

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Sharpener: Ferramenta de correção automática de programas para o ensino de programação com workflow expandido.

Pedro Morello Abbud



São Carlos – SP

Sharpener: Ferramenta de correção automática de programas para o ensino de programação com workflow expandido.

Pedro Morello Abbud

Orientador: Prof. Dr. Dilvan de Abreu Moreira

Coorientador: Prof. Dr. Orlando de A. Figueiredo

Monografia final de conclusão de curso apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação.

Área de Concentração: Sistemas Computacionais

USP – São Carlos

Outubro de 2019

RESUMO

ABBUD, P. M.. *Sharpeners: Ferramenta de correção automática de programas para o ensino de programação com workflow expandido..* 2019. 14 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Resumo

Palavras-chave: Palavras chave 1, palavras chave 2.

ABSTRACT

ABBUD, P. M.. *Sharpeners: Ferramenta de correção automática de programas para o ensino de programação com workflow expandido..* 2019. 14 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Resumo em inglês

Key-words: Keyword 1, Keyword2.

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Motivação e Contextualização	9
1.2	Objetivos	10
1.3	Organização	10
	REFERÊNCIAS	11
ANEXO A	LINKS RELACIONADOS	13

INTRODUÇÃO

1.1 Motivação e Contextualização

Nas últimas décadas, o interesse por programação cresceu de forma extraordinária e cursos introdutórios mostram-se cada vez mais populares. No entanto, cursos de programação ainda são considerados difíceis e, com frequência, possuem taxas de desistência elevada. Iniciantes sofrem de uma vasta gama de déficits e dificuldades, desde a compreensão de constructos básicos de uma linguagem de programação a elaboração de planos para resolver um dado problema (ROBINS; ROUNTREE; ROUNTREE, 2003).

Um grande número de estudos concluem, de forma semelhante, que alunos, mesmo tendo aprendido a sintaxe e semântica de uma linguagem de programação, tendem a falhar em compor os blocos fundamentais estudados em programas válidos. Parte deste problema é abordado por Davies (1993), que faz uma distinção de duas partes essenciais no aprendizado em programação: conhecimento e estratégia. Denota-se conhecimento em programação como a parte declarativa do conhecimento, *e.g.* saber que existe um constructo chamado *for* na linguagem e seu propósito. Já estratégia em programação seria empregar de forma adequada o conhecimento em face de um problema, como o uso pertinente de um *for* em um programa.

Davies (1993) defende que grande parte das abordagens em ensino tendem a investir excessivamente na parcela de conhecimento em programação, muitas vezes deixando de lado o aspecto de estratégia.

Além desta falta de ênfase no ensino de estratégia em programação, Winslow (1996) afirma que, ainda que o aluno saiba resolver manualmente o problema, existe uma grande dificuldade na adaptação desta solução para seu equivalente em um programa de computador. Este fato corrobora com a teoria de Brooks (1983) em que se afirma que programação é o processo de construir mapeamentos entre o domínio problema, possivelmente passando por muitos domínios intermediários, para chegar finalmente no domínio da programação. Por esta teoria, o entendimento de um programa nada mais é que a reconstrução de ao menos uma parte destas relações entre domínios e que este processo é orientado a criação, confirmação e refinamento de hipóteses.

A complexidade da tarefa não só se concentra no ensino de conhecimento e estratégia em programação nem na elaboração de habilidades que relacionem o domínio do problema ao domínio da programação, mas na flexibilidade que o iniciante deve alcançar na resolução

de problemas. Desta forma, o processo de escrita de um programa deve ser oportunística e incremental, em que o plano adotado e problema sempre estão em constante reavaliação, e que isto pode resultar em desvios e reformulações do plano original (VISSER, 1990).

Para que seja possível instrumentalizar esta escrita oportunística e incremental, o aluno deve adquirir outras habilidades como *debugging* de programas e familiaridade com um ambiente de programação.

Em vista ao imenso desafio que é o ensino de programação a iniciantes, diversas ferramentas foram desenvolvidas para dar apoio pedagógico ao professor nesta tarefa. Uma razoável parcela destas ferramentas são sistemas de juízes *online*, que consistem em sistemas capazes de receber códigos desenvolvidos pelos usuários como resposta a um determinado problema, mas com a capacidade de dar um *feedback* imediato ao usuário, informando se o código está correto ou não, através de análise sintática e testes automatizados.

Já existe uma grande variedade em juízes *online* disponíveis na literatura científica, tais como: *The Huxley* [Paes *et al.* (2013)], *CodeBench* [Galvão, Fernandes e Gadelha (2016)], *UVa Online Judge* [Revilla, Manzoor e Liu (2008)], *feeper* [Alves e Jaques (2014)], *URI Online Judge* [Bez, Tonin e Rodegheri (2014)], *BOCA* [Campos e Ferreira (2004)], *RunCodes*, *Sphere Online Judge*, *HackerRank*, *CodeChef*, *InterviewBit*, *Kattis*, *LeetCode*, *CodinGame*, *CodeSignal*, *CodeWars*, *Exercism*, entre tantos outros.

Apesar da grande variedade e maturidade destas plataformas existentes, poucas tem como foco ser uma ferramenta de auxílio didático ao professor em sala de aula e praticamente nenhuma é de código aberto. Assim sendo, existe uma lacuna que pode ser preenchida por um sistema que tenha estes dois focos e dê liberdade ao professor a implementar técnicas de ensino que o convenha.

1.2 Objetivos

Pretende-se neste trabalho desenvolver uma prova de conceito de um sistema para apoio ao ensino de programação em salas de aula, que seja de código aberto e que forneça ferramentas necessárias para o professor conduzir e monitorar exercícios práticos dados em sala de aula.

1.3 Organização

No Capítulo 2 pretende-se fazer uma revisão bibliográfica das técnicas empregadas na implementação da prova de conceito do sistema *Sharpenner*. A seguir no Capítulo 3 descreve-se a especificação funcional do sistema tais como seus detalhes de implementação. Finalmente no Capítulo 4 apresentam-se os resultados e possíveis trabalhos futuros.

REFERÊNCIAS

- ALVES, F. P.; JAQUES, P. Um ambiente virtual com feedback personalizado para apoio a disciplinas de programação. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [S.l.: s.n.], 2014. v. 3, n. 1, p. 51. Citado na página 10.
- BEZ, J. L.; TONIN, N. A.; RODEGHERI, P. R. Uri online judge academic: A tool for algorithms and programming classes. In: IEEE. **2014 9th International Conference on Computer Science & Education**. [S.l.], 2014. p. 149–152. Citado na página 10.
- BROOKS, R. Towards a theory of the comprehension of computer programs. **International journal of man-machine studies**, Elsevier, v. 18, n. 6, p. 543–554, 1983. Citado na página 9.
- CAMPOS, C. P. de; FERREIRA, C. E. Boca: um sistema de apoio a competições de programação. 2004. Citado na página 10.
- DAVIES, S. P. Models and theories of programming strategy. **International Journal of Man-Machine Studies**, Elsevier, v. 39, n. 2, p. 237–267, 1993. Citado na página 9.
- GALVÃO, L.; FERNANDES, D.; GADELHA, B. Juiz online como ferramenta de apoio a uma metodologia de ensino híbrido em programação. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2016. v. 27, n. 1, p. 140. Citado na página 10.
- PAES, R. de B.; MALAQUIAS, R.; GUIMARÃES, M.; ALMEIDA, H. Ferramenta para a avaliação de aprendizado de alunos em programação de computadores. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [S.l.: s.n.], 2013. v. 2, n. 1. Citado na página 10.
- REVILLA, M. A.; MANZOOR, S.; LIU, R. Competitive learning in informatics: The uva online judge experience. **Olympiads in Informatics**, Institute of Mathematics and Informatics, v. 2, n. 10, p. 131–148, 2008. Citado na página 10.
- ROBINS, A.; ROUNTREE, J.; ROUNTREE, N. Learning and teaching programming: A review and discussion. **Computer science education**, Taylor & Francis, v. 13, n. 2, p. 137–172, 2003. Citado na página 9.
- VISSER, W. More or less following a plan during design: opportunistic deviations in specification. **International journal of man-machine studies**, Elsevier, v. 33, n. 3, p. 247–278, 1990. Citado na página 10.
- WINSLOW, L. E. Programming pedagogy—a psychological overview. **ACM Sigcse Bulletin**, ACM, v. 28, n. 3, p. 17–22, 1996. Citado na página 9.

LINKS RELACIONADOS

<<https://www.thehuxley.com>>: Página da plataforma *The Huxley*, que utiliza um sistema de juiz *online* para auxiliar professores em aulas de programação.

<<http://codebench.icomp.ufam.edu.br>>: Página da plataforma *CodeBench*, que utiliza um sistema de juiz *online*, de forma gamificada, para auxiliar professores em aulas de programação.

<<https://uva.onlinejudge.org>>: Página da plataforma *UVa Online Judge*, um sistema de juiz *online* desenvolvido pela *University of Valladolid*.

<<http://feeper.unisinos.br>>: Página da plataforma *UVa Online Judge*, um sistema de juiz *online* desenvolvido pela *Universidade do Vale do Rio dos Sinos*.

<<https://www.urionlinejudge.com.br>>: Página da plataforma *URI Online Judge*, um sistema de juiz *online* desenvolvido pela *Universidade Regional Integrada*.

<<https://www.ime.usp.br/~cassio/boca>>: Página da plataforma *BOCA*, um sistema de juiz *online* desenvolvido pelo *Instituto de Matemática e Estatística* para apoio em competições.

<<https://we.run.codes>>: Página da plataforma *RunCodes*, um sistema de juiz *online* desenvolvido pelo *Instituto de Matemática e Estatística*, para auxiliar professores em aulas de programação.

<<https://www.spoj.com>>: Página da plataforma *Sphere Online Judge*, um sistema de juiz *online* desenvolvido pela *Sphere Research Labs*.

<<https://www.hackerrank.com>>: Página da plataforma *HackerRank*, que utiliza um sistema de juiz *online* para auxiliar na contratação de funcionários para empresas.

<<https://www.codechef.com>>: Página da plataforma *CodeChef*, que utiliza um sistema de juiz *online*, desenvolvida pela empresa *Directi*, para competições em programação.

<<https://www.interviewbit.com>>: Página da plataforma *HackerRank*, que utiliza um sistema de juiz *online* para auxílio na preparação de entrevistas para empresas.

<<https://open.kattis.com>>: Página da plataforma *Kattis*, um sistema de juiz *online* desenvolvido para promover competições.

<<https://leetcode.com>>: Página da plataforma *LeetCode*, que utiliza um sistema de juiz *online* para auxílio na preparação de entrevistas para empresas.

<<https://www.codingame.com>>: Página da plataforma *CodinGame*, que utiliza um sistema de juiz *online*, de forma completamente gamificada, para ensino e auxiliar empresas na contratação de funcionários.

<<https://codesignal.com>>: Página da plataforma *CodeSignal*, que utiliza um sistema de juiz *online* para auxiliar empresas na contratação de funcionários.

<<https://www.codewars.com>>: Página da plataforma *CodeWars*, que utiliza um sistema de juiz *online* para auxiliar professores em aulas de programação e empresas na contratação de funcionários.

<<https://exercism.io>>: Página da plataforma *Exercism*, que utiliza um sistema de juiz *online*, para ensino. As submissões contam com feedback de instrutores voluntários.