

# MOJO: Uma Ferramenta de Auxílio à Elaboração, Submissão e Correção de Atividades em Disciplinas de Programação

José Osvaldo M. Chaves, Angélica F. Castro, Rommel W. Lima, Marcos Vinicius A. Lima, Karl H. A. Ferreira

Programa de Pós-Graduação em Ciência da Computação (PPgCC)  
Universidade do Estado do Rio Grande do Norte (UERN)  
Universidade Federal Rural do Semi-Árido (UFERSA)  
BR 110 – Km 46 – Bairro Costa e Silva, 59.625-620, Mossoró – RN, Brasil  
oswaldo.mesquita@gmail.com, angelica@ufersa.edu.br,  
rommelwladimir@uern.br, {marcos.engsoft, karlhansimuller}@gmail.com

***Abstract.** This paper proposes a tool that automates the process of preparation, submission and evaluation of practical programming, based on a process similar to that used in marathons programming. The presented environment integrates with Moodle and supports education whether in the classroom or distance learning mode.*

***Resumo.** Este artigo apresenta uma ferramenta que automatiza o processo de elaboração, submissão e correção de atividades práticas de programação, com base em um processo semelhante ao adotado em maratonas de programação. O ambiente apresentado se integra ao Moodle e apoia o ensino quer seja na modalidade presencial ou na modalidade de ensino à distância.*

## 1. Introdução

Durante os primeiros semestres dos cursos de graduação em computação, por exemplo, é observada uma quantidade bem relevante de alunos que reprovam, desistem ou obtêm um baixo rendimento nas disciplinas que focam o estudo de programação. Isso é ocasionado, na maioria dos casos, devido à complexidade do processo que é aprender e desenvolver lógica de programação [Mota *et al.* 2009].

Neste sentido, muitas ferramentas têm sido propostas para auxiliar no ensino de programação. Um exemplo disso é a utilização dos Ambientes Virtuais de Aprendizagem (AVA), e um forte representante desta categoria é o Moodle [Moodle 2013]. Porém, mesmo com o advento dessas ferramentas, algumas barreiras ainda são encontradas pelo professor no ensino de programação, 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 na modalidade de educação presencial, as ferramentas de auxílio existentes atualmente fornecem um ambiente que permite ao aluno criar seus algoritmos e codificá-los em alguma linguagem de programação, porém, é de responsabilidade do professor a elaboração, submissão, correção e fornecimento de *feedback* a todos os seus alunos, o que acaba causando uma sobrecarga de atividades ao professor.

Visando contribuir com a melhoria do processo de ensino e aprendizagem de programação, este artigo apresenta o Módulo de Integração com os Juízes *Online* (MOJO), uma ferramenta que integra os Juízes *Online* (JO) [Kurnia *et al.* 2001] ao Moodle, automatizando os processos de elaboração, submissão e correção das atividades práticas propostas pelo professor. O objetivo é fornecer uma ferramenta que facilite o gerenciamento de recursos e permita o devido acompanhamento dos alunos.

Para um melhor entendimento, o texto está organizado da seguinte forma: a seção 2 apresenta as características do Moodle e dos sistemas de JO, além de citar alguns trabalhos relacionados. A seção 3 apresenta e descreve o MOJO, a arquitetura de integração e o funcionamento da ferramenta e, por último, são apresentadas, na seção 4, as considerações finais e o que se pretende realizar em trabalhos futuros.

## **2. Referencial Teórico**

Nesta seção são descritos os principais ambientes utilizados como base para o desenvolvimento do MOJO, além de enumerar os trabalhos relacionados.

### **2.1. *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 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 [Pulino 2004].

A plataforma, que pode ser utilizada nas modalidades de EaD e de ensino presencial, conta com uma diversidade de recursos educacionais, permitindo larga flexibilidade para configuração e utilização. O seu desenvolvimento extremamente modular permite a fácil inclusão de novos recursos que melhor o adaptem às reais necessidades de quem o utiliza. Neste contexto, a ferramenta oferece a professores e alunos um ambiente capaz de reunir as 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 com outras ferramentas.

### **2.2. *Juízes Online* (JO)**

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 *et al.* 2001].

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

Os JO são muito utilizados em maratonas de programação e podem ser encontrados na Internet, como exemplos podem ser citados o SPOJ Brasil [Sphere

Research Labs 2013] e URI *Online Judge* [URI Erechim 2013], ambos utilizados no desenvolvimento do MOJO. Nesses sistemas são disponibilizados vários problemas a serem resolvidos e submetidos. Dessa maneira, um usuário seleciona a linguagem de programação a ser utilizada na escrita do código e envia sua solução para ser avaliada.

### 2.3. Trabalhos Relacionados

O uso de ambientes virtuais para dar suporte à educação, mais especificamente às atividades práticas de programação, já vem sendo explorado há alguns anos. Em um contexto semelhante ao apresentado neste artigo, alguns trabalhos foram realizados, como é o caso do *BOCA Online Contest Administrator* [Campos e Ferreira 2004] e do *JOnline*, iniciativa de Santos e Ribeiro (2011), ambos explicados a seguir.

O *BOCA Online Contest Administrator* é um sistema que foi desenvolvido para ser utilizado na maratona de programação da Sociedade Brasileira de Computação (SBC). Sua ideia principal é disponibilizar aos alunos e professores um sistema de apoio como os utilizados em maratonas de programação. Ele possui uma interface *Web* em que cada participante possui um login próprio e, utilizando esse login, é possível submeter a resolução de um problema e observar o placar da competição em tempo real, dentre outras características.

O sistema do *JOnline* possui um juiz *online* ao qual é possível acessar problemas para serem resolvidos e submeter códigos-fonte para a visualização do resultado. Além disso, o sistema adiciona funcionalidades didáticas ao juiz – que auxiliam o discente no processo de aprendizagem de programação. Essas funcionalidades são: a apresentação de dicas (em língua portuguesa) para correção de erros de compilação do código enviado, organização de problemas por categoria e grau de dificuldade, votação do nível de dificuldade dos problemas pelos usuários, além da programação colaborativa.

Em geral, as ferramentas de auxílio existentes são produzidas para um fim específico, sem levar em consideração futuras integrações entre ambientes ou a expansão de suas funcionalidades. E, embora os trabalhos aqui citados possam contribuir para auxiliar no ensino e aprendizagem de programação, eles não estão integrados a nenhum AVA, como o Moodle, por exemplo.

Neste artigo, em complemento aos trabalhos aqui citados, é apresentado um ambiente que, integrando os JO ao Moodle, fornece de forma coesa o auxílio necessário ao professor no que diz respeito à elaboração, submissão e correção de atividades de programação, resultando em maior agilidade nas atividades do professor, ganho de tempo na elaboração das questões submetidas e um *feedback* mais rápido ao aluno. O professor poderá se aproveitar do ganho de tempo para dar uma maior atenção aos seus alunos, reduzindo assim a desmotivação e abandono do curso pelos mesmos.

## 3. Módulo e Integração

Um dos desafios mais cansativos, e demorados, enfrentados por um professor de programação é a elaboração e correção das atividades práticas de seus alunos, pois o professor muitas vezes trabalha com mais de uma turma e cada turma contendo vários alunos e, além disso, ainda precisa gerenciar seu tempo para correção de trabalhos, provas, seminários, dentre outras tarefas.

Neste sentido, este artigo apresenta uma ferramenta (MOJO) que funciona como um novo módulo integrado ao Moodle, visando auxiliar o professor no processo de elaboração, submissão e correção de questões e no rápido *feedback* sobre o resultado das mesmas, proporcionando um ambiente coeso onde estejam integrados o Moodle e os JO. Para um melhor entendimento, na Figura 1 é mostrada a arquitetura do ambiente.

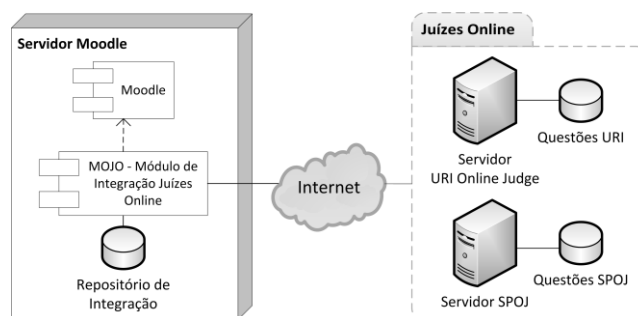


Figura 1. Arquitetura do ambiente

Ainda conforme a Figura 1 é possível observar de um lado os JO em seus respectivos servidores e com seus repositórios de questões. E do outro lado, no servidor Moodle, encontra-se o MOJO com seu próprio repositório local de questões.

### 3.1. Módulo de Integração com os Juizes *Online* (MOJO)

O MOJO é a ferramenta, propriamente dita (encapsulada em um módulo), responsável pela integração. Este módulo é o responsável pela comunicação e interação que irá ocorrer entre o Moodle e os JO envolvidos nas operações.

Em um primeiro momento, o MOJO irá fazer uma carga inicial em seu repositório local de questões. O módulo fará isso por meio de requisições *Web* onde serão obtidas as questões de cada um dos JO. Com as questões armazenadas em sua base de dados, o MOJO fornecerá as informações necessárias para utilizar esses dados como atividades no Moodle.

Com o módulo carregado com as questões dos juizes, o professor poderá visualizar uma tela com a lista de questões disponíveis e selecionar a mais apropriada para submeter a seus alunos. E para a correção das soluções enviadas pelos alunos, o MOJO fará nova requisição *Web* ao juiz responsável pela questão, enviando o código do aluno e recuperando o resultado da avaliação feita pelo juiz. Por fim, disponibilizando este resultado para visualização no Moodle.

### 3.2. Funcionamento da Ferramenta

O funcionamento da ferramenta ocorre de forma distinta para cada um dos diferentes envolvidos no processo: o professor, o aluno e o Juiz *Online*. Este funcionamento é explicado a seguir:

- **Para o professor:** (i) o professor, por meio do Moodle, irá acessar o MOJO, onde encontrará as diversas questões organizadas em uma lista, e ao clicar na questão poderá conferir todos os dados relativos à questão, como por exemplo, nome do juiz responsável, código e descrição da questão; (ii) o professor seleciona a questão (que contém em sua descrição todas as informações necessárias para a sua realização), e

então a submete para resolução pelos alunos; (iii) em posse do resultado da avaliação o professor poderá dar o devido *feedback* ao aluno.

- **Para o aluno:** (i) o aluno irá acessar o Moodle e verificar a existência de novas questões; (ii) existindo uma nova questão, o aluno realiza a codificação do problema, e depois de terminada, envia sua resposta para avaliação.
- **Para o Juiz *Online* responsável pela questão:** (i) após o aluno submeter sua resposta da questão, o módulo de integração entra em contato com o Juiz *Online* responsável pela questão respondida e envia a resposta para avaliação; (ii) o Juiz *Online* responsável realiza os devidos processos de avaliação automática para a questão e devolve o resultado da avaliação realizada.

É importante lembrar que o professor terá acesso aos códigos submetidos pelos alunos para consulta e que as questões ficarão armazenadas no repositório local para posterior reutilização.

### 3.3. Requisitos necessários à Integração

No que diz respeito à base de dados, percebeu-se a necessidade de criação de novas tabelas para a base de dados do MOJO, sendo essas novas tabelas responsáveis por interagir com as tabelas já existentes do Moodle. A Figura 2 exibe um modelo onde estão identificadas as novas tabelas que precisaram ser criadas, e suas interações.

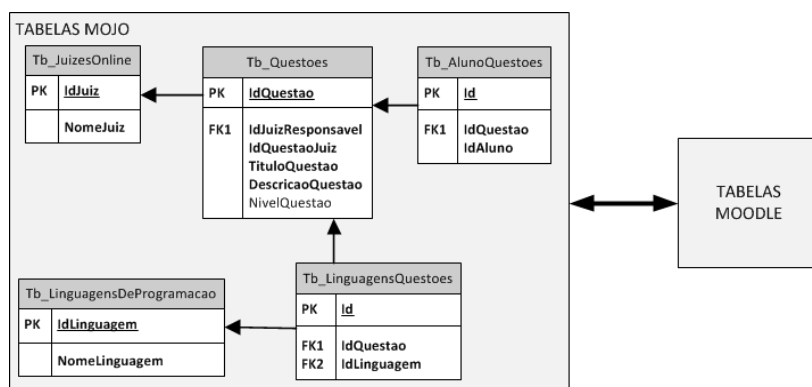


Figura 2. Interação entre as novas tabelas criadas

## 4. Considerações Finais

A integração dos JO com o Moodle, por meio do MOJO, visa diminuir consideravelmente a sobrecarga de trabalho na correção de códigos-fonte por parte dos professores, bem como reduzir o tempo necessário para correção e apresentação dos resultados das atividades desenvolvidas pelos alunos. Como resultado, espera-se a melhoria na qualidade do ensino e aprendizagem das disciplinas de programação, tendo em vista que, o tempo do professor com atividades de administração e de gestão de recursos pode ser reduzido, com isso, espera-se uma maior disponibilidade para dar uma maior atenção ao aluno.

Além disso, apesar de outros fatores poderem ser responsáveis por desistências nos cursos, considera-se que, com o aumento da disponibilidade de tempo, o professor

poderá atuar de maneira a evitar o desestímulo ou eventual sensação de abandono por parte do aluno.

Devido sua recente concepção, o MOJO ainda não passou por uma avaliação prática (fora do âmbito laboratorial), porém, já se trabalha em uma forma de avaliar seu desempenho junto a alunos e professores. A avaliação se dará por meio da utilização da ferramenta em uma turma regular que esteja cursando alguma disciplina de programação, obtendo, assim, os dados relativos à sua utilização para posterior divulgação e publicação dos mesmos.

Como trabalhos futuros pretende-se integrar ao MOJO um editor de código-fonte para que o aluno possa desenvolver seu código no próprio módulo, além de uma ferramenta de avaliação automática, que auxilie o aluno na edição e avaliação de seus códigos, onde o próprio aluno poderá obter um *feedback* antes mesmo de submeter seu código para avaliação pelo juiz. Além de, também, permitir ao professor elaborar suas próprias questões, caso ele opte por não utilizar as questões do MOJO.

### **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 PPgCC da UERN e UFERSA, por toda infraestrutura oferecida.

### **Referências**

- Campos, C. P; Ferreira, C. E. (2004) “BOCA: Um sistema de apoio para competições de programação”. In: Workshop de Educação em Computação, Anais da Sociedade Brasileira de Computação, Salvador-BA, 2004.
- Kurnia, A; Lim, A.; Cheang, B. (2001) “*Online Judge*”. Computer & Education, v. 36, n. 4, maio 2001, pp 299-315.
- MOODLE. (2013) “*Modular Object-Oriented Dynamic Learning Environment*”. Disponível em: <<http://moodle.org/>>. Acesso em 16 de mar de 2013.
- Mota, M. P; Pereira, L. W. K; Favero, E. L. (2009) “*JavaTool: Uma Ferramenta Para Ensino de Programação*”. In: Anais do XX Simpósio Brasileiro de Informática na Educação. Florianópolis-SC, 2009.
- Pulino, F. A. R. (2004) “Introdução ao Moodle - Ambiente de Aprendizagem (Módulo 1)”. Disponível em: <[http://ead.faculdamarista.com.br/file.php/1/modulo01-moodle\\_1.pdf](http://ead.faculdamarista.com.br/file.php/1/modulo01-moodle_1.pdf)>. Acesso em 15 de mar de 2013.
- Santos, J. C. S; Ribeiro, A. R. L. “*JOnline: proposta preliminar de um juiz online didático para o ensino de programação*”. In: Anais do XXII Simpósio Brasileiro de Informática na Educação. Aracaju-SE, 2011.
- Sphere Research Labs. (2013) “SPOJ Brasil”. Disponível em <<http://br.spoj.pl/>>. Acesso em 16 de mar de 2013.
- URI Erechim. (2013) “*URI Online Judge*”. Disponível em: <<http://www.urionlinejudge.com.br>>. Acesso em 16 de mar de 2013.