

# Modelo para Comparar Desempenho e Percepção de Dificuldades de Aprendizagem de Alunos em Cursos de Introdução à Programação

Daniel Calife, Ricardo Nakamura

Escola Politécnica da Universidade de São Paulo (USP)  
São Paulo – SP – Brasil

calife@gmail.com, ricardonakamura@usp.br

**Abstract.** *Introduction to Programming courses are mandatory in computer science and related areas curricula and are increasingly being offered in several other areas. In these courses the difficulties faced by students and their fails rates are always high. Many studies attack these problems without trying to understand or help the perception of students in relation to these difficulties. This paper proposes a model to compare the difference between the performance of students and their perception of their difficulties. With this model it is possible to detect this gap, allowing the professor to take actions in order to improve student performance.*

**Resumo.** *Cursos de Introdução à Programação são oferecidos de forma obrigatória em currículos relacionados à computação e estão sendo cada vez mais oferecidos em diversas outras áreas. Nestes cursos as dificuldades enfrentadas pelos alunos e as taxas de reprovação são sempre elevadas. Muitos estudos atacam estas dificuldades sem procurar entender ou auxiliar a percepção dos alunos em relação à estas dificuldades. Este trabalho propõe um modelo para comparar a diferença entre o desempenho dos alunos e sua percepção de dificuldades. Com esse modelo é possível detectar esta disparidade, permitindo ao professor tomar ações para que o desempenho dos alunos seja aprimorado.*

## 1. Introdução

Em cursos superiores na área de Ciência da Computação, o ensino de programação é obrigatório e é abordado em diversas disciplinas por quase toda a extensão do curso. Em cursos de outras áreas, como Engenharia, Matemática, Design etc, que não tem o foco principal em computação, estão sendo cada vez mais oferecidas disciplinas introdutórias à programação, dado o contexto da tecnologia sempre presente e auxiliando estas áreas.

Independente da grande área do curso, as dificuldades dos alunos e a taxa de reprovação em cursos de programação sempre são elevadas, o que leva a vários pesquisadores tentarem encontrar quais são as principais dificuldades dos alunos em aprender, e, as dos professores em ensinar os conceitos fundamentais de programação [Lahtinen, Ala-Mutka e Järvinen, 2005]. Estes estudos têm uma grande variação de abordagens, que podem abranger desde análises bastante específicas, como a análise de padrões de codificação [Fujiwara et al, 2012], discussões pedagógicas do ponto de vista

dos professores [Gomes, Mendes, 2014], e também, análises mais amplas como o planejamento e arquitetura do curso de programação [Pears et al, 2014] e as causas que levaram ao sucesso ou fracasso do ponto de vista dos alunos [Hawi, 2010].

Estas pesquisas que apresentam as dificuldades no ensino e aprendizagem de programação normalmente levam em conta a percepção dos alunos em relação às suas dificuldades em tópicos específicos, como em estruturas de decisão e métodos, e também a percepção destas dificuldades, dos alunos, pelos professores com experiência no ensino de programação. Embora muitas das percepções das dificuldades coincidam, os professores normalmente afirmam que os alunos são otimistas em relação as dificuldades que eles enfrentam, e alegam que os alunos têm maiores dificuldades do que eles percebem [Milne e Rowe, 2002] [Sarpong, Arthur, e Amoako, 2013] [Piteira e Costa, 2013]. Assim, estas pesquisas seguem o caminho tendo essa diferença, entre a percepção do aluno e a real dificuldade, como certa, sem se dedicar à entender melhor porque ela existe, se realmente existe, e como identificar estas discrepâncias precocemente para direcionar melhor os alunos.

Com a crescente adoção de ambientes de ensino *online* e de cursos com cada vez mais recursos *online* e digitais é possível levantar dados de desempenho dos alunos através de técnicas de extração e de análise de dados educacionais. No contexto deste trabalho, a definição de dados analíticos educacionais foi adotada conforme Bienkowski, Feng e Means (2012) como *Learning Analytics*, que são técnicas para construir diversos modelos para auxiliar o ensino *online*. Como por exemplo, a modelagem do usuário, indicando o que ele sabe, o que o motiva, qual seu comportamento, quão satisfeito ele está etc, à partir de dados sobre sua interação com sistemas computacionais.

Desta forma, o objetivo deste trabalho é propor um modelo que possa avaliar e comparar o desempenho dos alunos de graduação em uma disciplina introdutória de programação com a percepção das dificuldades que ele está enfrentando. Este modelo também pode ser utilizado como base para correlecionar o desempenho inicial dos alunos com a sua possível situação final de sucesso ou falha no curso.

Para isso, dados sobre o desempenho em exercícios, avaliações e percepção dos alunos serão extraídos e analisados, conforme metodologia descrita na seção 3 deste artigo.

## **2. O Curso de Introdução à Programação**

Esta pesquisa foi utilizada tendo como objeto os alunos da disciplina de Introdução à Programação de Jogos que faz parte do Curso de Design Games da Universidade Anhembi Morumbi, oferecida no 1º semestre de 2016. Curso que tem como principal área o Design, portanto a maioria dos alunos e disciplinas não tem um perfil voltado para exatas ou programação, e esta disciplina é o primeiro contato que eles têm com o tema. A disciplina apresenta as técnicas e conceitos fundamentais programação através da construção de algoritmos, da programação estruturada, a definição de métodos/funções, bem como a utilização e armazenamento de dados em memória através de variáveis e estruturas homogêneas como vetores e matrizes.

A disciplina, que tem oferecimento semestral, é dividida em 20 semanas com 4 horas/aula presenciais cada, totalizando 80 horas/aula. As atividades presenciais são

todas realizadas em ambiente de laboratório, permitindo assim a apresentação de conceitos e na sequência a aplicação prática e exercícios sobre o tema. Como ferramenta de gerenciamento e de apoio às aulas é utilizado o LMS Blackboard [Blackboard, 2016].

A ferramenta para a aplicação prática da programação escolhida para o curso foi o Processing [Processing, 2016], que possui uma linguagem baseada em Java e permite codificar programas interativos e visuais de forma bastante rápida.

Os alunos são avaliados de duas formas, a primeira, com aspecto mais contínuo é composta por avaliações parciais, com conteúdos específicos, exercícios realizados em sala e em casa e listas ou pequenos projetos de implementação. A segunda avaliação é uma prova presencial que abrange todo o conteúdo da disciplina. A metodologia utilizada neste artigo utilizou as primeiras avaliações parciais e exercícios aplicados aos alunos.

### **3. Metodologia Empregada**

Para realizar a extração dos dados utilizados foram realizados quatro experimentos: os dois primeiros para identificar o desempenho dos alunos em tópicos específicos na disciplina de programação, o terceiro para o aluno classificar a sua percepção de dificuldade nestes mesmos tópicos e o quarto, uma avaliação parcial que cobre o conteúdo visto até aquele momento.

Estes experimentos foram aplicados à 25 alunos de graduação em Design de Games, participando da disciplina de Introdução à Programação de Jogos, conforme descrita na seção 2.

A descrição mais detalhada dos experimentos realizados estão à seguir:

Experimento 1: Os dados levantados à partir de questionários com questões objetivas sobre o tópico apresentado em sala de aula. Estes questionários são formados por 7 questões de múltipla escolha e a nota é determinada pelo número de acertos que o aluno obteve. Os alunos tem um prazo de até 7 dias para responder estas questões e poderão utilizar quaisquer recursos para tal. Estes questionários são disponibilizados online pelo LMS utilizado, e podem ser respondidos, somente uma vez, à qualquer momento, desde que respeitado o prazo limite.

Experimento 2: Uma pequena avaliação presencial com duas questões objetivas de múltipla escolha e uma dissertativa onde o aluno deve desenvolver um pequeno código à partir de um problema dado. Os alunos poderão utilizar os recursos disponíveis na aula, menos a utilização de compiladores, como o do Processing, para testar as respostas ou códigos.

Experimento 3: Os alunos respondem à um questionário no estilo likert com uma escala de 1 à 5 (1 – discordo totalmente, 5- concordo totalmente) identificando se aprendeu um determinado tópico.

Experimento 4: Avaliação parcial presencial sobre os tópicos apresentados até o momento, sendo composta de seis questões objetivas de múltipla escolha e quatro questões dissertativas onde o aluno deve desenvolver um pequeno código, totalizando assim dez questões e acertos possíveis. Nesta avaliação não foi permitido nenhum tipo de consulta ou utilização de quaisquer outros recursos.

A separação dos tópicos abordados em sala de aula e também nos experimentos foi realizada da seguinte forma:

- Exercícios de Lógica e Resolução de Problemas;
- Tipos de Variáveis, Operadores e Expressões;
- Estruturas de Seleção (*if/else*, *switch*);
- Modularização (métodos, parâmetros e retornos);

Após a extração dos dados foi realizada uma limpeza nos dados eliminando alunos que não tenham realizado pelo menos 50% dos questionários propostos nos experimentos 1 e 2 ou que não tenham realizado a avaliação parcial presencial ou a avaliação da percepção de dificuldades. Como resultado desta limpeza 8 alunos foram removidos do experimento, destes, 5 alunos por motivo de desistência do curso e 3 alunos por não estarem regularmente matriculados desde o princípio da disciplina.

### 3.1 Variáveis aleatórias

Após a extração e limpeza dados, estes foram organizados de forma a facilitar a criação do modelo proposto neste trabalho. Assim as variáveis aleatórias foram definidas da seguinte forma, neste contexto:

**Questionários** - soma de todos os questionários, dos quatro tópicos, aplicados nos experimentos 1 e 2, totalizando assim 40 questões, e o mesmo número de acertos possíveis.

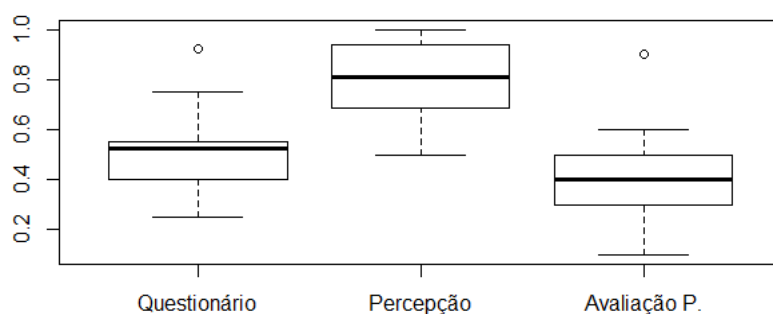
**Percepção** - soma das avaliações das percepções de dificuldades do experimento 3, nos quatro tópicos apresentados, formando uma escala *likert* com variação de 4 até 20. Consideramos que quanto é o valor da percepção menor é a dificuldade sentida pelos alunos.

**Avaliação Parcial** - número de acertos, 10 no máximo, conseguidos pelo aluno no experimento 4.

Estas três variáveis aleatórias foram normalizadas em uma escala entre 0 e 1 para facilitar as comparações entre seus valores. À partir da análise destes dados espera-se responder com objetividade à hipótese de que a percepção das dificuldades enfrentadas pelos alunos no aprendizado de programação é diferente do seu desempenho, ou de suas dificuldades reais, como descrito nas pesquisas levantadas. Desta forma a hipótese nula afirma que as médias dos Questionários e da Percepção são iguais, a hipótese alternativa afirma que são diferentes.

## 4. Análise e Discussão dos Resultados

Nesta seção serão apresentados e discutidos os resultados da aplicação da metodologia proposta na seção 3. O gráfico da figura 1 apresenta a comparação dos diagramas de caixa das variáveis aleatórias levantadas.

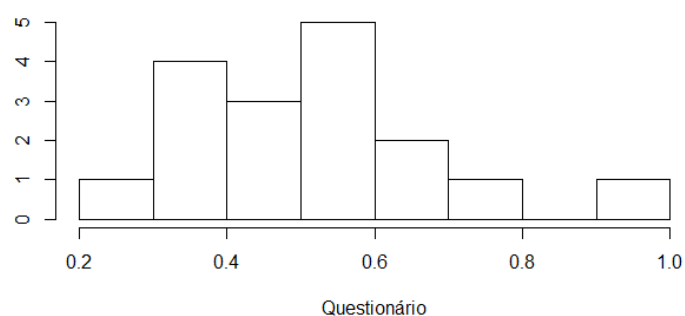


**Figura 1 – Diagrama de Caixas**

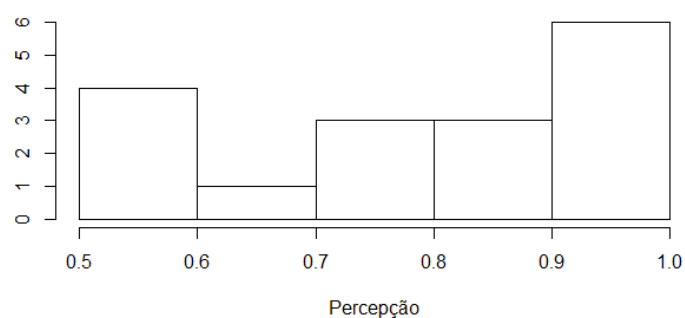
Na figura 1 podemos analisar que o valor que representa a mediana da percepção de dificuldades dos alunos encontra-se acima das demais medianas, assim como a discrepância entre os outros quartis representados no gráfico.

A figura 2 demonstra os histogramas das três variáveis avaliadas:

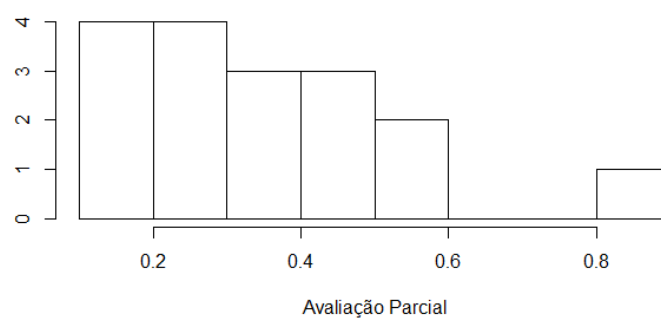
**Histograma de Questionário**



**Histograma de Percepção**



**Histograma de Avaliação P.**

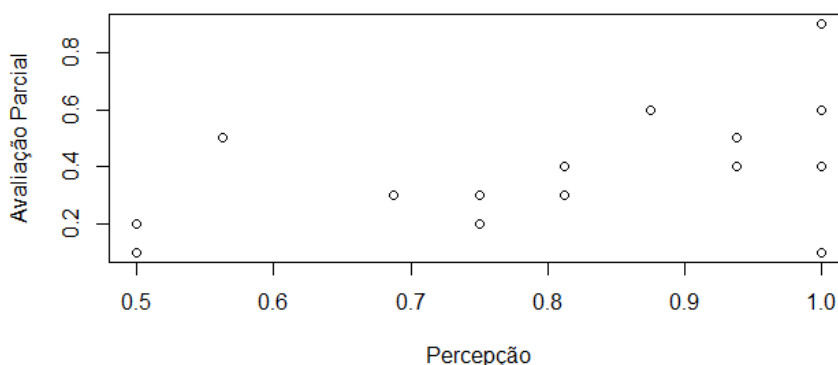


**Figura 2 – Histogramas de Questionários, Percepção e Avaliação Parcial.**

A principal característica à ser notada nos histogramas acima é que, embora, estejam em uma escala de 0 à 1 eles apresentam valores discretos e não descrevem uma distribuição com características de Normal, que poderia facilitar sua análise e comparação entre médias.

#### 4.1 Correlação entre as Variáveis

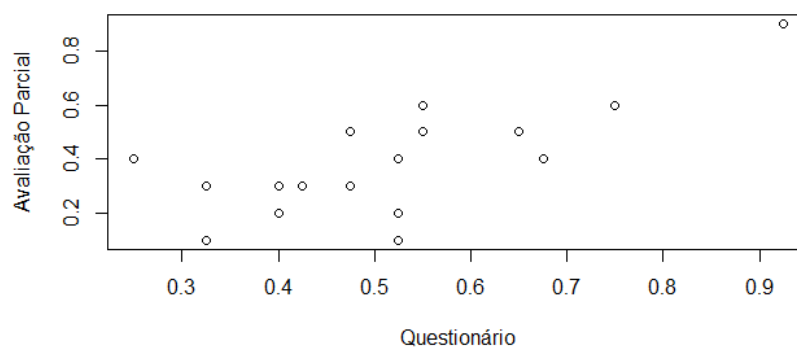
Na figura 3, à seguir, é mostrado a relação entre os valores da Avaliação Parcial e a Percepção de Dificuldades dos alunos.



**Figura 3 –Avaliação Parcial em relação à Percepção de Dificuldades.**

No gráfico da figura 3 podemos analisar que os valores da Avaliação Parcial estão relacionados à valores mais altos de Percepção dos alunos. A correlação, fraca, entre estas duas variáveis é dada aproximadamente por  $\rho \cong 0,41$ .

A figura 4 apresenta o gráfico com a relação entre os valores da Avaliação Parcial e o Questionário realizado pelos alunos.



**Figura 4 –Avaliação Parcial em relação ao Questionário.**

Os valores entre a Avaliação Parcial e o Questionário realizado pelos alunos apresentam uma proximidade maior entre seus valores, conforme demonstrado na figura 4, assim a correlação entre estas duas variáveis tem um aspecto mais forte, sendo dada aproximadamente por  $\rho \cong 0,72$ .

### 4.3 Modelo para a Comparação entre as Médias

Como foi demonstrado anteriormente as distribuições das variáveis aleatórias que estamos analisando neste trabalho não seguem o modelo de distribuição Normal. Nas três distribuições os valores são discretos e representam a quantidade de acertos em questões objetivas ou então uma escala de percepção de dificuldade dos alunos. Assim, a distribuição Binomial poderia representar este modelo, uma vez que ela representa o número de sucessos, dado uma sequência  $n$  de experimentos, com uma probabilidade  $p$  de sucesso,  $Binomial(n, p)$ .

Considerando a quantidade de acertos  $x$  do aluno temos o seguinte modelo de probabilidade:

$$Pr(x|n, p) = \binom{n}{x} p^x (1 - p)^{n-x}$$

Como a probabilidade de sucesso pode variar de acordo com o aluno e a própria questão, o parâmetro  $p$  foi modelado de acordo com uma distribuição  $Beta(\alpha, \beta)$ . Chegando assim à uma distribuição Beta-Binomial, com o seguinte modelo:

$$Pr(x|n, \alpha, \beta) = \binom{n}{x} \frac{B(x + \alpha, n - x + \beta)}{B(\alpha, \beta)}$$

Para encontrarmos o intervalo de confiança da média de cada uma das variáveis aleatórias, precisamos encontrar os parâmetros  $\alpha$  e  $\beta$  do modelo apresentado, assim a Verossimilhança, dada por  $L$ , do modelo apresentado é descrita por:

$$L(x, n, \alpha, \beta) = \prod_{i=1}^m \binom{n}{x_i} \frac{B(x_i + \alpha, n - x_i + \beta)}{B(\alpha, \beta)}$$

Onde  $n$  é o número de questões,  $m$  é o número de alunos e  $x$  é a quantidade de sucessos.

Seguindo o modelo *Bayesiano* foi utilizado o estimador Máxima Verossimilhança. Para estimarmos os parâmetros  $\alpha$  e  $\beta$ , definidos por  $\theta$ , para cada uma das variáveis aleatórias, foi utilizado o algoritmo de otimização da subida da montanha, definido por:

$$\theta_{t+1} = \theta_t + \alpha_t \nabla_{\theta}(\theta_t)$$

O algoritmo atualiza os parâmetros desejados utilizando o gradiente definido previamente. Desta forma a tabela 1 lista os parâmetros  $\theta$  estimados para cada distribuição, antes da normalização de seus valores:

**Tabela 1, Parâmetros de  $\alpha$  e  $\beta$  estimados.**

	$\alpha$	$\beta$
Questionário	5,36	4,98
Percepção	4,15	1,07
Avaliação Parcial	5,34	8,37

Utilizando os parâmetros estimados na tabela 1 foram realizadas 100.000 simulações da distribuição para cada variável aleatória e calculada a média normalizada de cada uma delas. Utilizando a confiança determinada por 95%, os intervalos de confiança da média para cada uma está descrita na tabela 2 à seguir:

**Tabela 2, Intervalo das Médias Normalizadas.**

	<i>Intervalo de Confiança da Média</i>
Questionário	0,43 à 0,59
Percepção	0,70 à 0,88
Avaliação Parcial	0,29 à 0,47

#### 4.4 Testes de Hipóteses

Conforme descrito anteriormente neste artigo algumas pesquisas indicam que a percepção dos alunos em relação à suas dificuldades reais no aprendizado de programação é menor do que a dificuldade observada à partir de seu desempenho em avaliações de programação.

De acordo com a metodologia empregada, os valores da variável aleatória que representa a percepção da dificuldade dos alunos representa a percepção de facilidade ou de assimilação de conteúdo, portanto deveria refletir diretamente em seu desempenho nos questionários e nas avaliações.

Assim, neste primeiro momento, iremos testar a hipótese de que a média da percepção da dificuldade dos alunos é maior que a média do resultado obtido nos questionários. Desta forma definimos as seguintes hipóteses:

$$H_0 : \bar{x}_{questionarios} \geq \bar{x}_{percepcao}$$

$$H_A : \bar{x}_{percepcao} > \bar{x}_{questionarios}$$

Utilizando os intervalos de confiança apresentados previamente na tabela 2, podemos rejeitar a  $H_0$  deste primeiro teste.

Alternativamente podemos testar mais uma hipótese de que a média da percepção da dificuldades dos alunos é maior que a média obtida na avaliação parcial. Portanto, consideremos as seguintes hipóteses:

$$H_0 : \bar{x}_{avaliacao} \geq \bar{x}_{percepcao}$$

$$H_A : \bar{x}_{percepcao} > \bar{x}_{avaliacao}$$

Neste segundo teste de hipótese a  $H_0$  também pôde ser rejeitada de acordo com o intervalo das médias presentes na tabela 2.

Após a execução destes testes de hipóteses nota-se claramente que existe uma forte tendência de a percepção das dificuldades sofridas pelos alunos que estão iniciando seus estudos em programação serem otimistas, apresentados valores maiores que as notas recebidas nos questionários e nas avaliações.



## 5. Conclusão

Neste artigo foi apresentada a metodologia e modelos para comparar a dificuldade percebidas pelos alunos com o seu desempenho na disciplina de introdução à programação. Através deste modelo foi possível corroborar com a afirmação de que a percepção dos alunos tende a ser otimista, mesmo quando os alunos receberam as notas e correções de exercícios propostos.

A metodologia empregada pode ser generalizada e utilizada em outros contextos, onde é possível avaliar o desempenho dos alunos através de questionários com perguntas objetivas.

Em casos de aplicação deste modelo para cursos com conteúdo e avaliação totalmente *online* é necessário validar a aplicação dos experimentos 2 e 4, que neste trabalho foram realizados de forma presencial.

Este modelo pode ser utilizado para avaliar precocemente esta discrepância e ajudar os professores a tomarem ações visando a melhoria de desempenho de seus alunos. Um dos possíveis trabalhos futuros na área de *Learning Analytics* é o auxílio em tempo real para que os alunos melhorem sua percepção das dificuldades, e procurem corrigir e reforçar estes tópicos para melhorar seu desempenho durante as avaliações do curso [Duval, 2011].

Dada as correlações observadas das variáveis aleatórias, onde foi verificada uma forte correlação entre as notas do questionário com a nota da avaliação, é possível refinar um modelo de regressão que permita prever o desempenho dos alunos a partir destes questionários iniciais.

## Referências

- Bienkowski, M., Feng, M., Means, B. (2012) "Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics: an Issue Brief". U.S. Department of Education.
- Blackboard Brasil website. (2016) <http://blackboard.grupoa.com.br/>, Maio.
- Duval, E. (2011) "Attention please!: learning analytics for visualization and recommendation". In Proceedings of the 1st International Conference on Learning Analytics and Knowledge (LAK '11). ACM, New York, NY, USA, 9-17
- Gomes A. e Mendes, A. (2014) "A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations," 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, Madrid, pp. 1-8.
- Hawi, N. (2010) "Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course". Computers & Education. 54, 4, 1127-1136.
- K. Fujiwara, K., Fushida, K., Tamada, H., Igaki, H. e Yoshida, N. (2012) "Why Novice Programmers Fall into a Pitfall?: Coding Pattern Analysis in Programming Exercise," Empirical Software Engineering in Practice (IWESep), 2012 Fourth International Workshop on, Osaka, pp. 46-51.

- Lahtinen, E., Ala-Mutka, K. e Järvinen, H. (2005) “A study of the difficulties of novice programmers”. SIGCSE Bull. 37, 3 , 14-18.
- Milne, I. e Rowe, G. (2002) “Difficulties in Learning and Teaching Programming: Views of Students and Tutors”. Education and Information Technologies 7, 1 , 55-66.
- Pears, A., Seidman, s., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M. e Paterson, J. (2007) “A survey of literature on the teaching of introductory programming.” In Working group reports on ITiCSE on Innovation and technology in computer science education (ITiCSE-WGR '07), Janet Carter and June Amillo (Eds.). ACM, New York, NY, USA, 204-223.
- Piteira, M. e Costa, C. (2013) “Learning computer programming: study of difficulties in learning programming”. In Proceedings of the 2013 International Conference on Information Systems and Design of Communication (ISDOC '13). ACM, New York, NY, USA, 75-80.
- Processing.org. (2016) <https://processing.org/>, Maio.