7ª JORNADA CIENTÍFICA E TECNOLÓGICA DO IFSULDEMINAS
4º SIMPÓSIO DE PÓS-GRADUAÇÃO
12 de novembro de 2015 | Poços de Caldas - MG

ANÁLISE E IMPLANTAÇÃO DE FERRAMENTAS PARA CORREÇÃO AUTOMATIZADA DE ALGORITMOS

Douglas Castilho[1]; Alysson Eduardo[2]

RESUMO

O processo de aprendizagem de linguagens de programação depende, dentre outras coisas, da aplicação do conhecimento adquirido em sala nas tarefas práticas propostas pelo professor. No ensino de linguagens de programação, esse processo, apesar de importante, se torna inviável para o docente, pois o tempo gasto na correção se torna impraticável, diante do volume de alunos e da quantidade de atividades produzidas. Neste contexto, este projeto tem como objetivo a análise e implantação de uma ferramenta que auxilie o professor, proporcionando a correção automática de algoritmos e programas desenvolvidos pelos alunos. Além de proporcionar ao professor várias métricas para avaliação durante a correção das atividades, a ferramenta tende a diminuir falhas e equívocos durante o processo de correção humanizado, mostrando um grande potencial no auxílio ao professor.

INTRODUÇÃO

O ensino de linguagens de programação é fundamental na área da computação. Uma das principais metas das linguagens de programação é permitir que programadores tenham uma maior produtividade, permitindo expressar suas intenções mais facilmente do que quando comparado com a programação em nível

[1] Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais – Campus Poços de Caldas. Poços de Caldas/MG - E-mail: douglas.braz@ifsuldeminas.edu.br
[2] Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais – Campus Passos. Passos/MG. E-mail: alyssonedu10@gmail.com

de máquina. Aprender a programar é muito semelhante a estudar um novo idioma. O primeiro passo é escolher uma linguagem de programação, que possui uma sintaxe e estrutura própria. O segundo passo é entender a sintaxe desta linguagem e produzir algoritmos que resolvam alguns problemas específicos. Um professor deste conteúdo frequentemente propõe exercícios e atividades para fixação do conhecimento por parte do aluno. Tão importante quanto a aplicação destas atividades é o feedback que o professor oferece ao aluno na correção.

Para um bom desempenho de uma turma que está aprendendo uma linguagem de programação, além de aulas teóricas, o professor deve fornecer meios para que os alunos consigam testar o que foi aprendido em sala de aula. No entanto, esses meios nem sempre são suficientes para o aluno, e a alternativa geralmente utilizada é a disponibilização de tarefas para serem pesquisadas e resolvidas fora da sala de aula. Como o número de alunos geralmente é alto, o volume de atividades que devem ser corrigidas pelo professor é acaba sendo grande. Assim, é comum observar uma sobrecarga de trabalho do professor para as atividades de avaliação e esclarecimento de dúvidas (MOTA, 2009). Além disso, no processo de correção humanizado de algoritmos podem ocorrer falhas e equívocos. Porém, em um processo de correção automatizado, a probabilidade disso ocorrer diminui.

Neste contexto, este projeto tem como principal objetivo o desenvolvimento de uma ferramenta para auxiliar o professor na tarefa de corrigir de forma justa, completa e eficiente os projetos e trabalhos submetidos pelos alunos. Para tanto, é necessário um sistema online de correção automatizada que aponte níveis de acerto, além de outros parâmetros que podem ser definidos pelo professor, tais como tempo máximo de execução dos programas. Muitas vezes, as ferramentas de automatização encontradas na literatura não abrangem todas as linguagens de programação que a instituição oferece no curso de informática, e que irá oferecer no curso de Ciência da Computação. A maioria das ferramentas já existentes trabalha com a linguagem de programação JAVA. Ferramentas já consolidadas, como o BOCA (DE CAMPOS, 2004), já atuam nessa área de suporte a submissão de trabalhos. Porém, seu foco é no apoio à competições e maratonas de programação, e que exigem um ou vários juízes, dispensando a correção automática. Outra ferramenta já disponível é o SAmbA (NOBRE, 2002), que trabalha com a linguagem de programação PASCAL, completando uma ferramenta já existente, a AmCorA (MENEZES E CURY,1999).

MATERIAL E MÉTODOS

Para o desenvolvimento deste projeto, faz-se necessário também um conhecimento sobre a teoria dos compiladores. O princípio de um compilador é a tradução de uma linguagem fonte para uma linguagem alvo (NETO,1993). Geralmente a linguagem fonte é de alto nível, como C, JAVA, C++, e a linguagem-alvo é um código de máquina. Seu funcionamento simplificado pode ser dividido em três etapas:

- **Análise Sintática** - O analisador sintático é o cerne do compilador, responsável por verificar se os símbolos contidos no programa fonte formam um programa válido ou não. (DELAMARO, 2004).
- **Análise Léxica** - Sua função é “varrer” o código-fonte buscando símbolos léxicos, é também de sua responsabilidade a remoção de espaços, tabulações e comentários. (PRICE,2000).
- **Análise Semântica** - O papel do analisador semântico é prover métodos pelos quais as estruturas construídas pelo analisador sintático possam ser avaliadas ou executadas (WATSON, 1989).

Após a compilação de um código fonte, o programa gerado pode ser executado na plataforma de destino. Este programa será utilizado para realizar a verificação da corretude do algoritmo, que pode ser afirmada quando o algoritmo funciona corretamente com relação à determinada especificação.

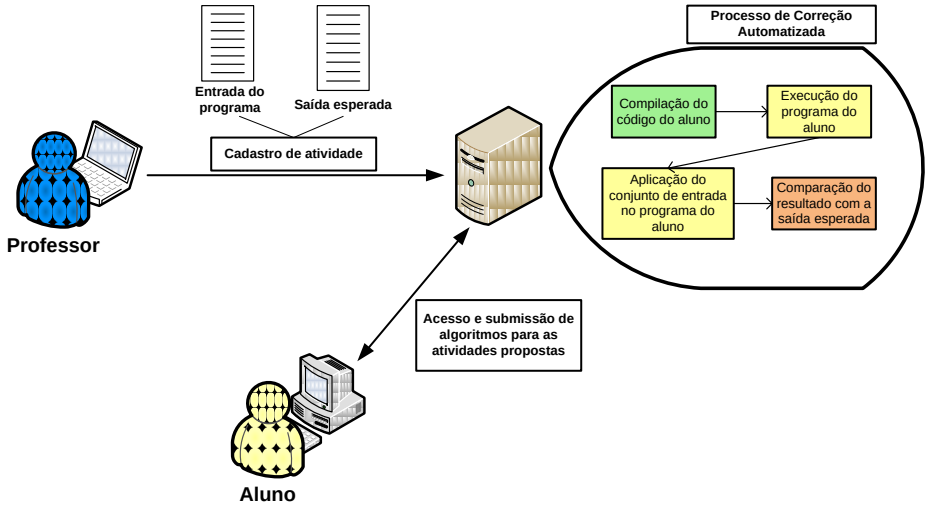

Figura 1: Processo de correção automatizado

A Figura 1 ilustra o processo de correção automatizado utilizado neste trabalho. O professor, ao cadastrar uma atividade, deve elaborar um conjunto de entradas para serem inseridas no programa do aluno, assim como um conjunto de saída esperado, i.e., resultado que o programa do aluno deveria retornar para o respectivo conjunto de entradas. O aluno poderá acessar as atividades e submeter seus algoritmos para serem testados pela ferramenta. Além disso, podemos ver também na Figura 1 as etapas necessárias para a correção automatizada de um programa.

O módulo *online* do sistema é feito através da ferramenta BOCA, que é responsável pelo cadastro das atividades pelo professor, cadastro das entradas e saídas esperadas, e o acesso do aluno para submissão dos trabalhos. Foram realizadas algumas adaptações no código, uma vez que o BOCA foi desenvolvido para competições de programação. O BOCA possui licença *GNU Public License 3*, e todos os requisitos e aspectos legais para utilização do código fonte foram devidamente respeitados. O módulo responsável pela correção automatizada é *offline,* foi desenvolvido na linguagem JAVA, e é capaz de corrigir programas desenvolvidos nas linguagens C e JAVA, podendo ser executado nos sistemas operacionais Linux e Windows. Os módulos *online* e *offline* são independentes, o que permite que o professor receba os trabalhos dos alunos da forma que achar conveniente, e utilize isoladamente o módulo *offline* para realizar as correções.

RESULTADOS E DISCUSSÃO

A ferramenta de correção foi testada através de *programas exemplo*, e também com um estudo de caso utilizando programas reais, relacionados com trabalhos e exercícios propostos pelo professor coordenador em suas disciplinas. Os testes da ferramenta foram feitos em duas etapas. Primeiramente foram realizados testes utilizando *programas exemplo,* com objetivo de avaliar o desempenho do sistema quando submetido a um volume grande de processamento. Em seguida, foram realizados testes utilizando programas reais, enviados previamente pelos alunos ao professor. Estes programas fazem parte de um conjunto de atividades avaliativas de disciplinas relacionadas à programação.

Nos testes de desempenho, a ferramenta se mostrou estável e concisa, dentro dos parâmetros de execução, mesmo quando requisitada a avaliar milhares

de programas simultaneamente. Nos testes realizados em ambiente real, a ferramenta apresentou alta precisão na correção dos códigos dos alunos. Todos os programas corretos foram classificados desta forma, e todos os incorretos também foram identificados. Alguns erros na correção são ocasionados pela inserção de caracteres que não deveriam estar contidos nas saída dos programas, como frases indicando os resultados, ao invés da resposta básica solicitada pelo professor. Este problema está relacionado com a falta de padronização na saída dos programas feitos pelos alunos, que comumente exibem mensagens diferentes da saída padrão solicitada. Alguns exemplos dessa divergência são mostrados na Tabela 1, contendo resultados desenvolvidos por alunos:

Tabela 1: Comparação entre respostas corretas e erradas

Saída com a Resposta Correta	Saída com a Resposta Errada
0.66	*O resultado é 0.66*
Sim *7 7 6 7 7 6 7 7 6*	*Sim, e os valores dos pratos são 7 7 6* *7 7 6 7 7 6*
4.59	*4.59* *Digite qualquer tecla para sair*

Para amenizar este problema, foi desenvolvido um módulo que calcula a similaridade entre a resposta do aluno e a resposta esperada. Este valor é exibido juntamente com a sinalização da corretude do programa, e ajudará o professor a identificar possíveis erros causados pela má formatação da resposta.

Além do desenvolvimento do sistema de correção automatizada e adaptação do BOCA, foi elaborado um manual de instalação do software BOCA, com o intuito de facilitar a implantação deste, que é a base para submissão dos trabalhos. Este manual encontra-se disponível no sítio do sistema[3]. O sistema ainda não se encontra disponível para utilização, mas em breve será disponibilizado para os professores do IFSULDEMINAS, e posteriormente para toda comunidade acadêmica.

CONCLUSÕES

Os objetivos propostos para o trabalho foram concluídos, porém existem várias melhorias no sistema que devem ser desenvolvidas. Foi adaptado o software BOCA, ferramenta utilizada para competições e maratonas de programação, para

[3] www.castilho.bz/sistema_correcao

permitir que os professores cadastrem as atividades, e também para permitir que os alunos submetam seu projetos e trabalhos para avaliação. O módulo do software BOCA é *online*. Além disso, foi desenvolvido um sistema *offline* de correção automatizada, que realiza a compilação e execução dos trabalhos dos alunos, e verifica se os resultados gerados condizem com aquilo que é esperado pelo professor. Foram realizados testes com trabalhos reais enviados por alunos de algumas disciplinas, e o resultado das avaliações automáticas equivalem às avaliações realizadas pelos professores, exceto algumas avaliações onde o aluno não formatou a resposta de acordo com o esperado, como mostrado na Tabela 1.

O desenvolvimento e a melhoria contínua da ferramenta serão realizados em projetos futuros. Dentre estas melhorias, podemos citar: implantar análise de similaridade entre códigos e árvores sintáticas dos programas, para encontrar possíveis cópias de trabalhos; desenvolvimento de um sistema para a submissão dos trabalhos, eliminando a necessidade da utilização o software BOCA, que irá proporcionar maior flexibilidade na manutenção do sistema todo; módulo de teste para os alunos, que poderão testar seus trabalhos antes de submeterem para avaliação do professor; inclusão de outras linguagens de programação.

REFERÊNCIAS

DELAMARO, Marcio. Como Construir um Compilador Utilizando Ferramentas Java. São Paulo: Novatec, 2004. 308 p. p. 4. ISBN 85-7522-055-1

DE CAMPOS, Cassio P.; FERREIRA, Carlos E. BOCA: um sistema de apoio a competições de programação. 2004.

MENEZES, C; CURY, D. AmCorA: Um Ambiente Cooperativo para a Aprendizagem Construtivista Utilizando a Internet .SBIE, 1999.

MOTA, M. P., DE BRITO, S. R., MOREIRA, M. P., & FAVERO, E. L. (2009, November). Ambiente Integrado à Plataforma MOODLE para Apoio ao Desenvolvimento das Habilidades Iniciais de Programação. In Anais do Simpósio Brasileiro de Informática na Educação (Vol. 1, No. 1).

NOBRE, Isaura Alcina Martins; MENEZES, Crediné Silva de. Suporte à Cooperação em um Ambiente de aprendizagem para Programação (SAmbA). In: Anais do Simpósio Brasileiro de Informática na Educação. 2002. p. 337-347.

PRICE, Ana M. A.; Toscano, Simão Sirineo. Implementação de Linguagens de Programação: Compiladores: Série de Livros Didáticos Número 9. Porto Alegre: Sagra Luzzatto, 2000. 195 p. ISBN 978-85-241-0639-2

WATSON, Des. High-Level Languages and Their CompilersWokingham, Reino Unido: Addison-Wesley, 1989. 337 p. ISBN 0-201-18489-3