



FERRAMENTAS ONLINE QUE APOIAM O ENSINO E A APRENDIZAGEM DE LINGUAGENS DE PROGRAMAÇÃO NA EDUCAÇÃO A DISTÂNCIA

Rosemary Pessoa Borges; Carla Katarina de Monteiro Marques; Rommel Wladimir de Lima

Programa de Pós-Graduação em Ciência da Computação – PPgCC UERN/UFERSA – Mossoró, RN – Brasil

marybezerraleao@gmail.com; carla.katarina@gmail.com; rommel.lima@gmail.com

Resumo: O ensino e a aprendizagem das linguagens de programação é um tema que vem sendo continuamente discutido. Espera-se que as disciplinas dessa área ocorram de forma dinâmica, porém, nem sempre os professores conseguem realizar um acompanhamento eficiente. Isso ocorre pela complexidade intrínseca a análise de códigos, além do fato de que cada discente tem seu ritmo de desenvolvimento. Se em turmas presenciais é possível identificar essas dificuldades, em turmas a distância, onde há separação física e temporal, a problemática se intensifica. Estudos apontam que nessas disciplinas os alunos tendem a ter maior índice de reprovação, o que leva a desmotivação para continuar no curso ou mesmo na área. Existem várias hipóteses que são exploradas com objetivo de melhorar o desempenho e motivação dos discentes. Dentre essas focamos aqui na hipótese que defende que um rápido retorno sobre as atividades desenvolvidas mantém o interesse e permite aos alunos ter tempo hábil para avaliar seus pontos falhos e assim poder investir neles e melhorar seu desempenho. Com base nessa hipótese, foram realizadas pesquisas sobre ferramentas que auxiliam o professor na avaliação de código de forma automatizada, reduzindo assim a sobrecarga do docente no acompanhamento dos alunos e agilizando o retorno necessário. Esse trabalho se constitui em uma pesquisa preliminar que objetiva integrar ferramentas como essas ao Moodle, de forma transparente aos professores e alunos. Aqui será apresentado o resultado dessa pesquisa exploratória sobre as ferramentas que realizam avaliação de código e que já foram utilizadas de forma integrada em algum ambiente de ensino/aprendizagem.

Palavras-chave: ferramentas *online*, ensino de programação, retorno automatizado, educação a distância.

Introdução

Em cursos da área de computação, há sempre a oferta de disciplinas sobre linguagens de programação, pois, essas são disciplinas compostas de conteúdos que constituem a base para muitas linhas dentro da área de informática, capacitando os alunos com relação a lógica e a resolução de problemas (Mutiaiwani e Juwita, 2014). Diversas pesquisas revelam que nessas disciplinas os alunos tendem a ter maior índice de reprovação, o que os levam a desmotivação para continuar na disciplina, no curso ou até mesmo na área. Por isso, o ensino de linguagens de programação é um tema que merece atenção, levando os professores e equipe pedagógica ao questionamento sobre quais procedimentos, técnicas e ferramentas podem auxiliar no ensino e aprendizagem dessa disciplina.



Se em turmas presenciais já é possível visualizar essas dificuldades, em turmas a distância, onde há separação física e temporal, onde nem sempre os diálogos são completamente compreendidos por todos, a problemática se intensifica.

Com isso, a discussão sobre essa questão se torna fundamental para o sucesso dos discentes que cursam essas disciplinas. Dessa forma, é importante analisar metodologias que garantam o sucesso do ensino de linguagens de programação, além de colocá-las em prática e realizar constante avaliação dos resultados. Nesse contexto, uma questão é suscitada: como é possível acompanhar e avaliar o desempenho do aluno nesse cenário, sabendo que a carga de trabalho dos professores é intensificada uma vez que o código é um elemento complexo e cada aluno tem seu ritmo de desenvolvimento?

Uma das hipóteses defendidas é de que o retorno influencia diretamente no desempenho e motivação dos alunos no ensino presencial e, mais ainda, nos cursos a distância. Hricko (2002), destaca que as frustrações dos alunos de cursos a distância são originárias da falta de retorno em tempo hábil, instruções ambíguas e descentralizadas.

Outros autores, como Salomão e Watanabe (2013), com base em suas pesquisas, afirmam que os docentes devem ter um conjunto de competências para manter seus alunos motivados, dentre essas competências destacam-se a presteza quanto às atividades em geral (esclarecendo as dúvidas) e o retorno das atividades solicitadas (proporcionando *retorno*).

Segundo Fluminhan et al. (2014), na educação a distância o retorno é aguardado ansiosamente pelos alunos, muito mais do que em outras modalidades de ensino. Quando o docente não registra o retorno, os alunos interpretam negativamente, levando-os a desmotivação e até mesmo a evasão do curso.

Partindo dessa linha, há diversas opções de metodologias que proporcionam um retorno rápido aos alunos. Uma delas propõe a utilização de ferramentas automatizadas, como os juízes *online*. As ferramentas pertencentes a essa categoria realizam o papel de avaliadores de códigos de forma que, uma vez submetido o problema, o juiz *online* analisa um conjunto de dados de entrada/saída e compara com as respostas esperadas. Em seguida, exhibe o resultado da avaliação possibilitando que o discente reelabore e envie novamente o código.



Diante do contexto apresentado, tem-se como objetivo explorar nesse trabalho as ferramentas que realizam a correção automatizada de código, como os juízes *online*, bem como as funções específicas de cada uma delas. Caracterizando essas ferramentas, temos subsídios para a posterior escolha da que mais se adequa a integração com a plataforma Moodle.

Para atingir esse propósito, esse trabalho está organizado da seguinte forma: primeiramente é apresentada a metodologia utilizada nessa pesquisa e seu objetivo final. Em seguida, há uma explanação sobre o conceito fundamental de juízes *online*, para que se possa ter entendimento sobre como funcionam as ferramentas pertencentes a essa categoria. Após esse tópico, serão apresentadas ferramentas, que se enquadram na categoria de juízes online, ou que forneçam funções semelhantes. Por fim, serão apresentadas, de forma sucinta, as características gerais que mais se destacam.

Metodologia

Nesse trabalho será apresentado o resultado de uma pesquisa preliminar que tem por objetivo integrar ferramentas, que realizam a avaliação automatizada de código, ao ambiente de aprendizagem Moodle. Para isso, foi executada uma pesquisa exploratória, buscando identificar ferramentas que fornecem tal serviço, avaliando as principais características e destacando informações relevantes que possam auxiliar na escolha de uma delas. No primeiro momento, foi realizada uma pesquisa a fim de identificar as ferramentas, que realizam a avaliação automatizada de código, que já tenham sido utilizadas em conjunto com a plataforma Moodle. Para compreender de que forma cada uma dessas ferramentas fornece o serviço, foi necessário realizar a simulação de uso, através de uma análise exploratória da ferramenta. Os resultados desse processo são apresentados na próxima seção.

Juízes online

Juízes online são aplicações que recebem código do usuário, em uma linguagem de programação previamente definida. É responsável por compilar, executar e apresentar o resultado obtido a partir do código enviado. Para a realização desse processo, os juízes online podem receber dados de entrada para execução do código, em seguida realizam o processamento e por fim, comparam os resultados obtidos com aqueles esperados. As respostas enviadas pelos juízes online são padronizadas com base nessas comparações, podem ser do tipo certo, errado, erro de compilação ou de execução, entre outras (Chaves, 2014).



Diversos juízes online estão disponíveis de uso gratuito ou não, cada um possuindo suas especificidades implementadas em forma de serviços. Nesse trabalho foram abordados as seguintes: BOCA *Online Contest Administrator*, Ideone, ProgTest, SOPJ BR, URI *Online Judge* e UVA *Online Judge*.

Ferramentas online para correção automática de código

Existem diversas ferramentas online que tem por objetivo fornecer suporte a correção automática de código. Parte delas se propõe a fornecer serviços e funcionalidades através de interfaces acessíveis pela *web*. Outras, buscam reutilizar serviços acoplando novas funcionalidades as já existentes, proporcionando assim novas aplicações. Nessa seção, serão apresentadas algumas dessas ferramentas, destacando e classificando de acordo com as características gerais.

Embora as ferramentas pesquisadas tenham funcionalidades em comum, como a avaliação automática de código, observou-se que possuem objetivos e aplicabilidades distintas. Dessa forma, cada ferramenta possui suas especificidades implementadas em forma de serviços.

A ferramenta BOCA, por exemplo, é um sistema de apoio a competições de programação que conta com juízes online para executar o processo de julgamento das submissões. Ele auxilia no controle de variáveis importantes durante a competição, como o tempo para submissão, controle de entrada e saída de dados, linguagem de programação, submissão de código e avaliação final (Campos, 2011).

Já a Ideone foca em fornecer a funcionalidade de compilação online de código, de forma segura executando o código em um servidor remoto. Seu uso é restrito ao acesso manual através de um navegador não sendo autorizado, pelo seu mantenedor, o uso massivo da ferramenta. Dessa forma há um limite para o envio de código fonte e ao atingir esse limite, o usuário deverá adquirir uma licença de uso para continuar a usufruir da ferramenta. Apesar dessa limitação, ela se destaca pelo amplo suporte a diversas linguagens de programação e por proporcionar interface simples e intuitiva (Ideone, 2013).

É possível ainda combinar diversas bibliotecas em uma aplicação para obter serviços mais amplos. Como exemplo tem-se o conjunto de bibliotecas Sandbox que foram concebidas para serem utilizadas como um módulo de um sistema de juiz *online* (Sandbox, 2016).



Algumas ferramentas funcionam como repositório de questões, como é o caso da SPOJ BR, da URI *Online Judge* e da UVA *Online Judge*. A SPOJ BR é uma versão brasileira do *Sphere Online Judge* (SPOJ), e disponibiliza um repositório de problemas regionais, de seletivas, de olimpíadas e maratonas de programação, em português. Além disso, a SPOJ BR oferece fóruns de discussão, *ranking* de classificação, além de *links* de notícias e informações (SPOJ BR, 2014).

A URI *Online Judge* (2013), além de algumas funcionalidades semelhantes ao SPOJ BR, disponibiliza um módulo para professores onde se pode criar disciplinas limitando, ou não, seu acesso a determinado grupo. É possível ainda disponibilizar listas de exercícios e realizar o acompanhamento do progresso dos estudantes, possibilitando retorno em tempo real.

Apesar da ferramenta UVA *Online Judge* (2016) fornecer funcionalidades semelhantes as da SPOJ BR e da URI *Online Judge*, não há possibilidade do professor inserir novas questões no repositório, ficando limitado as questões fornecidas pela ferramenta. Em contra partida, não limita a quantidade de usuários nem de submissões, inclusive, de acordo com o aumento da demanda, a correção será atribuída a múltiplos servidores para atender a todos em tempo razoável.

Existem ferramentas que permitem ao professor e/ou aluno criarem suas próprias definições de testes para avaliação do código, é o caso da ProgTest (Souza et al., 2011) que utiliza um programa referência, também denominado oráculo, que deve ser fornecido pelo professor para avaliação das atividades dos alunos, bem como casos de teste, utilizando diferentes ferramentas para testes como JUnit e CUnit. Fornece também um módulo para professores, onde se pode criar disciplinas e associar alunos a essas disciplinas e direcionar as atividades.

Resultados

Na tabela 1, é possível ter uma visão geral das ferramentas *online* analisadas. Em evidencia estão as características que diferenciam cada uma das ferramentas discutidas ao longo desta seção.



Tabela 1 – Ferramentas online para correção automática de código

Características	Ferramentas					
	BOCA	Ideone	ProgTest	SPOJ BR	URI	UVA
Gratuito	X	X Com restrições	X	X	X	X
Autenticação	X	X	X	X	X	X
Interface web	X	X	X	X	X	X
Inserção de questões	X	-	X	X	X	-
Auxílio online (materiais, tutoriais, fóruns, etc.)	X	- Possui FAQ	-	X	X	X
Ranking de classificação	X	-	-	X	X	X
Linguagens suportadas	C, C++ e Java	+ 60 linguagens	C e Java	+ 50 linguagens	C, C++, Java e Phyton	C, C++, Java e Pascal

No geral, todas as ferramentas aqui analisadas restringem o acesso as funcionalidades a apenas usuários autenticados e fornecem acesso através de navegadores *web* ou *web services*. A função do *ranking* de classificação varia de acordo com a ferramenta, algumas classificam os usuários de acordo com o código submetido, outras classificam as questões por nível de facilidade ou percentual de acertos.

Considerações finais

Após a realização dessa pesquisa, chega-se a conclusão de que proporcionar rápido retorno com relação as atividades desenvolvidas, em disciplinas de linguagens de programação, aumenta o desempenho e motivação dos alunos. Durante a pesquisa observou-se vários estudos de caso e relatos de experiência que apresentavam dados comprovando a que essa abordagem proporciona resultados positivos, seja no ensino presencial ou a distância.

Mas, antes de tomar a decisão sobre qual ferramenta adotar, é importante ter o conhecimento sobre quais características e funcionalidades cada uma implementa, visto que possuem especificidades bem distintas, embora tenham o mesmo objetivo final que é verificar o código. Somente assim, é possível ter os subsídios necessários para que essa tomada de decisão seja satisfatória e de fato obtenha-se sucesso ao implantar no cenário pretendido.



Como trabalhos futuros, serão realizadas pesquisas sobre como ocorre a integração dessas ferramentas com o ambiente virtual de aprendizagem Moodle, disponibilizando a funcionalidade de avaliação automática e repositório de questões obtidos a partir de juízes *online* e permitindo que o professor visualize o andamento individual, assim como da turma, visando facilitar o acompanhamento.

Referências

CAMPOS, C. P. **BOCA Online Contest Administrator**. Disponível em: <<http://www.ime.usp.br/~cassio/boca/boca/doc/html/manualjudge.en.html>> Acesso em: 11 de abr. de 2016.

CHAVES, J. O. M. **Uma ferramenta de apoio ao processo de ensino-aprendizagem em disciplinas de programação de computadores por meio da integração dos Juízes Online ao Moodle**. 2014. 80 f. Dissertação (Mestrado em Ciência da Computação). Universidade do Estado do Rio Grande do Norte. Universidade Federal Rural do Semi-Árido. Mossoró, RN. 2014.

FLUMINHAN, C. S. L. ; ARANA, A. R. A. ; FLUMINHAN JUNIOR, A. **A importância do feedback como ferramenta pedagógica na educação à distância**. Colloquium Humanarum , v. 10, p. 721-728, 2013. Disponível em: <<http://www.unoeste.br/site/enepe/2013/suplementos/area/Humanarum/Educação/A%20IMPORTANCIA%20DO%20FEEDBACK%20COMO%20FERRAMENTA%20PEDAGÓGICA%20NA%20EDUCAÇÃO%20À%20DISTANCIA.pdf>> Acesso em: 11 de abr. de 2016.

HRICKO, M. **Student Retention in Distance Education**. In: PHILLIPS, V. *Motivating & Retaining Adult Learners On-line*. 2002. Disponível em: <<http://www.geteducated.com/images/pdfs/journalmotivateretain.pdf>>. Acesso em: 11 de abr. de 2016.

IDEONE. **I deone™API**. Disponível em< <http://ideone.com/files/ideone-api.pdf> > Acesso em: 11 de abr. de 2016.

MUTIAWANI, V.; JUWITA. *Developing e-learning application specifically designed for learning introductory programming*. In: **Information Technology Systems and Innovation (ICITSI)**. International Conference, Bandung. 2014.

SALOMÃO, L. F. S.; WATANABE, R. H.. *Evasion in distance education courses offered by an organization of Brazilian Army: Actions to reduce*. In: **International Conference on Information Systems and Technology Management (CONTECSI)**, 10., 2013, São Paulo. Anais... São Paulo: 2013. Disponível em: <



<http://www.contecsi.fea.usp.br/envio/index.php/contecsi/10contecsi/paper/download/3539/2061> >
Acesso em: 11 de abr. de 2016.

SANDBOX. **Sandbox**. Disponível em: < <https://github.com/openjudge/sandbox> > Acesso em: 11 de abr. de 2016.

SOUZA, A. M.; POLONIA, A. C. Tutoring in Distance Education: New Proposals, Challenges and Reflections. In: **International Journal of Educational Excellence**, v. 1, p. 53-80, 2015. Disponível em: < http://www.suagm.edu/umet/ijee/pdf/1_2/miranda_souza_costa_polonia_ijee_1_2_53-80.pdf > Acesso em: 11 de abr. de 2016.

SOUZA, D. M; MALDONADO, J. C; Barbosa, E. F. ProgTest: An environment for the submission and evaluation of programming assignments based on testing activities. In: **Anais do XXIV Conference on Software Engineering Education and Training**, Honolulu. 2011.

SPOJ BR. **SPOJ (Sphere Online Judge) Brasil**. Disponível em: < <http://br.spoj.com> >. Acesso em: 11 de abr. de 2016.

URI. **URI Online Judge**. Disponível em: <http://www.urionlinejudge.com.br> >. Acesso em: 11 de abr. de 2016.

UVA. **UVA Online Judge**. Disponível em: < <https://uva.onlinejudge.org/> >. Acesso em: 11 de abr. de 2016.