

A GRAMÁTICA DE GRAFOS COMO ALTERNATIVA PARA O DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL NA EDUCAÇÃO BÁSICA

**BRAZ ARAUJO DA SILVA JUNIOR¹; LUCIANA FOSS²; SIMONE ANDRÉ DA
COSTA CAVALHEIRO³**

¹Universidade Federal de Pelotas – badsjunior@inf.ufpel.edu.br

²Universidade Federal de Pelotas – luci.foss@gmail.com

³Universidade Federal de Pelotas – simone.costa@inf.ufpel.edu.br

1. INTRODUÇÃO

A descrição do conceito de pensamento computacional como o conjunto de processos e estratégias para a formulação e resolução de problemas utilizando os fundamentos do amplo campo da ciência da computação (WING, 2006) incentivou o questionamento sobre o que é a ciência da computação e o quanto ela pode contribuir para a solução de problemas ao longo de todas as áreas de conhecimento humano (LEE, 2011).

Jeannette Wing expõe que as habilidades necessárias para um profissional da ciência da computação não têm sua utilidade restrita à sua área, são habilidades de propósito geral presentes até mesmo em situações cotidianas e que são fundamentais para todos (WING, 2006). A abrangência do pensamento computacional conquistou interesse de grandes empresas como Google e Microsoft. A Google lançou seu programa online sobre pensamento computacional, além de disponibilizar cursos de pensamento computacional para educadores (Google, 2015). E foi com o patrocínio da Microsoft, que a Universidade de Carnegie Mellon fundou seu Centro de Pensamento Computacional (Carnegie Mellon, 2015).

Tendo o pensamento computacional adquirido tal visibilidade, líderes da educação primária, secundária e superior colaboraram com a *International Society for Technology in Education* (ISTE) e a *Computer Science Teachers Association* (CSTA) para desenvolver uma definição operacional de pensamento computacional (ISTE, 2015). A partir desta colaboração foram criadas também ferramentas para o ensino e aprendizado ao longo da educação básica, das habilidades que constituem o pensamento computacional (como simulação, automação, algoritmos e procedimentos), estruturadas em um quadro de progressão. (CSTA, 2015).

O ensino sobre o pensamento computacional para educadores já é proposto por grandes empresas como a *British Broadcasting Corporation* (BBC) (BBC Education, 2015) e instituições de ensino superior renomadas como a Universidade de Harvard (Harvard GSE, 2015) e o *Massachusetts Institute of Technology* (edX, 2015), que se interessaram pela temática e passaram a desenvolver projetos, guias e cursos online focando-a. Ensino de programação para educação básica também já é abordado por diversas entidades como o grupo *Lifelong Kindergarten* com o projeto *Scratch* (MIT Media Lab, 2003) e a code.org, que conta com o apoio de grandes empresas como a Microsoft e o Facebook (Code.org, 2015). Ela promoveu em 2013 a campanha *Hour of Code* que se transformou em um movimento global para a desmitificação da programação (Code.org, 2013) e inspirou outras organizações a disponibilizarem aulas de programação para públicos variados, como a Hora do Código com a

Khan Academy, que ensina código para crianças e adolescentes a partir de 8 anos (Khan Academy, 2015).

Porém, a maioria destas propostas baseiam-se no paradigma imperativo da programação, que apesar de ser eficiente, pode não ser o ideal no mundo atual. Isto porque hoje em dia é fundamental o desenvolvimento de sistemas que sejam confiáveis, e para isso é necessária uma forma rigorosa e eficaz de modelagem, projeto e análise destes sistemas, que pode ser atingida através do uso de técnicas baseadas em formalismos matemáticos, os métodos formais (CRAIGEN; GERHART; RALSTON, 1993).

É interessante também que não possuam difícil legibilidade, sejam intuitivos, por isto a proposta deste trabalho é explorar a Gramática de Grafos (GG), que possui especificação visual, e consegue expressar de forma natural construções complexas, como as de distribuição, concorrência e paralelismo, por exemplo (EHRIG et al., 1997). E usá-la como alternativa para o ensino de programação e, consequentemente, o desenvolvimento das habilidades fundamentais do pensamento computacional na educação básica.

2. METODOLOGIA

Para atingir o objetivo deste trabalho serão estudadas e analisadas as propostas já existentes para o ensino de programação na educação básica, como as plataformas digitais interativas: *Robo Mind* (Research Kitchen, 2005); *Scratch* (MIT Media Lab, 2003); *Khan Academy* (Khan Academy, 2015); e *Kodu Game Lab* (Microsoft, 2009). Deverão ser estudadas e analisadas também quais abordagens visuais permitem descrever concorrência, paralelismo e/ou outras estruturas complexas de forma natural além da proposta por este trabalho. Adicionalmente, serão realizados estudos e análises das ferramentas disponíveis para a especificação e simulação de Gramática de Grafos, como a *Attributed Graph Grammar* (AGG) e o *Graphs for Object-Oriented Verification* (GROOVE).

E então se construirá, através dos estudos e resultados das análises mencionadas acima, uma proposta de ensino de programação com Gramática de Grafos para a educação básica. Também será realizada uma análise comparativa entre a proposta a ser construída e as já existentes, para que possam ser ressaltados e discutidos todos os pontos críticos e relevantes da proposta, bem como suas vantagens e desvantagens em relação às demais.

3. RESULTADOS E DISCUSSÃO

Como o discutido em (WING, 2008), o pensamento computacional tem influenciado pesquisas em praticamente todas as disciplinas e, a tendência é ser usado cada vez mais ampla e intensamente. Mas, se ele será utilizado em todo lugar, ele atingirá, direta ou indiretamente, todo mundo. Isto nos leva a uma necessidade de ensino abrangente do pensamento computacional. E caso queiramos assegurar uma base sólida e comum de entendimento e aplicação do pensamento computacional, então isto deverá ser feito desde os primeiros anos, ainda na infância.

Atualmente, grande parte dos sistemas computacionais que temos contato diariamente já possui enorme complexidade e estes sistemas tendem a se tornar cada vez mais e mais complexos, em função de garantirem eficiência e segurança em seus objetivos. Este contato crescente com tal complexidade nos leva a necessidade de não só introduzir o pensamento computacional, mas ensiná-lo de uma maneira em que estruturas complexas (como concorrência e paralelismo)

fluam de forma natural, para que nos familiarizemos com a complexidade sem que ela nos pareça um esforço muito assustador.

Esta é a importância da exploração de métodos de introdução do pensamento computacional que permitam construções naturais de estruturas complexas como a proposta por este trabalho, com Gramática de Grafos.

4. CONCLUSÕES

Este artigo relatou uma proposta que segue um caminho diferente dos comumente encontrados norteando o ensino de ciência da computação aplicado a alunos na educação básica. A proposta de um trabalho a ser desenvolvido acerca do desenvolvimento das habilidades fundamentais do pensamento computacional através de uma nova abordagem, ensinando-se programação com Gramática de Grafos. Com isto, incentiva-se a possibilidade de alternativas viáveis ao ensino da programação dentro do ensino primário, visando demonstrar a viabilidade de paradigmas de computação além do imperativo.

5. REFERÊNCIAS BIBLIOGRÁFICAS

WING, J. A Vision for the 21st Century: Computational Thinking. **Communications Of The ACM**, New York, v.49 n.3, 33–35, 2006.

WING, J. **Computational Thinking and Thinking About Computing**, 2008. Acessado em 24 jul. 2015 Online. Disponível em <http://www.cs.cmu.edu/~CompThink/papers/Wing08a.pdf>

LEE, I., Martin, F., Denner, J., Malyn-Smith, J., Werner, L. Computational thinking for youth in practice. **ACM Inroads**, New York, v.2, n.1, p.32-37, 2011.

Carnegie Mellon. **Center for Computational Thinking**. University of Carnegie Mellon, 2015. Acessado em 24 de jul. 2015. Online. Disponível em: <http://www.cs.cmu.edu/~CompThink/>

Google. **Exploring Computational Thinking**. Google, 2011. Acessado em 24 de jul. 2015. Online. Disponível em: <https://www.google.com/edu/resources/programs/exploring-computational-thinking/>

Google. **Computational Thinking for Educators**. Google, 2013. Acessado em 24 de jul. 2015. Online. Disponível em: <https://computationalthinkingcourse.withgoogle.com/preview/>

edX. **Introduction to Computational Thinking and Data Science**, 2015. Acessado em 24 de jul. 2015. Online. Disponível em: <https://www.edx.org/course/introduction-computational-thinking-data-mitx-6-00-2x-0>

Harvard Graduate School of Education. **Computational Thinking | Defining**, ScratchEd, 2015. Acessado em 24 de jul. 2015. Online. Disponível em: <http://scratched.gse.harvard.edu/ct/defining.html>

Massachusetts Institute of Technology Media Lab. **Scratch – Imagine, Program, Compartilhe**, Scratch, 2003. Acessado em 24 de jul. 2015. Online. Disponível em: <https://scratch.mit.edu/>

Research Kitchen. **RoboMind.net – The new way to learn programming**, 2005. Acessado em 24 de jul. 2015. Online. Disponível em: <http://www.robomind.net/pt/>

Microsoft. **What is Kodu?**, 2009. Acessado em 24 de jul. 2015. Online. Disponível em: <http://www.kodugamelab.com/about/>

British Broadcasting Corporation Education Dep. **Introduction to Computational Thinking**, 2015. Acessado em 24 de jul. 2015. Online. Disponível em: <http://www.bbc.co.uk/education/guides/zp92mp3/revision>

International Society for Technology in Education. **Operational Definition of Computational Thinking for K-12 Education**, 2010. Acessado em 24 de jul. 2015. Online. Disponível em: <http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2>

Computer Science Teacher Association, International Society for Technology in Education and National Science Foundation. **Computational Thinking Leadership Toolkit**. Computer Science Teacher Association, 2010. Curriculum. Acessado em 24 de jul. 2015. Online. Disponível em : <http://www.csta.acm.org/Curriculum/sub/CompThinking.html>

Khan Academy. **Hora do Código | Khan Academy**, 2015. Acessado em 24 de jul. 2015. Online. Disponível em: <https://pt.khanacademy.org/hourofcode>

Code.org. **Anybody can learn | Code.org**, 2015. Acessado em 24 de jul. 2015. Online. Disponível em: <https://code.org/>

Code.org. **Uma hora do Código para cada aluno**, 2013. Acessado em 24 de jul. 2015. Online. Disponível em: <http://hourofcode.com/pt/>

EHRIG, H.; HECKEL, R.; KORFF, M.; LOWE, M.; RIBEIRO, L.; WAGNER, A.; CORRADINI, A. **Handbook of graph grammars and computing by graph transformation**, River Edge, NJ, USA, v. I, p. 247 – 312, 1997.

CRAIGEN, D.; GERHART, S.; RALSTON, T. **An International Survey of Industrial Applications of Formal Methods**: Volume 1 Purpose, Approach, Analysis, and Conclusions.