

## Uma Ferramenta Baseada em Juizes Online para Apoio às Atividades de Programação de Computadores no Moodle

José Osvaldo Chaves, PPgCC-UERN/UFERSA, oswaldo.mesquita@gmail.com

Angélica Castro, PPgCC-UFERSA, angelica@ufersa.edu.br

Rommel Lima, PPgCC-UERN, rommelwladimir@uern.br

Marcos Vinicius Lima, PPgCC-UERN/UFERSA, marcos.engsoft@gmail.com

Karl Hansimuller Alelaf Ferreira, PPgCC-UERN/UFERSA,  
karlhansimuller@gmail.com

**Resumo** – Os Juizes Online são sistemas que compilam, executam e testam códigos-fonte realizando a avaliação automática destes códigos. Este método é utilizado em muitos concursos e maratonas de programação. Estes sistemas podem ser facilmente encontrados na Internet, onde disponibilizam vários problemas a serem submetidos para resolução, e são tão atraentes que muitos alunos começam a praticar suas habilidades em programação neles. Com base nisto, este artigo apresenta uma ferramenta para integrar os Juizes Online ao Moodle, fornecendo a praticidade de um Ambiente Virtual de Aprendizagem junto a um sistema de avaliação automática.

**Palavras-chave:** juizes online, programação, moodle, avaliação automática.

### A Tool Based on Online Judges to Support Computer Programming Activities on Moodle

**Abstract** – Online Judges are systems that compile, execute and test source code performing the automatic assessment of these codes. This method is used in many contests and programming marathons. These systems can be easily found on the Internet, where provide several problems to be submitted for resolution and they are so attractive that many students begin to practice their skills in programming them. Based on this, this paper presents a tool to integrate Online Judges into Moodle, providing the convenience of a Virtual Learning Environment together with a system of automatic assessment.

**Keywords:** online judges, programming, moodle, automatic assessment.

#### 1. Introdução

Os programas de natureza algorítmica funcionam basicamente da seguinte maneira: obtêm como entrada um padrão de dados devidamente formatado e, a partir desses dados, realizam o devido processamento (compilação e execução do código). Após este processamento, os resultados são apresentados de maneira formatada em uma saída padronizada. Dessa forma, é possível que a avaliação dos códigos-fonte seja feita de maneira automática por uma ferramenta que gere os dados de entrada e por outra que obtenha, verifique e compare os resultados obtidos (dados de saída).

Os sistemas de Juízes *Online* são conhecidos por realizarem este processo de avaliação automática. Nestes sistemas o código-fonte é compilado, executado e testado com o objetivo de verificar sua correteza (Zhigang *et al.*, 2012). Durante a execução do código do programa, os Juízes *Online* utilizam dados formatados como entrada do programa. Em seguida, esses dados são processados e é realizada a comparação dos resultados obtidos com os resultados esperados, dando uma resposta apropriada com base nestas comparações (certo, errado, erro de execução, erro de compilação). Este é o método utilizado em muitos concursos de programação, tais como ACM *Collegiate Programming Contest International* (ACM-ICPC, 2013) e *TopCoder* (Inc., 2013).

Nestes sistemas são disponibilizados vários problemas para resolução. Dessa maneira, um usuário seleciona o problema que pretende resolver, seleciona a linguagem de programação, a ser utilizada na escrita do código, e envia a sua solução para ser avaliada. Os Juízes *Online* são facilmente encontrados na Internet e como exemplos podem ser citados o *Timus Online Judge* (Timus Online Judge Team, 2013), o SPOJ Brasil (Sphere Research Labs, 2013) e o *URI Online Judge* (URI Erechim, 2013), estes dois últimos utilizados no desenvolvimento deste trabalho.

A ideia de integrar estes sistemas ao Moodle (Kumar *et al.*, 2011) surgiu da exigência de um maior envolvimento do professor de disciplinas de programação. Pois, normalmente o professor fica sobrecarregado por ter que elaborar, submeter, avaliar e fornecer o *feedback* necessário das questões aos alunos (levando em consideração que este professor possui uma turma extensa de alunos) – o que acaba por impactar negativamente na assistência dada à turma.

Com o objetivo de contribuir com a melhoria do processo de ensino e aprendizagem em disciplinas de programação de computadores, este artigo apresenta o Módulo de Integração com os Juízes *Online* (MOJO), uma ferramenta para integrar os Juízes *Online* ao Moodle. O ambiente integrado automatiza o processo de Elaboração, Submissão e Avaliação (aqui chamado de processo ESA) das atividades de programação propostas pelo professor. O objetivo é fornecer um ambiente coeso que facilite o gerenciamento de recursos e permita realizar o devido acompanhamento dos alunos, além de disponibilizar um maior número de questões para utilização nestas disciplinas.

O MOJO, como o próprio nome sugere, é um módulo integrado ao Moodle. Na arquitetura geral do ambiente integrado, o Moodle fica sendo o responsável por fornecer a interface e o conjunto de funcionalidades necessárias à gestão e ao acompanhamento das atividades de programação, e o MOJO fica como o responsável pela comunicação e interação com os Juízes *Online*.

Para um melhor entendimento do texto, este artigo está organizado da seguinte forma: a seção 2 apresenta as características dos dois ambientes integrados pela ferramenta (Moodle e Juízes *Online*), além de citar alguns trabalhos relacionados. A seção 3 apresenta e descreve o MOJO, sua arquitetura interna e como ocorre o seu funcionamento. Finalizando, são apresentados, na seção 4, os resultados e as considerações finais abordando o que se pretende realizar em trabalhos futuros.

## 2. Ambientes da Integração e Trabalhos Relacionados

Nesta seção são descritos os ambientes integrados pelo MOJO, além de enumerar alguns dos trabalhos relacionados.

## 2.1 Moodle

Moodle é um acrônimo para *Modular Object-Oriented Dynamic Learning Environment* e faz parte do grupo de Ambientes Virtuais de Aprendizagem (AVA), esses ambientes permitem a produção de conteúdos e canais variados de comunicação, bem como o gerenciamento de dados e controle total de informações.

O Moodle foi desenvolvido pelo australiano Martin Dougiamas em 1999, possui tradução para mais de 40 idiomas, e conta com uma comunidade colaborativa de usuários que, atualmente, soma mais de um milhão de participantes espalhados por mais de 200 países, inclusive no Brasil. Essa comunidade, formada por professores, pesquisadores, e, principalmente, programadores, mantém um Portal na *Web* que funciona como uma central de informações, discussões e colaborações.

A plataforma conta com uma diversidade de recursos educacionais, permitindo larga flexibilidade para configuração e utilização. Por meio do seu desenvolvimento extremamente modular é possível, ainda, a inclusão de novos recursos que melhor se adaptem às necessidades de quem utiliza o ambiente. É importante observar que o Moodle é a plataforma oficial do Ministério da Educação (MEC) para as escolas públicas brasileiras (Martins e Giraffa, 2008), e se mostrou, também, uma importante ferramenta auxiliar no ensino superior, podendo ser utilizada tanto na modalidade de ensino à distância como na modalidade de ensino presencial.

A ferramenta oferece a professores e alunos um ambiente capaz de reunir a maioria das informações e eventos relevantes, associados a uma disciplina de determinado curso. Diante deste cenário e com base no grande potencial oferecido para a criação de novas funcionalidades, além de sua ampla utilização, fica justificado o porquê da escolha do Moodle para integração com outras ferramentas.

## 2.2 Juízes *Online* Utilizados

Nesta seção são apresentados os dois Juízes *Online* que fazem parte desta primeira versão da ferramenta MOJO, e as características que levaram a seleção dos mesmos.

### 2.2.1 SPOJ *Brasil*

O SPOJ Brasil é a versão brasileira do *Sphere Online Judge* (SPOJ). Suas principais características são:

- Possui interface amigável;
- Disponibiliza fóruns de discussão, ranking de classificação, além de links de notícias e informações;
- Não disponibiliza funcionalidades via *Web Service*;
- Seus problemas suportam várias linguagens de programação, dentre elas C, C++ e Java.

### 2.2.2 URI *Online Judge*

O URI *Online Judge* também conta com uma versão em português. Suas principais características são:

- Possui interface amigável;
- Disponibiliza fóruns de discussão, ranking de classificação, além de tutoriais e materiais extras;

- Não disponibiliza funcionalidades via *Web Service*;
- Possui um repositório dividido e classificado por categorias;
- Seus problemas suportam as linguagens C++ e Java.

Para esta primeira versão do MOJO optou-se por utilizar estes dois juízes devido as suas interfaces amigáveis, número de questões disponibilizadas e por possuírem versão em português.

### 2.3 Trabalhos Relacionados

O uso de ferramentas como suporte ao ensino e aprendizagem, mais especificamente em disciplinas de programação, já vem sendo explorado há alguns anos. Esta seção relaciona algumas dessas ferramentas.

Entre os ambientes que fornecem apoio à submissão e avaliação automática de atividades de programação destacam-se a *WEB-CAT* (Edwards, 2004) e a *PROGTEST* (Souza *et al.*, 2012). *WEB-CAT* é um ambiente *Web* que visa incentivar o desenvolvimento orientado a testes, apoiando a submissão e avaliação automática de atividades de programação. As ferramentas de teste e de análise são disponibilizadas na *WEB-CAT* por meio de *plugins*. Cada *plugin* possui critérios de avaliação e relatórios próprios, associados á uma linguagem de programação particular. Dentre as linguagens suportadas, encontram-se Java, C, C++ e Pascal.

A *PROGTEST*, semelhantemente a *WEB-CAT*, é um ambiente *Web* automatizado para apoio a submissão e avaliação de atividades práticas de programação, e também se baseia em atividades de teste de *software*. Atualmente, dá suporte a apenas duas linguagens de programação (Java e C) e utiliza um programa referência (programa oráculo) que deve ser fornecido pelo professor para avaliação das atividades dos alunos, além de utilizar diferentes ferramentas para testes.

Embora esses trabalhos tragam importantes estudos para auxiliar no ensino de programação, as ferramentas desenvolvidas neles não estão integradas em um ambiente que forneça outros recursos educacionais como, por exemplo, suporte a gestão de conteúdo e ferramenta de discussão. Estes que são considerados importantes recursos, principalmente no que diz respeito a disciplinas ministradas a distância. E em alguns casos específicos, além do professor ter que criar programas referências para auxiliar na correção das atividades, tem-se ainda a limitação de se trabalhar restrito a poucas linguagens de programação.

Uma iniciativa que utiliza Juízes *Online* é a de Santos e Ribeiro (2011) que propõe o *JOnline* um sistema que possui um Juiz *Online* próprio no qual é possível acessar problemas para serem resolvidos e submeter códigos-fonte para a visualização do resultado. Além da programação colaborativa o sistema ainda adiciona funcionalidades didáticas ao juiz, auxiliando o aluno no processo de aprendizagem. Porém, mesmo com estas vantagens, o *JOnline* também não está integrado a um ambiente como, por exemplo, o Moodle que forneça outros importantes recursos.

Em um contexto semelhante ao abordado neste artigo, algumas iniciativas foram realizadas no sentido de integrar recursos de apoio a disciplinas de programação ao ambiente Moodle, como é o caso da iniciativa de Sirotheau *et al.* (2011), do BOCA-LAB (França e Soares, 2011) e do *Onlinejudge* para Moodle (Onlinejudge, 2013). Em Sirotheau *et al.* (2011), com o objetivo de contribuir para uma melhor compreensão do estudante no aprendizado de programação, a ferramenta conhecida

como *JavaTool* (Mota *et al.*, 2009) foi integrada ao Moodle juntamente com o avaliador automático desenvolvido por Moreira e Favero (2009). Proporcionando uma maneira de visualizar e simular programas no Moodle e permitindo a combinação de técnicas para avaliação da complexidade do código. Desta forma, colaborando para uma melhor avaliação e *feedback* das atividades.

O BOCA-LAB foi desenvolvido no Departamento de Engenharia de Teleinformática (DETI) da Universidade Federal do Ceará (UFC) e surgiu da extensão de outro sistema – o BOCA (Campos e Ferreira, 2004). O BOCA-LAB foi integrado ao Moodle por meio de *Web Services*. 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.

O *Onlinejudge* para Moodle é composto por *plugins* e também foi desenvolvido para gerenciar a submissão e avaliação de códigos-fontes no Moodle. Ele pode ser integrado a outras duas aplicações: a *Sandbox* (Sandbox, 2013) e a *Ideone* (Sphere Research Labs, 2013). A *Sandbox* se restringe a submissões que executem em C/C++ no ambiente Linux. Já a *Ideone* permite escrever códigos-fonte em aproximadamente 40 linguagens de programação diferentes, sendo executados diretamente a partir do navegador. Entretanto, a *Ideone* é uma aplicação comercial e permite submissão de apenas 2000 códigos-fonte por mês em uma conta gratuita (Zhigang *et al.*, 2012). Sem as integrações o *Onlinejudge* suporta apenas as linguagens C e C++.

Todos os trabalhos citados contêm importantes contribuições para o ensino de programação. Porém, mesmo com esse auxílio, o professor ainda sofre com a sobrecarga de atividades no que diz respeito ao processo de elaboração, submissão, avaliação e *feedback* de uma turma extensa de alunos.

Neste artigo, em complemento aos demais trabalhos, é apresentado um ambiente que, integrando os Juízes *Online* ao Moodle, forneça: (1) auxílio necessário ao professor no que diz respeito ao processo ESA de atividades de programação; (2) acompanhamento de resultados através de uma mesma interface disponível no ambiente virtual; (3) *feedback* mais rápido ao aluno; (4) suporte a diversas linguagens de programação.

### 3. Módulo de Integração com os Juízes *Online* (MOJO)

O MOJO é a ferramenta, propriamente dita (encapsulada em um módulo que é instalado no Moodle), responsável pela integração. Este módulo é o responsável pela comunicação e interação que irá ocorrer entre o Moodle e os Juízes *Online* envolvidos nas operações. A Figura 1 ilustra como fica a arquitetura final de integração entre os ambientes por meio do MOJO. E a seção seguinte apresenta a arquitetura interna da ferramenta.

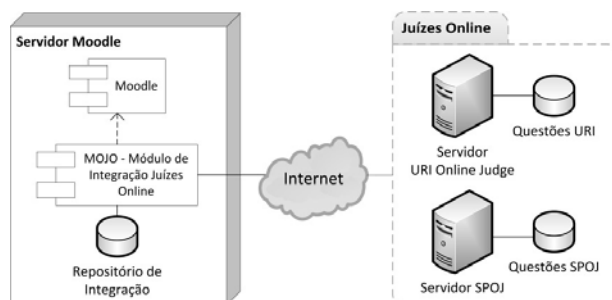


Figura 1 – Integração dos Ambientes com o MOJO

### 3.1 Arquitetura Interna do MOJO

Para que o MOJO garanta a integração entre os dois ambientes (Moodle e Juízes Online), sua arquitetura é composta por dois módulos que se comunicam entre si: o Módulo Principal e o Módulo de Carga e Atualização, além de um Repositório de Integração, detalhados nas subseções seguintes e ilustrados na Figura 2.

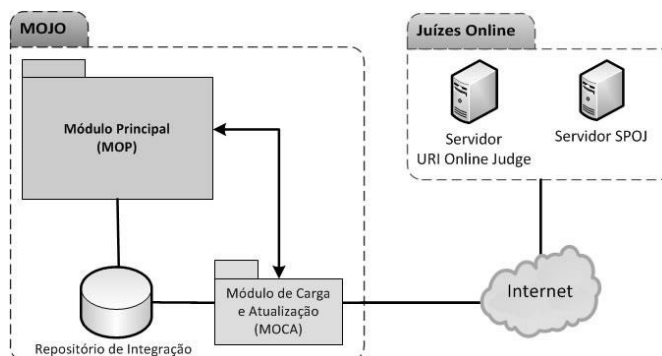


Figura 2 – Arquitetura Interna da Ferramenta

#### 2.2.1 Módulo Principal (MOP)

O MOP, como o próprio nome sugere, é o principal módulo da ferramenta, é ele quem gerencia, controla e fornece todas as funcionalidades necessárias ao MOJO, para disponibilização das questões e dos resultados no Moodle. É o MOP quem controla as ações do Módulo de Carga e Atualização.

#### 2.2.2 Módulo de Carga e Atualização (MOCA)

O MOCA é o responsável pela carga de questões no repositório local (Repositório de Integração) da ferramenta e por sua constante atualização. Em um primeiro momento, o MOJO irá realizar uma carga inicial de questões em seu repositório local. Esta carga é feita pelo MOCA que interage, por meio de requisições *Web*, com o repositório de questões dos Juízes Online.

O MOCA também monitora o repositório de questões dos juízes a fim de saber se novas questões foram adicionadas. Caso existam novas questões, é realizada a atualização no Repositório de Integração do MOJO.

#### 2.2.3 Repositório de Integração

O Repositório de Integração consiste de um conjunto de tabelas específicas, criadas exclusivamente para o MOJO, pois foi uma necessidade identificada para um bom funcionamento das funcionalidades da ferramenta. Estas tabelas interagem diretamente com as tabelas do Moodle, propiciando a integração, alvo desta pesquisa. É neste repositório onde dados como, por exemplo, as questões dos juízes ficarão armazenadas, assim como também o resultado das mesmas.

### 3.2 Processo ESA com o MOJO

Com o repositório carregado com as questões dos juízes, o MOJO vai propiciar uma forma de colaboração, ainda que indireta, entre o Moodle e os Juízes Online, uma vez que estes não interagem diretamente entre si. Neste processo, cada envolvido (professor, aluno e Juiz Online) tem um papel diferente e importante.

O processo ESA, citado no decorrer deste artigo, nada mais é que o processo realizado pelo professor ao Elaborar, Submeter e Avaliar uma atividade de programação. É este processo que a ferramenta automatiza.

Para um melhor entendimento, a Figura 3 ilustra o fluxo do processo ESA com o MOJO, e cada uma de suas etapas é explicada a seguir.

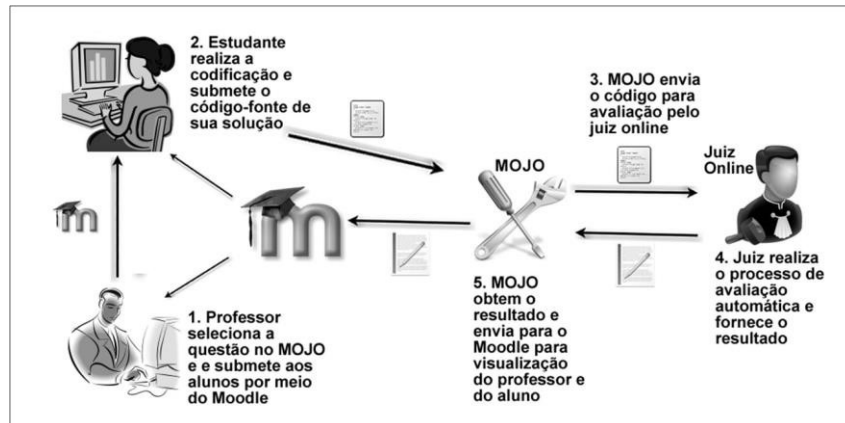


Figura 3 – Fluxo do Processo ESA utilizando o MOJO

De acordo com a Figura 3:

- **Etapa 1:** o professor define a questão e a submete (Elaboração e Submissão) no Moodle para resolução pelos alunos;
- **Etapa 2:** o aluno visualiza, desenvolve e submete uma solução para a questão;
- **Etapa 3:** o MOJO entra em contato com o juiz responsável pela questão e envia o código-fonte para avaliação;
- **Etapa 4:** o Juiz *Online* 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.

Com a possibilidade de visualizar os resultados, o professor poderá realizar o devido acompanhamento de seus alunos. Ele poderá optar por visualizar os resultados por um aluno específico ou visualizar os resultados por questão.

Vale ressaltar que o professor terá acesso aos códigos-fonte submetidos pelos alunos para consulta, e as questões ficam armazenadas no Repositório de Integração para reutilização no futuro.

### 3.3 Tecnologias Utilizadas no Desenvolvimento do MOJO

Devido ao fato de nenhuma API (*Application Programming Interface*) ou serviço *Web* (*Web Service*) ser fornecido por qualquer um dos juízes utilizados nesta versão do MOJO, foi necessário realizar implementações bem específicas para cada um dos juízes. Para estas implementações, e com o objetivo de facilitar a integração, foi utilizada a linguagem de programação PHP (The PHP Group, 2013), mesma linguagem com a qual a plataforma Moodle foi desenvolvida. Em conjunto com a linguagem PHP, também foi utilizada *JavaScript* (W3Schools, 2013).

Para a base de dados optou-se por utilizar o *PostgreSQL* (The PostgreSQL Global Development Group, 2013), por sua característica *open source* (*software* livre). Pois percebeu-se a necessidade de criação de novas tabelas para possibilitar a integração do MOJO ao Moodle, dando origem ao Repositório de Integração. Estas novas tabelas,

ilustradas na Figura 4, tem a finalidade de armazenar informações individualizadas dos alunos, permitindo o correto direcionamento do *feedback* gerado pela ferramenta.

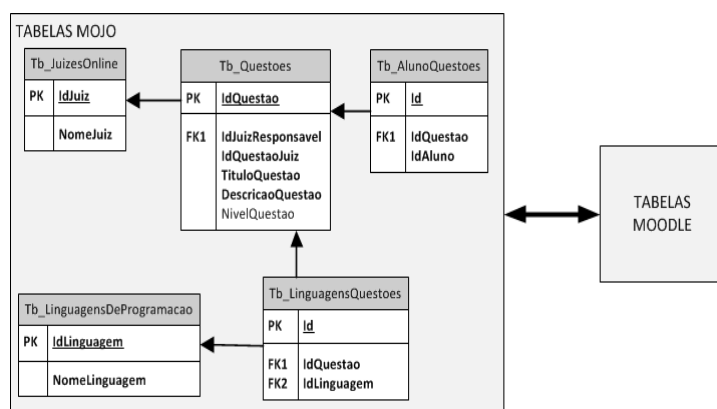


Figura 4 – Tabelas do Repositório de Integração

## 4. Resultados e Considerações Finais

O MOJO tem como objetivo diminuir a sobrecarga de trabalho do professor no processo ESA de atividades de programação. Como resultado, espera-se a melhoria na qualidade do ensino e aprendizagem de programação, tendo em vista que reduzindo a sobrecarga de tarefas do professor, o mesmo poderá fazer uso desse ganho de tempo para realizar um melhor acompanhamento de seus alunos.

Para coleta dos resultados iniciais da ferramenta, foi realizada uma conversa informal com 7 (sete) professores que lecionam alguma disciplina de programação, onde foi apresentado o MOJO e seus principais objetivos. Por meio desta conversa, foram obtidos os resultados mostrados a seguir.

### 4.1 Resultados Iniciais

- i) 71,4% dos professores concordaram que a ferramenta pode reduzir o tempo gasto pelo professor no processo ESA de atividades de programação;
- ii) 28,6% dos professores sentiram falta da possibilidade de elaborarem suas próprias atividades e poderem avaliá-las automaticamente;
- iii) 42,8% dos professores fizeram observações quanto à legibilidade do código;
- iv) 28,6% dos professores fizeram observações sobre a clareza nos resultados retornados pela ferramenta (resposta dos juízes);
- v) 85,7% dos professores concordaram que a ferramenta pode ajudar a fornecer um *feedback* mais rápido ao aluno;
- vi) 100% dos professores concordaram que com o a diminuição na sobrecarga de tarefas do professor, será possível fazer um melhor acompanhamento dos alunos, principalmente àqueles que sentem mais dificuldades.

Pode-se perceber, com base nos resultados obtidos, que o *feedback* dado pela ferramenta deve ser melhorado, assim como deve-se atentar a possíveis problemas na legibilidade do código. Observa-se, ainda, que alguns professores preferem elaborar suas próprias atividades, e que a avaliação automática destas também pode ser bem útil. Estas observações serão tratadas na seção de trabalhos futuros.

Ainda foram realizados alguns testes em laboratório quanto à velocidade em que é obtido o resultado da avaliação das questões. Nestes testes, realizados para ambos os



juízes utilizados na ferramenta, o tempo de resposta era de apenas alguns segundos ou até mesmo menos de um segundo (para problemas mais básicos), mas deve-se frisar que foram testes realizados em laboratórios com um número reduzido de usuários.

Para uma melhor avaliação do desempenho da ferramenta junto a professores e alunos, esforços estão sendo realizados com o intuito de providenciar a disponibilização do MOJO em uma turma regular, com o objetivo de verificar a aplicação prática da ferramenta e possíveis ajustes a serem realizados.

## 4.2 Trabalhos Futuros

Como trabalhos futuros, pretende-se integrar ao MOJO um Laboratório Virtual de Programação, como, por exemplo, o VPL (VPL, 2013). Este laboratório virtual irá auxiliar o aluno na edição e avaliação de seus códigos, além de permitir ao professor elaborar suas próprias atividades e avaliá-las automaticamente. Com este laboratório virtual também se pretende tratar os problemas de legibilidade e do *feedback* retornado na avaliação da atividade, ambos apontados nos resultados deste trabalho. A inclusão de outros Juízes *Online* para ampliar o repositório do MOJO também está prevista.

## Agradecimentos

Os autores agradecem a CAPES e a FAPERN pela concessão das bolsas de pesquisa e pelo apoio financeiro para realização da mesma, 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.

## Referências

- ACM-IPC. **The ACM-ICPC International Collegiate Programming Contest**. Disponível em <<http://icpc.baylor.edu/>> Acesso: ago. de 2013.
- CAMPOS, C. P; FERREIRA, C. E. BOCA: Um sistema de apoio para competições de programação. In: **WORKSHOP DE EDUCAÇÃO EM COMPUTAÇÃO, 12**, Salvador-BA, 2004.
- EDWARDS, S. H. Using software testing to move students from trial-and-error to reflection-in-action. **SIGCSE Bulletin**, v. 36, n. 1, março 2004, pp 26–30.
- FRANÇA, A. B; SOARES, J. M. Sistema de apoio a atividades de laboratório de programação via Moodle com suporte ao balanceamento de carga. In: **SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE), 22**, Aracaju-SE, 2011.
- INC., T. **TopCoder**. Disponível em <<http://www.topcoder.com/>> Acesso: ago. de 2013.
- KUMAR S; GANKOTIYA, A. K; DUTTA, K. A Comparative Study of Moodle with other e-Learning Systems. In: **INTERNATIONAL CONFERENCE ON ELECTRONICS COMPUTER TECHNOLOGY, 3**, Kanyakumari: IEEE, 2011.
- MARTINS, C; GIRAFFA, L. M. M. Capacit@ndo: uma proposta de formação docente utilizando o Moodle. **RENOTE. Revista Novas Tecnologias na Educação**, v.7, p.1-8, 2008.
- MOREIRA, M. P; FAVERO, E. L. Um Ambiente Para Ensino de Programação com Feedback Automático de Exercícios. In: **WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 17**, Bento Gonçalves-RS, 2009.

- MOTA, M. P; PEREIRA, L. W. K; FAVERO, E. L. JavaTool: Uma Ferramenta Para Ensino de Programação. In: **SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE)**, 20, Florianópolis-SC, 2009.
- ONLINEJUDGE. **Onlinejudge**. Disponível em <<https://github.com/hit-moodle/onlinejudge/>> Acesso: ago. de 2013.
- SANDBOX. **Sandbox**. Disponível em <<https://github.com/openjudge/sandbox/>> Acesso: ago. de 2013.
- SANTOS, J. C. S; RIBEIRO, A. R. L. JOnline: proposta preliminar de um juiz online didático para o ensino de programação. In: **SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE)**, 22, Aracaju-SE, 2011.
- SIROTHEAU, S; BRITO, S. R; SILVA, A. S; ELIASQUEVICI, M. K; FAVERO, E. L; TAVARES, O. L. Aprendizagem de iniciantes em algoritmos e programação: foco nas competências de autoavaliação. In: **SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE)**, 22, Aracaju-SE, 2011.
- 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: **SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE)**, 23, Rio de Janeiro-RJ, 2012.
- SPHERE RESEARCH LABS. **IDE ONE**. Disponível em <<http://ideone.com/>> Acesso: ago. de 2013.
- SPHERE RESEARCH LABS. **SPOJ Brasil**. Disponível em <<http://br.spoj.pl/>> Acesso: ago. de 2013.
- THE PHP GROUP. **PHP: Hypertext Preprocessor**. Disponível em: <<http://www.php.net/>> Acesso: maio de 2013.
- THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL**. Disponível em: <<http://www.postgresql.org/>> Acesso: maio de 2013.
- TIMUS ONLINE JUDGE TEAM. **Timus Online Judge**. Disponível em: <<http://acm.timus.ru/>> Acesso: set. de 2013.
- URI ERECHIM. **URI Online Judge**. Disponível em: <<http://www.urionlinejudge.com.br>> Acesso: mar. de 2013.
- VPL. **Virtual Programming Lab**. Disponível em: <<http://vpl.dis.ulpgc.es>> Acesso: set. de 2013.
- W3SCHOOLS. **JavaScript Tutorial**. Disponível em: <<http://www.w3schools.com/js/default.asp>> Acesso: set. de 2013.
- ZHIGANG, S; XIAOHONG, S; NING, Z; YANYU, C. Moodle Plugins for Highly Efficient Programmin Courses. In: **MOODLE RESEARCH CONFERENCE**, 1, Heraklion, Crete-Greece, 2012.