

Uma proposta de um juiz *online* didático para o ensino de programação

Joanna Cecilia da Silva Santos, Admilson de Ribamar Lima Ribeiro

UFS - Universidade Federal de Sergipe

Departamento de Computação

Av. Marechal Rondon, S/N. Jardim Rosa Elze.

CEP 49100-000 São Cristóvão, SE

jc_joanna@yahoo.com.br , admilson@ufs.br

***Resumo.** O aprendizado de programação é de suma importância para a formação de profissionais na área de computação. Estabelecer maneiras de facilitar o ensino de programação torna possível a melhoria na qualidade de ensino de cursos em computação. Sendo assim, este artigo descreve uma proposta de juiz online didático para ser utilizado no ensino de programação. Essa proposta visa auxiliar discentes no aprendizado de programação.*

1. Introdução

O aprendizado de programação é de suma importância para a formação de profissionais na área de computação. Essa importância é proveniente, sobretudo, da adoção do modelo de ensino em que a programação é priorizada nas disciplinas introdutórias dos cursos de graduação nessa área [1]. Sendo assim, além de capacitar o indivíduo a utilizar a lógica de programação na resolução de problemas, a programação é a base para muitos campos em que a computação é aplicada, fator relevante em disciplinas avançadas [2].

Durante os primeiros anos dos cursos de graduação em computação é observada uma quantidade relevante de discentes que reprovam, desistem ou obtêm um baixo rendimento nas disciplinas iniciais focadas ao estudo de programação. Essas atitudes são ocasionadas devido ao processo complexo e exigente que é aprender e desenvolver lógica de programação para a maioria dos alunos [3].

Estabelecer maneiras de facilitar o ensino de programação torna possível a melhoria na qualidade de ensino de cursos em computação. Consequentemente, isso permite a redução da quantidade de reprovações em disciplinas que possuem como pré-requisito o conhecimento de programação.

Nesse contexto, é possível desenvolver ambientes *online* que facilitem esse aprendizado. Um ambiente muito comum utilizado em competições de programação são os chamados juízes *online* [4]. A função principal dos juízes *online* é a avaliação de códigos-fonte que foram enviados ao mesmo em uma determinada linguagem de programação. A avaliação feita por esses juízes gera respostas como: certo, saída errada, saída mal formatada, erro de compilação, erro em tempo de execução, entre outros [5].

Este artigo descreve uma proposta de desenvolvimento de um ambiente que facilite o ensino de programação através da utilização de um juiz *online*. Este juiz possui um funcionamento semelhante aos juízes *online* existentes, a exemplo de SPOJ (*Sphere Online Judge*) [6], UVA (*Universidad de Valladolid*) Online Judge [7] e BOCA *Online Contest Administrator* [5].

Este artigo está organizado como segue: na seção 2 são apresentados os trabalhos relacionados; na seção 3 é descrita a proposta do juiz *online*; na seção 4 descreve-se o estado atual do desenvolvimento do ambiente proposto; na seção 5 são explicitados os impactos esperados com a utilização desse juiz *online*; na seção 6, a última deste artigo, são feitas as considerações finais.

2. Trabalhos relacionados

A maioria dos programas de natureza algorítmica necessita apenas obter em sua entrada padrão dados devidamente formatados e, a partir desses dados, realizar processamento. Após o processamento, os resultados são apresentados de maneira formatada em sua saída padrão. Dessa forma, é possível que a avaliação de programas seja feita de forma automática utilizando um programa que gere os dados de entrada e outro que obtenha e verifique os resultados adquiridos [4].

Esse processo automático de avaliação é feito pelos chamados juízes *online*. Eles recebem código-fonte enviado pelo usuário e posteriormente compila e executa esse código. Durante a execução do programa, os juízes *online* utilizam dados formatados como a entrada do programa e, após o processamento, é feita a comparação dos resultados obtidos com os esperados.

Os juízes *online* são muito utilizados em maratonas de programação. Esses juízes podem ser encontrados na Internet. Entre eles pode-se citar o SPOJ [6] e UVA *Online Judge* [7].

Nesses juízes *online* são disponibilizados vários problemas a serem resolvidos e submetidos. Sendo assim, um usuário pode, além de enviar soluções dos problemas para serem avaliadas, selecionar a linguagem utilizada na escrita do código-fonte. É permitido ainda que os enunciados dos problemas possam ser obtidos em algum formato, a exemplo do formato PDF (*Portable Document Format*). Além disso, eles disponibilizam fóruns de discussão, *ranking* de usuários e estatísticas para cada problema (por exemplo, a quantidade de pessoas que resolveram o problema).

Outro trabalho relacionado consiste de um sistema que visa apoiar as competições de programação, o BOCA *Online Contest Administrator* [5]. Esse sistema foi desenvolvido para ser utilizado na Maratona de Programação da Sociedade Brasileira de Computação. Sua ideia principal é disponibilizar aos alunos e professores o sistema de apoio utilizado em maratonas de programação. Ele possui uma interface *Web* em que cada time participante da competição possui um *login* próprio e, utilizando esse *login*, é possível submeter a resolução de um problema, modificar informações de cadastro, enviar dúvidas aos juízes, observar o placar da competição em tempo real, solicitar a impressão de códigos-fonte e pedir ajuda do pessoal de *staff*. Além disso, é disponibilizada uma

interface de administração da competição que permite o gerenciamento de submissões, placar, dúvidas, usuários, problemas, linguagens, competição, entre outros.

Tendo em vista que o BOCA foi desenvolvido para funcionar na *Web*, ele pode ainda ser utilizado em competições distribuídas, ou seja, os times poderiam estar em qualquer lugar desde que estivessem conectados à Internet. Nesse caso, o processamento é feito todo em um único local, significando que todos trabalham simultaneamente com a correção e submissão de dúvidas centralizadas. Dessa forma, é estabelecida uma maneira igualitária de avaliar as equipes, pois esta avaliação é feita pelo mesmo grupo de juízes.

Nos juízes *online* existentes é observado que seu desenvolvimento é voltado ao uso em competições de programação, dessa maneira, poucas funcionalidades didáticas são encontradas nos mesmos. Por exemplo, após a submissão da resolução de um problema, o usuário obtém um resultado que apenas indica se a resposta está certa/errada ou se houve algum erro em tempo de execução/compilação. Esse resultado, porém, não fornece mais informações que possam indicar onde o usuário errou, sendo uma responsabilidade do mesmo encontrar o seu erro e corrigi-lo.

Sendo assim, a proposta de juiz *online* apresentada nesse artigo, chamado JOnline, visa estender as funcionalidades dos juízes *online* adicionando características educativas que auxiliem os discentes no aprendizado de programação. Essas características são detalhadas na próxima seção.

3. O JOnline

O ambiente proposto, chamado JOnline, possui um juiz *online* ao qual é possível acessar problemas para serem resolvidos e submeter código-fonte para a visualização do resultado. Além disso, o JOnline adiciona também funcionalidades didáticas ao juiz *online* que auxiliam o discente no processo de aprendizado de programação. Essas funcionalidades são: a apresentação de dicas (em língua portuguesa) para corrigir erros de compilação do código-fonte enviado, apresentação de casos de testes que geram resultados errados, organização de problemas por assunto e grau de dificuldade e votação do nível de dificuldade dos problemas pelos usuários, além da programação colaborativa.

A primeira funcionalidade educativa adicionada consiste na apresentação dos resultados da submissão de tal forma que facilite o usuário a encontrar os seus erros. Esses erros podem ser de três tipos: compilação, lógica e execução.

Os erros de compilação são causados pelos erros de sintaxe do código-fonte. Eles são gerados pelo compilador e apresentados ao usuário. Sendo assim, o JOnline processa os erros gerados pelo compilador e apresenta sugestões (dicas em língua portuguesa) de como resolvê-los. Um exemplo dessa situação é mostrado na figura 1. Nessa figura observa-se um erro de compilação (causado pela não-inclusão de um cabeçalho ao programa) com uma dica de solução do problema.

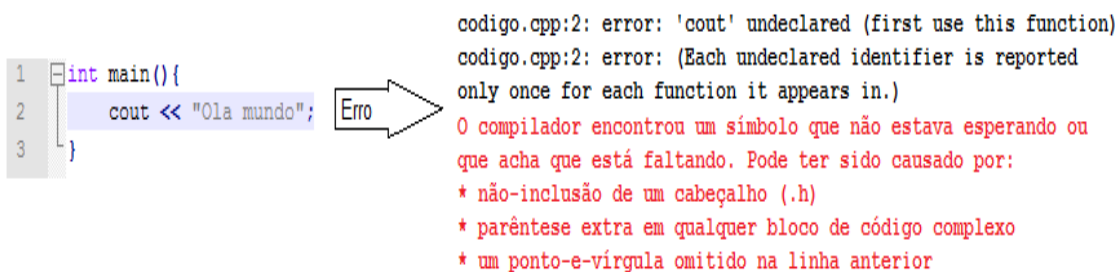


Figura 1. erro de compilação causado pela não-inclusão de um cabeçalho ao programa com uma dica (destacada em vermelho) de solução do problema.

A ajuda aos usuários na correção dos erros de execução e de lógica é feita através da apresentação dos casos de testes (entradas) que geraram resultados errados (incompatíveis com a resposta esperada) ou falhas na execução normal do programa. Além disso, pode-se utilizar *backtrace* para auxiliar o usuário na depuração do programa. *Backtrace* é uma listagem das chamadas de função na *thread* ativa naquele instante.

Para tornar o acesso aos problemas mais amigável ao usuário, os problemas são organizados por assunto abordado ou quanto ao seu grau de dificuldade. A avaliação do nível de dificuldade pode ser feita pelos usuários utilizando uma enquete e, de acordo com essa votação, os problemas são organizados.

A programação colaborativa disponibilizada pelo JOnline permite que dois ou mais discentes possam alterar código-fonte simultaneamente, ou seja, um código-fonte pode ser compartilhado entre um grupo de usuários tornando possível que os mesmos editem, compilem, executem e testem o programa resultante desse código-fonte (desde que estejam conectados ao ambiente). Na figura 2 é apresentado um protótipo simplificado dessa funcionalidade.

Observando a figura 2, nota-se que a ideia é possibilitar a realização simultânea das ações (edição, compilação, etc...) do usuário. Essa possibilidade favorece a realização de atividades práticas em grupo. Além disso, também é observada a disponibilização de um bate-papo para que esses usuários possam interagir e discutir suas ideias.

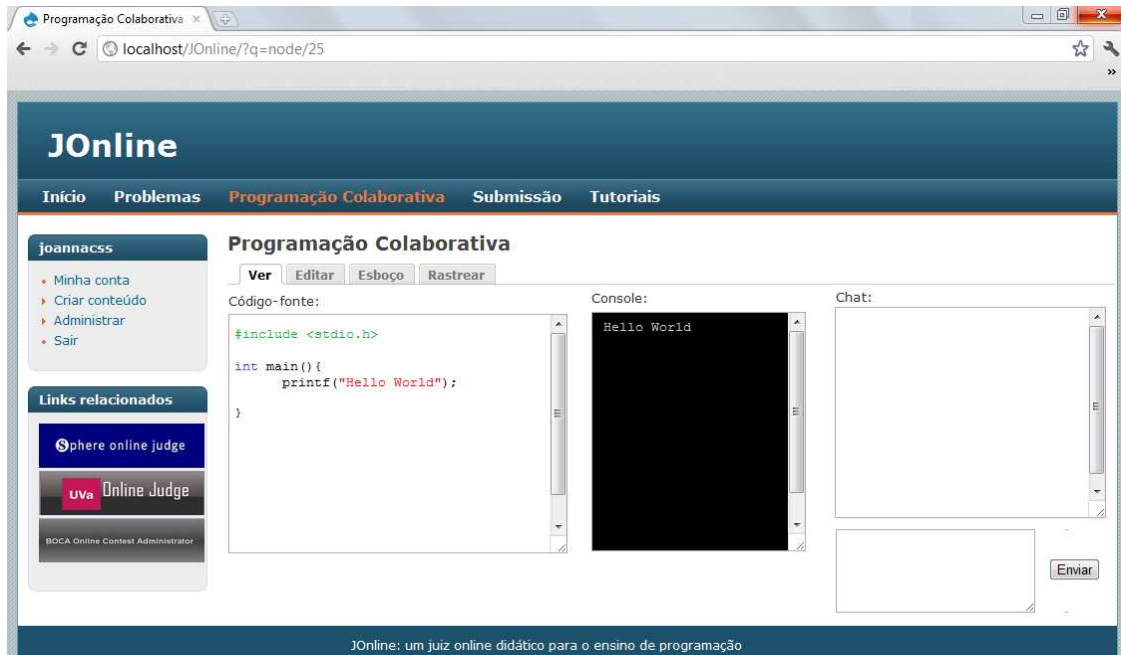


Figura 2. Protótipo da utilização da programação colaborativa

Os usuários do JOnline podem ser professores e alunos. Essa distinção caracteriza as funcionalidades que cada tipo de usuário pode acessar no ambiente. Na figura 3 é mostrado um diagrama de casos de uso [8] do JOnline

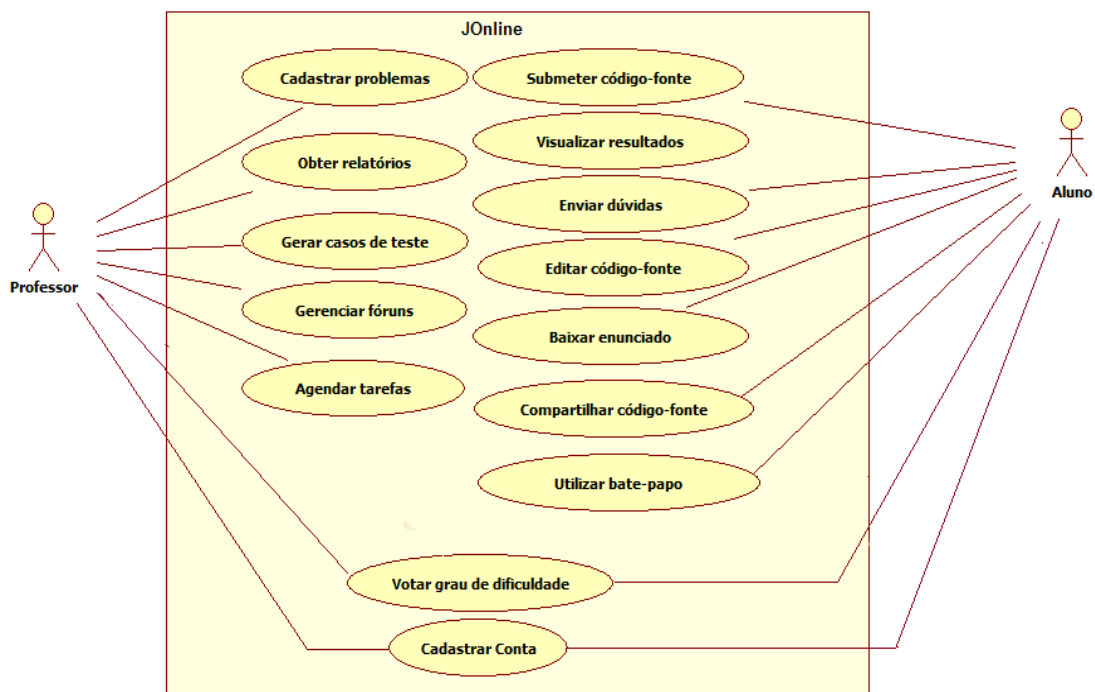


Figura 3. diagrama de casos de uso do JOnline

Conforme observado na figura 3, ao professor é disponibilizada a obtenção de relatórios de submissão para um determinado problema ou usuário, definição das entradas de teste

e saídas esperadas para os problemas, além do gerenciamento de problemas e alunos, entre outras. Ao aluno é fornecida a possibilidade de submissão de código-fonte, resolução de problemas utilizando a abordagem colaborativa, visualização de resultados, etc.

4. Desenvolvimento inicial do ambiente - JOnline

4.1. Ferramentas e tecnologias utilizadas na construção do ambiente

Devido à grande quantidade de informação e a necessidade de atualização constante do ambiente, o mesmo foi desenvolvido utilizando o Drupal [9]. O Drupal é uma ferramenta de gerenciamento de conteúdo (CMS - *Content Management System*) capaz de abstrair detalhes de programação, fornecendo uma interface amigável para a manutenção do *site*. Uma característica favorável nesse CMS é a possibilidade de instalação de módulos para a adição de funcionalidades ao *site*.

O Drupal necessita da instalação de um servidor de banco de dados e um servidor HTTP (*HyperText Transfer Protocol*) [10] compatível para funcionar. Sendo assim, é utilizado como servidor de banco de dados o MySQL [11] e o Apache[12] como servidor de HTTP.

Ainda no contexto *Web*, é utilizado PHP (*Hypertext Preprocessor*) [13] e Javascript [14] para tornar as páginas *Web* mais interativas com o usuário. Também é utilizado AJAX (*Asynchronous JavaScript and XML*) [15] para realizar solicitações assíncronas de informações e, assim, não há a necessidade de novo carregamento completo da página para a atualização das informações apresentadas. Além disso, a biblioteca jQuery da linguagem Javascript é utilizada para agilizar o desenvolvimento *Web* [16].

A linguagem de servidor (*server-side*) utilizada para efetuar os processos de compilação, execução e obtenção de dados é a linguagem Java [17].

Tendo em vista que o JOnline deve compilar e executar programas de usuários, é necessário que o servidor que hospeda esse ambiente possua um compilador de C/C++ instalado. Sendo assim, o JOnline foi hospedado em um servidor com um sistema operacional Ubuntu, uma versão do GNU/Linux baseado na distribuição Debian [18].

4.2. O ambiente desenvolvido

Para atender os requisitos do ambiente, o desenvolvimento do mesmo é dividido em três camadas principais: 1) camada de gerenciamento de conteúdo e usuários, 2) camada de processamento da solicitação e 3) camada de análise e obtenção de dados. Na figura 4 é possível observar as camadas do sistema bem como as interações entre elas. As funções dessas camadas são descritas a seguir:

- **camada de gerenciamento de conteúdo e usuários:** como o próprio nome sugere, essa camada é responsável por gerenciar o conteúdo e os usuários cadastrados no ambiente utilizando o CMS Drupal. Ela abrange o banco de dados que armazena informações sobre os usuários e problemas do ambiente. Também contém uma subcamada que visa apresentar conteúdo de forma amigável ao usuário.

- **camada de processamento da solicitação:** essa camada foi desenvolvida utilizando um conjunto de códigos em PHP, Javascript e AJAX. Ela recebe as solicitações da camada inferior e as processa. A partir do resultado desse processamento ela encaminha as respectivas solicitações à camada superior. Sendo assim, quando a camada superior retorna o resultado, esse é devolvido como resposta à camada inferior. Dessa maneira, essa camada realiza a comunicação entre a camada de gerenciamento de conteúdo e usuários com a camada de análise e obtenção de dados.

- **camada de análise e obtenção de dados:** essa camada contém a lógica de processamento do juiz *online*. Nela são realizadas as operações de compilação/execução de programas, obtenção de erros, entre outros. Essa camada é desenvolvida utilizando a linguagem Java.

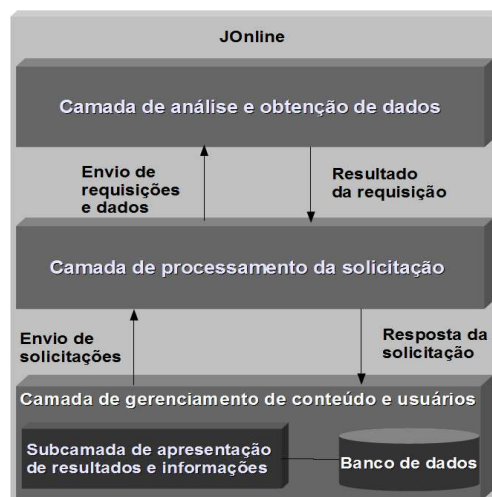


Figura 4. esquema das camadas do juiz desenvolvido. Nesta figura são mostradas as interações das camadas.

A partir do desenvolvimento progressivo das camadas descritas anteriormente, foi possível obter, até o momento, a possibilidade de cadastro de usuários, atribuindo-lhes os papéis de professor ou aluno de acordo com o usuário cadastrado. Além disso, já é possível acessar problemas, baixar documentos PDF contendo o enunciado de algum problema, submeter código-fonte, acessar tutoriais de uso do ambiente e participar de fóruns. Na figura 5 é possível observar a página inicial do JOnline.

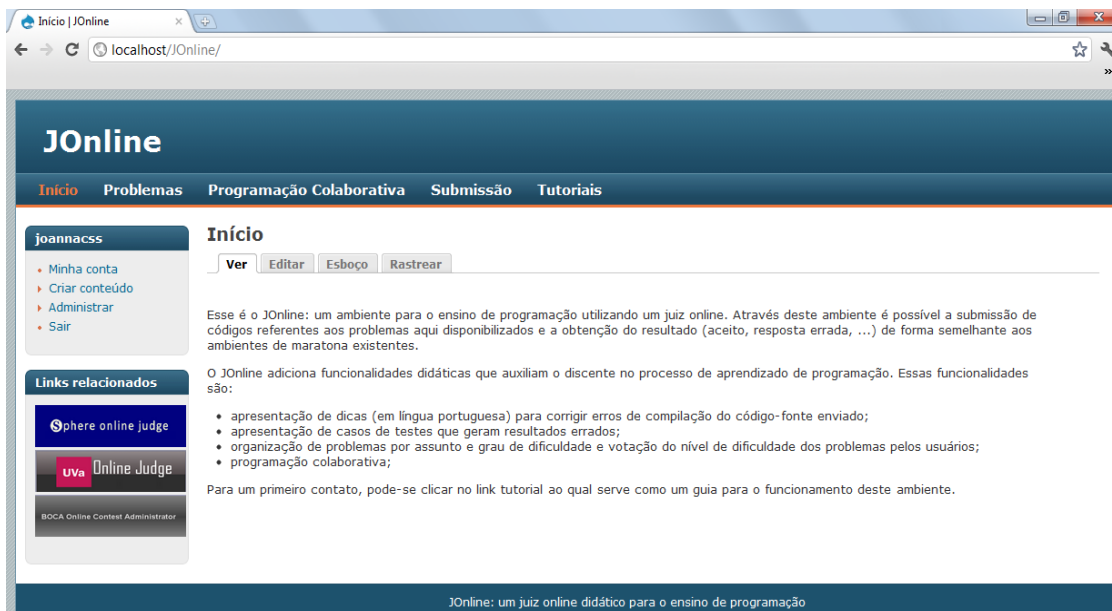


Figura 5. Página inicial do JOnline

O juiz online dessa primeira etapa de desenvolvimento aceita programas escritos em linguagem C/C++.

5. Impactos esperados

A proposta aqui descrita visa ajudar os alunos no aprendizado de programação e o professor na correção de exercícios. O ambiente proposto pode ainda ser utilizado tanto no ensino à distância, quanto no suporte às disciplinas presenciais.

Após o desenvolvimento total do ambiente proposto, espera-se que este contribua de forma significativa na formação acadêmica de indivíduos na área de computação. Espera-se também que esse ambiente possa estimular o aluno na participação de maratonas de programação e criar uma cultura colaborativa através do uso da programação colaborativa disponibilizada pelo ambiente.

6. Considerações finais

Com a proposta descrita nesse artigo é possível obter um ambiente capaz de oferecer suporte ao aprendizado de programação. Este ambiente auxilia o professor no processo de automatização da avaliação de códigos, e o aluno no processo de aprendizagem de desenvolvimento de programas.

O ambiente proposto está em fase de desenvolvimento. No estado atual, o juiz *online* suporta a avaliação de códigos-fonte nas linguagens C e C++. Posteriormente, esse ambiente suportará outras linguagens de programação. Com isto é possível que o aprendizado de programação possa ocorrer sob diferentes linguagens, sendo uma opção do aluno a escolha da linguagem que lhe seja mais familiar.

Atualmente já é permitido que os usuários possam se cadastrar, participar de fóruns,

recuperar documentos PDF contendo o enunciado de algum problema, submeter código-fonte e acessar os problemas e os tutoriais de uso do ambiente.

Referências

- [1] Tobar, C. M. et al., “Uma Arquitetura de Ambiente Colaborativo para o Aprendizado de Programação”, XII Simpósio Brasileiro de Informática na Educação, Vitória-ES, 2001.
- [2] Favero, E. L.; Moreira, M. P., “Um Ambiente para Ensino de Programação com *Feedback* Automático de Exercícios”, Anais do Workshop sobre Educação em Informática, Bento Gonçalves-RS, 2009.
- [3] Ferrandin, M.; Stephani, S. L., “Ferramenta para o ensino de programação via Internet”. I Congresso Sul Catarinense de Computação: UNESC – Criciúma, 2005.
- [4] Kurnia, A.; Lim, A.; Cheang, B., “Online Judge”, *Computer & Education*, vol. 36, No. 4, maio 2001, pp 299-315.
- [5] Campos, C. P.; Ferreira, C. E., “BOCA: um sistema de apoio para competições de programação”, Workshop de Educação em Computação, Anais do Congresso da SBC, Salvador-BA, 2004.
- [6] Sphere Research Labs, “Sphere Online Judge”, www.spoj.pl, julho 2011, 10.
- [7] Universidad de Valladolid, “UVA Online Judge”, <http://uva.onlinejudge.org/>, julho 2011, 10.
- [8] Booch, G.; Rumbaugh, J.; Jacobson, I., “UML - Guia do Usuário”, Tradução Fábio Freitas, Editora Campus, Rio de Janeiro, 2000.
- [9] Drupal.org, “Drupal – Open Source CMS”, <http://drupal.org/>, fevereiro 2011, 12.
- [10] W3C, “W3C HTML”, <http://www.w3.org/html/>, julho 2011, 10.
- [11] Oracle, “Mysql: The world's most popular open source database”, www.mysql.com, fevereiro 2011, 12.
- [12] Apache Software Foundation, “Apache HTTP Server Project.”, <http://httpd.apache.org/>, fevereiro 2011, 12.
- [13] The PHP Group, “PHP: Hypertext Preprocessor”, <http://www.php.net/>, julho 2011, 27.
- [14] Flanagan, D., “Javascript: the definitive guide: Activate Your Web Pages.” Ed. 6. O'Reilly Media, 2011.
- [15] W3C, “Scripting and Ajax - W3C”, <http://www.w3.org/standards/webdesign/script>, julho 2011, 10.
- [16] The jQuery Project, “jQuery: The Write Less, Do More, JavaScript Library”, <http://www.jquery.com/>, julho 2011, 27.

[17] Schildt, H., “Java: the complete reference”, 7 ed., McGraw-Hill, 2007.

[18] Ubuntu-Br, “Ubuntu-Br”, <http://www.ubuntu-br.org>, junho de 2011, 25.