

Uma Ferramenta de Auxílio ao Processo de Ensino-Aprendizagem para Disciplinas de Programação de Computadores

José Osvaldo Chaves,
Angélica Castro
LABSIM - PPgCC - UERN
Av. Francisco Mota, 572
Mossoró-RN-Brasil
+55-84-3317-8303

oswaldo.mesquita@gmail.com

Rommel Lima, Marcos
Vinicius Lima, Karl Ferreira
LORDI - PPgCC - UERN
BR110 - KM 48, R. Prof Ant. Campos
Mossoró-RN-Brasil
+55-84-3315-2196

mcc@uern.br

Agebson Façanha
Núcleo de Acessibilidade - IFCE
Av.13 de maio, 2081
Fortaleza-CE-Brasil
+55-85-3307-3791

agebson@ifce.edu.br

ABSTRACT

Online Judges are commonly used in programming marathons. They are famous for their repository of questions and automatic evaluation of source code. The idea of integrating these systems into Moodle came from the demand for more professor involvement in programming disciplines, and as a way to provide a greater number of issues for use. Usually the professor is overwhelmed by preparing, submitting, evaluating and providing the necessary feedback to the students of the issues - which can negatively impact the care given to the classes. Aiming to reduce this overhead, this paper presents MOJO, a tool to automate this process.

RESUMO

Os sistemas de Juízes *Online* são comumente utilizados em maratonas de programação. Eles são famosos pelo seu repositório de questões e avaliação automática de código-fonte. A ideia de integrar estes sistemas ao Moodle surgiu da exigência de um maior envolvimento do professor de disciplinas de programação, e como uma forma de disponibilizar um maior número de questões para utilização. Normalmente o professor fica sobrecarregado por elaborar, submeter, avaliar e fornecer o *feedback* necessário das questões aos os alunos – o que pode impactar negativamente na assistência dada às turmas. Visando diminuir esta sobrecarga, este artigo apresenta o MOJO, uma ferramenta para automatização desse processo.

Categories and Subject Descriptors

K.3.1 [Computers and Education]: Distance Learning

General Terms

Experimentation, Human Factors.

Keywords

MOJO, Programming Disciplines, Teaching, Learning.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

TISE 2013, December 9–11, 2013, Porto Alegre, RS, Brazil.

Copyright 2010 ACM 1-58113-000-0/00/0010 ...\$15.00.

1. RESUMO EXPANDIDO

As Tecnologias de Informação e Comunicação (TIC) têm contribuído de maneira significativa, ampliando as estratégias e proporcionando diferentes ferramentas e artefatos que apoiam o processo de ensino-aprendizagem [1]. A utilização dessas tecnologias têm sido de suma importância para auxiliar professores e alunos. Podem-se citar exemplos de sua utilização em disciplinas de programação de computadores, disciplinas essenciais em cursos de computação. Tais ferramentas, se bem aplicadas, servem de auxílio para facilitar o aprendizado de programação, reduzindo os índices de reprovação e o mau desempenho em outras disciplinas relacionadas.

Neste sentido, muitas ferramentas de TIC têm sido propostas para auxiliar o professor. Um exemplo prático disso é a utilização dos Ambientes Virtuais de Aprendizagem (AVA), e um forte representante desta categoria é o Moodle [2]. Porém, mesmo com o advento dessas ferramentas, alguns desafios ainda são encontrados pelo professor no ensino de programação como, por exemplo, a dificuldade de avaliar todos os exercícios de uma turma extensa de alunos em pouco tempo.

No que diz respeito à avaliação dos exercícios de programação, os sistemas de Juízes *Online* [3] são comumente utilizados em maratonas de programação, e conhecidos por seu amplo repositório de questões e pelo seu processo de avaliação automática de código-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 [4].

Em geral, quer seja na modalidade de Educação a Distância (EaD) ou na modalidade de educação presencial, os sistemas 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 das questões, avaliação e fornecimento de *feedback* ao aluno. Todo este processo, quando realizado para vários alunos em várias turmas, acaba causando desgaste (físico e mental) ao professor e uma sobrecarga de suas atividades.

Visando contribuir com a melhoria do processo de ensino-aprendizagem de programação, este artigo apresenta o Módulo de Integração com os Juízes *Online* (MOJO), uma ferramenta que integra os Juízes *Online* ao Moodle, fornecendo a praticidade de um AVA junto com um sistema de avaliação automática. O ambiente automatiza os processos de elaboração, submissão e avaliação das atividades práticas propostas pelo professor. O

objetivo é fornecer uma ferramenta que facilite o gerenciamento de recursos e permita realizar o devido acompanhamento dos alunos.

MOJO é um acrônimo para Módulo de Integração com os Juízes Online [5]. Este módulo é o responsável pela comunicação e interação que irá ocorrer entre o Moodle e os Juízes Online envolvidos nas operações.

Para garantir a integração entre os dois ambientes (Moodle e Juízes Online), a arquitetura interna do MOJO é composta por dois módulos que se comunicam entre si e controlam todas as funcionalidades da ferramenta: o Módulo Principal e o Módulo de Carga e Atualização, além de um Repositório de Integração, detalhados a seguir. A Figura 1 ilustra esta arquitetura interna, identificando os componentes do MOJO e como eles se interligam aos Juízes.

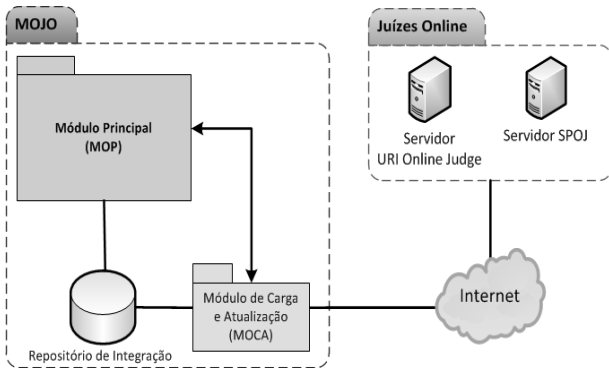


Figura 1. Arquitetura Interna do MOJO.

- **Módulo Principal (MOP):** 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.
- **Módulo de Carga e Atualização (MOCA):** responsável pela carga de questões no Repositório de Integração da ferramenta e por sua constante atualização. Em um primeiro momento, o MOJO irá fazer uma carga inicial em seu repositório. Esta carga é feita pelo MOCA que interage diretamente, por meio de requisições Web, com o repositório dos Juízes Online. O MOCA também monitora o repositório de questões dos juízes a fim de verificar se novas atividades foram adicionadas; em caso afirmativo, é realizada a atualização no repositório do MOJO.
- **Repositório de Integração:** consiste na base de dados exclusiva criada para o MOJO, pois foi identificada a necessidade de tabelas específicas para suas funcionalidades. Estas tabelas interagem diretamente com as tabelas do Moodle propiciando a real integração (objetivo da ferramenta). É neste repositório onde dados como, por exemplo, as questões dos juízes ficarão armazenadas, assim como também o resultado das mesmas.

O MOJO vai propiciar uma forma de colaboração entre estes dois sistemas Web, ainda que indiretamente, já que o Moodle e os Juízes Online não interagem diretamente entre si. Cada envolvido no processo (professor, aluno e Juiz Online) tem um papel

diferente e importante no processo de avaliação das questões. A Figura 2 ilustra o fluxo deste processo.

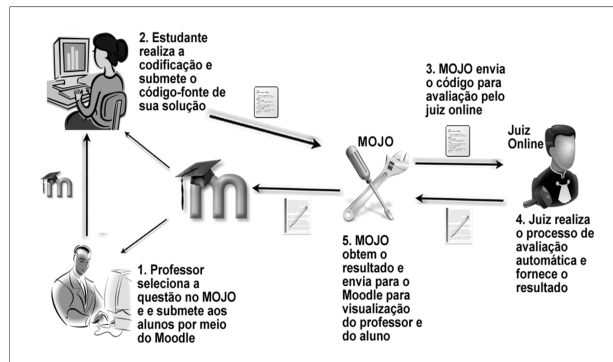


Figura 2. Fluxo do Processo com o MOJO.

Conforme a Figura 3, o processo se inicia com o professor definindo a questão (selecionada no MOJO) e a submetendo no Moodle para resolução pelos alunos (passo 1). Após submissão da questão, o aluno poderá visualizá-la e preparar sua solução (passo 2). Após submissão da solução pelo aluno, o MOJO entra em contato com o juiz responsável pela questão e envia o código-fonte para avaliação (passo 3). Em seguida, o Juiz Online realiza os devidos processos de avaliação automática para a solução proposta e devolve o resultado (passo 4).

No último passo do processo, o MOJO obtém o resultado da avaliação e o disponibiliza no Moodle para visualização do professor e do aluno (passo 5), conforme mostrado na Figura 3. Na tela de visualização dos resultados é possível observar o código, título da questão, resultado da submissão (aceito ou não-aceito), além da linguagem de programação selecionada e da data de submissão. O professor ainda poderá optar por visualizar os resultados de um aluno em específico ou visualizar os resultados de todos os alunos por questão.

#	Problema	Resposta	Linguagem	Time	Data Submissão
219242	1001	Extremamente Básico	Accepted	Java	0.096 08/08/2013 - 19:32:58
219185	1001	Extremamente Básico	Accepted	C++	0.000 08/08/2013 - 18:39:24
219179	1001	Extremamente Básico	Accepted	Java	0.096 08/08/2013 - 18:27:32
219173	1001	Extremamente Básico	Accepted	C++	0.000 08/08/2013 - 18:17:18
219170	1001	Extremamente Básico	Accepted	Java	0.092 08/08/2013 - 18:14:26
219162	1001	Extremamente Básico	Accepted	C++	0.000 08/08/2013 - 18:03:14

Figura 3. Tela de Visualização dos Resultados.

Com a possibilidade de visualizar os resultados, o professor poderá realizar o devido acompanhamento de seus alunos. Vale ressaltar que o professor terá acesso aos códigos submetidos pelos alunos para consulta, e as questões ficam armazenadas no repositório local para reutilização no futuro.

Devido ao fato de nenhuma API (Application Programming Interface) ou serviço Web (Web Service) ser fornecido por qualquer um dos juízes utilizados nessa versão do MOJO, foi necessário fazer implementações bem específicas para cada um

dos juízes individualmente. Para estas implementações e visando facilitar a integração com a plataforma Moodle, foi utilizada a linguagem de programação PHP [6], mesma linguagem utilizada no desenvolvimento da plataforma Moodle. Em conjunto com a linguagem PHP, também foi utilizada *JavaScript* [7].

Para a base de dados optou-se por utilizar o PostgreSQL [8], por sua característica *open source* (*software* livre). Ainda, percebeu-se a necessidade de criação de novas tabelas para possibilitar a integração do MOJO ao Moodle. Estas novas tabelas, mostradas na Figura 4, tem a finalidade de armazenar informações individualizadas dos alunos, permitindo o correto direcionamento do *feedback* gerado pela ferramenta.

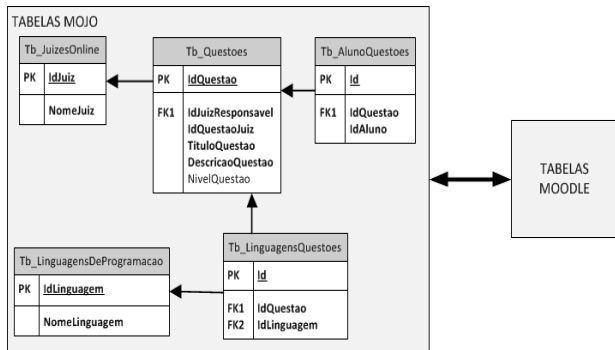


Figura 4. Tabelas do Repositório de Integração.

Por se tratar de uma ferramenta integrada ao Moodle, o ambiente pode ser utilizado em vários sistemas operacionais como, por exemplo, Windows e Linux, sendo executado a partir do próprio *browser* cliente.

Para coleta dos primeiros resultados da ferramenta, foi realizada uma conversa informal com 7 (sete) professores que lecionam disciplinas de programação, onde foi apresentado o MOJO e seus principais objetivos. Por meio dessa conversa, foram obtidos os seguintes resultados mostrados na Tabela 1.

Tabela 1. Resultados Iniciais.

Porcentagem	Significado
71,4%	Professores que concordaram que a ferramenta pode reduzir o tempo gasto pelo no processo de elaboração, submissão e avaliação de atividades.
28,6%	Professores que sentiram falta da possibilidade de elaborarem suas próprias atividades e terem as mesmas avaliadas automaticamente.
42,8%	Professores que fizeram observações sobre a legibilidade do código.
28,6%	Professores que fizeram observações sobre a clareza nos resultados retornados pela ferramenta (resposta dos juízes).
85,7%	Professores que concordaram que a ferramenta pode ajudar a fornecer um <i>feedback</i> mais rápido ao aluno.
100%	Professores que concordaram que com o a diminuição na sobrecarga de tarefas do professor, será possível fazer um melhor acompanhamento dos alunos.

Conforme a Tabela 1, é possível perceber, com base nos resultados dos dados coletados, que o *feedback* dado pela

ferramenta deve ser melhorado, assim como deve-se atentar para possíveis problemas de legibilidade do código. Observa-se, ainda, que alguns professores preferem elaborar suas próprias atividades, mas que a avaliação automática destas também pode ser bem útil.

Ainda foram realizados testes em laboratório quanto à velocidade que é dado o resultado das questões pelos juízes. Nos 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 foram testes realizados em laboratórios com um número reduzido de usuários.

Para uma melhor avaliação do desempenho da ferramenta junto a professores e alunos, já se trabalha, prioritariamente, no intuito de disponibilizar o MOJO em uma turma regular, com o objetivo de verificar a aplicação prática da ferramenta e possíveis ajustes a serem realizados.

O principal objetivo do MOJO ao integrar os Juízes Online com o Moodle é fornecer um ambiente *Web* coeso que venha diminuir a sobrecarga de trabalho do professor na correção de atividades práticas de programação, bem como reduzir o tempo necessário para apresentação dos resultados dessas atividades.

Como resultado, espera-se a melhoria na qualidade do ensino-aprendizagem de programação de computadores, tendo em vista que o tempo do professor com atividades administrativas e de gestão de recursos pode ser reduzido, e com isso ele poderá fazer uso desse ganho de tempo para realizar um melhor acompanhamento de seus alunos.

Além disso, apesar de outros fatores poderem ser responsáveis pelas reprovações e desistências em cursos que têm programação como disciplina essencial, 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, evitando, assim, a possível evasão do curso.

Conforme dito anteriormente, o MOJO ainda não passou por uma avaliação prática (fora do âmbito laboratorial), porém, já se trabalha neste intuito. A avaliação deve ocorrer 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.

Como trabalhos futuros pretende-se integrar ao MOJO um Laboratório Virtual de Programação, a exemplo do VPL [9], que além de auxiliar o aluno na edição e avaliação de seus códigos, também permitirá ao professor elaborar suas próprias atividades e avaliá-las automaticamente. Com este laboratório virtual também se pretende melhorar os problemas de legibilidade e do *feedback* retornado na avaliação da atividade. A inclusão de outros Juízes *Online* também está prevista, ampliando ainda mais o repositório de questões do MOJO.

2. REFERÊNCIAS

- [1] Amaral, E. M. et al. 2011. *Laboratório Virtual de Aprendizagem: Uma Proposta Taxonômica*. In: CINTED-UFRGS, V. 9, n. 2.
- [2] Kumar, S. et al. 2011. A Comparative Study of Moodle with other e-Learning Systems. In: *International Conference on Electronics Computer Technology – ICECT*, 3. Kanyakumari: IEEE, 2011.

- [3] Zhigang, S., Xiaohong, S., Ning, Z., Yanyu, C. 2012. *Moodle Plugins for Highly Efficient Programmin Courses*. In: Proceeding of 1st Moodle Research Conference.
- [4] Campos, C. P., Ferreira, C. E. 2004. *BOCA: Um sistema de apoio para competições de programação*. In: XII Workshop sobre Educação em Computação, (WEI 2004), Salvador/BA.
- [5] Chaves, J. O. et al. 2013. MOJO: Uma ferramenta para auxiliar o professor em disciplinas de programação. In: *X Congresso Brasileiro de Ensino Superior a Distância (ESUD 2013)*, de 11 a 13 de jun/2013, Belém/PA.
- [6] The PHP Group. 2013. *PHP: Hypertext Preprocessor*. Disponível em: <http://www.php.net>. Acesso em 17 maio de 2013.
- [7] W3Schools. 2013. *JavaScript Tutorial*. Disponível em: <http://www.w3schools.com/js/default.asp>. Acesso em 13 jun de 2013.
- [8] The PostgreSQL Global Development Group. 2013. *PostgreSQL*. Disponível em: <http://www.postgresql.org>. Acesso em 17 maio de 2013.
- [9] VPL. 2013. *Virtual Programming Lab*. Disponível em: <http://vpl.dis.ulpgc.es/>. Acesso em 21 jul de 2013.