Revista de Informática Aplicada , Volume 9, Número 1, 2013

8

Um Módulo de Integração com Juízes On-line para Auxiliar Atividades de Programação

José Osvaldo Chaves, Angélica Castro [1]

Rommel Lima, Marcos Vinicius Lima, Karl Ferreira [2]

Resumo

Disciplinas de programação exigem um grande envolvimento do professor que, muitas vezes, não consegue realizar um acompanhamento eficiente do aluno. A longa espera do aluno para tirar dúvidas ou para apresentar seus resultados são elementos que podem contribuir para sua desmotivação. Devido à sobrecarga de atividades do professor, o uso de ferramentas automatizadas de apoio ao acompanhamento se torna uma boa alternativa. Dessa maneira, visando a contribuir com a melhoria das condições de ensino e aprendizagem em disciplinas de programação, este artigo propõe a automatização do processo de elaboração, submissão de atividades práticas de programação e de suas avaliações, com base em um processo semelhante ao adotado em maratonas de programação, apresentando um ambiente que, integrado ao Moodle, apoie o ensino de programação, tanto presencialmente como à distância.

Palavras-Chave: Disciplinas de programação, Professor, Ensino, Moodle .

Abstract

Disciplines that require a lot of programming needs an involved teacher who often cannot perform efficient monitoring of the student. The long wait for the student to ask questions or present their results are elements that don't contribute to motivation. Due to the overload of teacher's activities, the use of automated tools support monitoring becomes an interesting alternative. Thus, in order to contribute to the improvement of teaching and learning in programming disciplines, this article proposes to automate the process of preparing, submitting programming practices and their evaluations based on a process similar to that used in programming marathons, presenting an environment that integrated the Moodle supports programming teaching both in person and distance.

Keywords: Programming disciplines, Teacher, Teaching, Moodle.

[1] LABSIM - PPgCC - UFERSA Av. Francisco Mota, 572 Mossoró-RN-Brasil oswaldo.mesquita@gmail.com

[2] LORDI - PPgCC - UERN BR110 - KM 48, R. Prof Ant.Campos Mossoró-RN-Brasil ppgcc@uern.br

1 Introdução

A disciplina de programação é uma das disciplinas essenciais, principalmente, aos estudantes de computação, pois constitui a base para muitas áreas em que a informática pode ser aplicada. Um bom aprendizado dessa disciplina torna o indivíduo apto a utilizar a lógica de programação na resolução de diversos problemas, fator importante em disciplinas mais avançadas.

As dificuldades encontradas no aprendizado de programação refletem em altos índices de reprovação e consequentemente em mau desempenho do aluno em outras matérias que têm programação como base. Esse é um grave problema enfrentado pelas instituições de ensino superior no Brasil. Para se ter ideia da gravidade do problema, de acordo com o jornal Folha de São Paulo, que analisou dados do Ministério da Educação (MEC), existe uma alarmante taxa de 28%, em média, de evasão nos cursos de Ciência da Computação (TAKAHASHI, 2014) nessas instituições.

Durante os primeiros anos dos cursos de graduação em computação, por exemplo, é observada uma quantidade bem relevante de alunos que reprovam, desistem ou obtém um baixo rendimento nas disciplinas que focam o estudo de programação. Isso é ocasionado, na maioria dos casos, devido ao processo complexo e exigente que é aprender e desenvolver lógica de programação (FERRADIN, STEPHANI, 2005).

A dificuldade para se aprender programação pode ser consequência de vários fatores, como, por exemplo, uma fraca base matemática, dificuldades para a compreensão do problema e entendimento do assunto. Segundo Prior (2003), a habilidade de se programar computadores não pode ser adquirida sem um significante esforço em atividades práticas de laboratório.

Neste sentido, muitas ferramentas foram propostas para auxiliar o professor no ensino de programação, como, por exemplo, em Moreira e Favero (2009). Porém, mesmo com o advento dessas ferramentas, algumas barreiras ainda são encontradas pelo professor nesse processo, como por exemplo, a dificuldade de avaliar todos os exercícios de uma turma extensa em pouco tempo.

Em geral, quer seja na modalidade de Educação a Distância (EaD) ou no ensino presencial, os sistemas existentes atualmente fornecem um ambiente que permite ao aluno criar seus algoritmos e codificá-los em alguma linguagem de programação, porém, para o professor, torna-se difícil fornecer um feedback rápido e disponibilizar as devidas correções a seus alunos.

Um ambiente muito comum em competições de programação são os chamados Juízes On-line (KURNIA, LIM, CHEANG, 2001), que têm como principal função a avaliação de códigos-fonte. A avaliação feita por esses juízes gera respostas como: certo, errado, saída mal formatada, erro de compilação, erro em tempo de execução, dentre outras (FERREIRA, BOCA, 2004).

As ferramentas de auxílio existentes somadas às pesquisas na área, contribuem não apenas para minimizar os problemas de evasão e dificuldade do aprendizado de programação, como também podem melhorar a qualidade do processo de ensino. Contudo, muito ainda pode ser feito para que as inovações aconteçam com mais qualidade e credibilidade.

No cenário da tecnologia educacional, no que diz respeito ao ensino da disciplina de programação, os ambientes de auxílio existentes não são completos. Entretanto, é possível fazer a integração de dois ou mais ambientes distintos, de uma maneira complementar para um propósito comum. Desse modo, fazendo surgir um novo sistema, mas é nessa integração que se encontra um grande desafio e, embora seja um processo mais complexo, esse é o cenário encontrado mais comumente.

O texto está organizado da seguinte forma: a seção 2 aborda os trabalhos relacionados, apresentando soluções que buscam auxiliar o ensino de disciplinas que envolvam práticas de programação. A seção 3 apresenta as características do Moodle e porque ele é uma boa opção para fazer a integração com outros ambientes. Na seção 4, são apresentados os Juízes On-line, suas características e alguns exemplos de juízes. A seção 5 é responsável por mostrar a problemática e a justificativa que impulsionaram o desenvolvimento da pesquisa. A seção 6 apresenta e descreve o módulo e a arquitetura de integração e, por último, são apresentadas, na seção 7, as considerações finais abordando o que se espera como resultados e o que se pretende realizar em trabalhos futuros.

2 Trabalhos Relacionados

O uso de ambientes virtuais para dar suporte à educação, mais especificamente às atividades práticas de programação, vem sendo explorado há alguns anos. Em um contexto aproximado à pesquisa apresentada neste artigo, algumas iniciativas foram realizadas no sentido de integrar recursos de apoio a disciplinas de programação ao Moodle (KUMAR, GANKOTIYA, DUTTA, 2011), como o BOCA-LAB (FRANÇA, 2012) e a iniciativa de Sirotheau et al. (2011).

O BOCA-LAB foi desenvolvido no Departamento de Engenharia de Teleinformática (Deti) da Universidade Federal do Ceará (UFC) e surgiu da adaptação de um sistema utilizado em maratonas de programação – o BOCA [6]. O BOCA-LAB foi integrado ao Ambiente

Virtual de Aprendizagem (AVA) Moodle, no qual o Moodle forneceu a interface e o conjunto de funcionalidades necessárias à gestão e ao acompanhamento das atividades associadas ao laboratório de programação. A integração dos dois ambientes foi realizada por meio de Web Services (WS). A ferramenta é capaz de compilar e executar programas escritos em diversas linguagens de programação. Os programas submetidos são então avaliados quanto a erros de compilação e execução em um processo automático.

Em Sirotheau et al. (2011), com o objetivo de contribuir para uma melhor compreensão do estudante no aprendizado de programação, a ferramenta JavaTool (MOTA, PEREIRA, FAVERO, 2008), que propicia uma maneira de visualizar e simular programas, também foi integrada ao Moodle, juntamente com o avaliador automático de Moreira e Favero (2009), permitindo a combinação de algumas técnicas para avaliação da complexidade do código. Dessa maneira, colaborando para uma melhor avaliação e feedback das atividades.

Uma importante iniciativa é a de Souza et al. (2012), que mostra a evolução da ProgTest (SOUZA, MALDONADO, BARBOSA, 2011), um ambiente Web automatizado que apoia a submissão e avaliação de trabalhos práticos de programação, baseada em atividades de teste de software. A ProgTest, atualmente, dá suporte a apenas duas linguagens de programação (Java e C) e utiliza um programa referência (programa oráculo) fornecido pelo professor para avaliação dos trabalhos dos alunos (SOUZA, MALDONADO, BARBOSA, 2012), além de utilizar diferentes ferramentas para testes, tais como JUnit e CUnit.

Embora todos os trabalhos aqui citados contenham importantes contribuições para auxiliar no ensino das disciplinas de programação, eles ainda exigem que o professor gaste certo tempo na elaboração das atividades que serão submetidas aos alunos, ou seja, por mais auxílio que o professor tenha com essas ferramentas, ele ainda teria de dedicar uma boa parte de seu tempo para idealizar tais atividades.

Em alguns casos específicos, como é o caso da ferramenta ProgTest, além do professor ter de criar programas referências para auxiliar na correção das questões, tem-se ainda a limitação de se trabalhar restrito a poucas linguagens de programação.

Neste artigo, em complemento aos trabalhos aqui relacionados, é proposto um ambiente que forneça o auxílio necessário ao professor, no que diz respeito à elaboração, submissão e correção de atividades práticas de programação, resultando em maior agilidade nas atividades do professor, um ganho de tempo na avaliação das questões submetidas e um feedback mais rápido ao aluno.

Desse modo, a ferramenta propõe melhorar o ensino e aprendizagem de disciplinas de programação, pois o professor poderá utilizar-se do ganho de tempo para dar maior atenção aos seus alunos. Além disso, a integração dos ambientes vai evitar que o professor precise fazer uso de várias ferramentas tecnológicas para o acompanhamento do aluno, uma vez que a ferramenta proposta é instalada em um Ambiente Virtual de Aprendizagem.

3 Modular Object-Oriented Dynamic Learning Environment (Moodle)

O Moodle foi desenvolvido pelo australiano Martin Dougiamas em 1999, possui tradução para mais de 40 idiomas, e é classificado como um Ambiente Virtual de Aprendizagem (AVA) de software livre, ou seja, pode ser baixado, utilizado e/ou modificado por qualquer indivíduo em todo o mundo.

O Moodle tem uma comunidade de usuários colaborativa e conta, atualmente, com mais de um milhão de participantes espalhados por mais de 200 países, inclusive no Brasil. Essa comunidade, formada por professores, pesquisadores, administradores de sistema, designers instrucionais e, principalmente, programadores, mantém um Portal na Web, que funciona como uma central de informações, discussões e colaborações.

A plataforma em si é diversificada em recursos educacionais e permite larga flexibilidade para configuração e utilização. O seu desenvolvimento extremamente modular permite a fácil inclusão de novos recursos, que mais bem o adaptem às reais necessidades de quem o utiliza. Vale ressaltar que o Moodle é a plataforma oficial do Ministério da Educação (MEC) para as escolas públicas brasileiras (MARTINS, GIRAFFA, 2008), podendo ser utilizada tanto na modalidade de ensino a distância como na modalidade de ensino presencial.

Nesse contexto, a ferramenta oferece a professores e alunos um ambiente capaz de reunir a maioria das informações e eventos relevantes, associados a uma disciplina. O grande potencial oferecido para a criação de novas funcionalidades e sua ampla utilização justificam a integração do Moodle a outras ferramentas.

4 Juízes On-line

A maioria dos programas de natureza algorítmica necessita apenas obter como entrada um padrão de dados devidamente formatados e, a partir desses dados, realizar o devido processamento. Após o processamento, os resultados são apresentados de maneira formatada em uma saída padronizada. Dessa maneira, é possível que a avaliação de programas seja feita automaticamente,

utilizando uma ferramenta que gere os dados de entrada e outra que obtenha e verifique os resultados obtidos (KURNIA, LIM, CHEANG, 2001).

O processo de avaliação automática é feito pelos Juízes On-line. Esses sistemas recebem o código-fonte enviado pelo usuário e posteriormente compilam e executam esse código. Durante a execução do programa, os Juízes On--line utilizam dados formatados como a entrada do programa, processam esses dados e realizam a comparação dos resultados obtidos com os resultados esperados.

Os Juízes On-line são muito utilizados em maratonas de programação e podem ser encontrados na Internet, como, por exemplo, o SPOJ Brasil [14] e UVA Online Judge [15]. Nesses sistemas, são disponibilizados vários problemas de programação para resolução. Dessa maneira, um usuário seleciona a linguagem de programação a ser utilizada na escrita do código e envia a sua solução do problema para ser avaliada. Além disso, também são disponibilizados fóruns de discussão, ranking de usuários e tutoriais.

5 Problemática e Justificativa

No que diz respeito ao ensino de programação, as ferramentas existentes não promovem um auxílio completo a professores e alunos. Uma opção para minimizar esse problema é fazer a integração entre dois ou mais ambientes de maneira complementar, visando a obter um ambiente coeso e completo para esse fim. Porém, é nesta integração que reside um complexo desafio computacional, pois muitas vezes é necessário trabalhar com novas tecnologias e diferentes linguagens de programação.

O que se observa é que, em um contexto geral, as ferramentas de auxílio existentes já são produzidas para um fim específico, sem levar em consideração futuras integrações entre ambientes ou a expansão de suas funcionalidades.

Mesmo os Ambientes Virtuais de Aprendizagem atuais, que apresentam um conjunto de ferramentas de propósito geral e podem ser empregados para diversos cursos, raramente são concebidos com a perspectiva de extensão ou de integração com outras plataformas. Uma exceção a este modelo é o Moodle que possui documentação específica para a agregação de novas funcionalidades. Um forte argumento, se não o maior, que apoia a integração entre os Juízes On-line e o Moodle é o fato de evitar a reimplementação de todos os recursos de gerenciamento de disciplinas e cursos (IHANTOLA et al., 2010).

Um dos desafios mais cansativos, e demorados, enfrentados por um professor que ministra alguma disciplina que envolva prática de programação é a elaboração de atividades práticas para seus alunos, pois o professor muitas vezes trabalha com mais de uma turma e cada turma contendo vários alunos. Com isso, ele precisa gerenciar seu tempo para correção de trabalhos, provas, seminários, dentre outras tarefas.

Nesse contexto, a utilização dos Juízes On-line se mostra uma alternativa válida, pois esses sistemas contam com uma base de dados de questões pré-definidas com toda informação necessária à sua realização, além de procedimentos específicos para a avaliação dos códigos submetidos. Dessa maneira, o professor contaria com o auxílio dos Juízes On-line para utilização das questões e a avaliação delas. Segundo Carter et al. (2003), uma ferramenta que automatize esse processo poupa tempo do professor, que poderá utilizar o tempo ganho para realizar atividades que não podem ser automatizadas

6 Módulo de Integração com os Juízes On--line (Mojo)

Conforme mostrado na seção anterior, a elaboração das práticas pode se tornar um desafio para o professor e, com o auxílio dos Juízes On-line, esse problema pode ser resolvido. Nesse sentido, este artigo apresenta uma ferramenta que funciona como um novo módulo do Moodle para auxiliar o professor no processo de elaboração, submissão e avaliação de questões, proporcionando um ambiente coeso, onde estejam integrados o Moodle e os Juízes On-line. Para um melhor entendimento, a Figura 1 mostra a arquitetura do ambiente.

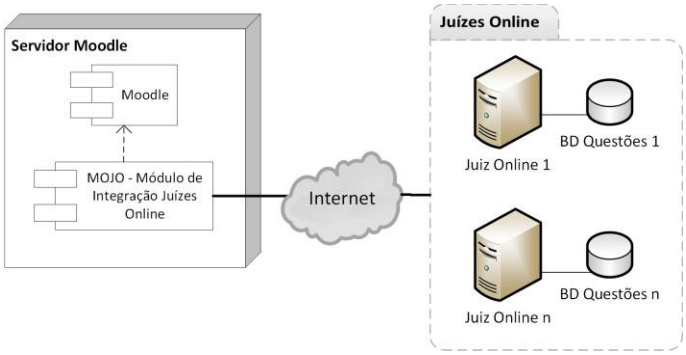

Figura 1: Arquitetura do ambiente

Conforme mostrado na Figura 1, observa-se que de um lado temos os Juízes On-line, com sua base de questões, e, do outro, no servidor Moodle, temos o módulo de integração com os juízes instalado no Moodle. A Figura 2 explora a arquitetura interna do módulo de integração.

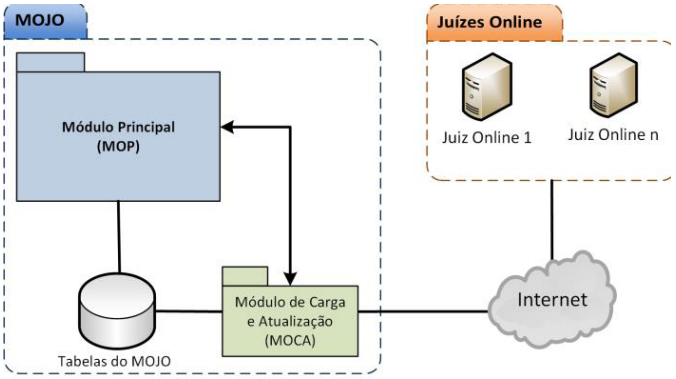

Figura 2: Arquitetura interna do módulo de integração

O Mojo (CHAVES et al., 2013) é a ferramenta propriamente dita (encapsulada em um módulo instalado no Moodle) responsável pela integração, e que vai fazer a comunicação entre o Moodle e os Juízes On-line envolvidos nas operações.

De acordo com a Figura 2, o Mojo é composto pelo MOP (Módulo Principal), que é o responsável por gerenciar todas as funcionalidades da ferramenta. Em um primeiro momento, o Mojo deve realizar uma carga inicial de questões em suas tabelas. Essa carga é realizada pelo outro módulo componente da ferramenta, o Moca (Módulo de Carga e Atualização), que é o responsável pela importação e atualização das questões nas tabelas que foram criadas exclusivamente para o Mojo.

Com as questões armazenadas, o Mojo fornece as informações necessárias para utilizar os dados dessas questões como uma atividade no Moodle. Essas questões são exibidas em uma lista, conforme mostra a Figura 3, na qual o professor pode selecionar e visualizar as informações da questão que ele julgar mais apropriada para submeter a seus alunos.

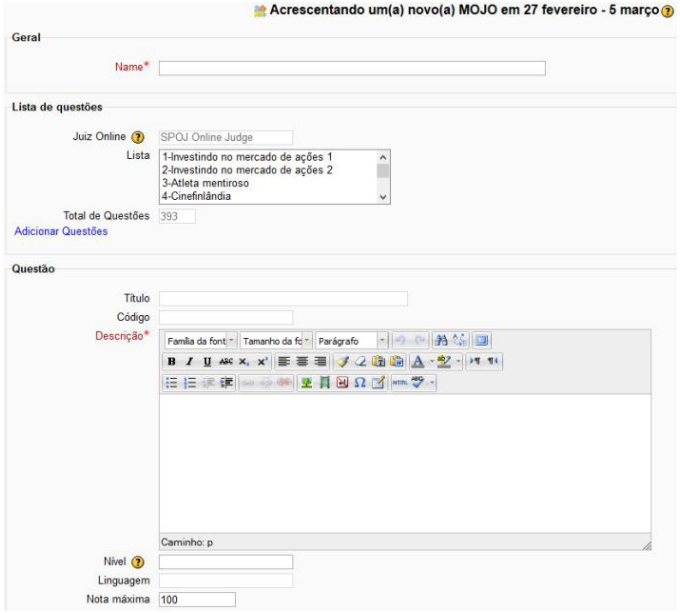

Figura 3: Lista de questões disponíveis no Mojo

Para a correção das soluções enviadas pelos alunos, o Mojo entra em contato com o juiz responsável pela questão, envia o código-fonte do aluno e recupera o resultado da avaliação feita pelo juiz. Por fim, disponibilizando esse resultado para visualização no Moodle.

O funcionamento da ferramenta ocorre de forma distinta para cada um dos diferentes envolvidos no processo: o professor, o aluno e o juiz on-line. Este funcionamento é ilustrado na Figura 4.

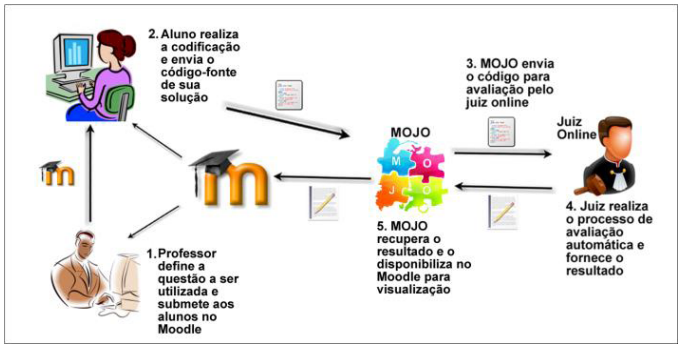

Figura 4: Funcionamento da ferramenta

Conforme ilustrado na Figura 4, o fluxo do processo se divide em 5 (cinco) etapas. Sintetizando bem cada uma delas, temos:

* **Etapa 1:** o professor, no Moodle, define a questão e a submete para resolução pelos alunos (Elaboração e Submissão).

* **Etapa 2:** o aluno visualiza, desenvolve e submete uma solução para a questão (Submissão).

* **Etapa 3:** o Mojo entra em contato com o juiz

responsável pela questão e envia o código-fonte para avaliação (Submissão).

* **Etapa 4:** o Juiz On-line realiza os devidos processos de avaliação automática para a solução proposta e devolve o resultado (Avaliação).

* **Etapa 5:** o Mojo obtém o resultado da avaliação e o disponibiliza no Moodle para visualização pelo professor e pelo aluno (Avaliação).

Em posse do resultado da avaliação, o professor pode dar o devido feedback ao aluno. É importante lembrar que o professor tem acesso aos códigos submetidos pelos alunos para consulta.

7 Considerações Finais

A integração dos Juízes On-line com o Moodle visa a diminuir, consideravelmente, a sobrecarga de trabalho na avaliação de códigos-fonte por parte dos professores. Como resultado, espera-se a melhoria na qualidade do ensino e aprendizagem de programação, tendo em vista que o tempo do professor com atividades de administração e de gestão de recursos pode ser reduzido e, com isso, espera-se uma maior disponibilidade, para dar uma maior atenção ao aluno.

O professor ficará livre da criação de questões, caso ele opte por utilizar uma das questões pré-definidas no juiz on-line, a proposta é exatamente esta: o professor verificará no juiz on-line se existe uma questão que ele julgue interessante para submeter a seus alunos.

Vale ressaltar que nos Juízes On-line existem diversas questões dos mais diferentes níveis de dificuldade, questões que se encaixam bem no aprendizado de turmas que estejam cursando uma disciplina de programação. Mas ainda assim o professor pode optar por não utilizar as questões disponibilizadas pelos Juízes On-line, e aí é onde se encontra uma das propostas de trabalhos futuros: uma maneira de o professor elaborar suas próprias questões e avaliá-las automaticamente, utilizando um laboratório virtual de programação.

O laboratório virtual também servirá como um editor de código-fonte, para que o aluno possa desenvolver seu código no próprio módulo. A integração com outros juízes também está prevista, aumentando ainda mais o número de questões disponibilizadas pelo Mojo.

A ferramenta já está disponível e pode ser encontrada em:

http://greenbeans.com.br/osvaldomesquita/mojo/

Além da ferramenta, é disponibilizado também um tutorial de instalação e utilização, um vídeo demonstrando sua utilização e alguns artigos relativos.

Agradecimentos

Os autores agradecem à Capes e à Fapern pelo apoio financeiro para realização da pesquisa, e em especial ao Programa de Pós-Graduação em Ciência da Computação (PPgCC) da Universidade do Estado do Rio Grande do Norte (Uern) e Universidade Federal Rural do Semi-Árido (Ufersa), por toda infraestrutura oferecida.

8 Referências

[1] TAKAHASHI, F. Matemática e ciências da computação têm alta taxa de abandono. Disponível em: http://www1.folha.uol.com.br/folha/educacao/ult305u546576.shtml. Acesso em: mar. 2014.

[2] FERRADIN, M; STEPHANI, S. L. Ferramenta para o ensino de programação via Internet. In: Anais do I Congresso Sul Catarinense de Computação, Criciúma, 2005.

[3] PRIOR, J. C. Online assessment of SQL query formulation skills. In Anais do V Australasian Conference on Computing Education, Adelaide, p. 247-256, 2003.

[4] MOREIRA, M. P; FAVERO, E. L. Um Ambiente Para Ensino de Programação com Feedback Automático de Exercícios. In: Anais do XXIX Congresso da Sociedade Brasileira de Computação, Belém, p. 429-438, 2009.

[5] KURNIA, A; LIM, A.; CHEANG, B. Online Judge. Computer & Education, 36(4):299-315, 2001.

[6] S, C. P; FERREIRA, C. E. BOCA: Um sistema de apoio para competições de programação. In: Anais do XXV Congresso da Sociedade Brasileira de Computação, Salvador, 2004.

[7] KUMAR S; GANKOTIYA, A. K; DUTTA, K. A Comparative Study of Moodle with other e-Learning Systems. In: Anais do III International Conference on Eletronics Computer Technology, Kanyakumari, p. 414-418, 2011.

[8] FRANÇA, A. B. Sistema de apoio a atividades de laboratório de programação com suporte ao balanceamento de carga e controle de plágio. Dissertação de Mestrado, Universidade Federal

do Ceará, mar. 2012.

[9] SIROTHEAU, S et al. Aprendizagem de iniciantes em algoritmos e programação: foco nas competências de autoavaliação. In: Anais do XXII Simpósio Brasileiro de Informática na Educação, Aracaju, p. 750-759, 2011.

[10] MOTA, M. P; PEREIRA, L. W. K; FAVERO, E. L. JavaTool: Uma Ferramenta Para Ensino de Programação. In: Anais do XVIII Congresso da Sociedade Brasileira de Computação, Porto Alegre, 2008.

[11] SOUZA, D. M; MALDONADO, J. C; BARBOSA, E. F. Aspectos de Desenvolvimento e Evolução de um Ambiente de Apoio ao Ensino de Programação e Teste de Software. In: Anais do XXIII Simpósio Brasileiro de Informática na Educação, Rio de Janeiro, 2012.

[12] SOUZA, D. M; MALDONADO, J. C; BARBOSA, E. F. ProgTest: An environment for the submission and evalua-

tion of programming assignments based on testing activities. In: Anais do XXIV Conference on Software Engineering Education and Training, Honolulu, 2011.

[13] MARTINS, C; GIRAFFA, L. M. M. Capacit@ndo: uma proposta de formação docente utilizando o Moodle. Revista Novas Tecnologias na Educação, 6(1):1-8, 2008.

[14] SPHERE RESEARCH LABS. SPOJ Brasil. http://br.spoj.com/, mar. 2014.

[15] UNIVERSIDAD DE VALLALOLID. UVA Online Judge. Disponível em: http://uva.onlinejudge.org/. Acesso em: mar. 2014.

[16] IHANTOLA, P et al. Review of recent systems for automatic assessment of programming assignments. In: Anais do X Koli Calling International Conference on Computing Education Research, Koli, 2010.

[17] CARTER, J et al. ITICSE working group report: How shall we assess this?. SIGCSE Bulletin, 35(4):107-123, 2003.

[18] CHAVES, J. O. M. et al. Uma Ferramenta Baseada em Juízes Online para Apoio às Atividades de Programação de Computadores no Moodle. Revista Novas Tecnologias na Educação, 11(3):1-8, 2013.