

Investigação do Currículo *Bootstrap* no Ensino de Matemática por meio da Programação de Computadores

Daniel A. Karling¹, Helio H. L. C. Monte-Alto¹

¹Universidade Federal do Paraná (UFPR) – Setor Palotina – Palotina, PR – Brazil

{heliohenrique,daniel.karling}@ufpr.br

Abstract. *Computer Science researchers argue that computer programming may help in teaching Mathematics. Bootstrap is a curriculum for teaching algebraic concepts for middle- and high-school classes by teaching programming, taking advantage, among other approaches, from the functional programming paradigm. Therefore we analyse the employment of functional languages and the effectiveness and suitability of Bootstrap by means of an experimental study with a sample of students from public schools. The results of the study indicate the potential of the approach in algebra teaching, motivating the use of this approach in Brazil and its application in the classroom, thus contributing to the improvement of Mathematics teaching and the introduction of Computer Science in schools.*

Resumo. *Educadores e pesquisadores da área de Computação argumentam que a programação de computadores pode auxiliar no ensino de Matemática. A abordagem curricular Bootstrap visa ensinar conceitos algébricos na Educação Básica através do ensino de programação, utilizando, dentre outras abordagens, o paradigma de programação funcional. Portanto, buscamos analisar o emprego de linguagens funcionais e a efetividade e adequabilidade do Bootstrap mediante um estudo experimental com uma amostra de alunos da rede pública estadual. Os resultados do estudo indicam o potencial da abordagem no ensino de álgebra, motivando o uso desta abordagem no Brasil e sua aplicação em sala de aula, contribuindo, assim, para a melhoria do ensino de Matemática e a introdução da Ciência da Computação nas escolas.*

1. Introdução

Linguagens de Programação proporcionam ambientes de expressão de raciocínio visando a solução de problemas [de Lima 2009]. De acordo com [Valente 1999], “o computador pode ajudar no processo de conceituação e no desenvolvimento de habilidades importantes para a sobrevivência na sociedade do conhecimento se for usado como um dispositivo para ser programado”. Várias pesquisas defendem que conhecimentos sobre programação são importantes para estudantes que se interessam por outras áreas além da Ciência da Computação, desenvolvendo novas habilidades como o raciocínio lógico, resolução de problemas e pensamento algorítmico [Papert 1972, Wing 2006].

Entretanto, a programação é uma disciplina que, muitas vezes, não é vista com bons olhos por alunos ingressantes em cursos superiores na área de Computação e Engenharias [Rebouças et al. 2010]. A aprendizagem de conceitos e métodos para a construção de programas de computador não é trivial, requerendo o uso de habilidades de

alto nível e muito raciocínio abstrato [Castro et al. 2008]. A elaboração de algoritmos é uma das competências mais difíceis de serem desenvolvidas, principalmente para aqueles que acabam de ingressar em cursos desta área [Júnior et al. 2005]. Tal dificuldade pode vir por consequência de falhas na metodologia de ensino nesses cursos.

A abordagem de ensino de álgebra por meio de programação *Bootstrap* [Schanzer et al. 2015], já amplamente difundida nos EUA, porém ainda pouco conhecida no Brasil, baseia-se na ideia de ensinar Matemática por meio da programação de computadores. Deste modo, aprimora-se o ensino de Matemática de uma forma mais instigante e motivadora, e ao mesmo tempo introduz-se a Ciência da Computação. O currículo baseia-se em um modelo de computação e de raciocínio que visa permitir uma transferência de aprendizagem efetiva entre as duas disciplinas (Computação e Matemática). Destaca-se, entre as abordagens utilizadas pelo *Bootstrap*, a utilização do paradigma de programação funcional, como proposto por [Chakravarty e Keller 2004, Schanzer et al. 2015]. Tal paradigma é defendido por apresentar características que oferecem maior proximidade com conceitos matemáticos aprendidos na Educação Básica, pois baseia-se na definição e aplicação de funções matemáticas.

Este artigo apresenta um estudo sobre a abordagem *Bootstrap*, assim como sobre as ideias e metodologias de ensino adotadas e sua efetividade no ensino de álgebra e programação de computadores na Educação Básica. A Seção 2 apresenta a metodologia utilizada neste trabalho. A Seção 3 apresenta em detalhes o currículo *Bootstrap* e seu referencial teórico. A Seção 4 apresenta os resultados e discussão. Por fim, apresentam-se as considerações finais.

2. Metodologia

A metodologia adotada para este estudo consiste nas seguintes etapas: (I) investigação dos conceitos e abordagens utilizadas no currículo *Bootstrap*, por meio da revisão de seu plano de ensino e material didático; (II) tradução dos planos de aula e material didático; (III) revisão bibliográfica sobre o uso de linguagens funcionais no ensino de programação; e (IV) planejamento e execução de um estudo experimental a fim de verificar a efetividade da abordagem para alunos do 9º Ano do Ensino Fundamental ao 2º Ano do Ensino Médio de um colégio público.

Portanto, a pesquisa iniciou-se com leitura e tradução dos planos de aula do *Bootstrap* a fim de se obter um entendimento sobre a abordagem, assim como iniciar a preparação do material em Português, sem o qual torna-se muito difícil sua aplicação no Brasil. Embora o uso da programação funcional no ensino de programação seja defendido por vários pesquisadores da área [Felleisen 2001, Van-Roy e Haridi 2004, Abelson et al. 1996, Chakravarty e Keller 2004] e adotado em várias universidades, ainda é obscurecido pelo uso mais comum do paradigma imperativo e, portanto, pouco conhecido por muitos acadêmicos. Assim, foi realizada uma revisão bibliográfica sobre o uso de programação funcional em disciplinas iniciais de algoritmos e programação a fim de se obter o estado da arte sobre o assunto na literatura. Por fim, visando validar a efetividade do *Bootstrap* no ensino de álgebra, foi realizado um estudo experimental no formato de aulas experimentais extra-classe.

3. Desenvolvimento

3.1. O Currículo *Bootstrap*

O *Bootstrap* [Schanzer et al. 2015] apresenta um currículo designado a ensinar tópicos-chave de álgebra e geometria por meio do desenvolvimento, pelos próprios alunos, de jogos eletrônicos contendo um jogador, um alvo e algum perigo. Diferentemente da maioria dos cursos de programação, “o *Bootstrap* utiliza álgebra como o veículo para a criação de imagens e animações” [Schanzer et al. 2015], eliminando, deste modo, a necessidade do aluno ter que aprender a pensar como a máquina (conforme o paradigma imperativo) e se concentre apenas em definir funções matemáticas para resolver problemas computacionais.

Estudos experimentais já foram realizados com a abordagem nos EUA, obtendo-se resultados que corroboram com a hipótese de que o currículo é realmente eficaz no ensino de conceitos de álgebra [Schanzer et al. 2015]. Esse estudo foi feito comparando-se dados iniciais sobre os conhecimentos dos alunos aos dados finais, após o término do ensino, por meio da aplicação de pré-teste e pós-teste. Uma tentativa de replicar o experimento no Brasil será apresentada na Seção 3.3.

Atualmente o projeto *Bootstrap* já conta com dois módulos. O *Bootstrap:Algebra*, anteriormente chamado de *Bootstrap:1*, é o currículo que tratamos neste trabalho. Ele consiste na aplicação de conceitos matemáticos e princípios rigorosos de programação para criar um jogo eletrônico simples com 3 personagens: um jogador, um alvo e um perigo. Para isso, eles desenham cada um dos personagens e utilizam conceitos algébricos para fazer a detecção de colisão, o tratamento de comandos do teclado e a determinação de como os personagens se movem e interagem.

Além desse primeiro currículo, existe também sua continuação: O *Bootstrap:Reactive*, que se aprofunda um pouco mais na programação e introduz mais alguns conceitos matemáticos, tais como relações funcionais complexas, aleatoriedade e noções mais avançadas de trigonometria. Atualmente, dois novos currículos se encontram em desenvolvimento: o *Bootstrap:DataScience* e o *Bootstrap:Physics*. O primeiro tem por objetivo introduzir os alunos à Ciência de Dados, que introduz conceitos fundamentais de estatística, e se encontra em fase de teste piloto em várias escolas nos EUA. O segundo consiste em utilizar conceitos de física no desenvolvimento de programas, e se encontra em fase de desenvolvimento.

Este artigo foca apenas no módulo *Bootstrap:Algebra*, que é dividido em 9 unidades. Cada unidade tem por objetivo desenvolver um incremento do jogo, sendo que para cada nova parte desenvolvida são apresentados os conceitos de programação e os conceitos de Matemática necessários para implementá-la. Desse modo, cada unidade introduz, em paralelo, novos conceitos do jogo, de programação e de Matemática, como detalhado na Tabela 1.

Uma característica interessante do *Bootstrap*, que também o diferencia de outras metodologias de ensino de programação na Educação Básica, é o fato de ser elaborado para ser aplicado em sala de aula, e não como atividade extra-classe (como, por exemplo, o *Code Club*¹). O *Bootstrap* tem a intenção de que professores de Matemática aprendam a abordagem e a utilizem para ensinar tópicos de álgebra e geometria, utilizando a

¹<http://www.codeclubbrasil.org.br>

Tabela 1. Estrutura do currículo: cada unidade introduz conceitos do jogo, de programação e de matemática em paralelo. [Schanzer et al. 2015]

Unidade	Característica do Jogo	Conceito de Programação	Conceito de Matemática
1	localização dos elementos na tela	expressões, Círculos de Avaliação	coordenadas
2	criação de texto e imagem	operações em <i>strings</i> e imagens	domínio, imagem, tipos de dados
3-5	fazendo imagens se moverem	definição de funções, exemplos (testes)	múltiplas formas de representar funções: como fórmulas e como tabelas
6	determinar quando os elementos do jogo estão fora da tela	Valores e operações <i>booleanas</i>	inequidades
7	responder a comandos do teclado	estrutura condicional	funções definidas por partes
8	detecção de colisão	(nada novo)	Teorema de Pitágoras
9	polindo jogos para apresentação	revisões de código	explicação de conceitos matemáticos para os outros

programação como uma ferramenta motivadora e facilitadora para a aprendizagem de Matemática [Lopes et al. 2016]. Portanto, o currículo é alinhado ao CCSM (*Common Core Standards for Mathematics*, do Inglês: Diretrizes Básicas Comuns para a Matemática), que é o conjunto de diretrizes estabelecidas para o planejamento curricular da disciplina de Matemática nos Estados Unidos. Desta forma, o *Bootstrap* abrange a maior parte do conteúdo algébrico estabelecido para o 7º Ano do Ensino Fundamental americano até a disciplina chamada de Álgebra 2, tornando-o facilmente integrável às aulas de Matemática que cobrem esse conteúdo. O *Bootstrap* também é uma implementação do modelo CCSMP (*Common Core Standards for Mathematical Practice*, do Inglês: Diretrizes Básicas Comuns para a Prática de Matemática), oferecendo recomendação pedagógica explícita em todos os oito padrões de prática de matemática estabelecidos nos EUA. Além disso, o *Bootstrap* satisfaz vários dos padrões da CSTA (*Computer Science Teacher's Association*, do Inglês: Associação dos Professores de Ciência da Computação), um conjunto de diretrizes definidas nos EUA para o ensino de Computação na Educação Básica [Bootstrap 2017]. Esforços vêm sendo empregados para criar diretrizes similares ao CSTA no Brasil, tais como as diretrizes propostas por [Von Wangenheim et al. 2014].

A *Code.Org*², organização sem fins lucrativos que visa encorajar pessoas a aprender Ciência da Computação, fez uma parceria com a equipe do *Bootstrap* e criou um currículo baseado nele, o *CS in Algebra*. A principal diferença entre ambos é que o *CS in Algebra* utiliza uma linguagem de blocos similar ao *Scratch*³, enquanto que o *Bootstrap* original utiliza a linguagem BSL (*Begginer Student Language*), uma linguagem funcional

²<https://code.org>

³<https://scratch.mit.edu>

e textual. O *CS in Algebra* também implementa as diretrizes CCSM, CCSMP e CSTA, conforme detalhado em [Code.Org 2017].

O *Bootstrap* primeiramente introduz os Círculos de Avaliação como uma maneira de ensinar a compôr expressões algébricas, como exemplificado na Figura 1(a). Um Círculo de Avaliação possui as seguintes regras: todo círculo contém uma função (ou operador) no topo e os operandos devem ser descritos abaixo, da esquerda para a direita. Desta maneira, pode-se começar a ensinar a ideia de expressões aninhadas e precedência, além de introduzir uma noção do que é uma função matemática. Tal abordagem também é interessante por proporcionar uma visão mais abstrata das expressões sendo, inclusive, de mais fácil transição para uma sequência de códigos em BSL. Torna-se mais simples compreender conceitos algébricos desta forma do que visualizando uma expressão escrita de maneira convencional.

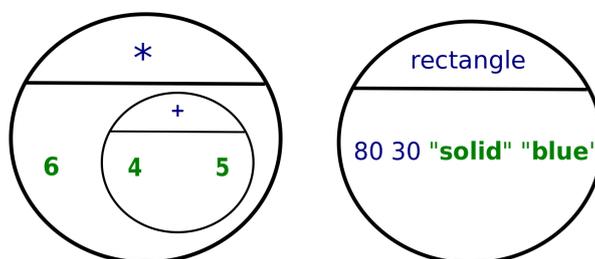


Figura 1. (a) À esquerda, exemplo de Círculo de Avaliação para a expressão aritmética $6 \times (4 + 5)$ [Schanzer et al. 2015]; (b) à direita, exemplo de Círculo de Avaliação que desenha um retângulo azul preenchido de tamanho 80x30.

O *Bootstrap* utiliza a linguagem funcional BSL, que é baseada na linguagem *Racket*⁴, para ensinar programação aos alunos. A linguagem apresenta uma sintaxe que permite uma tradução facilitada dos Círculos de Avaliação para expressões válidas na linguagem. O círculo apresentado na Figura 1(a), por exemplo, é traduzido para a expressão pré-fixada $(*\ 6\ (+\ 4\ 5))$, equivalente à expressão infixa $6 \times (4 + 5)$. Apesar da notação pré-fixada parecer estranha, no início, ela se mostra muito adequada para introduzir o conceito de funções, uma vez que considera-se que qualquer operação é feita por meio da aplicação de funções. Por exemplo, no decorrer do curso o aluno aprende que pode escrever uma expressão que desenha um retângulo na tela (por meio de uma chamada à função `rectangle`) do mesmo modo como escreve uma expressão aritmética, tal como exposto na Figura 1(b).

O projeto apresenta, ainda, uma abordagem pedagógica para a resolução de problemas inspirada em [Polya 1978], chamada de Receita de Projeto (do inglês, *Design Recipe*), que auxilia os alunos a pensarem em uma definição passo-a-passo para resolver um problema. As Receitas de Projeto incorporam conceitos-chave em padrões de problemas algébricos, de modo que os alunos sejam capazes de trabalhar com diferentes tipos de funções, analisando previamente conceitos de domínio e imagem, entrada/saída e representação simbólica [Schanzer et al. 2015].

À medida que os alunos desenvolvem seus jogos, vai surgindo a necessidade da

⁴*Racket* é uma linguagem de programação generalista baseada em *Lisp* e *Scheme*. Mais informações em <http://racket-lang.org/>.

introdução de diferentes conceitos. Um deles é o Teorema de Pitágoras, utilizado para que o jogo detecte colisões entre os elementos do jogo. Além disso, logo de início os alunos são introduzidos ao plano cartesiano, pois o jogo desenvolvido por eles ocorre em um cenário 2D no qual é necessário considerar a posição dos personagens nos eixos x e y . Como o aprendizado é focado em objetivos (fazer o jogo funcionar), tudo torna-se muito natural e proveitoso.

Os planos de aula e apostila do *Bootstrap:Algebra* estão totalmente traduzidos e revisados, e a tradução do *Bootstrap:Reactive* encontra-se em fase final de revisão.

3.2. O Paradigma Funcional no Ensino da Programação

Um paradigma de programação determina a visão que o programador possui a respeito da estruturação e execução do programa. A programação funcional é um paradigma de programação que trata a computação como a avaliação de funções matemáticas evitando estados ou dados mutáveis. Tal paradigma enfatiza a aplicação de funções, em contraste com a programação imperativa, que enfatiza mudanças de estado em um programa [Hudak 1989].

Para que haja transferência de conhecimentos entre duas disciplinas são necessárias ligações estruturais profundas entre os domínios e instruções explícitas sobre como aplicar conceitos de uma disciplina na outra [Bransford e Schwartz 1999]. O paradigma funcional baseia-se em conhecimentos que o aluno deve estar familiarizado desde o Ensino Médio [Castro et al. 2008], tais como funções e mapeamento entre domínios, permitindo que os alunos se concentrem na elaboração de soluções e sua descrição formal, capacitando-os a resolverem problemas mais complexos logo no início de um curso.

O modelo tradicional de ensino de programação, segundo [Borges 2000], não é muito atrativo, e dificilmente consegue motivar o aluno pela disciplina. Esse modelo, além de tudo, é distinto ao que o aluno aprende em Matemática, implicando, desta maneira, que o aluno tenha que pensar de modo diferente, como uma máquina que executa passos prévia e linearmente escritos.

Para justificar a abordagem, [Chakravarty e Keller 2004] destacam três objetivos principais dos cursos introdutórios de Computação: (i) transmitir técnicas elementares de programação (prática); (ii) introduzir conceitos essenciais de Computação (teórica); e (iii) promover o desenvolvimento de habilidades de resolução de problemas analíticos (metodológico). Propõe-se que as linguagens funcionais permitem abordar bem esses três aspectos de forma integrada. As bases semânticas limpas da linguagem ajudam nos primeiros dois objetivos, e o alto nível de expressividade da linguagem é um fator essencial no que diz respeito ao terceiro aspecto.

Uma vez que o foco principal de um curso de programação é a solução de problemas, torna-se vantajosa a utilização de uma linguagem de programação que permita declarar soluções de modo que o sucesso da aprendizagem dependa mais da habilidade de abstração, aprendida com maior facilidade neste paradigma, do que da habilidade de codificação. Além disso, a maior similaridade semântica do paradigma funcional com a Matemática, comparada ao paradigma imperativo, pode tornar o primeiro contato do aluno com a programação menos doloroso e mais satisfatório. A subsequente introdução dos demais paradigmas, como o imperativo, torna-se, então, mais facilitada, uma vez que

o aluno tenha entendido como compôr soluções em forma de algoritmos no paradigma funcional.

3.3. Planejamento e Execução do Estudo Experimental

A fim de verificar a validade do *Bootstrap* no ensino de álgebra por meio da programação, planejamos e executamos um estudo experimental. Pretende-se, por meio dele, testar a seguinte hipótese: “Alunos que completaram o currículo *Bootstrap* passaram a ter um desempenho melhor em Matemática, especialmente na resolução de problemas algébricos”.

Para tanto, foram convidados, como participantes das aulas experimentais, alunos entre o 9º Ano do Ensino Fundamental e o 2º Ano do Ensino Médio de um colégio da rede pública de ensino, para os quais foi aplicado um pré-teste com questões de álgebra e problemas algébricos. No total, 7 (sete) alunos se ofereceram voluntariamente para participar do estudo. No entanto, apenas 4 (quatro) alunos continuaram até o final do curso, o que representa, de fato, uma amostra muito baixa. Após a execução das aulas experimentais, um pós-teste com questões similares e dificuldade equivalente ao pré-teste foi aplicado, a fim de verificar se houve melhora significativa do conhecimento e habilidades em álgebra dos participantes.

A primeira parte dos testes apresentou um caráter bem prático, não necessitando interpretar as questões, mas apenas substituir as chamadas de funções pelos respectivos resultados. O restante do teste foi composto de exercícios interpretativos, também chamados de problemas algébricos (ou *word problems*, em Inglês), em que o participante devia indicar o domínio e imagem da função que resolve o problema, assim como a fórmula da função e uma tabela com entradas e saídas das funções. Um exemplo de um desses problemas é apresentado na Figura 2.

10. O rótulo da lata de tinta que Chang comprou dizia que 1 galão de tinta cobre 90 metros quadrado. A função $\text{metros}(g)$ representa o número de metros quadrados que g galões podem cobrir.

a. Qual é o domínio e qual é a imagem de 'metros'?

domínio : _____ imagem: _____

b. Preencha a coluna de resultado para a função $\text{metros}(g)$, completando os dois exemplos fornecidos para mostrar como o número de metros quadrados que podem ser pintados se relacionam ao número de galões fornecidos.

$f(2)$	
$f(3)$	

c. Escreva a função $\text{metros}(g)$ que representa o número de metros quadrados que g galões cobrirão.

$\text{metros}(g) =$ _____

Figura 2. Exemplo de problema algébrico aplicado no pré-teste.

4. Resultados e Discussão

Após a aplicação do pós-teste, constatou-se uma melhora significativa na nota média em relação ao pré-teste, elevando de 29% para 62,2% de acertos no teste em geral. A análise individual perante o desempenho no pós-teste mostrou uma evolução em todos os alunos, conforme mostrado na Figura 3, embora não tenha sido muito significativa em alguns casos.

Na primeira etapa dos testes (aplicação de funções), no pré-teste houve pouco mais de 62% de acertos, em média, já no pós-teste a média ficou em 90% de acertos.

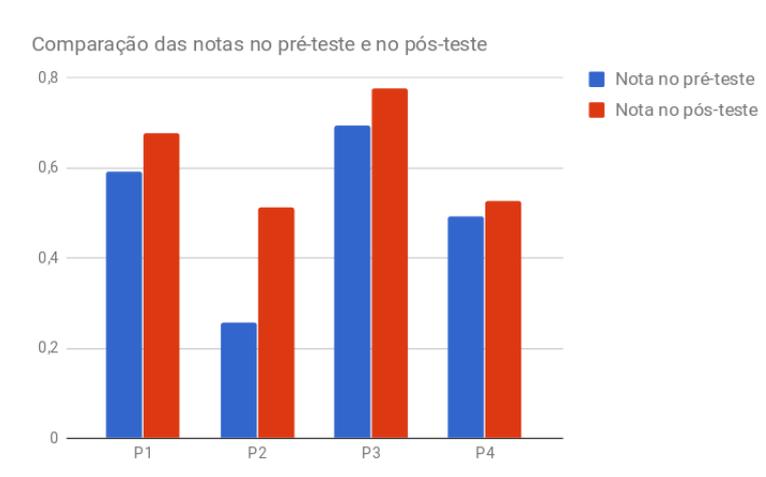


Figura 3. Comparação entre as notas dos participantes (P1, P2, P3 e P4) no pré-teste e pós-teste.

Na segunda etapa (problemas algébricos) houve uma taxa de acerto inferior a 22% no pré-teste e próxima a 32% no pós-teste.

Conclui-se que, embora não seja possível afirmar, por meio de teste estatístico, que a aplicação do curso foi muito significativa, por conta do baixo número de participantes, as evidências indicam que houve uma melhora em suas habilidades e conhecimentos algébricos.

4.1. Ameaças à Validade

Houve baixo interesse por parte dos alunos convidados para participar, portanto conseguiu-se apenas uma amostra pequena para as aulas experimentais. Além disso, dos 7 voluntários iniciais, apenas 4 participaram do curso até o final. Embora tenham havido resultados positivos, esse tamanho de amostra é muito pequeno para assegurar que a abordagem apresentou significância estatística. Apesar disso, é possível assumir uma certa validade a partir de alguns pressupostos relacionados à execução do estudo experimental: (i) todos os participantes eram da mesma turma, portanto possuíam conhecimentos comparativamente equivalentes de álgebra; e (ii) os alunos não tiveram, durante o período em que ocorreu o curso experimental, contato, na escola, com o conteúdo abordado. Pressupõe-se, também, que os alunos não tenham estudado em casa por conta própria os conteúdos de álgebra.

A dificuldade em conseguir voluntários e em retê-los até o final do curso é também um indicativo de que a aplicação do curso é mais adequada se for realizada de modo integrado às aulas de Matemática, conforme o propósito principal do *Bootstrap*. Deste modo, a participação não dependeria de voluntários para realizar o curso em horário extra-classe, e também evitar-se-ia a evasão.

A principal dificuldade encontrada para aplicar o experimento em sala de aula é a falta de interesse dos professores de Matemática pelo uso da programação. Um projeto de extensão relacionado a este trabalho vêm tentando fornecer cursos de programação para professores de Matemática como uma forma de incentivá-los e prepará-los para o uso de metodologias como o *Bootstrap* em sala de aula [Monte-Alto et al. 2017]. Também

é importante mencionar a dificuldade que os professores de muitas escolas têm em levar os alunos para um laboratório de informática, que muitas vezes se encontra em situação precária, com computadores que não funcionam e acesso lento à *Internet*.

5. Conclusão e Trabalhos Futuros

É interessante observar como o uso do paradigma de programação funcional possui potencial, tanto no ensino introdutório de programação no Ensino Superior, quanto no ensino de Matemática e programação na Educação Básica. A abordagem curricular *Bootstrap* para o ensino de álgebra por meio de programação de computadores também parece promissora, levando-se em consideração, ainda, o interesse que o desenvolvimento de jogos de computador pode despertar nos alunos, tornando-os mais motivados. Acreditamos que os resultados decorrentes do estudo experimental são positivos quanto à efetividade do currículo *Bootstrap*, estimulando, assim, a introdução deste currículo na educação brasileira e contribuindo para a melhoria do ensino de Matemática e para a introdução e incentivo ao ensino de Ciência da Computação na Educação Básica. No entanto, a abordagem só será realmente efetiva se for aplicada em sala de aula, o que requer que se desperte o interesse dos professores de Matemática pelo seu uso. Iniciativas de extensão vêm sendo empregadas para que os professores possam ser incentivados a utilizar essas metodologias em sala de aula. Pretende-se também, como trabalho futuro, aplicar o estudo experimental com o *Bootstrap* diretamente em sala de aula. Deste modo, será possível obter resultados mais significativos no que diz respeito à efetividade da abordagem.

Referências

- Abelson, H., Sussman, G. J., e Sussman, J. (1996). *Structure and interpretation of computer programs*. Justin Kelly.
- Bootstrap (2017). Standards alignment. Disponível em <http://www.bootstrapworld.org/materials/Standards.shtml>. Último acesso em 6 de junho de 2017.
- Borges, M. (2000). Avaliação de uma metodologia alternativa para a aprendizagem de programação. In *Anais do VIII Workshop de Educação em Computação–WEI*.
- Bransford, J. D. e Schwartz, D. L. (1999). Rethinking transfer: A simple proposal with multiple implications. *Review of Research in Education*, 24(1):61–100.
- Castro, T., Fuks, H., Spósito, M. A. F., e de Castro Jr, A. N. (2008). Análise de um Estudo de Caso para Aprendizagem de Programação em Grupo. *Revista Iberoamericana de Tecnologias del Aprendizaje*, 4(2):155–160.
- Chakravarty, M. M. T. e Keller, G. (2004). The risks and benefits of teaching purely functional programming in first year. *J. Funct. Program.*, 14(1):113–123.
- Code.Org (2017). Code.org standards alignment. Disponível em <https://code.org/curriculum/docs/algebra/standards>. Último acesso em 6 de junho de 2017.
- de Lima, M. R. (2009). Construcionismo de papert e ensino-aprendizagem de programação de computadores no ensino superior. Master's thesis, Universidade Federal de São João Del-Rei - MG.

- Felleisen, M. (2001). *How to design programs: an introduction to programming and computing*. MIT Press.
- Hudak, P. (1989). Conception, evolution, and application of functional programming languages. *ACM Comput. Surv.*, 21(3):359–411.
- Júnior, J. C. R. P., Rapkiewicz, C. E., Delgado, C., e Xexeo, J. A. M. (2005). Ensino de algoritmos e programação: Uma experiência no nível médio. In *XIII Workshop de Educação em Computação (WEI'2005)*, pages 2351–2362, São Leopoldo, RS.
- Lopes, C. C., dos Santos Guedes, M. d. S., Sousa, E. A. D., de Souza, R. P., Pereira, I. B., et al. (2016). O ensino de algoritmos e lógica de programação como uma ferramenta pedagógica para auxiliar a aprendizagem de matemática: Um relato de experiência. In *Anais do Workshop de Informática na Escola*, volume 22, page 41.
- Monte-Alto, H. H., Schreiner, M. A., Dörr, J., Lisbôa, E., e Soares, J. P. (2017). Iniciativas para a disseminação da ciência da computação na educação básica no município de palotina - paraná. *Extensão em Foco*, 1(13).
- Papert, S. (1972). Teaching children to be mathematicians versus teaching about mathematics. *International Journal of Mathematical Education in Science and Technology*, 3(3):249–262.
- Polya, G. (1978). *A arte de resolver problemas*, volume 2. Interciência, Rio de Janeiro.
- Rebouças, A. D. D. S., Marques, D. L., Costa, L. F. S., e Silva, M. A. A. (2010). Aprendendo a Ensinar Programação Combinando Jogos e Python. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 1, pages 1–10.
- Schanzer, E., Fisler, K., Krishnamurthi, S., e Felleisen, M. (2015). Transferring skills at solving word problems from computing to algebra through bootstrap. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education, SIGCSE '15*, pages 616–621, New York, NY, USA. ACM.
- Valente, J. A. (1999). *O computador na sociedade do conhecimento*. UNICAMP/NIED, São Paulo.
- Van-Roy, P. e Haridi, S. (2004). *Concepts, techniques, and models of computer programming*. MIT press.
- Von Wangenheim, C. G., Nunes, V. R., dos Santos, G. D., da Cruz Alves, N., Coan, E. S., e Mansur, C. (2014). Resumo de objetivos de aprendizagem de computação no ensino fundamental (currículo de referência csta/acm k-12). Technical report, Universidade Federal de Santa Catarina, Florianópolis - SC.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.