela concepção e condução das atividades de promoção da tríade universitária, sempre tutorados por um professor. Um grupo PET típico é formado por um professor tutor, doze alunos bolsistas e, opcionalmente, mais seis não bolsistas, escolhidos em rigorosos processos de seleção onde são avaliados tanto o perfil acadêmico do aluno quanto suas habilidades de trabalho colaborativo.

O grupo Programa de Educação Tutorial - Sistemas de Informação (PET-SI) foi estabelecido dentro do contexto do curso de SI da EACH no final de 2010. No seu projeto de formação, aprovado no Edital no 9/2010 (Programa de Educação Tutorial PET 2010 – MEC/SESu/SECAD), é apresentado como um dos objetivos de trabalho para este grupo *“idealizar e executar projetos relacionados à melhoria do ensino de graduação no curso de Sistemas de Informação, fortalecendo os mecanismos de ensino aprendido propostos pela escola”*. Além deste, outros objetivos são delineados, relacionados a questões de formação humanística, pesquisa e extensão e disseminação do conhecimento e da informação.

Na busca por alcançar estes objetivos, desde sua formação, o grupo vem continuamente contribuindo com o curso de graduação através da realização de atividades que, de maneira direta ou indireta, revertem em oportunidades para que os alunos da graduação como um todo (e não apenas aqueles que participam do grupo) melhorem sua formação¹⁰.

A implantação deste programa no curso de SI da EACH contribui com o atendimento das metas de melhoria da qualidade da graduação, como estabelecido pelo MEC quando da concepção do programa, e com a segunda meta do curso de SI aqui previamente mencionada. Observando o trabalho realizado pelo grupo, é notório que o programa contribui para a criação de uma cultura acadêmica diferenciada no aluno que participa do grupo. E, de maneira mais tímida, percebe-se que estes alunos atuam como agentes multiplicadores desta cultura junto aos demais, como discutido por Peres e Boscaroli (2011).

¹⁰Todas as atividades realizadas pelo grupo PET-SI estão resumidamente descritas em seu sítio na WEB: <http://www.each.usp.br/petsi/>.

4. Experiências

Desde a criação do curso de SI, alunos e professores têm se preocupado com o oferecimento de um curso completo e de qualidade. De 2005 a 2008 (os quatro primeiros anos do curso), os principais esforços ficaram concentrados na composição do corpo docente e proposição de disciplinas optativas. A partir de 2009, com a formatura das primeiras turmas do curso, os esforços foram direcionados para o aperfeiçoamento e complementação da formação dos alunos.

Esta seção apresenta duas iniciativas ligadas ao aperfeiçoamento das habilidades de programação, raciocínio lógico matemático e trabalho em equipe: a criação e oferecimento das disciplinas optativas Desafios de Programação I e II e a criação e execução do Campeonato de Programação para Calouros (BxComp). Essas iniciativas, na realidade, acompanham outras relacionadas a diferentes linhas de atuação que, juntas, compõem uma série de ações que vêm sendo realizadas no intuito de aperfeiçoar o curso de SI e oferecer uma formação de qualidade nas mais diferentes perspectivas. Exemplos de outras iniciativas são: o apoio à fundação da empresa júnior “SI USP Jr”, o incentivo à participação dos alunos na Olimpíada de Inovação da USP, e ações para internacionalização da graduação por meio de suporte aos alunos para participação em eventos internacionais e em programas de intercâmbio acadêmico.

4.1. Desafios de Programação I e II

Em 2007, no terceiro ano do curso de SI, foi criada a disciplina optativa Desafios de Programação, cuja ementa previa a intensa execução de atividades de programação, em laboratório, supervisionadas por um professor e contextualizadas em um ambiente desafiador e dinâmico. A ementa da disciplina foi inspirada em competições como a Maratona Internacional de Programação e a Olimpíada Brasileira de Informática.

Até o início de 2009 (quinto ano do curso), tal disciplina não havia sido oferecida e nenhuma outra atividade extracurricular específica para aperfeiçoamento das habilidades relacionadas à programação havia ocorrido. Nesse mesmo ano, a Comissão de Coordenação do Curso definiu, no plano trienal de metas, a criação e realização de tais atividades, motivando os professores a ofertar a disciplina e propor outras ações correlatas.

No segundo semestre de 2009, antes do primeiro oferecimento da disciplina, foi criado um grupo de estudos em algoritmos visando a avaliar o interesse dos alunos pelo assunto e, eventualmente, preparar times para concorrerem na etapa brasileira da Maratona Internacional de Programação. O grupo era composto por pouco mais de 20 alunos e um professor, que se reuniam semanalmente no período da tarde para discutir um ou dois problemas computacionais extraídos do conjunto de problemas do *Uva Online Judge*. Por se tratar de uma atividade extracurricular, nos períodos de provas e entregas de trabalhos a frequência dos alunos às reuniões do grupo diminuía. Mesmo assim, esta primeira experiência foi bastante interessante. No primeiro semestre de 2010, um novo grupo de estudos de algoritmos foi formado e, desta vez, o número de alunos aumentou para 30. Novamente a iniciativa foi considerada positiva, embora continuasse não sendo prioritária para os alunos, devido ao seu caráter extracurricular. Ambas as experiências mostraram que os alunos estavam interessados em realizar estudos referentes a desafios de programação, motivando o oferecimento da disciplina Desafios de Programação.

No segundo semestre de 2010 a disciplina Desafios de Programação foi oferecida pela primeira vez. Optou-se por oferecê-la à tarde para que alunos de todos os períodos e anos, contanto que atendessem aos pré-requisitos da disciplina, pudessem cursá-la. Foi o primeiro semestre em que o curso ofereceu uma disciplina optativa à tarde e esta teve mais de 70 alunos inscritos para as 60 vagas oferecidas e desde então, ambas as disciplinas têm sido ofertadas em semestres alternados¹¹.

Devido à grande procura pela disciplina, foi criada uma segunda disciplina na mesma linha - Desafios de Programação II, porém com requisitos mais avançados, podendo assim tratar de problemas (desafios) mais complexos. Como consequência, a primeira disciplina foi renomeada para Desafios de Programação I. A descrição resumida de ambas as disciplinas é apresentada na Tabela 1.

Como pode ser observado na Tabela 1, cada uma das duas disciplinas relacionadas a desafios de programação exige o cumprimento de um pré-requisito. Isso significa que para um aluno estar apto a cursar a disciplina, ele precisa ter cursado as disciplinas referentes a Algoritmos e Estrutura de Dados. Essa exigência é necessária para garantir condições mínimas para que os alunos estejam capacitados a resolver os problemas que a eles serão apresentados durante as aulas, e também para possibilitar que situações de competição sejam simuladas.

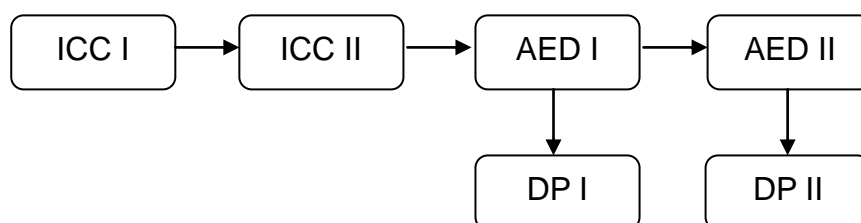
Tabela 1. Descrição resumida das disciplinas Desafios de Programação I e II¹²

	Desafios de Programação I	Desafios de Programação II
Pré-requisitos	Algoritmos e Estruturas de Dados I	Algoritmos e Estruturas de Dados II
Carga Horária	60 horas	
Objetivo	Desenvolver habilidades de resolução de problemas computacionais em ambiente colaborativo e sob pressão, nos moldes de concursos internacionais e nacionais de programação.	
Programa	Disciplina em laboratório de computação, com intensa atividade de programação. Os problemas de programação incluem estruturas de dados, ordenação, aritmética, álgebra, combinatória, teoria dos números, <i>backtracking</i> , grafos, programação dinâmica e geometria computacional.	
Metodologia	Toda aula é dividida em três partes. Durante os 15 primeiros minutos de cada aula, diferentes soluções dos problemas da aula anterior são apresentadas e discutidas. Durante os 15 minutos seguintes, os enunciados dos problemas da aula atual são apresentados. No restante da aula, os alunos se organizarão em grupos e tentarão resolver o maior número de problemas que conseguirem.	
Livro texto	S. S. Skiena, M. A. Revilla. <i>Programming Challenges: The Programming Contest Training Manual</i> . Springer, 2003.	

¹¹A avaliação constante na Seção 4 do presente artigo é referente à oferta da disciplina Desafios de Programação I, oferecida no segundo semestre de 2010.

¹²A descrição completa da disciplina está disponível em <https://uspdigital.usp.br/jupiterweb/>, sob os códigos ACH2107 e ACH2108.

A linha de pré-requisitos relacionada a este contexto está ilustrada pela Figura 1, note que as disciplinas de Algoritmos e Estruturas de Dados exigem, por sua vez, o cumprimento de outras disciplinas como pré-requisito. Observe que é possível cursar Desafios de Programação II sem antes ter cursado Desafios de Programação I, mas quatro disciplinas precisam ser cumpridas para capacitar o aluno a cursar Desafios de Programação II. A disciplina de Computação Orientada a Objetos, embora não integre a linha de pré-requisitos referente às disciplinas de Desafios de Programação, também aborda conceitos que são úteis aos alunos na resolução dos desafios.



Legenda:

- ICC I - Iniciação à Ciência da Computação I
- ICC II - Iniciação à Ciência da Computação II
- AED I - Algoritmos e Estrutura de Dados I
- AED II - Algoritmos e Estrutura de Dados II
- DP I - Desafios de Programação I
- DP II - Desafios de Programação II

Figura 1: Linha de Pré-Requisitos referente às disciplinas de Desafios de Programação.

4.2. BxComp

O BxComp – Campeonato de Programação para Calouros do Curso de Sistemas de Informação, foi idealizado e organizado pelos alunos do grupo PET-SI, com supervisão da professora tutora do grupo e de mais três professores do curso de SI, atuantes nas disciplinas das áreas de “lógica de programação” e “algoritmos e estrutura de dados” do curso. O campeonato foi concebido especificamente para os alunos ingressantes do curso mediante a percepção dos próprios alunos do grupo PET-SI sobre três aspectos:

- inexistência de uma atividade do estilo de olimpíadas ou maratona de programação na qual todos os alunos ingressantes fossem elegíveis para participar¹³;
- a carência de atividades extracurriculares que pudessem preencher o “tempo livre” dos alunos calouros, uma vez que estes geralmente não participam de atividades de iniciação científica e também não podem realizar estágios extracurriculares;

¹³Embora a Olimpíada Brasileira de Programação admita como competidores alunos recém ingressantes no ensino superior, nem todos os alunos são elegíveis por terem finalizado o Ensino Médio a mais de um ano antes de serem aprovados no vestibular. E, a exigência dos desafios que compõe a Maratona de Programação também inibe aqueles que são ainda iniciantes no assunto.

- a possibilidade de executar uma tarefa que pudesse incentivar os alunos a se dedicarem mais ao estudo dos conceitos de programação e, desta forma, contribuir para a melhoria do processo de aprendizado dos alunos da graduação, como esperado pelos preceitos do Programa de Educação Tutorial.

No entanto, logo no início do processo de preparação do campeonato, percebeu-se que a atividade tinha o potencial para contribuir com o aprimoramento da habilidade de resolução de problemas através da programação, motivando tanto o aluno que **participa** do campeonato a se dedicar com mais afinco ao estudo da disciplina de programação, quanto o aluno que o **organiza** a se dedicar à revisão de conceitos e aprimoramento das habilidades de formulação e resolução de problemas (uma vez que estes últimos eram os responsáveis por preparar os desafios de programação¹⁴ que seriam usados nas etapas do campeonato).

Somado a estas finalidades, está também o intuito de promover atividades de integração dos alunos calouros ao contexto do curso de Sistemas de Informação, tanto no que tange à área de atuação do curso quanto no que diz respeito à melhoria das relações interpessoais entre os alunos (calouros e veteranos).

Para alcançar os objetivos delineados, a realização do campeonato compreendeu duas fases:

- **preparação:** desenvolvida durante três meses antes do início do campeonato, compreendendo atividades de elaboração de regulamento, alinhamento de nível de dificuldade de desafios junto aos professores das disciplinas, elaboração de homepage¹⁵ e preparação de infraestrutura (reserva de laboratórios, preparação do ambiente computacional com o compilador JAVA e suporte com o uso do software BOCA¹⁶);
- **realização:** com duração de quatro meses, foi composta por dez etapas, onde uma etapa se refere à reunião das equipes participantes (alunos calouros) em um laboratório, para desenvolvimento das soluções dos problemas apresentados a elas, sob supervisão dos alunos que organizam a atividade (alunos veteranos);

A fim de regulamentar a atividade, os alunos que prepararam o campeonato elaboraram uma série de regras, dentre as quais se destacam: as equipes deveriam ser compostas por no mínimo dois e no máximo quatro alunos; apenas alunos ingressantes no ano de 2011 poderiam participar do campeonato; a pontuação das equipes seria calculada de forma cumulativa (pontos corridos); as equipes poderiam faltar em no máximo três etapas; nenhum material adicional (livros ou acesso à internet) seria permitido durante a realização das etapas; cada etapa teria duração máxima de duas horas; em uma etapa poderia ser solicitada a resolução de um ou mais desafios.

Ao final de cada etapa, os alunos proponentes da atividade (os veteranos) devem estar à disposição dos alunos participantes (calouros) para mostrar a solução esperada para o desafio, bem como mostrar os erros que as equipes cometeram durante a etapa. Com base nessa interação, dicas sobre como proceder nas etapas seguintes para evitar

¹⁴Para essa preparação os alunos usavam, e em alguns momentos adaptavam, desafios de programação disponíveis em repositórios existentes na WEB, em notas de aulas de professores, ou em exercícios disponibilizados em livros.

¹⁵<http://www.each.usp.br/petsi/BxComp2011>

¹⁶<http://www.ime.usp.br/~cassio/boca/>

algum erro e explicações sobre alguns dos conceitos cobrados nos desafios da etapa eram, posteriormente, disponibilizadas na homepage do campeonato.

Com o objetivo de conferir um caráter lúdico ao ambiente, as equipes receberiam no início de cada etapa uma quantidade de bexigas equivalente ao número de desafios propostos, sendo que, a cada submissão de resolução correta de um desafio, a equipe deveria estourar uma bexiga¹⁷.

Nesta primeira edição do campeonato participaram 57 alunos, organizados em 15 equipes. Foram preparados 21 desafios, sendo que em média, em cada etapa eram apresentados dois desafios aos competidores. A preparação dos desafios, realizada pelos alunos organizadores, envolvia: a escolha de desafios publicados em repositórios especializados existentes na WEB; se necessário, a adequação do desafio para atender ao nível de complexidade estabelecido para a etapa; a resolução do desafio tanto para validar a adequação do problema à etapa, verificando se não existe a necessidade de aplicar algum conceito ou recurso de programação ainda não trabalhado em sala de aula, quanto para preparar a parte da codificação que é oferecida pronta para os competidores; a criação dos casos de teste para apresentação aos competidores e a criação dos casos de teste para correção das soluções submetidas pelos competidores. Para fins de ilustração, a Figura 2 traz um exemplo de um desafio usado no campeonato¹⁸.

A opção por realizar uma atividade de longa duração, distribuída durante todo o semestre, conferiu a essa experiência um diferencial em relação às outras competições de programação (que têm a duração de um dia), aproximando esta iniciativa de uma competição também realizada no curso de Sistemas de Informação, porém na área de Engenharia de Software (com duração mais longa) [Fantinato et al. 2011].

Cabe observar que, embora o grupo organizador tenha trabalhado para oferecer uma premiação às equipes mais bem colocadas, tratava-se de uma premiação simbólica (medalhas e brindes para os membros das equipes vencedoras e certificado de participação para todos os participantes). Assim, o maior motivador para os alunos calouros foi a possibilidade de participar de uma atividade extraclasse, lúdica, desafiadora e correlata ao objeto de interesse de aprendizado dos alunos – conceitos de lógica de programação e resolução de problemas.

¹⁷Essa “brincadeira” é uma alusão ao recebimento de bexigas por parte das equipes participantes de maratonas de programação, quando da apresentação da resolução correta para um desafio.

¹⁸Esse desafio, em especial, foi formulado pelos alunos do grupo PET-SI, com base em alguns outros similares presentes no repositório <http://olimpiada.ic.unicamp.br/>.

Encrypted Images

Your mission, if you decide to accept it, is to create an image drawing program. The image will consist of characters and spaces.

Input and Output:

Your program will get the information for the image from the input file. This file will contain lines of characters which your program must interpret to draw an image. Each row in the file comprises a series of numbers and characters, always taken pairwise, where the leading number (greater than or equal to zero) indicates how many times the character will be used. Be careful, the lowercase letter "b" will be used in the input file just to represent spaces in the image (also following the same number-character format). The character "!", the only unpaired character (that is, not taken pairwise), will be used to represent different rows in the image. There is no limit to the number of rows in an image or the number of images in a file. All of the input files are disposed following these rules. Happy drawing!

Simple Input

Ex1.

```
1X2a1b1X!4A1b0u1X
```

Ex2.

```
1T1b3&2T!1T2b1T1b2T!1T1b1T2b2T!1T3b1T1b1T!2*1T3b1T!1T3b1T1b1T!3*4T
```

Ex3.

```
3P1b3E1b3T4b3S1b3I!1P1b1P1b1E4b1T5b1S4b1I!3P1b3E2b1T2b3b3S2b1I!1P3b1
```

Simple Output

Ex1.

```
Xaa X
AAAA X
```

Ex2.

```
T &&&TT
T T TT
T T TT
T T T
**T T
T T T
***TTTT
```

Ex3.

```
PPP EEE TTT SSS III
P P E T S I
PPP EEE T SSS I
P E T S I
P EEE T SSS III
```

Figura 2: Exemplo de um desafio aplicado no campeonato. Em algumas etapas, uma variável nova era apresentada aos competidores. Na etapa onde este desafio foi apresentado, a novidade foi a apresentação do enunciado do problema em língua inglesa.

5. Avaliação

Como forma de avaliação das iniciativas acima descritas, foram distribuídos questionários que deveriam ser respondidos anonimamente pelos alunos. Devido a questões de oferecimento e objetivo de cada iniciativa, bem como do fato do BxComp estar em sua primeira edição, os questionários, além de diferirem em parte, não foram aplicados simultaneamente, não tendo sido também respondidos pela mesma amostra de alunos, havendo um período de dois semestres entre a avaliação da disciplina Desafios de Programação e do campeonato BxComp.

Vale mencionar que, por razão do momento de realização dessa pesquisa, apenas os dados relacionados à disciplina Desafios de Programação I e ao BxComp estavam disponíveis para análise e foram usados para as discussões no presente artigo. Por fim, é importante ressaltar novamente que nenhuma das iniciativas é obrigatória, isto é, o BxComp é uma atividade extracurricular e as disciplinas de Desafios de Programação I e II são optativas. Conforme será destacado na Seção 6, estas atividades têm atraído, principalmente, alunos que possuem desempenho acadêmico acima da média de suas turmas.

5.1. Desafios de Programação

Participaram da avaliação 52 alunos, de um total de 64, correspondendo aos alunos que cursaram a disciplina durante o ano de 2010 (segundo semestre). Foram respondidas questões referentes tanto à análise da atividade em si, como por exemplo “o grau de dificuldade dos problemas apresentados”, quanto à análise do efeito da atividade sobre a percepção de aprendizado, instigando os alunos a avaliarem, por exemplo, o conjunto de habilidades que eles acreditam terem sido desenvolvidas durante a disciplina. De maneira geral, a iniciativa foi muito bem avaliada pelos alunos, conforme demonstram os gráficos na Figura 3.

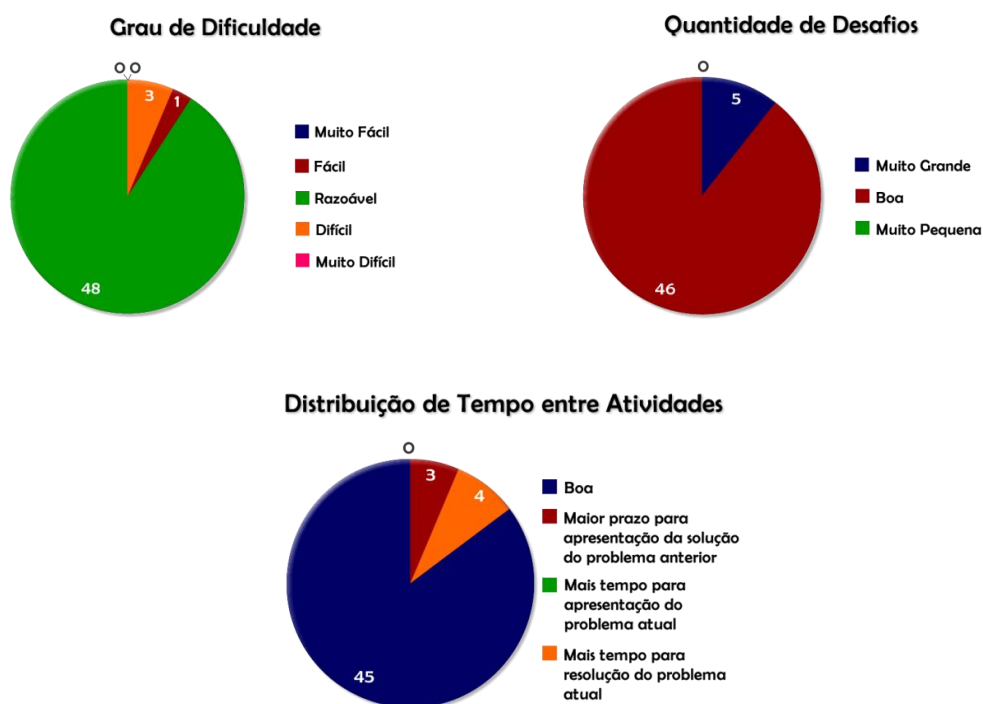


Figura 3. Adequação da disciplina Desafios de Programação

De acordo com os resultados, a quantidade e grau de dificuldade ficaram no ponto médio da escala (uma Likert balanceada de 5 e 3 categorias, respectivamente), ou seja, os problemas foram considerados adequados em sua quantidade e grau de dificuldade. Além disso, a distribuição do tempo entre as atividades também foi considerada boa por 86,5% dos estudantes (contra 13,5% que apontaram a necessidade de mais tempo para apresentação das soluções ou para a resolução dos problemas atuais).

Quando questionados sobre a contribuição da disciplina para sua formação no curso de SI, 100% dos alunos afirmaram ter a disciplina contribuído para sua formação, sendo que 98,1% demonstraram interesse em cursar outras disciplinas que apresentassem a mesma dinâmica. Quanto ao grau de contribuição, 78,8% afirmaram terem não somente complementado conhecimento de outras disciplinas, mas também aprendido novos conteúdos, enquanto que 21,2% disseram terem apenas complementado conteúdos já vistos. Nenhum dos alunos afirmou não ter obtido ganho algum com a disciplina.

Com relação às notas dos alunos nas disciplinas das quais DP I depende, não foi encontrada diferença significativa entre as médias de notas nas disciplinas ICC I e ICC II, obtidas pelos alunos que cursaram DP I, em comparação às médias obtidas pelos que não cursaram tal disciplina, conforme apresentado na Tabela 2. Houve, contudo, uma diferença significativa em relação à disciplina AED I (pré-requisito imediato de DP I), em que as médias diferiram de maneira significativa ($t\text{-student} = 3,1285$; $p < 0,01$). Isso mostra que o conjunto de alunos então cursando DP I correspondia às maiores médias em AED 1. Para essa análise assumiu-se um intervalo de confiança de 95%.

Tabela 2: Relação das médias obtidas por alunos que cursaram e alunos que não cursaram Desafios de Programação I

	Alunos que cursaram DP I		Alunos que não cursaram DP I		<i>t-student</i>	<i>p</i>
	Média	Mediana	Média	Mediana		
ICC I (2009 ¹⁹)	4,6	4,8	3,7	3,7	1,6621	< 0,11
ICC II (2009)	6,8	7,1	6,4	6,5	1,1037	< 0,28
AED I (2010)	6,4	6,6	4,9	5,3	3,1285	< 0,01

5.2. BxComp

Participaram da avaliação 20 alunos, de um total de 53, correspondendo aos alunos que participaram do BxComp no segundo semestre de 2011. Foram respondidas questões subjetivas acerca da validade da iniciativa e sua contribuição para a integração dos ingressantes com seus demais colegas e veteranos. A iniciativa foi muito bem recebida pelos alunos, que foram unânimes em afirmar que não apenas ela foi válida, mas que deveria se repetir no ano seguinte, com a próxima turma de calouros. A totalidade dos participantes afirmou também ter o torneio contribuído para a evolução de seu aprendizado.

¹⁹ Os anos correspondem à época ideal em que os alunos cursaram essas disciplinas, conforme a árvore de pré-requisitos, de modo a que conseguissem fazer Desafios I no segundo semestre de 2010.

Dentre os principais fatores positivos, destacam-se o desenvolvimento do trabalho em equipe (apontado por 45% dos participantes) e o exercício da lógica e prática de conhecimentos já adquiridos (apontado por 40%). Também foram citados o aprendizado adquirido, a aproximação do aluno a um ambiente de competição, e o treinamento em programar sob pressão, dentre outros. Entre os fatores negativos, prevaleceram os operacionais, como a existência de alguns enunciados confusos (45%) e algumas falhas no sistema de correção (30%). A Figura 4 apresenta um resumo dos fatores apontados como positivos e negativos no torneio.

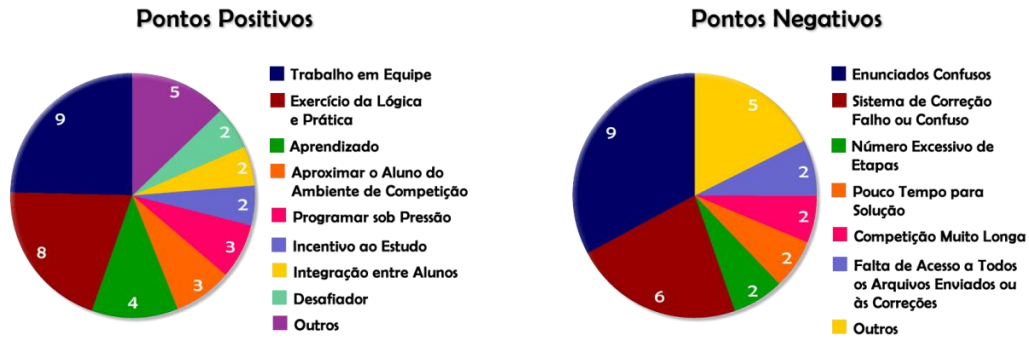


Figura 4. Pontos positivos e negativos do BxComp

Com relação à contribuição dessa iniciativa para a integração dos alunos com seus companheiros de torneio, demais ingressantes, e outros alunos do curso de Sistemas de Informação, os resultados demonstram uma divisão de opiniões, conforme ilustrado na Figura 5. Segundo esses resultados, muito embora a iniciativa contribua para a interação dos participantes (55% de afirmativas contra 40% de negativas), ela parece não contribuir para a integração com os demais ingressantes do curso (60% de negativas contra 35% de positivas), sendo bastante ambígua sua contribuição para a integração com os demais alunos do curso (45% de afirmativas contra 50% de negativas).

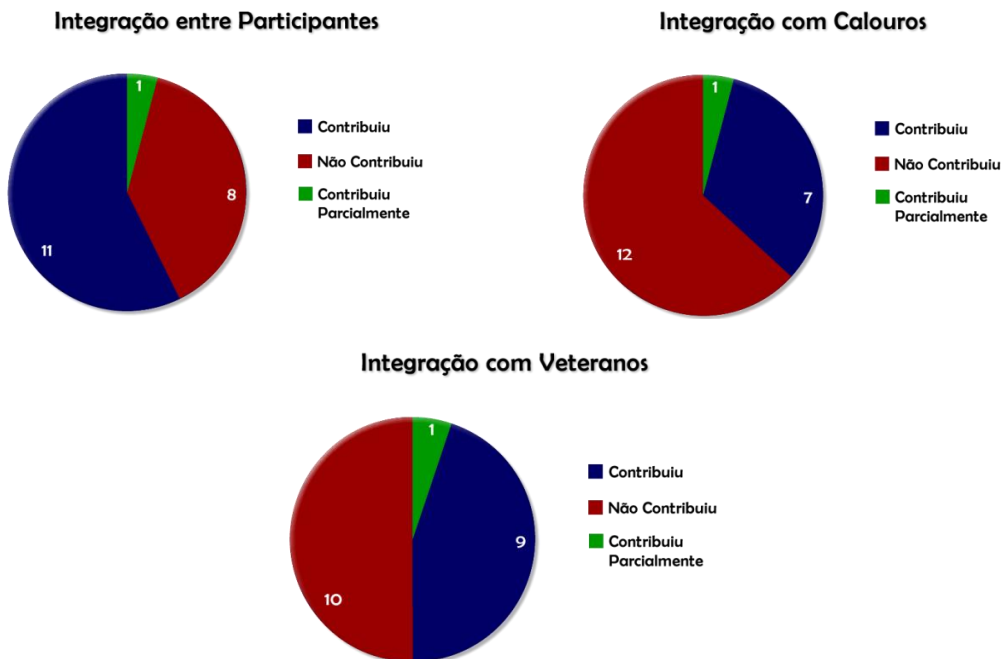


Figura 5. Contribuição do BxComp para integração dos alunos

No que tange ao desempenho dos participantes do BxComp na disciplina de ICC I, disciplina esta cujo conteúdo serve de base para o campeonato, uma diferença significativa foi encontrada entre o grupo dos participantes e dos não participantes ($t\text{-student} = 7,2619, p < 0,01$). Enquanto a média, em ICC I, dos alunos que participaram posteriormente do BxComp ficou em 7,1 (mediana 7,3), a média dos que não participaram ficou em 4,7 (mediana 5,0). Portanto, assim como observado na disciplina DP I, notou-se também aqui que o conjunto de alunos participando do torneio correspondia às maiores médias na disciplina que fornecia sua base imediata. Para essa análise assumiu-se um intervalo de confiança de 95%.

6. Lições Aprendidas

O oferecimento das disciplinas DP I e DP II tem bons reflexos na qualidade do curso de SI. Além da grande procura que essas disciplinas optativas vêm tendo desde seu primeiro oferecimento, mostrando que os alunos do curso têm o desejo de melhorar suas habilidades em programação e resolução de problemas, elas contribuíram para despertar o interesse nos alunos para participarem da Maratona de Programação. Mais do que despertar o interesse, a disciplina serve também como um “treinamento”: já na segunda participação do curso de SI na Maratona de Programação, os alunos se classificaram para a final nacional. Este excelente resultado foi um grande motivador para que outros alunos se interessassem por cursar as disciplinas e participar da competição (para o primeiro semestre de 2012, 113 alunos se inscreveram na disciplina Desafios de Programação II, concorrendo às 60 vagas oferecidas; também para este ano, a quantidade de alunos que se mostraram interessados em participar da Maratona foi bastante grande exigindo a execução de uma prova seletiva dentro do curso, visto existir uma quantidade máxima de equipes que um curso pode enviar).

Uma lição importante aprendida sobre o grupo de estudos de algoritmos e o oferecimento das disciplinas de Desafios de Programação é, justamente, que este tipo de iniciativa pode gerar um ciclo virtuoso de interesse e participação dos alunos. Os resultados e demandas dos alunos surgiram tão rapidamente que percebemos que não estávamos totalmente preparados para atendê-los, principalmente quanto à preparação de material e de um ambiente para treino intensivo para a Maratona de Programação. Nos últimos meses, em um esforço conjunto de professores e alunos, estamos organizando e preparando novos conjuntos de problemas, bem como material didático complementar para atender e essa nova demanda.

O BxComp trouxe benefícios para todos os alunos – participantes e organizadores. As respostas ao questionário de avaliação confirmam que o BxComp contribuiu para o desenvolvimento de habilidades, tendo sido muito bem aceito pelos participantes, inclusive com recomendação para uma nova edição. Cabe observar que alguns participantes se preparavam para o campeonato procurando problemas na internet e resolvendo-os, ou ao menos estudando as resoluções, estabelecendo para si mesmos um tipo de “treino”.

Ainda assim, a diferença significativa entre as médias na disciplina de ICC I, disciplina em que os alunos têm o primeiro contato com uma linguagem de programação, de quem participou do BxComp e de quem não participou indica que sua atratividade ainda está restrita ao grupo com as maiores notas em programação. Resultado similar foi também encontrado com relação aos alunos que cursaram Desafios de Programação I. Nesse caso, contudo, a diferença foi observada somente

com relação às médias na disciplina requisito de DP I, ou seja, AED I. Para as demais disciplinas (que embora não constituíssem pré-requisitos imediatos, faziam parte da cadeia de pré-requisitos que levava a Desafios) não foi encontrada diferença significativa. Embora compreensível, seria interessante tentar homogeneizar esses conjuntos, atraindo também alunos com menor aproveitamento nas disciplinas que compõem os requisitos tanto para o BxComp quanto para a disciplina de Desafios de Programação.

Nesse ponto, mesmo que esses dados apontem para uma possível existência de viés de seleção, a partir do qual somente os alunos com as maiores médias se interessariam por essas iniciativas, há que se ter em mente que é um desafio aumentar o número de interessados, sem que seja necessário tornar obrigatória a participação de todos os alunos.

Além disso, o curso de Sistemas de Informação é um curso preparado para atender a Computação como uma atividade meio. Assim, como previsto no currículo de referência da SBC, este curso deve atender a inserção da Computação no ambiente organizacional, que demanda dos profissionais da área um perfil sócio-técnico. Por esta razão, os perfis dos alunos que entram neste curso variam entre aqueles que entendem o curso como um braço da Computação, e aqueles que esperam do curso uma abordagem mais próxima à área da Administração. Como já destacado na Seção 3, o profissional da área de Sistemas de Informação necessita de um bom embasamento técnico-computacional, ou não estará apto a entender e gerenciar, a contento, os processos computacionais existentes em uma organização. Dado os diferentes perfis de alunos existentes no curso, é válido o incentivo à participação de atividades extracurriculares nas diferentes subáreas de formação. Entretanto, a obrigatoriedade da participação poderia, ao contrário do esperado, atuar como um desestímulo ao estudante.

Já os alunos organizadores, diante da necessidade de preparação dos problemas para cada etapa, se depararam com um desafio até então inédito em seu processo de aprendizado: a complexidade de escolher desafios com níveis de dificuldades adequados, traduzir e ou modificar enunciados sem que o mesmo ficasse dúbio ou incompleto e, principalmente, preparar corretamente casos de testes com uma boa cobertura do problema. Também durante este processo, os alunos organizadores se depararam, em alguns momentos, com a necessidade de assumir um erro e conduzir um processo de reparação como, por exemplo, anular um desafio que não ficou tão bem preparado quanto se achava que estivesse.

7. Conclusão

Ainda que diferentes em seus questionamentos, ambas as avaliações apresentam um importante ponto em comum, expresso unanimemente pelos alunos: a contribuição das iniciativas para a formação do aluno. De fato, a aceitação de ambas iniciativas foi expressiva, declarada por 98,1% dos alunos de Desafios de Programação, que afirmaram ter interesse em cursar outra disciplina com a mesma dinâmica e pela totalidade das respostas dos alunos participantes do BxComp, que concordaram que essa iniciativa deveria ser repetida com a próxima turma de calouros.

O grupo de estudos sobre algoritmos e o posterior oferecimento da disciplina Desafios de Programação deram segurança para que os alunos do curso participassem da Maratona Brasileira de Programação. Em 2010 o curso de Sistemas de Informação da

EACH enviou um time que obteve um resultado mediano na etapa local, não se classificando para a final nacional. Já em 2011, três times do curso participaram da etapa local e um se classificou para a final nacional, ficando em 35^o lugar entre os 536 times que participaram da competição. Em 2012, doze times demonstraram interesse em participar da competição e foi necessário realizar uma prova seletiva para selecionar os cinco times que irão representar o curso na etapa regional da Maratona.

Obviamente muitas características ainda precisam ser aprimoradas nestas iniciativas, como bem mostrou a avaliação feita por meio dos questionários. Possivelmente, a cada oferecimento de disciplina ou a cada versão do campeonato, outros quesitos de avaliação surgirão e melhorias serão implementadas. Objetivamente, alguns direcionamentos já estão sendo implementados para melhorar as próximas versões destas experiências, entre eles destacam-se:

- para as versões seguintes da disciplina Desafios de Programação II foi solicitada uma bolsa para um aluno monitor, de preferência um dos alunos que já participaram da Maratona de Programação, para auxiliar durante as aulas, bem como para passar dicas de treinamento aos alunos da disciplina;
- também foram solicitadas duas bolsas de um ano para que os bolsistas ajudem a preparar e organizar materiais de estudo para a Maratona de Programação e montar um banco de problemas;
- as versões subsequentes do BxComp têm grandes chances de serem organizadas com a participação de alunos que, na primeira versão estavam na posição de participantes; espera-se que esta dinâmica possibilite que alguns dos problemas observados pelos próprios alunos seja resolvido também por eles; além disso, alguns quesitos como número de etapas e processo de preparação de enunciados de desafios devem ser revistos a fim de evitar o descontentamento evidenciado na avaliação;
- a inserção no rol de experiências avaliadas como essas aqui apresentadas, a série de sessões de Dojô de Programação que também compõem experiências para complementar o ensino de programação; essas sessões de Dojô já vem sendo realizadas, conduzidas por alunos monitores com experiência em programação e com participação de alunos ingressantes do curso;
- desenvolver análises referentes também à disciplina Desafios de Programação II, possibilitando uma comparação entre ela e as demais disciplinas a ela correlatas;
- uma uniformização do método de avaliação das iniciativas, permitindo assim uma melhor comparação entre elas.

As experiências aqui discutidas trouxeram resultados relevantes para o curso de Sistemas de Informação e, embora se saiba que cada curso de graduação possui suas particularidades inerentes ao seu contexto de atuação (região, perfil de professores, grade curricular, etc), motivar o aluno a dedicar tempo extra para aprender e praticar conceitos de programação com atividades similares a essas aqui discutidas, pode render efeitos positivos.

Agradecimentos

O grupo PET-SI agradece o suporte do Programa de Educação Tutorial – MEC/SESu. Os autores agradecem aos desenvolvedores do software BOCA.

Referências

- Anderson, J. R.; Skwarecki, E. (1986) The automated tutoring of introductory computer programming. *Communications of the ACM*, vol. 29, n. 9; p. 842-849. New York, NY, USA.
- Anton-Rodriguez, M.; Martinez-Zarzuela, M.; DiezHiguera, J. F.; Perozo-Rondon, F. J.; Diaz-Pernas, F. J.; de la Torre-Diez, I. e Gonzalez-Ortega, D. (2011) Usage testing of game based learning in computer programming courses. In *Proceedings of Promotion and Innovation with New Technologies in Engineering Education (FINTDI 2011)*, p. 1-8. Teruel, Spain.
- Digiampietri, L. A. ; Peres, S. M. ; Silva, B. B. C. ; Teodoro, B. ; Silva Júnior, D. F. P. ; Pereira, G. V. A. ; Borges, G. O. ; Santos, M. V. S. ; Wagner, P. K. e Barros, V. A. (2012) Complementando o Aprendizado em Programação: Experiências no Curso de Sistemas de Informação da USP. Anais VIII Simpósio Brasileiro de Sistemas de Informação, p. 779-790. São Paulo, Brasil.
- Baber, S.; Hoelzeman, D.; Cunningham, B.; Massengale, R. (2011) Programming Contest Hosting: Conference Tutorial. *Journal of Computing Sciences in Colleges*, vol. 26, n. 5, p. 113-115. Consortium for Computing Sciences in Colleges, USA.
- Bergin, S.; Reilly, R. (2005) The influence of motivation and comfort-level on learning to program. In *Proceedings of 17th Workshop of the Psychology of Programming Interest Group (PPIG 2005)*, p. 293-304. Brighton, UK.
- Boersen, R.; Phillips, M. (2006) Programming contests: Two innovative models from New Zealand. In *Proceedings of Perspectives on Computer Science Competitions for (High School) Students*. <http://www.bwinf.de/competition-workshop/>; January 25-28,2006.
- Dagiene, V. (2006) Competition in information technology - learning in an attractive way. In *Proceedings of Perspectives on Computer Science Competitions for (High School) Students*. <http://www.bwinf.de/competition-workshop/>; January 25-28,2006.
- Dijkstra, E. W. (1989). On the Cruelty of Really Teaching Computing Science. *Communications of the ACM*, Vol.32, p. 1398-1404.
- Fantinato, M.; Chaim, M. L.; Morandini, M.; Peres, S. M.;Tuesta, E. F. (2011). Applying a contest to improve learning in the information systems development - an interdisciplinary and extracurricular approach. In *Informatics in Education*, vol. 10, n. 2, p.195–212. Vilnius, LT
- Goldbloom, A. (2010) Data Prediction competitions: Far more than just a bit of fun. In *Proceedings of International Conference on Data Mining Workshops (ICDMW 2010)*, p. 1384-1385. Sydney Australia.
- Gomes, A.; Mendes, A. J. (2007). Learning to program - difficulties and solutions. In *Proceedings of the International Conference on Engineering Education*. Coimbra, Portugal.

- Halim, S.; Halim, F. (2010) *Competitive Programming*. Lulu Press Inc., 152p. Raleigh NC, USA
- Hare, B.; Cain, J. e Cigas, J. (2011) Organizing and Hosting a Student Programming Contest. Panel Discussion. *Journal of Computing Sciences in Colleges*, vol. 26, n. 5, p. 233-234. Consortium for Computing Sciences in Colleges, USA.
- Jenkins, T. (2002). On the Difficulty of Learning to Program. In Proceedings 3rd LTSN for Information and Computer Science Conference, p. 65-71, Loughborough, UK.
- Jolly, J; Jacob, C. (2012) A Study of Problem Based Learning Approach for Undergraduate Students. *Asian Social Science*, vol. 8, n. 15, p. 157-164, Canadian Center of Science and Education.
- Lumsden, L. S. (1994) Student motivation to learn. In Eric Digests, number 92. ERIC Clearinghouse on Educational Management, 1787 Agate Street, Eugene, OR 97403.
- Maehr, M. L.; Midgley, C. (1991) Enhancing student motivation: A school wide approach. *Educational Psychologist*, vol. 26. n. 3-4, p. 399-427.
- Mahmoud, Q. H.; Dobosiewicz, W.; Swayne, D. (2004) Redesigning introductory computer programming with HTML, JavaScript, and Java. In *Proceedings of the 35th Technical Symposium on Computer Science Education (SIGCSE 2004)*, p. 120-124. Norfolk, Virginia, USA.
- Masiero, P. C. (2010) Projeto pedagógico do curso de Sistemas de Informação. Universidade de São Paulo.
- Mergendoller, J. R.; Maxwell, N. L.; Bellissimo, Y. (2006) The Effectiveness of Problem-Based Instruction: A Comparative Study of Instructional Methods and Student Characteristics. *Interdisciplinary Journal of Problem-based Learning*, v.1, n.2, p. 49-69.
- Nishimura, T.; Kawasaki, S.; Tominaga, H. (2011) Monitoring system of student situation in introductory C programming exercise with a contest style. In *Proceedings of International Conference on Information Technology Based Higher Education and Training (iTHET 2011)*, p. 1-6. Kusadasi, Izmir, Turkey.
- O'Grady, M. J. (2012) Practical problem-based learning in computing education. *ACM Transactions on Computing Education*, vol.12, n. 3, p.10:1-10:16 , ACM.
- Paulik, M. J.; Krishnan, M. (2001) A competition-motivated capstone design course: the result of a fifteen-year evolution. *Transactions on Education*, vol. 44, n. 1, p. 67-75.
- Pereira, H. B. B; Zebende, G. F.; Moret, M. A. (2010) Learning computer programming: Implementing a fractal in a Turing Machine. *Computers & Education*, vol. 55, p. 767-776, Elsevier.
- Peres, S. M.; Boscarioli, C. (2011) Valorizando o ambiente acadêmico da graduação com o Programa de Educação Tutorial do Ministério da Educação. In: Elisabete M. Aguiar Pereira, Gabriela Celani, Dora Maria Grassi-Kassisse.. (Org.). In *Inovações curriculares: experiências no ensino superior*. Campinas: FE/Unicamp, p. 536-559.
- Prince, M. (2004) Does Active Learning Work? A Review of the Research. *Journal of Engineering Education*, vol. 93, n. 3, p. 223-231.

- Queirós, R. A. P.; Leal, J. P. (2012) PETCHA: a programming exercises teaching assistant. In *Proceedings of the 17th ACM Annual conference on Innovations and Technology in Computer Science Education (ITiCSE 2012)*, p. 192-197. Haifa, Israel.
- Revilla, M. A.; Manzoor, S.; Liu, R. (2008) Competitive Learning in Informatics: The UVa Online Judge Experience. *Olympiads in Informatics*, Vol. 2, p. 131–148. Vilnius LT.
- Savery, J. R. (2006). Overview of Problem-based Learning: Definitions and Distinctions. *Interdisciplinary Journal of Problem-based Learning*, v. 1, n.1, p. 9-20.
- SBC (2003). Currículo de referência da SBC para cursos de graduação em computação e informática. Documento aprovado em Assembleia Geral da SBC.
- Sleeman, D. H. (1986) Special Section on The Challenges of Teaching Computer Programming. *Communications of ACM*, vol. 29, n. 9, p. 840-841. New York, NY, USA.
- Wang, G.; Feng, R. (2010) Research and Reformation of Online Practice Oriented Teaching in Programming Foundation Course. In *Proceedings of 10th International Conference on Computer and Information Technology (CIT 2010)*. p. 2018-2024. Bradford UK.