SISTEMA DE CORREÇÃO AUTOMÁTICA DE CÓDIGO-FONTE

Fernanda Sartori Beltrame[1], Guilherme Alberto Wachs Lopes[2]
[1,2] Centro Universitário FEI
(uniffbeltrame,gwachs)@fei.edu.br

Resumo: Na área da Computação, muitos sistemas de correção de exercícios são usados até mesmo para correção automática de códigos-fonte. Contudo, um dos principais empecilhos do uso dessas ferramentas está na ineficiência da interface para elaboração e uso do professor em sua disciplina. Nesse projeto, é proposto um sistema de correção automática de códigos-fonte para que o professor possa elaborar seu próprio corretor de maneira prática, a partir da comparação entre a saída do sistema construído pelo aluno e a saída esperada.

1. Introdução

Na área da educação, diversas abordagens são propostas visando o progresso da aprendizagem. Uma dessas abordagens que, aliado ao desenvolvimento computacional, apresenta destaque é o *Online Judge* (OJ).

Online Judge é um tipo de sistema que possibilita a correção automática de exercícios, muito utilizado, também, em competições, por exemplo. Um sistema muito conhecido é o site *www.urionlinejudge.com*, que apresenta uma lista de problemas computacionais classificados em categorias e faz a correção automática para cada aluno [1].

Esses tipos de sistemas apresentam três principais vantagens em relação ao sistema de ensino tradicional. A primeira delas é que o aluno tem o retorno imediato sobre a corretude do seu programa. A segunda é que pode-se utilizar o sistema mesmo não estando presente na faculdade. E, a última, é que seu desempenho pode ser medido com o passar do curso.

Além dos benefícios acima citados, sistemas OJ são vantajosos quando se trata de facilitar o trabalho do professor, pois torna o processo de correção, avaliação individual de cada aluno e projeção de novos exercícios mais acessível [4], [5], [6], [7].

Apesar desses sistemas de correção automática apresentarem esses benefícios, ainda não se tem, até o momento, um sistema como esse focado na usabilidade do professor em disponibilizar exercícios e materiais tal como o *Moodle*. Esse é um importante ponto negativo que faz com que a implantação desses sistemas se torne inviável em ambientes acadêmicos. Assim, nesse projeto, é proposto um sistema de correção automática de programas computacionais que seja focado não só na utilização do aluno, mas principalmente, na utilização do professor. Esse sistema permitirá hospedagem de arquivos, cálculo de notas e configuração do sistema de correção para cada exercício.

2. Sistema Proposto

A metodologia de execução do projeto é focada em duas vertentes: Professor e Aluno.

A vertente focada na usabilidade professor foi confeccionada para que o educador possa submeter exercícios tal qual como no *Moodle*, definindo critérios como data de entrega, entrada esperada para o sistema, código de substituição, saída para comparação e, por exemplo, arquivos PDF com as instruções para sua execução.

A vertente focada no aluno, por sua vez, é dividida em cinco partes: *Upload*, Compilação, Execução, Comparação e *Feedback*. A primeira etapa, chamada de *Upload*, é responsável pela interface de comunicação entre o código-fonte do exercício que o aluno resolveu e o sistema de correção. A segunda etapa, chamada de Compilação, é responsável pela compilação do código-fonte em um programa executável. A terceira etapa, a Execução, irá executar a aplicação do aluno em um *container*, isto é, em um mecanismo de máquina virtual. A quarta etapa, chamada de Comparação, irá comparar a saída gerada pelo arquivo executável com a saída esperada do programa. Por fim, a última etapa, chamada de *Feedback*, é responsável por organizar todas as informações de erro e exibir ao aluno. A Figura 1 ilustra essas etapas.

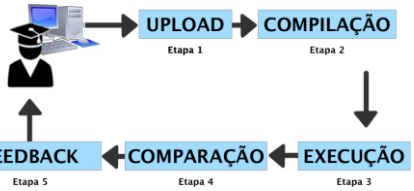

Figura 1: Diagrama Das Fases Do Projeto

Etapa 1) Através da aplicação web integrada ao Banco de Dados, os alunos logram da possibilidade de submeterem os exercícios propostos pelos professores, bem como de realizarem o *download* do material necessário para sua devida realização. O *upload* de arquivos é uma fase essencial do sistema, uma vez que a correção do código-fonte só é possível quando o sistema tem acesso ao arquivo enviado pelo aluno. Enviar as respostas propostas possibilita, junto à compilação e correção automática, que o aluno tenha uma melhor percepção dos seus erros, bem como aumento da assimilação dos conteúdos abordados.

Etapa 2, 3 e 4) A interface de Correção Automática foi dividida em fases, ilustradas pela Figura 2.

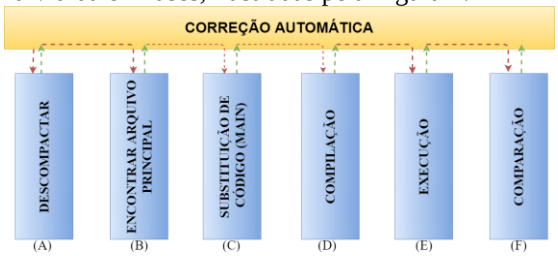

Figura 2: Diagrama da Etapa de Correção Automática.

(A) Descompactar: Após enviado, cada arquivo passa por uma etapa de verificação de compactação e em caso positivo, ele automaticamente será convertido,

prevalecendo a hierarquia de pastas do arquivo original, e movido para uma pasta no servidor na qual as próximas etapas ocorrerão. Caso a verificação não aponte que o arquivo submetido esteja compactado, o envio do aluno será redirecionado a uma pasta no servidor, assim como o arquivo após a descompactação.

(B,C) Substituição e Localização: Para que a correção seja realizada, a saída gerada do código enviado pelo aluno deve ser compatível com a esperada pelo professor (enviada na criação do laboratório). Antes que a saída seja gerada, o professor possui duas opções: manter o código-fonte do aluno original ou substituir parte do código por um fornecido pelo professor na criação do laboratório, assim, é possível que funções mais específicas criadas pelos alunos sejam testadas. Nessa etapa, serão realizadas as seguintes funções:

a) **Localização:** Localiza-se qual o arquivo principal do projeto baseado na linguagem optada pelo aluno. Após localizado o arquivo, automaticamente é identificada a região do código na qual será realizada a substituição do código-fonte por aquele enviado pelo professor, em caso da opção de substituição ter sido escolhida pelo professor.

b) **Substituição:** Nesta etapa, a modificação do código-fonte original é efetuada.

(D) Compilação: Nessa etapa, a linguagem escolhida pelo aluno para a execução do código-fonte é consultada no Banco de Dados, visto que a compilação ocorre em caso de *C++*, *C* ou *Java*. Em *Python*, por sua vez, o código é apenas executado. De acordo com a linguagem, um arquivo saída é gerado com os erros de compilação, caso ocorram, para o *feedback* ao aluno.

(E) Execução: Após compilados, arquivos "*.exe*", "*.java*" são criados para as linguagens *C/C++* e *Java*, respectivamente. Para *Python*, a execução além de gerar o código de máquina, a análise sintática é realizada, produzindo um arquivo contendo os erros lógicos e sintáticos do programa, caso existam. Com a compilação/execução realizada, os arquivos de execução são executados produzindo a saída do código do aluno. Essa saída, por sua vez, é salva no Banco de Dados para a etapa a seguir.

(F) Comparação: Neste ponto, a saída do aluno é comparada com a saída esperada do professor (cadastrada previamente), retornando um valor de zero a um, representando a sua compatibilidade. Para fins de correções mais justas, definimos que se o valor de compatibilidade for maior ou igual a 70%, o código-fonte enviado é considerado "aprovado", caso contrário, é considerado "reprovado". Se o aluno não enviar nenhuma submissão, setamos a "resposta" do compilador como "nenhuma tentativa". A performance do programa não será levada em conta.

Etapa 5) O objetivo dessa etapa é notificar ao aluno seus erros de execução do exercício proposto, bem como seu desempenho ao evoluir os códigos-fonte.

4. Resultados e Conclusões

Como resultado do desenvolvimento de um sistema *Online Judge* temos: ampliação da assimilação e habilidade de execução de conteúdos pelos alunos; possibilidade do ensino remoto; e diversificação da plataforma a ser utilizada, como citado nos trabalhos [2], [3], [4] e [5].

A implementação de sistemas OJ traz benefícios aos professores, visto que os auxiliam a traçar uma estratégia sobre os erros mais comuns cometidos por alunos, corroborando para o melhor planejamento das aulas e assimilação dos conteúdos por parte dos alunos como afirmado em [4]. Neste mesmo trabalho, os autores relatam que utilizando a ferramenta, os professores têm a possibilidade de constatar quais alunos mais se destacam nos assuntos abordados, podendo selecioná-los como ajudantes a alunos que apresentam maiores dificuldade, a fim de maximizar a taxa de compreensão.

Implementar sistemas OJ não constitui a obsolescência do papel do professor, visto que esses sistemas apenas verificam o conhecimento e a capacidade profissional do aluno, não sendo uma plataforma de aprendizagem, embora contribuam para o crescimento do aprendizado do aluno [2]. Portanto, salienta-se que o papel do professor é de suma importância para transmissão de conteúdos e construção da base curricular.

5. Referências

[1] M. Selivon, J. L. Bez, N. A. Tonin, and R. Erechim, "Uri online judge academic: Integração e consolidação da ferramenta no processo de ensino/aprendizagem".

[2]X. Lu, D. Zheng, and L. Liu, "Data driven analysis on the effect of online judge system", 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 573-577, 2017.

[3] J. Wu, S. Chen, and R. Yang, "Development and application of online judge system", 2012 International Symposium on Information Technologies in Medicine and Education, vol. 1, pp. 83-86, 2012.

[4] H.Wu, Y. Liu, L. Qiu, and Y. Liu, "Online judge system and its applications in c language teaching", 2016 International Symposium on Educational Technology (ISET), pp. 57-60, 2016.

[5] A. Kurnia, A. Lim, and B. Cheang, "Online judge", Computers & Education, vol. 36, pp. 299-315, 2001.

[6] J. de San Pedro, J. Carmona, J. Cortadella, and J. Petit, "Integrating formal verification in an online judge for e-learning logic circuit design", in SIGCSE, 2012.

[7] X. Du, C. Yi, Y. Wei, S. Feng, and Z. Gong, "Design of automata online judge", 2010 2nd International Conference on Information Engineering and Computer Science, pp. 1-4, 2010.

6. Agradecimentos

À instituição Centro Universitário FEI e à Telefônica/Vivo pelo financiamento e infraestrutura necessária.

[1] Aluno de IT do Centro Universitário FEI. Projeto com vigência de 09/17 a 08/18.