

# Análise dos Principais Problemas que Afetam Alunos de Programação: uma investigação empírica no Estado da Paraíba.

Túlio H. Costa<sup>1</sup>, Frederico M. Bublitz<sup>1</sup>

<sup>1</sup>Departamento de Computação – Universidade Estadual da Paraíba (UEPB)  
Campina Grande PB - Brasil

tuliohcosta@gmail.com, fredbublitz@uepb.edu.br

**Abstract.** *It is very common that Students of Computing face difficulties in programming disciplines. Due to this difficulty, the number of students that failure in such disciplines and consequently abandons the course has been raised. To contour this problem, it is necessary to investigate and analyze what are the difficulties that influence students in this process. In this article, we present the preliminary results of a field survey conducted with students of Computing in the State of Paraíba, where we proposes an initial discussion about the factors that lead students to have a low performance in the disciplines of Programming.*

**Resumo.** *Os estudantes de Computação costumam apresentar dificuldades em disciplinas de programação, o que acaba contribuindo para a elevação do grau de reprovação e evasão em disciplinas da área. Para contornar esse problema, faz-se necessário investigar e analisar as causas que influenciam os alunos no processo de ensino/aprendizagem. Neste artigo, apresentam-se os resultados preliminares de uma pesquisa de campo realizada com alunos de cursos de Computação no Estado da Paraíba, onde se propõe uma discussão inicial acerca dos fatores que levam os alunos ao baixo rendimento nas disciplinas de Programação.*

## 1. Introdução

O ensino superior, de um modo geral, tem seu foco na formação profissional e científica dos alunos. Nos cursos de Computação (e.g., Licenciatura em Computação, Ciência da Computação, Engenharia da Computação e Análise de Sistemas), as disciplinas de Programação são consideradas as bases para uma boa formação ao aluno, abrindo um leque para uma série de áreas tais como Engenharia de Software e Teoria da Computação de acordo com a *ACMC*<sup>1</sup>.

Embora componham a base de qualquer curso de Computação, um dos grandes problemas enfrentados nesses cursos recai justamente no ensino/aprendizagem das disciplinas ligadas à Programação. Grande parte dos estudantes apresenta dificuldade em entender os conceitos iniciais, aumentando, cada vez mais, o índice de evasão e reprovação nas disciplinas desta área [Prietch e Pazeto, 2010].

Existem alguns trabalhos que tentam minimizar esses problemas, em sua maioria focados no fator motivacional do aluno [Scaico *et al.* 2012, Barbosa *et al.* 2011, Souza 2009]. Segundo Rapkiewicz *et al.* [2006], um dos problemas mais significativos do ensino de Programação é o costume de decorar o conteúdo, que é agravado com a proposição constante de problemas com

---

<sup>1</sup> The ACM Computing Classification System, <http://www.acm.org/class/1998/ccs98.html>.

enunciados textuais. Zanini e Raabe [2012] relatam a forma com a qual a maior parte dos enunciados dos problemas são apresentados aos alunos, geralmente de forma objetiva e sem contextualização.

Em face ao que foi exposto, fica claro que existe uma necessidade de contornar os problemas que têm afetado o desempenho dos alunos. Para isso, deve-se antes de qualquer coisa, investigar e identificar as causas que influenciam o baixo rendimento dos mesmos em disciplinas relacionadas ao ensino de Programação. Somente dessa forma se torna possível atacar o problema em sua origem aumentando a possibilidade de reverter esse quadro.

Neste artigo, apresentam-se resultados preliminares de um experimento realizado com turmas de Programação em cursos de Computação do Estado da Paraíba. O artigo está organizado da seguinte forma: na Seção 2 abordam-se as principais hipóteses que possivelmente explicam os problemas enfrentados pelos alunos de Programação; na Seção 3 descreve-se o material e o método utilizado para a realização do experimento; na Seção 4 é feita uma análise preliminar dos dados obtidos com o experimento; e por fim, na Seção 5 são apresentadas as conclusões deste trabalho.

## **2. Qualificação do Problema**

Diante das dificuldades encontradas por alunos de Programação, faz-se necessário analisar o ponto central dessa problemática, o porquê de o aluno não estar aprendendo e que fatores estão influenciando para que isso aconteça. Mas será que o problema está relacionado apenas ao aluno? E os professores? E a instituição? O problema parece ser bem mais amplo.

Ao analisar o papel do professor, nota-se que caso ele seja o responsável pela problemática em questão, deduz-se que ele pode não conseguir transmitir o conteúdo como deveria. Isso pode ser explicado de acordo com as seguintes vertentes: (i) o professor não domina o conteúdo; (ii) o professor não possui uma formação condizente com o cargo que ocupa; (iii) o professor não tem didática; (iv) a instituição não oferece condições para o professor.

Na última vertente, pode-se eximir a totalidade da culpa do professor, que passa a encontrar obstáculos impostos pela instituição de ensino. A estrutura oferecida pela instituição, que inclui elementos como sala de aula informatizada e biblioteca, influencia no processo de desenvolvimento do aluno.

Relacionando a problemática ao próprio aluno, possivelmente pode-se explicá-la seguindo algumas hipóteses: (i) O aluno utiliza métodos inadequados de estudo; (ii) Desmotivação por parte do aluno; (iii) Recursos insuficientes para o aprendizado extraclasse; (iv) Atraso ou evasão às aulas; e (v) O aluno não consegue acompanhar o conteúdo.

## **3. Material e Método**

Com o objetivo de identificar e analisar os fatores que interferem no desempenho de alunos em disciplinas de Programação a ponto de prejudicar o processo ensino-aprendizado como um todo, propôs-se a realização de uma pesquisa de campo por meio de um experimento envolvendo alunos de turmas de Programação em cursos de Computação.

O experimento consiste em um questionário de pesquisa composto por trinta e duas questões objetivas, sendo estas divididas em quatro etapas que representam os objetivos da

pesquisa, descritas na Seção 4. Foram obtidos dados de trinta e cinco alunos de cursos de Computação os quais faziam parte de instituições sediadas no Estado da Paraíba.

A coleta de dados foi realizada no período de 2/Junho a 2/Julho de 2012, contando com a presença de alunos dos primeiros períodos letivos - que já tiveram contato com a disciplina de algoritmos - ao último do curso. A participação dos alunos dos primeiros períodos no experimento foi condicionada à época a qual o experimento foi realizado, possibilitando que aqueles que estavam ao término do semestre letivo participassem do mesmo, contando com uma experiência inicial em disciplinas de lógica de Programação e algoritmos.

## **4. Resultados**

O experimento foi classificado em quatro etapas, sendo estas descritas como: (1) Conhecer o aluno; (2) Conhecer as condições da instituição; (3) Conhecer as práticas e as formas de estudo do aluno; (4) Conhecer as dificuldades dos alunos;

### **4.1. Conhecer o Aluno**

Nesta etapa do experimento são consideradas as características da instituição em que o aluno se encontra matriculado, formas e condições de locomoção, turnos de estudo e período o qual o aluno se encontra matriculado, carga horária de vínculos empregatícios caso possua, tempo de uso semanal de um computador e o tipo de internet utilizado pelo aluno (se possuir) em sua residência.

Dentre os alunos participantes do experimento, tem-se o total de 91% residentes na cidade da instituição de ensino. Em sua totalidade, 46% dos participantes se locomovem por meio de ônibus coletivo, o que possivelmente acarreta em atrasos às aulas prejudicando-os no processo de ensino/aprendizagem, o que caracteriza a hipótese “Atraso ou evasão às aulas”.

Embora em sua maioria os participantes do experimento residam na cidade a qual a instituição de ensino se localiza, a distância entre a residência dos mesmos e a instituição foi levada em conta. O tempo médio de locomoção entre os dois pontos citados anteriormente foi questionado no experimento, o qual revela que 43% dos participantes levam de 30 minutos a 1 hora para chegar ao destino, enquanto 37% levam até 30 minutos para realizar o percurso. 20% dos alunos levam de 1 a 2 horas ou mais para completar o percurso, gerando um possível cansaço, o que também os leva à hipótese “Atraso ou evasão às aulas”.

A condição de locomoção dos participantes quanto ao cumprimento de horários também foi avaliada no experimento. Ao todo 68% afirmaram que seus meios de locomoção geralmente ou sempre são pontuais, enquanto 32% geralmente ou sempre se atrasam, o que possivelmente explica a dificuldade enfrentada por alunos de Programação novamente por meio da hipótese “Atraso ou evasão às aulas”.

Outro fator abordado no experimento foi a questão de os alunos possuírem vínculos empregatícios, os quais possivelmente podem caracterizar a hipótese “Atraso ou evasão às aulas” por proximidade de horários entre trabalho e aulas ou a hipótese “O aluno não consegue acompanhar o conteúdo” descrita na Seção 2.5 gerada pelo cansaço devido à carga horária. O experimento mostrou que 54% dos participantes possuem emprego, dentre eles, 79% possuem uma carga horária de 31 a 44 horas semanais ou superior.

### **4.2. Conhecer as Condições da Instituição**

O objetivo desta segunda etapa é conhecer as condições que a instituição a qual o aluno se encontra matriculado tem a oferecer ao mesmo. São abordadas questões sobre o auxílio aos alunos por meio de salas de aula informatizadas, e a caracterização das mesmas quanto aos tipos de computadores, acesso à Internet, presença dos *softwares* necessários às aulas e horários disponíveis para o suporte aos alunos. Ainda nesta etapa, é discutida uma questão referente à disponibilidade de livros necessários às aulas.

O primeiro item abordado correspondeu à presença de salas de aula informatizadas e suas características presentes. Os participantes em sua totalidade responderam que suas instituições apresentavam tais salas, com a presença de Internet, porém o acesso às salas era possível apenas no horário das aulas. 60% dos participantes destacaram os computadores como adequados (razoáveis ou de última geração), enquanto os outros 40% caracterizaram os computadores como defasados (lentos).

O segundo item abordado referiu-se aos computadores quanto à presença dos *softwares* necessários às aulas neles instalados. Ao todo, 55% dos participantes relataram que os computadores das salas de aula informatizadas de suas instituições não possuem ou raramente possuem os *softwares* em questão, enquanto 37% afirmaram que possuem, mas alguns sistemas funcionam inadequadamente. A menor parcela dos participantes, cerca de 8% relataram que os computadores possuem os *softwares* e funcionam adequadamente. Esses dados confirmam que as instituições precisam urgentemente de melhorias para que esse índice seja amenizado.

Por fim, no último item abordado sobre este tópico, os participantes foram questionados se os mesmos conseguem obter os livros indicados por professores nas bibliotecas da instituição. Os resultados obtidos não foram os melhores, 43% dos entrevistados afirmaram não frequentar a biblioteca de suas instituições, 23% raramente conseguem os livros pois, ou os mesmos estão ocupados ou a biblioteca não possui o livro indicado pelo professor, e apenas 34% conseguem obter os livros requisitados.

### **4.3. Conhecer as Práticas e as Formas de Estudo do Aluno**

Nesta etapa, o aluno é questionado quanto às suas práticas de estudo frente a problemas gerais e específicos de Programação. São abordadas questões sobre o tempo semanal dedicado à resolução de problemas, o cenário de estudo rotineiro utilizado pelo aluno, estratégias para resolução de problemas, verificação de correção das soluções de problemas, estratégias utilizadas para testes, formas de tirar dúvidas, e por fim, sobre a atitude do aluno ao não conseguir resolver um problema.

Os resultados do experimento mostram que 54% dos participantes resolvem problemas voltados à Programação fora do expediente de aula apenas quando há avaliação agendada, esporadicamente (menos de 1 hora), ou simplesmente não resolvem, levando-os à hipótese “O aluno utiliza métodos inadequados de estudo” descrita na Seção 2.1, enquanto 31% costumam resolver problemas entre 2 e 6 horas semanais, sobrando a pequena fatia de 15% dos participantes que superam a marca das 6 horas semanais.

Os participantes também relataram suas estratégias utilizadas para a resolução de problemas de Programação: cerca de 60% ao se depararem com um problema procuram entendê-lo para em seguida transformá-lo em um algoritmo; 20% pesquisam por exemplos prontos na Internet, em livros, etc. enquanto 14% partem diretamente para a resolução do algoritmo sem se

preocuparem em entender o problema a *priori*, o que os leva à hipótese “O aluno utiliza métodos inadequados de estudo”.

Outro fator relevante para o experimento foi checar como os participantes verificam a corretude de suas soluções. Ao todo, 69% deles a cada criação realizam a compilação do algoritmo e verificam se o resultado é o esperado, mostrando que os alunos não procuram entender o domínio do problema, validando a hipótese “O aluno utiliza métodos inadequados de estudo”, já 15% a cada criação executam o algoritmo manualmente e mentalmente antes de compilá-lo ou executa-o em modo *debug* verificando cada estado do algoritmo.

A estratégia para realização de testes foi outro fator abordado o qual apontou que 26% dos participantes não costumam realizar testes de suas soluções, destacando que os alunos não procuram ter a certeza que suas soluções estão realmente corretas, validando a hipótese “O aluno utiliza métodos inadequados de estudo”, assim como no caso de o aluno não conseguir solucionar um problema, o qual o experimento verificou que 20% procuram por soluções prontas e 14% desistem da questão.

#### **4.4. Conhecer as Dificuldades dos Alunos**

Por fim, nesta última etapa o aluno é questionado quanto ao seu conhecimento prévio, e o seu grau de atenção rotineiro. Algumas questões sobre o histórico em disciplinas relacionadas à Programação e cálculo são impostas aos alunos, que por conseguinte são questionados sobre seus níveis de raciocínio lógico de atenção e percepção nas aulas e atividades de Programação. Ao término da etapa, o aluno é indagado sobre seu interesse em resolver atividades previamente agendadas.

O experimento mostrou que 46% dos participantes perderam pelo menos uma vez alguma disciplina voltada à Programação e 34% perderam alguma disciplina voltada ao cálculo. No que tange a lógica, foi avaliada a capacidade de encontrar a solução para um problema de Programação em geral. Ao todo, 31% dos participantes relataram que normalmente conseguem visualizar uma solução para um determinado problema, mas não conseguem transformar em um algoritmo, 26% entendem o que a questão pede, mas não conseguem visualizar uma solução. Esses resultados mostram que os alunos possivelmente não possuem um conhecimento prévio necessário para obter um desempenho satisfatório em disciplinas de Programação, caracterizando a hipótese “O aluno não consegue acompanhar o conteúdo”.

Quanto ao nível de atenção dos participantes em aula, cerca de 40% não conseguem se concentrar ou conseguem por pouco tempo, algo que possivelmente pode ser explicado de acordo com uma das hipóteses: “Desmotivação por parte do aluno” ou “O aluno não consegue acompanhar o conteúdo”. Tratando do nível de percepção dos mesmos em problemas de Programação, o experimento mostra que 69% esporadicamente passam despercebidos por alguns detalhes, enquanto 11% frequentemente passam despercebidos. No geral, 80% dos participantes se consideram razoavelmente atenciosos em suas rotinas diárias, as quais as vezes se distraem ou se esquecem de algo importante.

Por fim, com o intuito de verificar a hipótese “Desmotivação por parte do aluno”, os participantes foram questionados sobre qual a atitude dos mesmos ao saberem que há uma atividade de Programação agendada. O resultado obtido revelou que: 14% frequentemente evitam fazer ou raramente fazem a atividade; 66% sempre tentam fazer, caso não consigam, procuram ajuda; 20% sempre tentam fazer, caso não consigam, desistem da atividade;

## 5. Conclusões

Neste artigo foram apresentados e discutidos alguns fatores que podem ser responsáveis pelo agravamento do índice de evasão e reprovação em disciplinas de Programação. Esses fatores foram identificados por meio de uma pesquisa de campo realizada com alunos de graduação em cursos relacionados à Computação.

A partir dos resultados obtidos com experimento tratado neste artigo, pôde-se identificar que tanto os alunos quanto os professores possuem problemas que precisam ser sanados para a melhoria do processo de ensino/aprendizagem. As instituições no que tange a estrutura a oferecer aos alunos possuem precariedades que por sua vez acabam os atingindo. Em se tratando dos alunos, diversos fatores foram abordados, porém o mais relevante foi a utilização de métodos inadequados de estudo que, segundo os dados do experimento, mostram que há uma deficiência na abordagem dos problemas de Programação.

Para que um panorama mais abrangente seja criado, é preciso ainda que sejam feitas análises estatísticas mais detalhadas, algo que já está sendo. Além disso, é preciso ampliar o número de instituições participantes, a fim de se ter uma visão mais clara da situação não só no Estado da Paraíba, mas a nível nacional. Para isso, estamos usando esse experimento também para aprimorar a forma de aplicação e análise dos dados.

## 6. Referências Bibliográficas

- Barbosa, Leônidas S. & Fernandes, Teresa C. B. & Campos, André M. C. (2011) "Takkou: Uma Ferramenta Proposta ao Ensino de Algoritmos". XXXI Congresso da Sociedade Brasileira de Computação - WEI XIX Workshop Sobre Educação em Computação. Natal, pp. 1-10.
- Prietch, S. S & Pazeto, T. A. (2010). "Estudo sobre a Evasão em um Curso de Licenciatura em Informática e Considerações para Melhorias". Anais do VIII Workshop de Educação e Informática Bahia-Alagoas- Sergipe - WEIBASE 2010, Maceió/AL.
- Rapkiewicz, C. E. & Falkemback, G. & Seixas, L. & Santana dos Santos, N. & Cunha, V. V. & Klemann, M. (2006) "Estratégias Pedagógicas no Ensino de Algoritmos e Programação Associadas ao Uso de Jogos Educacionais". RENOTE - Revista Novas Tecnologias na Educação, Vol. 4(2), pp. 1-11.
- Scaico, Pasqueline & Marques, Diego L. & Melo, Leandro de A. & Azevedo, Max A. & Mendes Neto, Sinval V. & Oliveira, Anderson & Alves Júnior, Josinaldo & Labanca, Marcelo & Scaico, Alexandre. (2012) "Um jogo para o ensino de programação em Python baseado na taxonomia de Bloom". In: XX Workshop sobre Educação em Computação - CSBC 2012, Curitiba.
- Souza, Cláudio M. (2009) "VisuAlg - Ferramenta de Apoio ao Ensino de Programação". Revista TECCEN Vol. 2(2), ISSN 1984-0993.
- Zanini, Adriana S. & Raabe, André L. A. (2012). "Análise dos enunciados utilizados nos problemas de programação introdutória em cursos de Ciência da Computação no Brasil". In: XX Workshop sobre Educação em Computação - CSBC 2012, Curitiba.