



UMA ANÁLISE DAS DIRETRIZES PARA ENSINO DE PENSAMENTO COMPUTACIONAL PROPOSTAS PELA SBC NA EDUCAÇÃO BÁSICA

Nathalia da Cruz Alves (nathalia.alves@posgrad.ufsc.br), Christiane Gresse von Wangenheim (c.wangenheim@ufsc.br), Jean Carlo Rossa Hauck (jean.hauck@ufsc.br), Adriano Ferreti Borgatto (adriano.borgatto@ufsc.br), Dalton Francisco de Andrade (dalton.andrade@ufsc.br)

1) Introdução

Com a crescente relevância da computação para a sociedade, o pensamento computacional tornou-se imprescindível para um cidadão preparado para os desafios do século XXI (GROVER e PEA, 2013). Atualmente, um número crescente de países está desenvolvendo currículos de referência para guiar o ensino de pensamento computacional na Educação Básica convergindo em termos de conceitos e sequenciamento, como o *K-12 Computer Science Framework* (CSTA, 2016) nos EUA.

No Brasil, a Sociedade Brasileira de Computação (SBC) elaborou diretrizes definindo as competências e habilidades para os eixos pensamento computacional, mundo digital e cultura digital que compõem a computação na Educação Básica (SBC, 2018). Observando diferenças das diretrizes da SBC com outros currículos internacionais levantam-se questões acerca do sequenciamento de conteúdos que deve possibilitar que os alunos aprendam e sejam bem-sucedidos nessa área.

2) Método

Com o objetivo de analisar as diretrizes da SBC em relação ao sequenciamento de conteúdo foi realizada uma análise via Teoria da Resposta ao Item em 2 etapas:

- 1. Avaliação da proficiência** em pensamento computacional por meio da análise do projeto App Inventor criado pelo aluno. A avaliação é realizada de forma automatizada usando a ferramenta CodeMaster (ALVES et al., 2019).
- 2. Análise via Teoria de Resposta ao Item** dos resultados da avaliação usando o modelo de Resposta Gradual (SAMEJIMA, 1969). Assume-se que itens com bom desempenho devem ser ensinados nos estágios iniciais e itens mais difíceis em estágios mais avançados.

3) Resultados

Mapeando os resultados com os conceitos e sequência propostos pela SBC (2018) pode-se observar diferenças entre o grau da dificuldade indicado pela análise da TRI e a alocação ao longo da Educação Básica.

Pontos da escala	Escala (TRI)									
5,5	Listas b4									
5,0	Sensores b4									
4,5										
4,0										
3,5	Persistência b4									
3,0	Laços b4		Sensores b3		Laços b3		Persistência b3		Conditioais b4	
2,5	Laços b2		Laços b3		Persistência b2		Persistência b3		Conditioais b4	
2,0	Nomeação b4		Listas b3		Persistência b2		Persistência b3		Conditioais b4	
1,5	Listas b2		Abstração b3		Abstração b4		Desenho b3		Des. Anim. b4	
1,0	Strings b3		Conditioais b3		Sincronização b2		Abstração b2		Sensores b2	
0,5	Operadores b3		Operadores b4		Nomeação b3		Abstração b2		Sensores b2	
0,0	Operadores b2		Variáveis b3		Nomeação b2		Eventos b4		Desenho b2	
-0,5	Variáveis b2		Strings b2		Eventos b3				Telas b3	
-1,0										
-1,5	Eventos b2									
-2,0										
-2,5	Telas b2									

Diretrizes SBC	Itens da Rubrica CodeMaster														
	Operadores	Variáveis	Strings	Nomeação	Listas	Persistência	Eventos	Laços	Conditioais	Sincronização	Abstração	Sensores	Desenho e Animação	Mapas	Telas
Ensino Fundamental															
1 Ano	Organização de objetos														
	Algoritmo: definição														
2 Ano	Identificação de padrões de comportamento														
	Modelos de objetos														
	Algoritmos: construção e simulação														
3 Ano	Definição de problemas														
	Algoritmo: seleção														
	Introdução à lógica														
4 Ano	Estruturas de dados estáticas: registros e vetores														
	Algoritmo: repetição														
5 Ano	Estruturas de dados dinâmicas: listas e grafos														
	Algoritmos sobre estruturas dinâmicas														
6 Ano	Tipos de dados														
	Linguagem visual de programação														
	Introdução à generalização														
	Técnicas de solução de problema: decomposição														
7 Ano	Automatização														
	Estruturas de dados: registros e vetores														
	Técnicas de solução de problemas: decomposição e reuso														
	Programação: decomposição e reuso														
8 Ano	Estruturas de dados: listas														
	Programação: listas e recursão														
	Técnicas de solução de problema: recursão														
	Paralelismo														
9 Ano	Estruturas de dados: grafos e árvores														
	Programação: generalização e grafos														
	Técnica de construção de algoritmos: Generalização														

4) Considerações finais

Os resultados indicam várias oportunidades de melhoria. A partir desses resultados identifica-se a necessidade da revisão dos conceitos e da sua sequência visando uma proposta que permita aos alunos aprender computação com sucesso de forma motivadora.