CORREÇÃO AUTOMÁTICA DE EXERCÍCIOS DE LÓGICA DE PROGRAMAÇÃO EM SISTEMAS VIRTUAIS DE APRENDIZAGEM

São Paulo – SP – Abril 2013

Elcio Abrahão – Universidade São Judas Tadeu – USJT - SP – elcioabrahao@usp.br

Categoria: C

Setor Educacional: 3

Classificação da Área de Pesquisa em EAD:

Macro: C / Meso: H / Micro: M

RESUMO

Um dos problemas no ensino de lógica de programação é a quantidade de exercícios propostos aos alunos. A falta de retorno sobre a correção é a maior reclamação dos alunos e uma fonte de trabalho extra para os professores. Em cursos realizados em ambientes virtuais de aprendizagem não existe um mecanismo para correção automática de exercícios introdutórios de lógica de programação, particularmente os feitos na linguagem de programação Portugol, dificultando a utilização prática, principalmente se for considerada a quantidade potencial de alunos. O objetivo deste trabalho é propor um *framework* para correção automática de exercícios em Portugol que possa ser utilizado por qualquer ambiente virtual de ensino que desenvolva um software cliente para o serviço. Nossa metodologia estudou outros trabalhos sobre correção automática, interface humana-computador e melhores práticas de interfaces em ambientes virtuais. O resultado obtido foi em *framework* que pode ser utilizado por qualquer ambiente virtual, independente de plataforma, através de *web service*. Conclui-se que o presente *framework* permite que instrutores de ambientes virtuais de aprendizagem cadastrem gabaritos de exercícios e que os alunos submetam seus exercícios e recebem como retorno a correção automática, sem interferência humana e que por fim todos tenham acesso a dados estatísticos gerados.

Palavras Chave: programação; *web service*; portugol; interface; A.V.A.

1- Introdução

Em cursos de programação de computadores é comum que os alunos sejam testados sobre suas habilidades de programação através de exercícios de lógica de programação. Através destes exercícios o instrutor pode verificar se o aluno está aplicando as técnicas corretas de programação, assim como se está aprendendo os conceitos de lógica e programação propostos. Para isso o aluno deve produzir um programa a partir de determinado enunciado e submeter à correção do professor. Independentemente do número de alunos em cada classe, os exercício devem ser corrigidos e as correções encaminhadas de volta aos alunos. A demora no retorno e a falta de qualidade da correção são as maiores reclamações dos alunos de cursos presenciais.

Nos ambientes virtuais de aprendizagem (A.V.A.), a quantidade potencial de alunos para os cursos de programação de computadores é muito maior que nos cursos presenciais. Neste caso, a utilização de exercícios de lógica de programação é ainda mais complicada, pois o volume de submissões demandaria um enorme esforço dos instrutores para gerar as correções com qualidade e em tempo hábil.

Não existe um sistema capaz de corrigir exercícios de lógica de programação em Portugol voltado para ambientes virtuais de aprendizagem que permita a submissão de gabaritos pelos instrutores, gerencie as submissões dos alunos e forneça estatísticas adequadas do processo.

A proposta deste trabalho é um *framework* para um Serviço de Correção Automática (S.C.E.) de exercícios introdutórios de lógica de programação em Portugol com arquitetura baseada em serviços que poderá ser acessado por qualquer A.V.A. que implemente um cliente para o *web service* .

Faz parte do *framework* a proposta de uma interface de usuário que implementa nos A.V.As as estruturas necessárias para desenvolvimento do cliente do serviço de correção tanto na visão do instrutor quanto do aluno.

O *framework* resultante deste trabalho poderá ser utilizado como ferramenta nos cursos de programação introdutória, libertando o professor de tarefas mecânicas e repetitivas, além de fornecer ao aluno uma correção de maior qualidade e mais rápida do que os métodos convencionais.

Trabalhos relacionados foram estudados na Seção 2. O levantamento dos problemas a serem resolvidos assim como a metodologia utilizada para desenvolver o framework é apresentado na Seção 3. Na Seção 4 o framework é apresentado e, finalmente, na Seção 5 a conclusão e a discussão de trabalhos futuros.

2- Trabalhos Relacionados

Segundo uma revisão feita por [2] os sistemas de correção automática podem ser classificados em três gerações: (a) Primeira Geração - Os primeiros sistemas de automação de testes datam da década de 60 porem sua utilização era limitada a laboratórios particulares. (b) Segunda Geração - A partir de 1989 e durante a década de 90 com o surgimento do sistemas orientados a ferramentas, primeiramente baseados em programação em linha de comando e mais tarde evoluindo para a utilização de interfaces gráficas. (c) Terceira Geração - A partir de 1999 até o presente, com os sistemas baseados em web, onde os alunos tem acesso a interfaces *on line*, disponíveis via Internet.

Em outra revisão realizada por [6], o autor cita a evolução percentual dos trabalhos publicados em conferências sobre ferramentas que dão suporte ao ensino de programação de computadores em diferentes períodos, ver tabela 1.

Período (ano)	1983 a 1993	1994 a 2003	2007
% trabalhos	18,0	24,6	44,0

Tabela 1. Participação dos trabalhos sobre ferramentas de suporte a educação em congressos de tecnologia na educação.

Um mecanismo para correção automática de exercícios introdutórios é sugerida por [12] e compreende as seguintes etapas: (a) a verificação sintática: compilação sem erros. (b) a verificação da presença de comandos de programação obrigatórios: evita que o aluno tente "simular" resultados. (c) a verificação da similaridade estrutural entre o programa do aluno e um conjunto de soluções gabarito e (d) a comparação entre as saídas apresentadas pelo programa do aluno e as saídas definidas como corretas pelo professor. Segundo [12] há uma sinergia entre as quatro técnicas utilizadas, que combinadas aumentam a robustez do mecanismo de correção, o que foi comprovado pelo autor em experimento conduzido com alunos.

A segurança em sistemas automáticos de avaliação de exercícios de programação foi amplamente discutida por [10], com uma preocupação especial com a submissão de programas com códigos maliciosos que possam comprometer o mecanismo de correção.

3- Material e Métodos

Segundo [3] um sistema interativo para apoiar os processos de ensino e aprendizagem possui dois componentes: aplicação e interface. A aplicação é o elemento responsável pela parte funcional do sistema que transforma dados de entrada em dados de saída. A interface é o elemento responsável por traduzir ações dos usuários em ativações das funcionalidades da aplicação, e é a interface que permite que essas ações sejam encaminhadas ao sistema (aplicação).

3.1- Interfaces para Instrutores e Alunos

A submissão de arquivos é uma prática comum na maioria dos gerenciadores de sistemas de aprendizagem (L.M.S.) [14]. Geralmente os L.M.S.s possuem uma ou mais funcionalidades que permitem ao aluno submeter arquivos ou redigir texto a ser armazenado no sistema através de uma interface adequada.

Vários trabalhos têm discutido sobre o desenvolvimento de interfaces para sistemas virtuais de aprendizagem como [3], [4] e [8].

Segundo [15] o projeto de interfaces é uma tarefa pouco estruturada e que não pode ser automatizada, sendo necessário o uso de heurísticas, ou seja, regras informais de julgamento que ajudam pessoas em uma rápida tomada de decisão. Ainda por recomendação de [15] o conjunto de heurísticas possui cinco objetivos básicos a serem alcançados que se baseiam nas qualidades de amigabilidade e usabilidade e são: facilidade de usar, facilidade de aprender, gestão e manipulação de erros, subjetiva satisfação e atração do usuário e adaptabilidade.

Com base nestes critérios o este trabalho apresenta uma sugestão da estrutura a ser considerada para satisfazer as interfaces de instrutor e aluno, apontando as funcionalidades que satisfazem o motor de correção selecionado.

3.2- Aplicação

O modelo de arquitetura proposto para o S.C.E. é orientada a serviço (S.O.A.) [7]. A aplicação responsável pelo motor de correção é implementada no formato de *web service* [16]. Qualquer A.V.A. pode implementar um cliente para o *web service* e acessar o motor de correção através da troca de arquivos XML [17] no formato determinado pelo sistema.

Na seleção da linguagem de programação para o presente trabalho, foram considerados os seguintes fatores: (a) usar a linguagem Portugol. (b) disponibilidade de um compilador e corretor automático já implementado e de código livre. (c) facilidade para portar o compilador para uma aplicação no formato *web service*. (d) corretor que implemente várias técnicas de correção conforme citado por [12]. (e) linguagem com número limitado de interações com o sistema operacional que evite problemas de segurança citados por [10].

A linguagem selecionada, o Portugol, e sua respectiva implementação Portugol Studio, foi desenvolvida por [10] e possui um sistema de correção que foi adaptado com base no trabalho de [12]. Possuir código fonte aberto conforme licença GNU GPL 3.0 [5] e é implementada em Java [11], podendo ser facilmente portada para *web service*.

Uma vez selecionada a linguagem foram agregados os componentes de software necessários para o gerenciamento dos gabaritos submetidos pelos professores, gerenciamento das submissões dos alunos e acumulo dos dados estatísticos sobre as submissões. Estes componentes é que diferenciam e caracterizam o *framework* proposto neste trabalho e permitem que o motor de correção seja exposto como serviço para os ambientes virtuais de aprendizagem.

No momento a implementação do *web service* está em andamento e o próximo passo será o desenvolvimento de um cliente com base no A.V.A. Sakai [13]. A implementação será disponibilizada como código aberto sobre a licença GNU GPL 3.0.

4- Resultados e discussão

O *framework* proposto neste trabalho está dividido em duas áreas: (a) interface para o sistema virtual Aprendizagem (A.V.A.); (b) Serviço de Correção de Exercícios (S.C.E.), ver Figura 1.

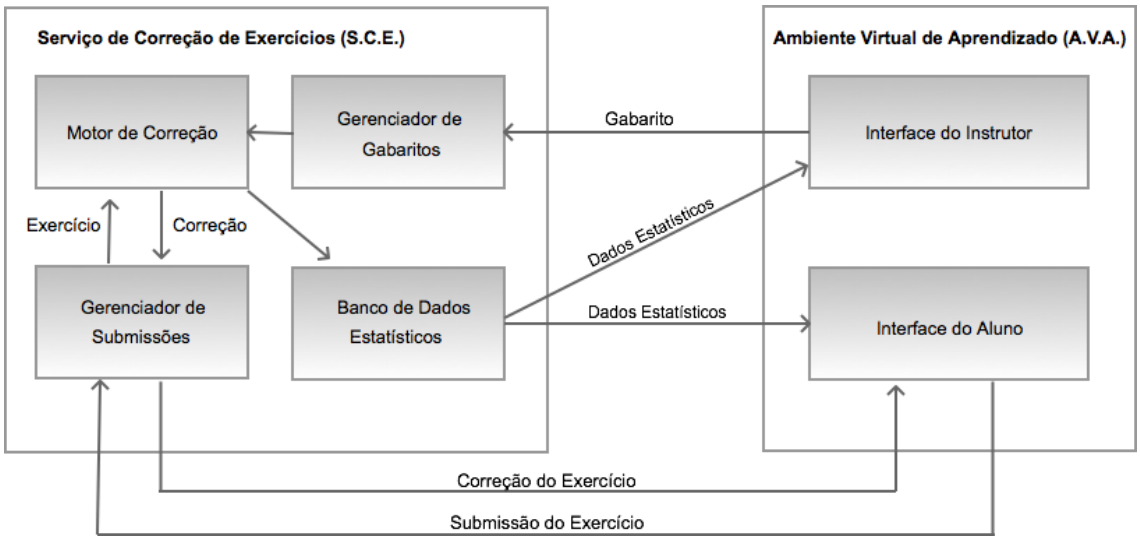

Figura 1. Framework para correção automática de exercícios de lógica de programação.

4.1- Interfaces do A.V.A

A interface do instrutor permite a criação e manutenção de listas de exercícios, que consistem em agrupamentos de questões que, por sua vez, são formadas pelo seu enunciado, valor da nota a ser atribuída e gabarito de soluções. O gabarito, por sua vez, é formado por um conjunto de valores de entradas válidas, conjunto de valores de resultados esperados e código fonte para aquela solução, ver Figura 2.

A interface do instrutor permite o controle da publicação das listas ou questões em datas de início e término pré-determinadas e marcar individual de uma questão como disponível ou não para o aluno do curso. A interface ainda permite que o instrutor informe a quantidade de vezes que uma questão pode ser submetida para o corretor e qual será a nota considerada para a avaliação: maior nota entre as submissões ou média das submissões.

O instrutor possui total acesso aos exercícios submetidos pelos alunos e pode consultar o código fonte de cada submissão a partir de sua própria interface.

As listas de exercícios podem ser armazenadas no formato proprietário do A.V.A. ou no padrão S.C.O.R.M. [1] o que permite a exportação ou importação das listas como objetos de aprendizagem.

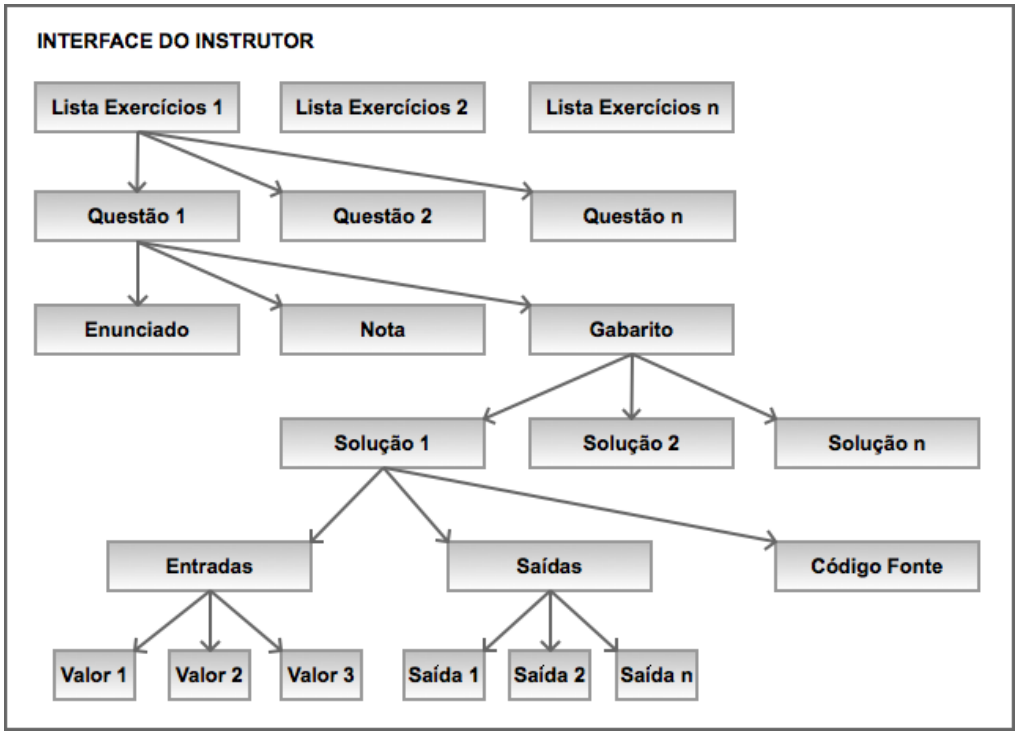

Figura 2. Estrutura da interface do instrutor.

O aluno só tem acesso a listas e questões que foram liberadas previamente pelo instrutor para aquela disciplina. Ao selecionar uma questão para resposta, o aluno é informado se já submeteu esta questão anteriormente e quantas submissões ainda podem ser feitas conforme determinado pelo instrutor. Para submeter uma questão o aluno deverá colar o código fonte da questão no respectivo campo da tela ou fazer o upload do arquivo em formato texto. Uma vez submetido o status da submissão passa de não submetido para submetido até o momento em que o S.C.E. devolva a submissão com a devida correção e nota, ver Figura 3.

Caso o aluno submeta novamente a mesma questão, deverá colar o código fonte no campo respectivo ou fazer o upload novamente. Neste caso o status da questão passa de submetida para ressubmetida.

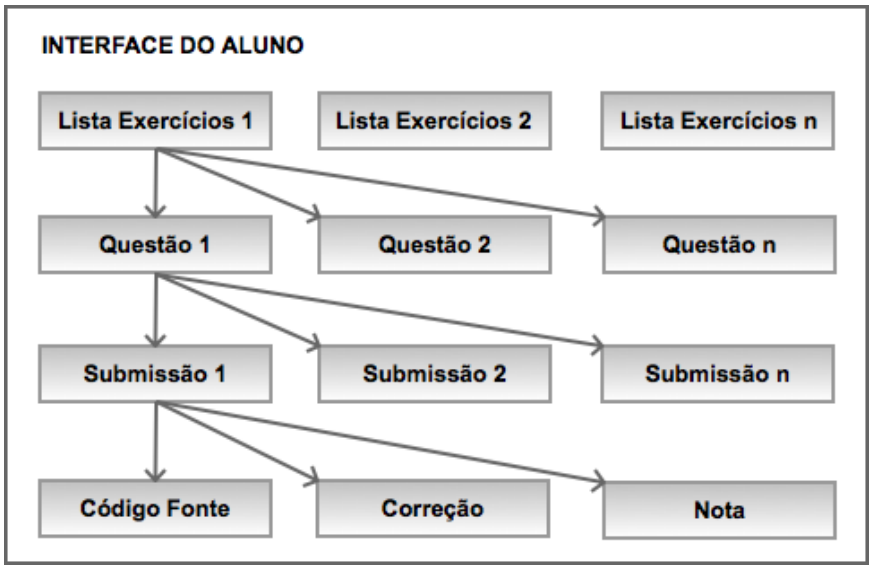

Figura 3. Estrutura da interface do aluno.

4.2- Serviço de Correção de Exercícios

O *web service* do S.C.E. expõe sua interface através três métodos remotos: (a) cadastro de gabarito, (b) submissão de exercícios, (c) consulta de dados estatísticos. As chamadas aos métodos são síncronas, portanto uma vez feita uma chamada o serviço vai devolver o resultado assim que terminar o processamento.

O S.C.E. é formado por quatro componentes: (a) gerenciador de gabaritos: recebe e processa os gabaritos através de arquivos XML que representam a estrutura da interface do instrutor, ver Figura 2. (b) gerenciador de submissões: recebe arquivos XML com a submissão do aluno, retransmite ao motor de correção e envia a resposta de volta ao aluno conforme estrutura da interface do aluno, ver Figura 3. (c) motor de correção: faz correção conforme descrito por [12] e (d) banco de dados de estatísticas: armazenam as informações para recuperação posterior.

5- Conclusões

Este trabalho apresenta um *framework* para a correção automática de exercícios introdutórios de lógica de programação na linguagem portugol baseado na arquitetura SOA. A solução viabiliza a correção de grandes quantidades de exercícios, melhorando a qualidade da correção e diminuindo o trabalho por parte do instrutor.

A proposta é independente de plataforma e a implementação sugerida é baseada em um compilador de portugol já existente que é adaptado para atuação como *web service.* Além de sugerir a arquitetura do *web service* o presente trabalho também sugere uma estrutura para a implementação das interfaces de instrutor e aluno da A.V.A.

Na presente fase, esta sendo desenvolvido o *web service* do serviço de correção de exercícios conforme descrito na seção 3 e em seguida será desenvolvido um cliente para o A.V.A. Sakai.

Em trabalhos futuros poderão ser exploradas outras linguagens de programação e outras funcionalidades do *framework* poderão ser desenvolvidas: submissão em massa, submissão assíncrona, mecanismo para proteção de ataques de negação de serviço, mecanismo para detecção de plágio e autenticação.

Referências

[1] A. D. Learning. "SCORM - http://www.adlnet.gov/scorm/scorm-2004-4th," 22/04/2013, 2013.

[2] D. Christopher, L. David, e O. James, “Automatic test-based assessment of programming: A review,” J. Educ. Resour. Comput., vol. 5, no. 3, pp. 4, 2005.

[3] J. Desconsi, S. R. Silveira, e S. d. C. Bertagnolli, "O desenho de interface para os ambientes virtuais de aprendizagem," 2010.

[4] G. G. Fernandes, e J. A. d. Casto Filho, "Avaliação da usabilidade da interface humano computador de ambientes virtuais de educação (AVE)," 2009.

[5] GNU. "http://www.gnu.org/licenses/gpl.html - GNU General Public License 3.0," 23/04/2013, 2013.

[6] P. Ihantola, T. Ahoniemi, V. Karavirta, e O. Seppälä, "Review of recent systems for automatic assessment of programming assignments." pp. 86-93.

[7] D. Krafzig, K. Banke, e D. Slama, "Enterprise SOA: Service-Oriented Architecture Best Practices", 1 ed.: Prentice Hall, 2004.

[8] P. S. R. Lima, S. R. d. Brito, O. F. Silva, e E. L. Favero, "Adaptação de Interfaces em Ambientes Virtuais de Aprendizagem com Foco na Construção Dinâmica de Comunidades," Novas Tecnologias na Educação, 2005.

[10] M. a. J. M. Luck, “A secure on-line submission system,” Software: Practice and Experience, vol. 29, no. 8, pp. 721--740, 1999.

[10] L. F. Noschang, “Adaptação do Portugol Core para Integração com Outras Ferramentas,” Centro de Ciências Tecnológicas de Terra e do Mar, Universidade do Vale do Itajaí, 2012.

[11] Oracle. "Linguagem de Programação Java - http://www.oracle.com/technetwork/java/index.html," 22/04/2013, 2013.

[12] F. D. Pelz, E. A. d. Jesus, e A. L. A. Raabe, “Um Mecanismo para Correção Automática de Exercícios Práticos de Programação Introdutória,” in 23º Simpósio Brasileiro de Informática na Educação (SBIE 2012), Rio de Janeiro, 2012.

[13] Sakai. "http://www.sakaiproject.org - Sakai Project," 23/04/2013, 2013.

[14] A. Soumplis, E. Koulocheri, N. Kostaras, N. Karausos, e M. Xenos, “Learning Management Systems and Learning 2.0,” Internation Journal of Web-Based Learning and Teaching Technologies, vol. 6, no. 4, pp. 1-18, 2011.

[15] R. Vicari, and G. A. Ascencio, “- Heurísticas Para O Projeto De Interfaces De Sistemas Educacionais,” - Ensaios e Ciência: Ciências Biológicas, Agrárias e da Saúde, vol. - 4, pp. - 75-97, 2000.

[16] W3C. "http://www.w3.org/TR/ws-arch/ - Web Service," 23/04/2013, 2013.

[17] W3C. "http://www.w3.org/XML/ - Extensible Markup Language (XML) 1.0 (Fifth Edition)," 23/04/2013, 2013.