

# Análise de um Estudo de Caso para Aprendizagem de Programação em Grupo

Thais Castro, Hugo Fuks, Marcos A. F. Spósito, Alberto N. de Castro Jr.

**Title—** A Case Study Analysis for Group Programming Learning.

**Abstract—** This paper describes a case study on group programming learning that has as central components the identification of specific demands for such domain and the requirement analysis for collaborative programming learning, using the record of activities and codes developed.

**Index Terms—** group programming learning, text mining, computer education, and technology for programming

## I. INTRODUÇÃO

A comunidade acadêmica de Ciência da Computação tem sempre procurado entender porque programação é uma atividade difícil e como as metodologias adotadas em cursos introdutórios influenciam a maneira pela qual os alunos aprendem a programar. Com respeito à aprendizagem, independentemente da área considerada, é sabido que o trabalho em grupo promove no indivíduo o desenvolvimento de habilidades e estratégias para a solução de problemas, o que é de extrema importância à construção do conhecimento nesse domínio [23].

O trabalho em grupo tem se provado ser uma necessidade tanto no mercado de trabalho quanto na educação. Do ponto de vista da indústria, a demanda crescente por produtos e serviços tem levado à competição exacerbada, forçando as empresas a buscarem convergência em suas atividades, de forma que evitem a duplicação de esforços para a mesma tarefa, priorizando assim a colaboração entre os vários times (equipes de trabalho) [3]. Do ponto de vista da educação, ao colaborar, os alunos podem ver o ponto de vista de seus pares e juntos construir uma solução para o problema apresentado. O efeito das contribuições proporcionado pela colaboração é facilmente percebido ao se acompanhar um

curso em andamento em um Ambiente Virtual de Aprendizagem (AVA), onde se podem analisar diferenças substanciais no refinamento das soluções registradas [1].

Este artigo descreve um estudo de caso conduzido com estudantes cursando o primeiro semestre de Ciência da Computação e Engenharia da Computação. O objetivo desse estudo de caso foi se verificar a adequabilidade e a aceitabilidade da programação em grupo, através de análises quantitativa e qualitativa do conteúdo desenvolvido por 9 grupos de alunos no contexto deste trabalho.

Na Seção 2 é apresentada uma revisão bibliográfica em aprendizagem de programação, enfatizando as dificuldades em se registrar os processos cognitivos envolvidos. Na Seção 3 é apresentado o estudo de caso supracitado juntamente com suas análises quantitativa e qualitativa. Finalmente, na Seção 4 apresentamos a conclusão de nosso trabalho.

## II. PROGRAMAÇÃO E SUA APRENDIZAGEM

A aprendizagem de conceitos e métodos para a construção de programas de computador não é trivial, uma vez que requer o uso de habilidades de alto nível e muito raciocínio abstrato. Em [8] é enfatizado que programar envolve mais raciocínio que qualquer outra habilidade. No entanto, programação também é uma tarefa de engenharia, pois lida com a produção de artefatos que devem satisfazer requisitos de qualidade e estar sujeitos a verificação.

Em [19] é destacado que em cursos introdutórios os alunos devem aprender técnicas de solução de problemas. O que acontece frequentemente é que os alunos acham muito difícil em aplicar suas habilidades previamente adquiridas. Isto termina por se tornar uma fonte de medo e frustração, resultando em evasão. No trabalho descrito em [7] é um esforço em se desenvolver um novo curso de programação baseado em sessões de laboratório. Muitas atividades foram planejadas como discussões online, exercícios de programação em pares, leituras de textos da Internet, anotações sobre reflexões, entradas em diários e colaborações utilizando o processo de revisão por pares para criticar as respostas dos colegas ao tópico em questão.

O último artigo trata da transformação de uma metodologia envolvendo aulas práticas e teóricas para outra que utiliza apenas aulas práticas, com atividades distintas bem distribuídas ao longo das sessões. Apesar da preocupação em se desenvolver questionários para criar nos alunos o hábito da reflexão, tal metodologia se provou ineficaz para detectar confusão na apreensão de conceitos. Como os exercícios de

T. Castro é uma aluna de doutorado da Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Brasil. Ela também é professora do Departamento de Ciência da Computação da Universidade Federal do Amazonas. E-mail: [tcastro@inf.puc-rio.br](mailto:tcastro@inf.puc-rio.br).

Hugo Fuks é professor associado do Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro, Brasil. E-mail: [hugo@inf.puc-rio.br](mailto:hugo@inf.puc-rio.br).

M. A. F. Spósito é professor no IFRR no Brasil. E-mail: [sposito@gmail.com](mailto:sposito@gmail.com).

A. N. Castro Jr. professor do Departamento de Ciência da Computação da Universidade Federal do Amazonas. E-mail: [alberto@dcc.ufam.edu.br](mailto:alberto@dcc.ufam.edu.br).

DOI (Digital Object Identifiers) Pendente.

programação foram conduzidos em duplas não há evidências de uma melhora no desempenho resultante desta técnica pois não houve registro das atividades dessas duplas ou mesmo um grupo de controle.

Prosseguindo com a linha de avaliação, o trabalho apresentado em [6] descreve uma combinação de algumas técnicas de avaliação em um curso introdutório de computação e demonstra por meio de análise estatística as diferenças e relacionamentos entre essas técnicas. O curso foi planejado Segundo a premissa de que antes que os alunos aprendam a programar eles devem ser capazes de resolver problemas. Conseqüentemente, primeiro os alunos resolvem problemas sem o uso de uma linguagem de programação e, posteriormente, aprendem como utilizá-la para representar soluções.

Após a fase inicial de solução de problemas, o curso continua com seis sessões de laboratório onde os alunos devem resolver problemas individualmente, sendo permitida a consulta ao outro membro da dupla sempre que eles acharem necessário. Vale ressaltar que a complexidade aumenta conforme as sessões avançam. Assim que esta fase termina é proposto um estudo de caso consistindo no desenvolvimento de um programa (normalmente um jogo) por pequenos grupos (2 a 4 membros). Uma avaliação da aprendizagem é conduzida comparando estatisticamente o desempenho dos alunos nas sessões de laboratório. Tal estudo de caso controlou práticas individuais sem consulta (como testes) sem consulta, o que foi conduzido duas vezes ao longo do curso.

A contribuição do trabalho acima mencionado é a análise estatística das correlações entre os mecanismos de avaliação utilizados. Embora seja importante saber se os alunos estão sendo avaliados pelos métodos mais eficientes para sua aprendizagem, outros fatos não são levados em conta, como o que ocorre com a natureza do trabalho desenvolvido em grupos – por exemplo, não há controle sobre o desenvolvimento de cada trabalho, tornando impossível saber se uma dada tarefa foi realizada por determinado aluno, o que causaria um erro nas correlações.

Novamente utilizando modelos de avaliação para ensinar programação, o trabalho descrito em [15] utiliza a pré-avaliação dos alunos como uma base para categorizá-los em estágios de aprendizagem conforme definido na taxonomia de Bloom. A partir daí o curso é formatado de forma que ofereça atividades diferenciadas para os alunos em diferentes estágios de treinamento.

Em um esforço para identificar aspectos que podem facilitar a aprendizagem de programação, o trabalho descrito em [9] afirma que através da resposta dos alunos a questões como “o que é programação?” é possível definir a ordem de apresentação dos paradigmas de programação. Os autores partem do princípio que os alunos precisam saber o que aprendizagem de programação realmente é a fim de que aprenda efetivamente. A maioria das respostas aferidas sugere que os alunos sejam primeiramente expostos para um raciocínio mais estruturado antes de serem apresentados às abstrações de objetos da POO.

O que é realmente necessário para facilitar a aprendizagem de programação ainda é uma questão em aberto. Embora os

artigos acima mencionados tentem encontrar uma resposta a essa questão, não há trabalhos na literatura revista que tenha sucedido em estabelecer métodos incontestáveis e técnicas para aprendizagem de programação em grupo. Por outro lado, houve iniciativas cujo foco é na criação e manutenção do interesse dos alunos no curso utilizando conceitos de extreme programming, já amplamente utilizado por times de desenvolvimento na indústria de software.

### III. UM ESTUDO DE CASO PARA UM CURSO INTRODUTÓRIO DE PROGRAMAÇÃO

No primeiro semestre de 2007 foi desenvolvido um estudo de caso em introdução à programação com duas turmas de alunos matriculados no primeiro semestre dos cursos de graduação em Ciência da Computação e Engenharia da Computação na Universidade Federal do Amazonas (UFAM).

O objetivo desse estudo de caso foi proporcionar aos alunos a experiência no desenvolvimento de soluções para problemas complexos através da distribuição de tarefas, negociação, composição de soluções parciais e refinamentos sucessivos. Isto foi alcançado trabalhando em grupos de até 5 alunos os quais eram também responsáveis pelo registro das atividades desenvolvidas ao longo das etapas do trabalho, utilizando um ambiente baseado no controle de versões chamado AAEP [1]. Ao final do curso, como um trabalho final, foi proposta uma tarefa em grupo. Conforme exigência da tarefa, a solução deveria ser acompanhada de um registro das interações entre os membros das equipes. Após a entrega e correção dessas tarefas, houve uma fase de análise e interpretação dos dados gerados baseados nos processos de classificação, codificação e tabulação.

Além do desenvolvimento de códigos e manutenção do registro das interações, os alunos também responderam a questionário, o qual foi submetido a uma análise quantitativa. Finalmente, eles puderam expressar livremente sobre suas dificuldades e sobre o desenvolvimento dos trabalhos, sendo esses comentários submetidos a uma análise qualitativa.

#### A. Análise quantitativa

O objetivo do questionário era descobrir o nível de aderência à colaboração na aprendizagem de programação como uma forma de mudar de práticas individuais para aprendizagem de programação em grupo através da interação entre os membros dos grupos. Os dados foram analisados de acordo com a distribuição absoluta sugerida em [14] e descrita na Tabela I.

Conforme destacado na análise mostrada na Tabela I, houve 9 grupos de respondentes e em todos os critérios houve uma predominância de respostas “Totalmente”, o que indica satisfação e sucesso na execução da tarefa em grupo. Na maioria dos critérios, a maioria das respostas corresponde à primeira coluna (Totalmente) indicando que houve uma tentativa de seguir a especificação do problema. No entanto, no critério “Busca por códigos similares” e “integração entre os membros do grupo” as respostas foram quase igualmente distribuídas entre as três colunas, indicando que os



e resultaram nos dados da Tabela II, que descreve as dificuldades sentidas pelos grupos e na Tabela III, que descreve as conclusões fornecidas pelos grupos, conforme descritas nos relatórios de desenvolvimento dos grupos.

TABELA III  
CONCLUSÕES FORNECIDAS PELOS GRUPOS

<i>Informantes</i>	<i>Descrição das conclusões fornecidas pelos grupos</i>
A	Foi muito produtivo; nós colocamos nosso conhecimento em prática; nós aprendemos como trabalhar em grupo.
B	Ensinou-nos a trabalhar como um time, ajudando cada participante a amadurecer e aprender como lidar com as dificuldades e diferenças dos outros; houve um comprometimento do grupo em todas as atividades realizadas.
C	Não relatado
D	Não relatado
E	Demandou um trabalho conjunto do grupo; demandou uma conexão e, sobretudo consenso entre todos os membros do grupo; comunicação entre os membros do time foi mantida principalmente via Internet (Chat e e-mail).
F	Não relatado
G	Não relatado
H	Não relatado
I	Uma discussão direcionada do grupo foi necessária para se encontrar uma solução viável.

Para clarificar textos difusos, ambíguos ou contraditórios envolvendo conteúdo relacionado ao contexto de aplicação, como, por exemplo, informação sobre autor, situações gerativas, etc, os pesquisadores utilizaram a técnica de análise explanatória de conteúdo. Baseados nessa análise, as percepções dos pesquisadores sobre as dificuldades sentidas pelos grupos são apresentadas na Tabela IV.

Considerando as conclusões relatadas pelos grupos, é possível formular a seguinte paráfrase explanatória: “é necessário se trabalhar em grupo e há uma demanda por

compromisso, esforço e acordo dos participantes. As atividades propostas foram benéficas e representaram uma boa oportunidade de pôr em prática o conhecimento em programação até então adquirido por todos”.

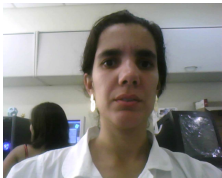
Os grupos experimentaram muitas dificuldades principalmente relacionadas à codificação em linguagem Haskell da solução proposta. Mesmo que o planejamento da solução tenha sido muito difícil para os alunos, já que esta habilidade (solução colaborativa de problemas) depende de coordenação e interação, as principais dificuldades relatadas foram as relacionadas às habilidades necessárias ao processo de codificação, utilização das técnicas de programação apropriadas e conhecimento da linguagem de programação específica. Isto resulta que aprendizagem de programação pode ser mais eficiente quando conduzida em grupo, desde que siga um modelo ou esquema que facilite este processo.

TABELA IV  
PERCEPÇÕES SOBRE AS DIFICULDADES SENTIDAS PELOS GRUPOS

<i>Informante</i>	<i>Percepções dos pesquisadores</i>
A	Interpretação da declaração de algumas questões.
B	Implementação do código; reunião de todo o time; alcance de um consenso; agregação de todas as soluções.
C	Implementação do código; agregação de todas as soluções.
D	Entendimento de como se constrói um programa; implementação do código; verificação de erros; utilização do interpretador Hugs.
E	Refinamento de soluções
F	Implementação do código
G	Utilização do interpretador Hugs; implementação do código;
H	Refinamento das soluções; utilização da recursão.
I	Documentação da atividade desenvolvida; planejamento da solução; implementação do código.

Finalmente, a maioria dos grupos relatou que a experiência de desenvolver uma atividade de programação em grupo foi positiva. Foi também observado que tão importante quanto a





**Thais Castro** é graduada pela UFAM (2000), mestre em Informática pela UFES (2003) e cursa doutorado em Informática na PUC-Rio. É professora assistente da UFAM, onde atua no grupo de pesquisa em Sistemas Inteligentes. Tem experiência na área de Ciência da Computação, com ênfase em Inteligência Artificial, Engenharia de Software e Informática na Educação, atuando principalmente nos seguintes temas: aprendizagem de programação, compreensão de programas, refatoração de programas, aprendizado mediado por agentes, engenharia semiótica aplicada a ambientes colaborativos e metodologias de ensino de computação.



**Hugo Fuks** é professor associado do Departamento de Informática da PUC-Rio, Pesquisador do CNPq e bolsista FAPERJ do programa Cientista do Nosso Estado. Obteve o doutorado em Computação pelo Imperial College London. Seu objetivo de pesquisa é dar suporte computacional à colaboração entre humanos. Na parte conceitual desenvolve uma Engenharia de Groupware baseado no modelo 3C de colaboração cujas dimensões são a comunicação, coordenação e cooperação. Na parte prática desenvolveu o LMS AulaNet, um ambiente de ensino e aprendizagem na web, e atualmente desenvolve tecnologia educacional em ambientes colaborativos no Second Life.



**Marcos André Fernandes Spósito** é graduado em Tecnologia em Processamento de Dados pela Universidade Tuiuti do Paraná (UTP-2000), Especialista em Gestão da Tecnologia da Informação pela Universidade Positivo (UP-2001), Especialista em Docência no Ensino Superior pela Faculdade Atual da Amazônia (2005) e Mestre em Informática com ênfase em Inteligência Artificial pela Universidade Federal do Amazonas (UFAM-2008). Atualmente é Professor efetivo do Instituto Federal de Educação, Ciência e Tecnologia de Roraima (IFRR), onde é presidente da Comissão Editorial da Revista Norte Científico, Coordenador do Curso de Pós-Graduação em Educação Profissional Integrada a Educação Básica na Modalidade de Educação de Jovens e Adultos e Pesquisador Orientador do Núcleo Tecnológico de Educação à Distância (NTEAD). Colaborador de vários grupos de pesquisa nas áreas de Tecnologia Educacional, Educação à Distância, Informática na Educação, Mídias Educacionais e Interação Humano Computador (IHC).



**Alberto Nogueira de Castro Júnior** possui graduação em Engenharia Civil pela UFAM (1986), graduação em Tecnologia Eletrônica pelo UTAM (1984), mestrado em Automação Industrial pela UFES (1993) e doutorado em Computer Science - Artificial Intelligence pela University of Edinburgh (1998). Atualmente é Professor adjunto da UFAM e membro de corpo editorial da Revista Brasileira de Informática na Educação (1414-5685). Tem experiência na área de Ciência da Computação, com ênfase em Metodologia e Técnicas da Computação, atuando principalmente nos seguintes temas: program synthesis, logic programming, intelligent interfaces, knowledge-based systems.