

# Pensamento Computacional e Educação Matemática: Relações para o Ensino de Computação na Educação Básica

Thiago Schumacher Barcelos<sup>1,2</sup> e Ismar Frango Silveira<sup>2</sup>

<sup>1</sup> Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP  
Av. Salgado Filho, 3501 – 07115-000 – Guarulhos – SP

<sup>2</sup> Universidade Cruzeiro do Sul  
Av. Dr. Ussiel Cirilo, 225 – 08060-070 – São Paulo – SP

tsbarcelos@ifsp.edu.br; ismar.silveira@cruzeirodosul.edu.br

**Abstract.** *The pervasiveness of computer devices in everyday situations poses a fundamental question about Computer Sciences as being part of those known as basic sciences. By doing this, it would be straightforward to consider it to be taught in initial series – not exactly the computation as a technique, but instead as a way of organizing thinking and problem solving. Under this perspective, there are inherent relationships among the knowledge, skills and attitudes that emanate from this field and those ones commonly related to Math. This paper aims to discuss this relationship by presenting a mapping between skills defined in the Brazilian Curriculum Guidelines and activities aimed at developing Computation Thinking skills described in the literature. By doing this, some challenges for education in Computer Sciences are identified.*

**Resumo.** *A pervasividade dos dispositivos computacionais no cotidiano traz à luz uma questão fundamental sobre a Ciência da Computação como parte das ciências básicas. Ao fazê-lo, é natural considerar o ensino de Computação em séries fundamentais – não como técnica, e sim como uma forma de organização de pensamento e de resolução de problemas. Sob tal perspectiva, há relações inerentes entre o conhecimento, habilidades e atitudes advindas deste campo e aquelas comumente relacionadas à Matemática. Este artigo discute essas relações a partir de um mapeamento das competências previstas nos padrões curriculares brasileiros com atividades que desenvolvem o Pensamento Computacional descritas na literatura. Desta maneira, alguns desafios para a educação em Computação são apontados.*

## 1. Introdução

Atualmente, a formação de profissionais para as áreas de Computação e Tecnologia da Informação (TI) apresenta sérios desafios. No Brasil, já há alguns anos verifica-se um déficit na formação de profissionais. Em um recente estudo da Associação Brasileira de Empresas de Tecnologia de Informação e Comunicação (BRASSCOM) estima-se que, em 2014, a demanda por profissionais de TI nos oito principais mercados (São Paulo, Rio de Janeiro, Distrito Federal, Paraná, Minas Gerais, Bahia, Pernambuco e Rio Grande do Sul) chegará a 78 mil profissionais, mas apenas 33 mil concluirão os cursos superiores da área.

Soma-se a este quadro a diminuição na procura por cursos superiores em Computação e TI e o alto índice de evasão nesses cursos, que seria de aproximadamente 87%, sem considerar ainda a ociosidade de vagas nas universidades e faculdades brasileiras (BRASSCOM, 2011). Porém, esse quadro crítico não é exclusividade do mercado brasileiro. Muratet *et al.* (2009) afirmam que no período de 2005 a 2009 houve uma redução de cerca de 20% no número de estudantes de Ciência da Computação na França; Crenshaw *et al.* (2008) reportam queda de 50% no interesse de estudantes americanos pela área ao longo dos anos 2000.

A falta do domínio adequado de conhecimentos matemáticos pelos alunos é um possível fator explicativo para a falta de interesse e evasão em cursos da área. Uma revisão da literatura permite apontar que diversos pesquisadores (CAMPBELL; MCCABE, 1984; WILSON; SHROCK, 2001; BEAUBOUF, 2002; SETTI, 2009) indicam possíveis correlações entre o conhecimento matemático prévio dos alunos e seu desempenho em cursos introdutórios de Computação, bem como a relevância de tópicos do conhecimento matemático para uma melhor compreensão e modelagem dos processos computacionais. Coincidentemente ou não, constata-se um baixo rendimento na aprendizagem de Matemática pelos alunos no ensino básico. É possível citar como exemplo os resultados do último SARESP (Sistema de Avaliação do Rendimento Escolar do Estado de São Paulo), que indicam que 58,4% dos egressos do ensino médio tem desempenho considerado abaixo do básico em Matemática (TARGINO, 2012). As dificuldades com o ensino de Matemática se repetem em outros países; no Reino Unido, a RSA (*Royal Society of Arts*) recentemente identificou que algumas universidades chegam ao ponto de não divulgar os pré-requisitos de Matemática necessários para a matrícula em disciplinas, temendo com isso afugentar os alunos (PAETON, 2012).

Por outro lado, discute-se também o aumento da abrangência do ensino de Computação, passando-se a considerá-la como uma ciência básica. Dessa forma, um subconjunto de competências e habilidades básicas relacionadas à área deveria ser desenvolvido pelos estudantes desde o ensino básico. Segundo Wing (2006), esse subconjunto de competências e habilidades pode ser denominado como *pensamento computacional*, envolvendo as competências relacionadas à abstração e decomposição de problemas de forma a permitir sua resolução usando recursos computacionais e estratégias algorítmicas.

Ao considerar a incorporação do pensamento computacional ao ensino básico, algumas questões se abrem para discussão. Poderia essa nova competência dar suporte a um aprendizado mais eficaz de áreas “críticas” como a Matemática, isto é, seria possível a transferência de competências entre domínios diferentes? Uma formação integrando pensamento computacional e Matemática eventualmente atrairia mais talentos para a área de Computação? Neste sentido, o presente artigo visa apresentar e discutir algumas tendências de pesquisa relacionadas ao ensino de Computação como ciência básica, discutindo suas possíveis relações com a Educação Matemática, em articulação com trabalhos da literatura. Nossa discussão procura sistematizar alguns caminhos para que as áreas possam se beneficiar mutuamente com o desenvolvimento de estratégias pedagógicas conjuntas.

O artigo está estruturado da seguinte maneira: na seção 2, discutimos a natureza da Computação enquanto ciência, suas relações com a Matemática e as definições de pensamento computacional apresentadas na literatura. Na seção 3, identificamos pontos

de intersecção entre as competências do pensamento computacional e as competências definidas para o ensino de Matemática nos Parâmetros Curriculares Nacionais para o Ensino Médio. Na seção 4, apresentamos possíveis caminhos de pesquisa envolvendo os temas tratados e na seção 5 trazemos algumas considerações finais.

## 2. A natureza da Computação e do pensamento computacional

O início da história da Computação moderna (isto é, dos avanços tecnológicos da Computação ao longo dos séculos XX e XXI) tem uma relação muito próxima com a viabilidade da realização de cálculos numéricos. O clássico artigo de 1936 de Alan Turing, “*On computable numbers, with an application to the Entscheidungsproblem*”, que define as bases estruturais da Computação como hoje é conhecida, é essencialmente uma prova matemática a respeito da viabilidade da realização de cálculos numéricos a partir de uma máquina conceitual, posteriormente denominada Máquina Universal de Turing. Porém, para viabilizar a construção de máquinas de computação eletrônica e o posterior desenvolvimento de linguagens de programação e compiladores, outras áreas do conhecimento foram incorporadas à Computação de maneira a estruturar o seu corpo de conhecimento (*Body of Knowledge*). Denning (2005) discute que, ao longo do seu desenvolvimento e maturação, as atividades da Computação buscam suporte nas ciências ditas naturais, na Engenharia e na Matemática. O método científico, baseado no princípio da experimentação, é utilizado no desenvolvimento de algoritmos heurísticos ou na definição de modelos para o processo de desenvolvimento de software. Já o projeto e desenvolvimento de software se constituem claramente como atividades de Engenharia, ou seja, da aplicação da técnica. Por sua vez a Matemática, com sua representação simbólica e sistema de dedução fundamentado em axiomas, é base para o estudo da complexidade de algoritmos e da análise numérica.

Apesar de buscar seus fundamentos em outras áreas, a Computação parece trazer mecanismos singulares de raciocínio para resolução de problemas, cujas aplicações ultrapassam as fronteiras da Computação em si – considere, por exemplo, as recentes aplicações de métodos computacionais a questões de pesquisa da Biologia e das Ciências Sociais, permitindo a análise de uma quantidade de dados muito superior ao que seria possível antes de atacar tais questões sob um ponto de vista computacional.

Como outras áreas do conhecimento podem se beneficiar de parte das competências específicas da Computação, torna-se conveniente então definir quais seriam tais competências. Assim, trazemos aqui algumas definições encontradas na literatura. Como mencionado anteriormente, Wing (2006) propõe agrupar essas competências específicas sob o nome de *pensamento computacional*<sup>1</sup>, definido através das seguintes características:

- *Conceituar ao invés de programar.* Resolver um problema aplicando o pensamento computacional significa reduzir problemas grandes e aparentemente insolúveis em problemas menores e mais simples de resolver. Isso exige a capacidade de pensar de forma abstrata e em múltiplos níveis, e não a mera aplicação de técnicas de programação;

---

<sup>1</sup> Nossa tradução para *computational thinking*

- *É uma habilidade fundamental e não utilitária.* O pensamento computacional não é uma habilidade mecânica ou utilitária, mas algo que permite a resolução de problemas diversos utilizando um recurso ubíquo na sociedade atual – os computadores – e por isso deveria ser desenvolvido por todos os estudantes;
- *É a maneira na qual pessoas pensam, e não os computadores.* A resolução de problemas através do pensamento computacional é um tratamento específico do problema de forma que ele possa ser resolvido por computadores, e não uma redução do raciocínio para simular o processamento do computador;
- *Complementa e combina a Matemática e a Engenharia.* A definição de Wing considera o aporte da Matemática e da Engenharia para a Computação, conforme mencionamos anteriormente, e reconhece as particularidades trazidas pelo enfoque computacional;
- *Gera ideias e não artefatos.* O pensamento computacional não deve ter necessariamente como resultado final a produção de software e hardware e reconhece que os conceitos fundamentais da Computação estarão presentes para resolver problemas em vários contextos do cotidiano;
- *Para todos, em qualquer lugar.* Por fim, o pensamento computacional pode ser útil para todas as pessoas, em diversas aplicações.

A mesma autora ainda argumenta que introduzir o pensamento computacional no ensino básico também significa fomentar o interesse pela área de Computação, mostrando sua versatilidade e relevância na resolução de problemas do mundo atual. Evidentemente, nem todos os alunos egressos do ensino básico farão uma graduação em Computação, mas essa estratégia educacional poderia criar e intensificar relações interdisciplinares da Computação com outras áreas. Podemos especular se, inclusive, teríamos melhores cientistas da computação no futuro.

O *CSTA K-12 Computer Science Standards* (CSTA, 2011) ressalta a necessidade de desenvolvimento de habilidades computacionais na Educação Básica, auxiliando no desenvolvimento de capacidades de resolução de problemas, dando suporte e se relacionando com outras ciências, além de atuar como um elemento motivador para os estudantes. Este currículo de referência sugere a formação em Computação de alunos da Educação Básica de acordo com uma abordagem que contempla conceitos, habilidades e competências, conforme abordado também por Bezerra e Silveira (2011).

Isbell *et al.* (2010) consolidaram os resultados de um debate sobre a natureza da Computação, com o objetivo de orientar o currículo de cursos de graduação. Desse debate resulta uma visão alternativa sobre o pensamento computacional. Na visão dos autores, o conceito de computação está centrado na produção de *modelos* representativos de um determinado domínio, modelos esses que devem ser expressos em uma determinada linguagem e que podem ser manipulados, interpretados e modificados por uma máquina. Nesse caso os conceitos chave que permeiam o pensamento computacional<sup>2</sup> são: modelos; abstração; interpretação (de dados ou padrões); escala e limites para o processamento; simulação de sistemas a partir de modelos e possibilidade

---

<sup>2</sup> Isbell *et al* optam por usar o termo *computationalist thinking* (“pensamento *computacionalista*”, em uma tradução livre) para ressaltar que a ênfase do conceito argumentado está na forma de pensar “de quem faz computação”. Do original: “(...) the mindset or way of thinking of computationalists (that is, those who do computing).”

de automação dos modelos. Podemos constatar algumas similaridades com a definição de Wing, especialmente nos conceitos de abstração e limites. No entanto, a ideia da construção de modelos permite uma aproximação em específico com a Matemática que será retomada mais adiante neste artigo.

Por sua vez, o trabalho de Basawapatna *et al.* (2011) define pensamento computacional baseando-se na identificação de padrões no funcionamento de agentes de software em jogos digitais. Os agentes são programados por estudantes em atividades didáticas, usando uma linguagem de programação visual e baseada na construção de blocos. Trata-se de uma proposta bem mais específica, porém seu objetivo é identificar a possível transferência, pelos alunos, do conhecimento dos padrões para o domínio da simulação de fenômenos físicos e biológicos. Dentre os padrões identificados, temos: geração, colisão, transporte, difusão e escalada (*hill climbing*). A questão da transferência de conhecimento é fundamental na discussão das relações entre o pensamento computacional e a Matemática, e este é o foco da próxima seção.

### **3. Relações entre Pensamento Computacional e Matemática no contexto dos PCNs**

A influência da tecnologia na sociedade e a conseqüente necessidade de contextualizá-la no ensino básico são observações presentes nas diretrizes dos Parâmetros Curriculares Nacionais para o ensino de Matemática: “O impacto da tecnologia na vida de cada indivíduo vai exigir competências que estão além do simples lidar com as máquinas” (PCN, 1999, p. 41). Coincidentemente, esse é o mesmo argumento utilizado para diferenciar as competências do pensamento computacional do letramento computacional, ou seja, a habilidade de (apenas) operar adequadamente os computadores. Ainda segundo os PCNs, o impacto da tecnologia exige um redirecionamento do ensino de Matemática, a partir do qual “o indivíduo possa se reconhecer e se orientar nesse mundo do conhecimento em constante movimento” (PCN, 1999, p. 41).

Entretanto, a articulação entre o pensamento computacional e a Matemática exige uma clara identificação dos momentos em que essa relação pode ocorrer ao longo do currículo escolar. Dessa forma, esta seção apresenta um mapeamento entre as competências para o ensino de Matemática presentes nas Orientações Curriculares Complementares aos PCNs (PCN+, 2002) e diversos trabalhos recentes que descrevem e avaliam estratégias didáticas envolvendo o pensamento computacional. Embora não seja exaustivo, esse mapeamento permite identificar nas tendências atuais de pesquisa três competências com um maior potencial de desenvolvimento conjunto.

#### **3.1. Articulação dos símbolos e códigos**

A leitura e interpretação dos símbolos, códigos e nomenclaturas da linguagem matemática fazem parte das competências esperadas dos alunos. Em particular, espera-se que o aluno seja capaz de traduzir uma situação dada em uma linguagem em outra; por exemplo, transformar situações dadas em linguagem discursiva em gráficos, tabelas, fórmulas e outras representações, e vice-versa. Dentro do domínio das diferentes representações permitidas pela linguagem matemática, também se espera que o aluno seja capaz de “traduzir” uma representação em outra como, por exemplo, converter os dados de uma tabela em um gráfico.

Representar a solução para uma determinada situação na linguagem algorítmica é uma das competências fundamentais do pensamento computacional (CSTA, 2011). A representação de algoritmos na forma procedural traz algumas semelhanças com a linguagem algébrica, em particular na representação de variáveis; porém, o algoritmo se constitui em um modelo dinâmico, em oposição à representação algébrica, que tipicamente é utilizada para expressar relações entre grandezas desconhecidas, relações essas que são estáticas (ISBELL *et al.*, 2010). Segundo Mor e Noss (2008), a natureza sequencial do algoritmo, definido através de procedimentos a serem seguidos passo a passo, o aproxima da linguagem discursiva. Dessa forma, representar um problema na forma algorítmica pode se constituir como uma etapa intermediária entre a narração verbal e a linguagem algébrica, podendo promover uma transição mais “suave” para a compreensão da linguagem matemática.

Uma evidência inicial do enriquecimento tanto da linguagem algorítmica quanto da matemática expressadas por alunos é apresentada por Lewis e Shah (2012). Os autores identificam uma correlação positiva entre as notas de alunos da quarta série em testes padronizados de conhecimento matemático e as notas dos mesmos alunos em testes sobre conceitos básicos a respeito da construção de algoritmos, implementados utilizando a plataforma Scratch<sup>3</sup> durante uma oficina extraclasse. Os autores apontam algumas possíveis sobreposições entre conteúdos matemáticos e conteúdos desenvolvidos nas aulas da oficina. A título de exemplo, apresentamos na Figura 1 uma das possíveis relações apresentadas.

No problema $y = x + 4$ , se $x$ vale 7, quanto vale $y$ ?	
Preencha os espaços em branco para desenhar uma forma geométrica de 5 lados:	 A Scratch script starting with 'quando lapis clicado' (when green flag clicked), followed by 'abaixe a caneta' (put pen down), a 'repita' (repeat) block with a blank input field, containing 'mova 10 passos' (move 10 steps) and 'vire 90 graus' (turn 90 degrees), and ending with 'levante a caneta' (put pen up).

**Figura 1. Exemplo de analogia entre as notações matemática e algorítmica, adaptado de (LEWIS; SHAH, 2012)**

O tópico matemático em questão refere-se às relações lineares entre grandezas, expressas no primeiro caso em uma equação com duas incógnitas onde o valor de  $x$  é conhecido. Pretende-se avaliar se o aluno compreende que o valor de  $y$  pode ser obtido com as informações dadas. No segundo caso, evidentemente há mais conteúdos em jogo – o problema remete ao desenho de um polígono regular, e exige o conhecimento adicional que a soma dos ângulos internos de um polígono deve ser de 360 graus, logo a cada passo da repetição o aluno deve comandar um giro de  $360/N$  graus, dependente do número  $N$  de lados do polígono a traçar, que também se constitui no número de passos da estrutura de repetição. Está presente aí o mesmo conceito de relação linear entre

<sup>3</sup> <http://scratch.mit.edu>

grandezas, porém a linguagem narrativa do algoritmo pode permitir que o aluno identifique e teste suas próprias hipóteses, construindo ativamente o seu próprio formalismo matemático a partir da construção do algoritmo.

### **3.2. Estabelecimento de relações e identificação de regularidades**

Em (PCN+, 2002) registra-se como uma competência esperada que o aluno seja capaz de “identificar regularidades em situações semelhantes para estabelecer regras, algoritmos e propriedades”. A partir dessa recomendação, é possível inferir que o aluno deve ser estimulado a assumir uma postura exploratória frente ao mundo e que, a partir dessa postura, estabeleça ativamente as relações matemáticas. Aqui também encontramos similaridades com as propostas da literatura para o pensamento computacional. Wing (2006) considera a identificação de padrões como uma das primeiras competências associadas ao pensamento computacional. Como vimos anteriormente, a definição de Basawapatna *et al.* (2011) é inteiramente baseada na competência de identificação de padrões.

A formação de sequências numéricas é um conteúdo matemático frequentemente explorado em experiências didáticas envolvendo a identificação de padrões associada a recursos computacionais. Mor e Noss (2008) relatam três episódios de atividades pedagógicas, todas envolvendo a identificação das regras de formação de sequências por alunos a partir da elaboração de algoritmos em um software educativo. A identificação da regularidade na formação das sequências surge na narrativa dos alunos, e sua transformação, inicialmente em algoritmo e depois na representação matemática, parece contribuir para essa identificação.

Porém, há relatos que essa transição pode não ser trivial; nesse sentido, Setti (2009) utiliza sequências de Fibonacci em uma atividade onde alunos ingressantes de um curso superior em Informática são solicitados a identificar a regra de formação da sequência, bem como a construir um algoritmo que calcule os termos da sequência. A autora conclui que a transição entre a representação matemática da formação da sequência e a sua representação algorítmica não é imediata para a maioria daqueles alunos. Além disso, aponta que as diferenças na representação semiótica de ambas as soluções pode estar na raiz do problema. Podemos especular se aqueles alunos, muito provavelmente oriundos de um ensino básico “tradicional”, enfrentariam essas mesmas dificuldades de alternar entre uma e outra representação se tivessem sido expostos aos fundamentos do pensamento computacional desde o início de sua formação.

### **3.3. Modelos explicativos e representativos**

A terceira competência a discutir refere-se à elaboração e interpretação pelos alunos de modelos e representações matemáticas para analisar situações. Segundo (PCN+, 2002), as situações devem estar tipicamente associadas ao cotidiano do aluno, tais como: cálculos de lucro e prejuízo envolvendo gráficos, estimação das intenções de voto em uma campanha eleitoral envolvendo conceitos de estatística e probabilidade, entre outros. Nosso interesse aqui se volta particularmente para a elaboração de modelos pelos próprios alunos, que constitui a estratégia denominada *modelagem matemática*.

Segundo Bassanezi (2002), um modelo matemático é útil do ponto de vista didático se expressa ideias de maneira clara e sem ambiguidades, além de possibilitar o

uso de elementos computacionais para calcular suas soluções numéricas. Essa concepção de modelagem matemática pode ser amplificada em conjunto com as competências do pensamento computacional. A modelagem e simulação de fenômenos define, inclusive, uma das áreas de competências que compõem os objetivos educacionais do pensamento computacional em (CSTA, 2011).

Atualmente o suporte computacional para atividades de modelagem matemática está tipicamente restrito ao uso de planilhas eletrônicas, de software para plotagem de gráficos e da navegação na Internet (DINIZ, 2007). Ou seja, o uso da tecnologia está restrito ao nível do letramento computacional, enquanto que a utilização de ferramentas de software mais poderosas pode permitir o desenvolvimento e aplicação de competências do pensamento computacional. Um exemplo desse direcionamento é relatado por Lee *et al.* (2011), que descrevem uma atividade didática em que alunos produzem e testam um modelo de contágio de doenças considerando a quantidade de alunos e a disposição física dos ambientes da sua escola. O modelo é testado utilizando uma linguagem de programação em blocos baseada na criação de agentes computacionais. Segundo De Souza *et al.* (2011), tais ferramentas de software podem prover uma expressividade rica o suficiente para permitir que os alunos explorem e testem modelos para fenômenos razoavelmente complexos. Os conceitos matemáticos associados a tais modelos (como a probabilidade, por exemplo) podem ser explorados e validados dinamicamente através da alteração de parâmetros na própria ferramenta.

#### **4. Alguns caminhos de pesquisa**

A partir do panorama exposto, apresentam-se alguns desafios de pesquisa para a comunidade brasileira de Educação em Computação, visando incorporar o pensamento computacional à educação básica. A partir da revisão da literatura, verifica-se que a intersecção ou equivalência de conteúdos da Matemática e da Computação no contexto da educação básica ainda é incipiente; apesar de alguns indícios da transferência de competências entre os dois domínios, é necessário um trabalho abrangente de mapeamento dessas competências no corpo de conhecimento da Matemática e da Computação. Com as relações identificadas na seção anterior, pretendemos contribuir com a sistematização e o debate acerca desse mapeamento.

Uma vez que o mapeamento de competências esteja claro, torna-se necessário o desenvolvimento e avaliação de sequências didáticas envolvendo os conteúdos associados. No tocante ao uso de estratégias de modelagem matemática, há um maior desafio e um maior potencial associado. Uma modelagem matemática “revisada” terá suas possibilidades de exploração de fenômenos e conceitos matemáticos ampliadas pelo suporte trazido pelo pensamento computacional e por ferramentas de software adequadas. Tais ferramentas devem ter uma linguagem de especificação expressiva o suficiente para modelar fenômenos, porém mantendo uma estrutura sintática simples para que iniciantes possam produzir resultados rapidamente, e progressivamente produzir modelos mais complexos a partir da exploração da ferramenta.

#### **5. Considerações finais**

O pensamento computacional define competências e habilidades que se tornam fundamentais para o efetivo domínio da tecnologia por todos em um mundo em que os dispositivos computacionais são cada vez mais pervasivos. Incorporar o pensamento

computacional à educação básica envolve a análise sistemática de sua potencial sinergia com outras áreas do conhecimento, como a Matemática. Dessa forma, neste artigo realizamos um mapeamento entre as competências da área de Matemática previstas nos padrões curriculares brasileiros e aquelas desenvolvidas em atividades didáticas envolvendo o pensamento computacional descritas na literatura.

O mapeamento realizado permite agrupar as atividades desenvolvidas até o momento em três competências relacionadas ao ensino de Matemática: articulação de símbolos, identificação de padrões e regularidades e construção de modelos representativos e explicativos. Embora não se constitua em uma classificação definitiva, o agrupamento proposto permite identificar as tendências de pesquisa na aplicação do pensamento computacional ao ensino básico. Além disso, torna-se possível sistematizar o desenvolvimento de sequências didáticas que possam associar a Matemática e o pensamento computacional. Dessa forma, os caminhos de pesquisa apresentados, uma vez atacados pela comunidade, permitirão uma identificação mais clara dos benefícios do pensamento computacional e a disseminação dos fundamentos da Computação para um público mais abrangente.

## Referências

ASSOCIAÇÃO BRASILEIRA DE EMPRESAS DE TECNOLOGIA DE INFORMAÇÃO E COMUNICAÇÃO. **Tecnologia sofre com evasão universitária**. Disponível em: <<http://www.brasscom.org.br/brasscom/content/view/full/5155>>. Acesso em: 3 fev. 2012.

BASAWAPATNA, A.; KOH, K. H.; REPENNING, A. *et al.* **Recognizing computational thinking patterns**. SIGCSE 2011. **Proceedings of SIGCSE 2011**. New York: ACM, 2011.

BASSANEZI, R. **Ensino-aprendizagem com modelagem matemática**. São Paulo: Contexto, 2002.

BEAUBOUF, T. Why computer science students need math. **SIGCSE Bulletin**, v. 34, n. 4, p. 57–59, dez 2002.

BEZERRA, L. N.; SILVEIRA, I. F. **Licenciatura em Computação no Estado de São Paulo: uma Análise Contextualizada e um Estudo de Caso**. XIX Workshop sobre o Ensino de Computação. **Anais do CSBC 2011**. Natal: SBC, 2011.

BRASIL. MINISTÉRIO DA EDUCAÇÃO. **Parâmetros Curriculares Nacionais: Ensino Médio**. Brasília: MEC/SEB, 1999.

BRASIL. MINISTÉRIO DA EDUCAÇÃO. **PCN+ Ensino Médio: Orientações Curriculares Complementares aos Parâmetros Curriculares Nacionais**. Brasília: MEC/SEB, 2002.

CAMPBELL, P. F.; MCCABE, G. P. Predicting the success of freshmen in a computer science major. **Communications of the ACM**, v. 27, n. 11, p. 1108–1113, nov 1984.

CRENSHAW, T. L.; CHAMBERS, E. W.; METCALF, H.; THAKKAR, U. **A case study of retention practices at the University of Illinois at Urbana-Champaign**. SIGCSE '08. **Proceedings of SIGCSE 2008**. New York: ACM, 2008.

- DENNING, P. J. Is computer science science? **Communications of the ACM**, v. 48, n. 4, p. 27–31, abr 2005.
- DINIZ, L. DO N. **O Papel das Tecnologias da Informação e Comunicação nos Projetos de Modelagem Matemática**. Dissertação (Mestrado em Educação Matemática). Rio Claro: UNESP, 2007.
- ISBELL, C. L.; STEIN, L. A.; CUTLER, R. *et al.* (Re)defining computing curricula by (re)defining computing. **SIGCSE Bulletin**, v. 41, n. 4, p. 195–207, jan 2010.
- LEE, I.; MARTIN, F.; DENNER, J. *et al.* Computational thinking for youth in practice. **ACM Inroads**, v. 2, n. 1, p. 32–37, fev 2011.
- LEWIS, C. M.; SHAH, N. **Building Upon and Enriching Grade Four Mathematics Standards with Programming Curriculum**. SIGCSE '12. **Proceedings of SIGCSE 2012**. New York: ACM, 2012.
- MOR, Y.; NOSS, R. Programming as mathematical narrative. **International Journal of Continuing Engineering Education and Life-long Learning**, v. 18, n. 2, p. 214–233, 2008.
- MURATET, M.; TORGUET, P.; JESSEL, J.-P.; VIALLET, F. Towards a serious game to help students learn computer programming. **International Journal of Computer Games Technology**, v. 2009, p. 3:1–3:12, jan 2009.
- PAETON, G. University maths “too difficult” for British students. **The Telegraph**, 10 fev 2012. Disponível em: <<http://bit.ly/LTozHv>>. Acesso em: 21 abr. 2012.
- SETTI, M. DE O. G. **O Processo de Discretização do Raciocínio Matemático na Tradução para o Raciocínio Computacional: Um Estudo de Caso no Ensino/Aprendizagem de Algoritmos**. Tese (Doutorado em Educação). Curitiba: UFPR, 2009.
- SOUZA, C. S. DE; GARCIA, A. C. B.; SLAVIERO, C.; PINTO, H.; REPENNING, A. **Semiotic traces of computational thinking acquisition**. **Proceedings of IS-EUD 2011**. Heidelberg: Springer-Verlag, 2011.
- TARGINO, R. Quase 6 em 10 alunos do ensino médio de SP saíram da escola com desempenho ruim em matemática em 2011. **UOL Educação**, 7 mar 2012. Disponível em: <<http://uol.com/blcqGH>>. Acesso em: 23 abr. 2012.
- THE CSTA STANDARDS TASK FORCE. **CSTA K-12 Computer Science Standards**, 2011. New York: ACM Computer Science Teachers Association. Disponível em: <<http://csta.acm.org/Curriculum/sub/K12Standards.html>>. Acesso em: 3 fev. 2012.
- WILSON, B. C.; SHROCK, S. **Contributing to success in an introductory computer science course: a study of twelve factors**. SIGCSE '01. **Proceedings of SIGCSE '01**. New York: ACM, 2001.
- WING, J. M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33–35, mar 2006.