

CPP: Corretor de Problemas de Programação

Aline R. de OLIVEIRA¹; Abner S. P. PALMEIRA²; Gustavo J. SILVA³; Aline M. D. VALLE⁴;

Paulo C. dos SANTOS⁵

RESUMO

Com a globalização e o avanço da tecnologia, o mercado de informática necessita cada dia mais de profissionais qualificados. Infelizmente esse mercado encontra-se em déficit de profissionais, uma das causas é a alta taxa de desistências dentre os ingressantes nos cursos da área. Uma grande dificuldade encontrada pelos alunos são os conteúdos das aulas de programação. Para solucionar esse problema é necessária atenção especial a essas aulas. Em vista disso, foi desenvolvido o CPP: Corretor de Problemas de programação, que tem por objetivo facilitar o processo de correção de exercícios de programação por parte dos professores, e maior possibilidade de prática para os alunos.

Palavras-chave: Problemas; Programação; Corretor; Judge.

1. INTRODUÇÃO

Na sociedade atual a informática é uma área muito importante e é imprescindível ter afinidade com as novas tecnologias. Grande parte das empresas públicas e privadas adotam computadores como principal fonte de trabalho, pelo fato que eles agilizam as atividades do dia-a-dia. Para isso ser efetivamente possível, são necessários profissionais da área da informática capacitados e prontos para lidarem com as adversidades do cotidiano.

Sabe-se que os conteúdos relacionados à programação de computadores fazem parte das disciplinas básicas nos cursos da área de informática. Nessas aulas os alunos aprendem sobre algoritmos e linguagens de programação, conteúdos essenciais em diversas carreiras em informática.

Além das vantagens que essas aulas trazem para área profissional do aluno, elas também aperfeiçoam o raciocínio lógico e aumentam os conhecimentos matemáticos dos alunos. Porém, alunos e professores encontram dificuldades no processo de ensino aprendizagem de programação (conforme descrito na seção 2). Para amenizar essas dificuldades, podem ser utilizadas ferramentas de apoio às aulas de programação, para um

¹ Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais – Campus Muzambinho. Muzambinho/MG - E-mail: alineregoli@gmail.com

² Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais – Campus Muzambinho. Muzambinho/MG - E-mail: abnerpalmeira@gmail.com

³ Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais – Campus Muzambinho. Muzambinho/MG - E-mail: gustavo.jose@muz.ifsuldeminas.edu.br

⁴ Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais – Campus Muzambinho. Muzambinho/MG - E-mail: aline.valle@ifsuldeminas.edu.br

⁵ Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais – Campus Muzambinho. Muzambinho/MG - E-mail: paulo.santos@muz.ifsuldeminas.edu.br

melhor aproveitamento e uma abordagem mais lúdica dos conteúdos aprendidos em sala de aula.

Em razão disto, foi desenvolvido um sistema de apoio às aulas de programação, o CPP: Corretor de Problemas de Programação, no qual o professor pode inserir questões elaboradas por ele, os alunos podem submeter seus códigos, e o sistema faz a correção automática das submissões.

2. FUNDAMENTAÇÃO TEÓRICA

Muitos estudantes enfrentam dificuldades na aprendizagem de programação. Segundo Moreira e Favero (2009) os motivos são:

- Hábitos de estudo pouco disciplinados e centrados em memorização;
- Conhecimentos prévios desestruturados, principalmente nos domínios matemáticos e lógicos;
- Abordagens pouco motivadoras;
- Conteúdos pouco relacionados ao cotidiano dos sujeitos;
- Dificuldades na compreensão do enunciado dos problemas;
- Forte carga de conceitos abstratos, entre outros.

Além disso, Tobar *et al.* (2001) dizem que grande parte dessa dificuldade se deve ao fato do professor não conseguir acompanhar efetivamente as atividades dos alunos em suas aulas de programação. Normalmente há nas turmas grande número de estudantes o que dificulta o acompanhamento efetivo de todos eles durante as aulas. Em decorrência disso, muitas vezes por não ter a supervisão ideal do professor, o aluno não tem suas dúvidas sanadas, o que acaba criando desinteresse pelo conteúdo, piorando ainda mais a situação.

Assim, as dificuldades nesse processo de aprendizagem podem ser vistas por duas perspectivas: pelos alunos, a pouca prática dos conteúdos estudados dificulta a aprendizagem e a memorização da matéria, e pelo professor, a dificuldade em corrigir uma numerosa quantidade de exercícios realizada por seus alunos.

3. MATERIAL E MÉTODOS

Os materiais utilizados para o desenvolvimento do sistema CPP foram: Linguagem de Programação Java 7, Ambiente de Desenvolvimento Integrado NetBeans 8.1, ferramenta de designer visual para bancos de dados Workbench.6.3.5, servidor banco de dados MySQL, e ferramenta de design para diagramas UML: Astah 7.0.

Foi utilizado o modelo de processo de software iterativo incremental, que Medeiros (2016) define como uma forma de entregar aos clientes pequenas funcionalidades, de modo que ele possa acompanhar mais efetivamente o processo de desenvolvimento. Assim é

possível obter maior interação do cliente no processo e obter maior qualidade no produto final.

Foram implementados os casos de uso mostrados na Figura 1. O sistema possui três atores: aluno, professor e administrador. No sistema o usuário (de acordo com o seu nível de acesso) pode Logar, Atualizar Perfil, Gerenciar Usuários, Gerenciar Turmas, Gerenciar Categorias, Gerenciar Questões, Submeter Códigos, Ver Submissões, Ver Relatórios de Questões. Para as questões submetidas foi implementado o corretor para códigos na linguagem de programação C.

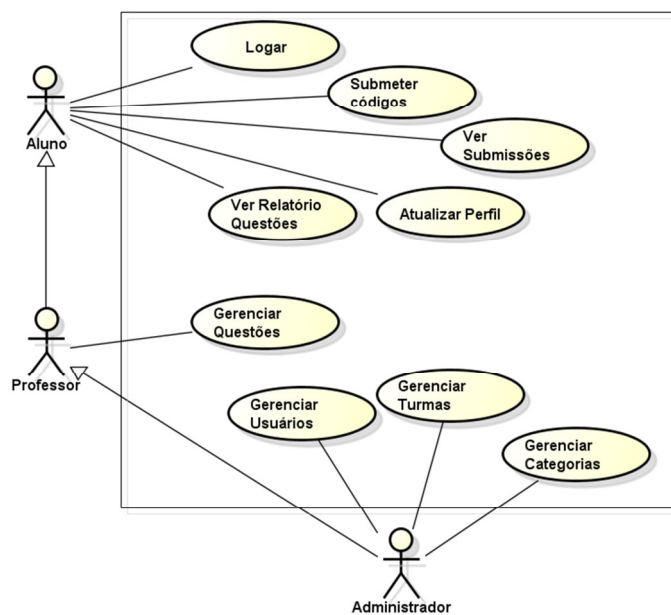


Figura 1 - Casos de Uso Sistema CPP

4. RESULTADOS E DISCUSSÕES

Após o *login* no sistema, o usuário tem acesso à um conjunto de funcionalidades disponíveis na interface principal do sistema, conforme Figura 2.

O sistema CPP permite que o professor crie problemas e associe-os às suas turmas. Estes problemas são criados e julgados no formato da Maratona de Programação, que de acordo com Ferrasa (2012) é “[...] um evento organizado pela Sociedade Brasileira de Computação desde o ano de 1996 [...] são propostos diversos problemas de lógica que devem ser resolvidos computacionalmente”. O objetivo é que o aluno resolvam problemas no formato de maratonas, visando também o treinamento para eventos nesta linha.

A Figura 3 apresenta a interface de submissão de problemas em linguagem C. Após o julgamento o sistema emite um veredito sobre a corretude da submissão, ou seja, a solução

apresentada para o problema. Os possíveis vereditos são: *Accepted* (Correto), *Wrong Answer* (Resposta Errada) e *Compilation Error* (Erro de compilação).

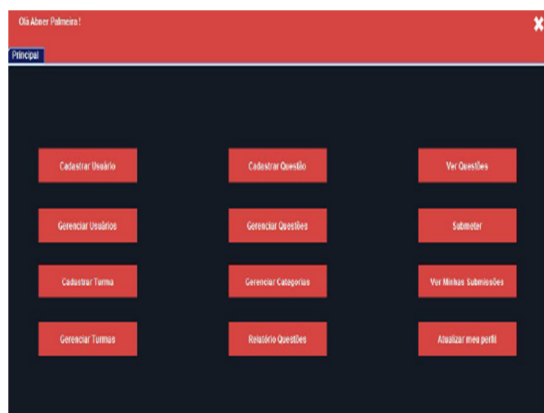


Figura 2: Interface principal do sistema

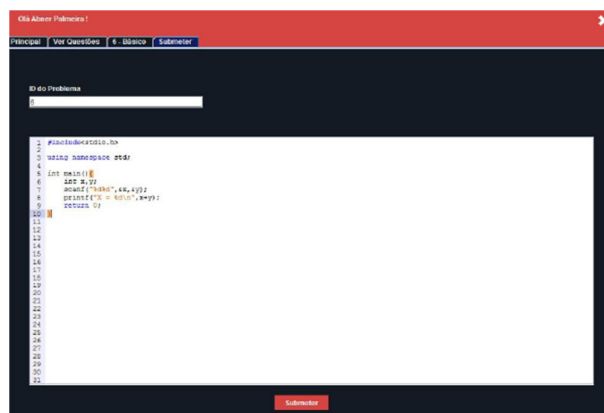


Figura 3: Interface para submissão de códigos

Foram feitos testes unitário no sistema, onde cada funcionalidade foi verificada separadamente, e testes de integração, onde todas as funcionalidades foram testadas em conjunto. Para o teste do corretor, foram resolvidos vários problemas e submetidos, o sistema se mostrou eficaz, fazendo o julgamento correto de todas as submissões em até oito segundos.

5. CONCLUSÕES

CPP pode ser uma ferramenta poderosa no auxílio nas disciplinas de Linguagens de Programação. Permitindo que os professores criem um repositório de questões e facilitando o trabalho de correção dos exercícios. Os alunos podem praticar suas habilidades de lógica e programação e, na medida em que resolvem problemas, ter *feedback* das soluções. São propostas de trabalhos futuros, testar o sistema CPP em um ambiente real e desenvolver o corretor para outras linguagens de programação.

REFERÊNCIAS

FERRASA, Marcelo; SOUZA, M. A. **Competições de raciocínio lógico e programação de computadores: um relato de experiência.** Anais do 10º CONEX – Conversando Sobre Extensão, 2012. Disponível em: <<http://goo.gl/XwyVhQ>> Acesso: 16 de Julho de 2016.

MOREIRA, Mireille Pinheiro; FAVERO, Eloi Luiz. **Um Ambiente Para Ensino de Programação com Feedback Automático de Exercícios.** In: Workshop Sobre Educação em Computação, 2009. Disponível em: <<http://goo.gl/OoxvKX>> Acesso: 16 de Julho de 2016.

MEDEIROS, Higor. **Introdução aos Processos de Software e o Modelo Incremental e Evolucionário.** Revista Devmedia. Disponível em: <<http://goo.gl/Ij0bwy>> Acesso: 16 de Julho de 2016.

TOBAR, C. M. et al. **Uma arquitetura de ambiente colaborativo para o aprendizado de programação.** In Anais do Simpósio Brasileiro de Informática na Educação (Vol. 1, No. 1, pp. 367-376), 2001. Disponível em: <<http://goo.gl/fsn0Kj>> Acesso: 16 de Julho de 2016.