

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

GUSTAVO CORREIA GONZALEZ

MONOGRAFIA

CAMPO MOURÃO

2016

GUSTAVO CORREIA GONZALEZ

Proposta de Trabalho de Conclusão de Curso de Graduação apresentado à disciplina de Trabalho de Conclusão de Curso 1, do Curso de Bacharelado em Ciência da Computação do Departamento Acadêmico de Computação da Universidade Tecnológica Federal do Paraná, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Igor Scaliante Wiese

Coorientador: Prof. Dr. Marco Aurélio Graciotto Silva

CAMPO MOURÃO

2016

Resumo

Gonzalez, Gustavo. . 2016. 9. f. Monografia (Curso de Bacharelado em Ciência da Computação), Universidade Tecnológica Federal do Paraná. Campo Mourão, 2016.

O elevado nível de reprovação em disciplinas onde são ensinados conceitos básicos de programação (Holcomb; Simone, 2016), em qualquer grau de ensino, é um problema enfrentado por muitos alunos e tem sido alvo de variadas pesquisas. Existe um conjunto de razões que estão relacionadas com a origem do problema, como o método de ensino e aprendizagem, falta de algumas competências e interesse por parte dos alunos (Sinclair *et al.*, 2015), e a própria dificuldade do tema.

Contexto: O desenvolvimento da pesquisa será realizado na Universidade Tecnológica Federal do Paraná, com alunos selecionados entre o primeiro e segundo ano da matéria de Algoritmos e Algoritmos e Estruturas de Dados 1.

Objetivo: Investigar o impacto do uso do mecanismo de dicas personalizadas no ensino de conceitos básicos de programação, mais especificamente em estrutura de condição e laço de repetição. Para atingir esse objetivo, foram criadas duas questões de pesquisas descritas como:

- **QP₁:** *Existe impacto do uso do mecanismo de dicas para minimizar erros e melhorar soluções de exercícios?*

Esta questão de pesquisa tem como objetivo investigar se o mecanismo de dicas irá ajudar os alunos a obterem melhores resultados na execução de exercícios de programação.

- **QP₂:** *Quais dicas personalizadas ajudam mais os alunos?*

Esta questão de pesquisa avaliará qual tipo de dica é melhor, podendo ser randomizadas em função do tipo do exercício, geradas a partir de erros cometidos em exercícios passados e sem dicas. Levando em consideração a fonte da dica gerada, podendo ser por alunos iniciantes, alunos experientes ou professores.

Método: Será desenvolvido um software para auxiliar o aprendizado de programação que utilizá um sistema de dicas personalizadas aos alunos no momento da realização de exercícios. Para validar a eficiência do software será realizando um estudo controlado com três grupos de

alunos distribuídos de forma homogênea em relação ao nível de conhecimento na programação. O primeiro grupo receberá dicas classificadas a partir de erros anteriores, o segundo será com dicas randomizadas levando em consideração o assunto abordado no exercício e o terceiro sem dicas.

Resultados esperados: Esperamos que o software realmente ajude no aprendizado dos alunos na programação tornando o ensino mais dinâmico e personalizado. Para que no futuro possa diminuir o número de reprovações nas matérias introdutórias de programação e em desistências do curso de Ciência da Computação.

Palavras-chaves: Aplicação, Dicas, Dicas Personalizadas, Learnersourcing, Crowdsourcing

Lista de figuras

Lista de tabelas

Sumário

1	Introdução	6
2	Referencial Teórico	8
	Referências	9

Introdução

O ensino de linguagens de programação tem o propósito de conseguir desenvolver nos alunos um conjunto de competências necessárias para conceber programas e sistemas computacionais capazes de resolver problemas reais. O insucesso na aprovação dos estudantes em disciplinas de programação, é um tema que tem sido alvo de muitos estudos (Bosse; Gerosa, 2015), (Cukierman, 2015), .

O problema do insucesso é evidenciado por (Lahtinen *et al.*, 2005) realizaram uma pesquisa com 559 alunos e 34 professores de diferentes universidades para estudar as dificuldades na aprendizagem de programação, como resultado obtido foi percebido que as questões mais difíceis na programação são: a compreensão de como projetar um programa para resolver uma tarefa determinada, dividir as funcionalidades em procedimentos e encontrar erros de seus próprios programas. Estas são as capacidades que o aluno deve obter para entender as maiores entidades do programa em vez de apenas alguns detalhes sobre eles. Já os conceitos de programação mais difíceis foram recursão, ponteiros e referências, tipos de dados abstratos, manipulação de erro. Os professores apontaram como sendo os conteúdos mais difíceis os mesmos que os alunos.

O objetivo desse trabalho é criar um software de código aberto para auxiliar na aprendizagem de programação, implementando um sistema colaborativo de dicas escritas pelos próprios usuários. Para investigar se o uso do mecanismo de dicas personalizadas é capaz de melhorar o rendimento dos alunos nos exercícios de estrutura de condição e laço de repetição.

No software serão implementadas funcionalidades que permitam o usuário resolver exercícios em qualquer linguagem de programação, fornecer dicas para a solução de exercícios, um diário para relatar as dificuldades enfrentadas durante a execução dos exercícios.

Para realizar a validação da ferramenta será realizado um estudo controlado na Universidade Tecnológica Federal do Paraná com três grupos de estudantes escolhidos aleatoriamente de forma homogênea em relação ao nível de conhecimento na programação, um grupo utilizará o software de dicas normal, outro irá dispor de dicas personalizado e o último utilizará o software sem o sistema de dicas personalizadas.

Os próximos capítulos estão organizados em três partes. O objetivo do capítulo de referencial teórico é apresentar e discutir os principais conceitos que envolvem este trabalho. Este procedimento é importante, pois discutiremos os pontos de vista de diversos autores, assim como diversas abordagens alternativas. O objetivo do capítulo de proposta é apresentar a metodologia que será utilizada para desenvolver o estudo e o software. O objetivo do capítulo de resultados preliminares é apresentar os avanços que o estudo obteve durante o seu desenvolvimento.

Referencial Teórico

aaaa

Referências

BOSSE, Yorah; GEROSA, Marco Aurélio. Reprovações e trancamentos nas disciplinas de introdução à programação da universidade de são paulo: Um estudo preliminar. In: *XXIII WEI-Workshop sobre Educação em Informática. Recife, Julho*, 2015.

CUKIERMAN, Diana. Predicting success in university first year computing science courses: The role of student participation in reflective learning activities and in i-clicker activities. In: *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, 2015. (ITiCSE '15), p. 248–253. ISBN 978-1-4503-3440-2. Disponível em: <http://doi.acm.org/10.1145/2729094.2742623>.

HOLCOMB, Kayla M.; SIMONE, Nevan F. The role of chronology in analyzing introductory programming assignments (abstract only). In: *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 2016. (SIGCSE '16), p. 725–726. ISBN 978-1-4503-3685-7. Disponível em: <http://doi.acm.org/10.1145/2839509.2851062>.

LAHTINEN, Essi; ALA-MUTKA, Kirsti; JÄRVINEN, Hannu-Matti. A study of the difficulties of novice programmers. In: *Acm. ACM SIGCSE Bulletin*, 2005. v. 37, n. 3, p. 14–18.

SINCLAIR, Jane; BUTLER, Matthew; MORGAN, Michael; KALVALA, Sara. Measures of student engagement in computer science. In: *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, 2015. (ITiCSE '15), p. 242–247. ISBN 978-1-4503-3440-2. Disponível em: <http://doi.acm.org/10.1145/2729094.2742586>.