

CONSTRUÇÃO DO PENSAMENTO COMPUTACIONAL ATRAVÉS DO DESENVOLVIMENTO DE GAMES

Rafael Marimon Boucinha, PGIE-UFRGS, rmboucinha@gmail.com
Christian Puhlmann Brackmann UFRGS/IFFAR, brackmann@iffarroupilha.edu.br
Dante Augusto Couto Barone, INF/PGIE - UFRGS, barone@inf.ufrgs.br
Ana Casali, Universidade Nacional de Rosário, acasali@fceia.unr.edu.ar

Resumo: O presente artigo apresenta uma proposta metodológica para construção do Pensamento Computacional através do desenvolvimento de games. Considerando que no Brasil o Pensamento Computacional não está inserido como disciplina na Educação Básica, a prática pedagógica foi desenvolvida através de um curso ofertado como atividade extraclasse, destinado a estudantes do Ensino Fundamental. Para verificar a eficiência do método de ensino-aprendizagem, os alunos foram avaliados com o Teste de Pensamento Computacional, antes e após o término do curso. Comparando os resultados obtidos observa-se que houve um incremento do Pensamento Computacional dos participantes da pesquisa, sendo este resultado estatisticamente significativo.

Palavras-chave: Pensamento Computacional. Avaliação. Metodologia de Ensino

CONSTRUCTION OF COMPUTATIONAL THINKING THROUGH THE DEVELOPMENT OF GAMES

Abstract: This paper presents a methodological proposal for the construction of Computational Thinking through the development of games. Considering that Computational Thinking is not included as a subject in Basic Education in Brazil, the pedagogical practice was developed through a course offered as an extra class activity for students of the Elementary School. To verify the efficiency of the teaching-learning method, the students were evaluated with the Computational Thinking Tests, before and after the end of the course. After comparing the results, it is observed that there was an increase in the Computational Thinking of the research participants, being this result statistically significant.

Keywords: Computational Thinking, Evaluation, Teaching Methodology

1. Introdução

Atualmente presenciamos uma onipresença da informática no cotidiano das pessoas, os computadores estão distribuídos em objetos distintos, nos quais utilizamos seu processamento mesmo sem perceber, fenômeno que Weiser (1991) denominou de computação ubíqua.

Diante deste contexto tecnológico, Prensky (2001) observou que por influência da tecnologia em seu cotidiano, os jovens teriam um novo perfil de aprendiz, à medida que estão acostumados a receber informações muito rápido, a trabalhar em mais de uma tarefa de forma paralela, gostam de estar conectados em rede e prosperam com base em gratificações instantâneas e recompensas frequentes. Para denominar esta nova geração que ingressa nas escolas "nativos" da linguagem digital dos computadores, videogames e da Internet, este autor propôs o conceito de nativos digitais.

Mitch Resnick (2012) ao discutir o conceito de nativos digitais, questiona a fluência digital destes jovens e explica que embora estes jovens tenham nascidos imersos num

ambiente tecnológico e demonstrem muita experiência e facilidade em interagir com tecnologias, não têm esta mesma habilidade para criar e se expressar utilizando novas tecnologias. Afirma que são consumidores de tecnologia e não produtores, comparando este fenômeno com alguém que sabe ler, mas não sabe escrever.

Uma visão ampliada de fluência digital, pressupõe que os alunos ultrapassem o simples domínio de TIC (Tecnologias da Informação e Comunicação), sendo necessária a compreensão de como os computadores funcionam e aprendizagem de formular problemas e expressar a sua solução de forma que um computador ou humano possam executar.

Este modo de pensar próprio das Ciência da Computação foi nomeado como Pensamento Computacional, termo que se popularizou a partir do artigo denominado “*Computational Thinking*” publicado em 2006 por Jeannette M. Wing (2006).

A necessidade de definir o conceito de Pensamento Computacional fez com que ISTE (*International Society for Technology in Education*) e CSTA (*Computer Science Teacher Association*, 2011) em colaboração com líderes do ensino superior, representantes da indústria da educação e de instituições escolares desenvolvessem uma definição operacional de Pensamento Computacional. De acordo com esta definição, Pensamento Computacional é um processo de resolução de problemas que inclui (mas não se limitam a) as seguintes características:

- Formulação de problemas de uma forma que nos permite usar um computador e outras ferramentas para ajudar a resolvê-los;
- Coleta e análise de dados;
- Representar dados através de abstrações como modelos e simulações
- Soluções automatizadas através do pensamento algorítmico (uma série de passos ordenados);
- Identificar, analisar e implementar soluções possíveis com o objetivo de alcançar a combinação mais eficiente e eficaz de recursos e passos;
- Generalizar e transferindo este processo de resolução de problemas para uma grande variedade de problemas.

O Pensamento Computacional tem sido comparado com a alfabetização do século XXI e possuir habilidades básicas de codificação tornou-se algo necessário para muitos postos de trabalho. Estudos da *European Commission* (2015) estimam que 90% das ocupações profissionais hoje em dia requerem competências digitais, incluindo programação. Em 2015, mais de 600.000 empregos na área de tecnologia com altos salários permaneceram vagos Estados Unidos. Para o ano 2018, é projetado que 51 por cento de todos os empregos serão em campos relacionados à Ciência da Computação (SMITH, 2016).

No Brasil a situação é semelhante, pois no último estudo publicado por Villela (2013), onde analisou indicadores de mão de obra em TI no Brasil, com base nos resultados obtidos foi projetado um déficit de cerca de 408 mil profissionais em 2022, com uma demanda estimada de cerca de 1,669 milhões contra 1,261 milhões efetivamente contratados. Comparando com estudos anteriores, verificamos um crescimento deste déficit, o qual, segundo o autor, é um dos principais obstáculos ao crescimento da Indústria Brasileira de Software e Serviços de TI.

A previsão desta carência de profissionais e o entendimento de que a Ciência da Computação (CC) é uma nova habilidade necessária para as oportunidades econômicas e a mobilidade social, fez com que o governo estadunidense destinasse 4 bilhões de dólares em financiamento para estados e 100 milhões de dólares diretamente para os

distritos escolares em seu próximo orçamento para expandir o ensino de CC nas escolas através da formação de professores, ampliando o acesso a materiais de ensino de alta qualidade e construção de parcerias regionais (Smith, 2016).

Grandes organizações têm sido incentivadas pelo governo americano a auxiliarem no ensino da Ciência da Computação em escolas. O Google, por exemplo, possui uma divisão denominada “Google na Educação”, a qual tem investido um montante considerável em programas e projetos para incentivar a introdução dos conceitos da CC e estimular a prática do Pensamento Computacional com estudantes da Educação Média.

Na Europa o ensino de Ciência da Computação já está integrado no currículo da Educação Básica de 15 países: Áustria, Bulgária, República Tcheca, Dinamarca, Estônia, França, Hungria, Irlanda, Lituânia, Malta, Espanha, Polônia, Portugal, Eslováquia e Inglaterra (Balanskat et. al., 2015).

Além dos financiamentos governamentais e dos programas de grandes empresas, observamos que têm surgido inúmeras iniciativas fora do ambiente escolar, tais como Code.org¹, Codecademy², CoderDojo³, Girls Who⁴ e Black Girls Code⁵, as quais têm auxiliado a popularizar discussões em torno da necessidade de introduzir Ciência da Computação no cotidiano dos adolescentes.

Acredita-se que o Pensamento Computacional não é uma habilidade exclusivamente relacionada ao curso Superior em CC e que o processo de resolução de problemas utilizados na CC possa ser generalizado e transferido para uma ampla variedade de problemas da vida cotidiana, embora não esteja cientificamente comprovado.

Cabe destacar que, no Brasil, o ensino de Pensamento Computacional não está estabelecido como parte do currículo do Ensino Básico (Brackmann, 2016) e as produções científicas evidenciam que este tema tem sido trabalhado em práticas pontuais sem avaliar seus efeitos.

Observa-se que são raros os estudos que utilizam instrumentos para mensurar o Pensamento Computacional e praticamente não existem pesquisas avaliando a eficácia das propostas pedagógicas. Diante disso, este trabalho pretende apresentar uma metodologia para ensino de Pensamento Computacional através de desenvolvimento de jogos, bem como a avaliação de sua eficácia.

2. Construção da metodologia de ensino-aprendizagem

Considerando que este tema não faz parte do currículo das escolas brasileiras, foi necessário adaptar atividades para nossa realidade e estruturar a prática pedagógica para ofertar um curso como atividade extraclasse. Para despertar o interesse dos alunos foi elaborado um curso denominado “Desenvolvimento de Games”, onde foram inseridos conceitos e práticas computacionais através da criação de pequenos jogos com Scratch.

O curso foi organizado e ministrado em 10 horas aulas, distribuídas em 5 encontros semanais de 2 horas cada. Com intuito de viabilizar a aprendizagem neste curto período de tempo, foi elaborado um plano de aula considerando os conhecimentos prévios necessários para cada lição.

¹ <https://www.code.org>

² <https://www.codecademy.com/pt>

³ <https://coderdojo.com>

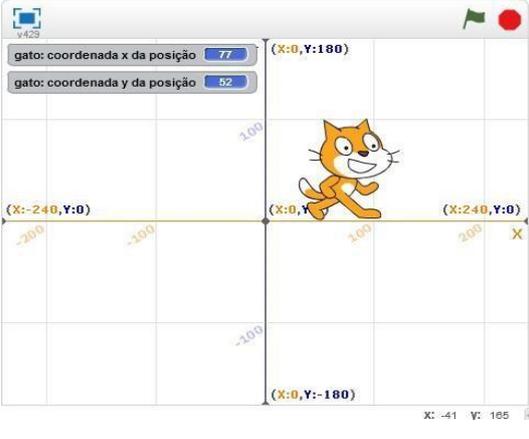
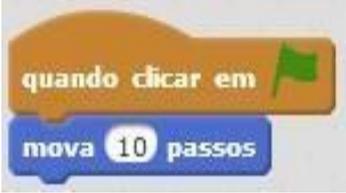
⁴ <https://girlswhocode.com>

⁵ <http://www.blackgirlscode.com/>

Conforme Ausubel et. al. (1980) para a aprendizagem ser significativa é necessário que o aluno esteja motivado, que o material seja potencialmente significativo e que possua os subsunçores necessários para aprender o conteúdo. Para Teoria da Aprendizagem Significativa, o subsunçor é um conhecimento existente na estrutura de conhecimentos do indivíduo e que permite dar significado a um novo conhecimento que lhe é apresentado ou por ele descoberto. Estes organizadores prévios são conhecimentos que têm a função de facilitar a aprendizagem sobre um domínio que pode ser completamente desconhecido, servindo como uma espécie de ponte cognitiva entre o que o aprendiz já sabe e o que ele deve saber.

Para a aprendizagem ser significativa, é importante que o conteúdo a ser aprendido possa se ligar a conceitos (subsunçores relevantes existentes na estrutura cognitiva), por este motivo o conceito de organizadores prévios tem especial importância nesta teoria. Para Ausubel (1980), os organizadores prévios verdadeiros são aqueles destinados a facilitar a aprendizagem significativa de tópicos específicos, ou ideias estreitamente relacionadas. Na Figura 1 podemos observar um exemplo de como esta teoria foi utilizada para o planejamento de cada aula.

Figura 1 - Subsunçor x Objetivo de aprendizagem

Subsunçor: Compreensão do plano cartesiano	Objetivo de aprendizagem: Aprender a movimentar objetos no palco do Scratch
	

Fonte: desenvolvido pelos autores.

Outro pressuposto teórico que norteou o planejamento das aulas foi a Teoria da Aprendizagem Experiencial de Kolb (1984), a qual descreve como a experiência é transformada em aprendizagem através de um ciclo de aprendizagem envolvendo experimentar, refletir, pensar e agir. Para Kolb (1984), aprender com a experiência é um processo de construção de conhecimento que envolve uma tensão criativa entre quatro modos de aprendizagem. Este processo é retratado como um ciclo de aprendizagem idealizado ou em espiral em que o aluno percorre as quatro fases:

- 1) Experiência Concreta: relacionada às experiências pessoais e os sentimentos envolvidos na situação de aprendizagem;
- 2) Observação Reflexiva: implica na solução de problemas revisando e refletindo sobre a experiência;

- 3) Conceituação Abstrata: o entendimento é baseado na compreensão intelectual de uma situação, ou seja, nas conclusões construídas com base na experiência, sendo o nível de abstração bastante elevado;
- 4) Experimentação Ativa: envolve a aprendizagem de forma ativa, onde os alunos planejam novas experiências, modificam variáveis e influenciam situações, para experimentar o que aprendeu e formular hipóteses.

Todas as lições do curso proposto por este projeto, foram planejadas para iniciar buscando resgatar experiências passadas dos alunos, para encontrar subsunçores e gerando discussões de experiências fora do contexto computacional. Posteriormente, os alunos interagem com fragmentos de código. No momento posterior eram indagados sobre o funcionamento do código e testavam modificações na programação. Por último, cada lição finalizava com desafios sobre o tópico abordado no Scratch. O Quadro 1 apresenta uma síntese de cada lição do Curso de Desenvolvimento de Games exemplificando a articulação dos subsunçores, experiências e objetivos de aprendizagem.

Quadro 1: Síntese do plano de Aula do Curso de Desenvolvimento de Games

Aula	Atividade	Descrição da Atividade	Conceitos/competências
1	Pré-teste	Teste de Pensamento Computacional	
	Interface do Scratch e Movimentação	Subsunçor: Conhecimento do Plano Cartesiano Experiência: Jogo Batalha Naval Objetivo: Movimentar atores no palco	Sequência, Evento
2	Repetir, Animar, Rotar	Subsunçor: rotação de figuras (direções), animação Experiência: Animação com série de desenhos Objetivo: Animar atores e utilizar laços de repetição	Iteração (ciclos), Gestão de eventos
	Condicionais	Subsunçor: Decisão com base em critério Experiência: Quando parar na Sinaleira Objetivo: utilizar condicional	Instruções condicionais
3	Variáveis e condicionais	Subsunçor: Conceito de variável Experiência: Placar de jogo de futebol Objetivo: Utilizar condições para controlar o valor de uma variável	Variáveis, Números aleatórios
	Condicional composto	Subsunçor: Mensagem diferente de acordo com critério Experiência: Boletim escolar (aprovado/reprovado) Objetivo: Utilizar múltiplas variáveis para controle do jogo	Instruções condicionais
4	Depuração	Subsunçor: conceito de condicional Objetivo: analisar o código e concluir a programação de terceiro	Identificação e eliminação de erros
	Clones	Subsunçor: Conceito de Clone Experiência: Fotocópia Objetivo: Uso de clones e alteração do tamanho dos atores	Clone

	Exemplos de Games	Subsuncor: Navegação no repositório do Scratch Experiência: Navegar no estúdio do curso Objetivo: Explorar projetos e reaproveitar código	Reutilização de código
5	Projeto Próprio	Objetivo: Desenvolver projeto com base em códigos explorados em sala de aula e pesquisa no Scratch	Compartilhamento
	Teste	Teste de Pensamento Computacional	

Fonte: desenvolvido pelos autores

Seguindo o exemplo do trabalho de Kologeski et. al. (2016) para motivar os alunos na realização de atividades, bem como sua permanência do curso, foi explicado que eles receberiam um certificado de conclusão, o qual foi entregue no encerramento do curso.

3. Metodologia

O presente estudo pode ser classificado como uma pesquisa quantitativa, porque recorre à linguagem matemática para descrever as causas de um fenômeno e as relações entre variáveis. Fonseca (2002, p. 20) explica que este tipo de pesquisa está centrado na objetividade e “influenciada pelo positivismo, considera que a realidade só pode ser compreendida com base na análise de dados brutos, recolhidos com o auxílio de instrumentos padronizados e neutros”. Em relação à natureza do estudo, este pode ser considerado uma pesquisa aplicada, tendo em vista que objetiva gerar conhecimentos para aplicação prática, dirigidos à solução de problemas específicos e envolve interesses locais.

Considerando os objetivos propostos, este estudo trata de uma pesquisa explicativa, a qual segundo Gil (2010) tem como preocupação central identificar os fatores que determinam ou que contribuem para a ocorrência dos fenômenos. Este é o tipo de pesquisa que mais aprofunda o conhecimento da realidade, porque explica a razão e o porquê das coisas.

A presente pesquisa pode ser classificada como um estudo quantitativo quase experimental, pois utilizou procedimentos estatístico, porém sem utilização de grupo de controle.

3.1. Amostra

Conforme Rudio (2007, p.31), uma pesquisa não é feita com todos os elementos que compõe uma população. Costuma-se neste caso, selecionar uma parte representativa dela, denominada “amostra”. Tendo em vista que os participantes do estudo foram convidados e adesão ocorria conforme interesse, a amostra foi intencional e não probabilística, uma vez que não foi escolhida de forma aleatória.

Para garantir que os alunos tivessem com o desenvolvimento do pensamento formal necessário para o acompanhamento das lições, o curso foi disponibilizado apenas para estudantes de ensino fundamental que estavam cursando a partir do sexto ano.

Matricularam-se neste curso 38 alunos pertencentes a 2 escolas particulares da cidade de Porto Alegre, RS. Considerando a livre adesão ao curso e a possibilidade de desistir em qualquer momento da pesquisa, 31 alunos concluíram o curso, sendo estes os sujeitos da pesquisa.

3.2. Instrumento

Para avaliar o Pensamento Computacional, os alunos realizaram um teste desenvolvido por Roman et. al. (2015), antes e após o término do curso. Este teste é composto por 28 itens com 4 alternativas de resposta sendo apenas uma alternativa correta. O tempo máximo para conclusão do teste é de 45 minutos.

Considerando que o Teste do Pensamento Computacional (TPC) foi desenvolvido em Espanhol, foi solicitada a autorização do autor para realizar sua tradução e posteriormente utilizá-lo nesta pesquisa. Para este teste, o Pensamento Computacional é a capacidade de formular e resolver problemas, baseando-se nos conceitos fundamentais da Computação e usando sintaxe-lógica de linguagens de programação de computador (e.g. sequências básicas, loops, iteração, condicionais, funções e variáveis). Na Figura 2 apresenta-se um modelo de questão utilizado no TPC.

Figura 2: Exemplo de questão do Teste de Pensamento Computacional

Fonte: Roman et. al. (2015), traduzido pelos autores

Os itens deste teste, estão alinhados com normas estabelecidas pela CSTA (Roman, 2015) para a ensino de CC no ensino fundamental, sendo dispostos em dificuldade crescente. Cada item aborda um ou mais dos setes conceitos computacionais a seguir:

- Direções básicas (4 itens);
- Loop: Repetir uma quantidade de vezes (4 itens);
- Loop: Repetir até uma condição (4 itens);
- Condicional simples: If (4 itens);
- Condicional composto: If/Else (4 itens);
- Enquanto: While (4 itens);
- Funções Simples: Functions (4 itens).

4. Análise dos dados

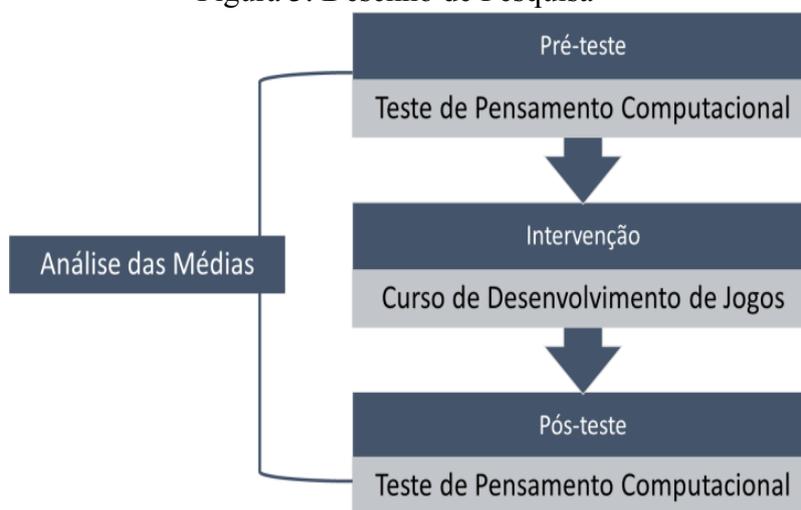
Para verificar se a metodologia utilizada no Curso de Desenvolvimento de Games propiciou um incremento no Pensamento Computacional dos alunos, foram formuladas as seguintes hipóteses:

- H0: A metodologia utilizada no Curso de Desenvolvimento de Games não propicia incremento do Pensamento Computacional;

- H1: A metodologia utilizada no Curso de Desenvolvimento de Games propicia incremento do Pensamento Computacional.

A fim de testar a hipótese nula, foi realizado o TPC antes do curso e após o término das aulas, resultando num desenho de pesquisa como apresentado na Figura 3.

Figura 3: Desenho de Pesquisa



Fonte: Desenvolvido pelos autores

Os resultados obtidos com este instrumento foram submetidos a procedimentos estatísticos a fim de testar a hipótese nula, ou seja, avaliar se houve diferença entre os resultados do pré-teste e pós-teste e se esta diferença era significativa em termos estatísticos. No presente estudo, a média dos alunos no pré-teste e pós-teste foram comparadas com o Teste T-Student pareado, considerando o nível de confiança de 95%. O objetivo deste teste foi verificar se a diferença entre as médias era realmente significativa ou apenas fruto de variações aleatórias. Observou-se que a média de acertos aumentou na segunda testagem, passando de 16,03 para 17,87. A Tabela 1 apresenta os resultados do teste para as duas amostras pareadas.

Tabela 1: Teste-t duas amostras em par para médias

	<i>Pré-teste</i>	<i>Pós-teste</i>
Média	16,03333	17,86667
Variância	20,86092	15,91264
Observações	30	30
Correlação de Pearson	0,66267	
Hipótese da diferença de média	0	
gl	29	
Stat t	-2,82593	
P(T<=t) uni-caudal	0,004224	
t crítico uni-caudal	1,699127	
P(T<=t) bi-caudal	0,008448	
t crítico bi-caudal	2,04523	

Fonte: Elaborado pelos autores

Considerando que o valor p relacionado com o Teste T foi menor do que o nível de significância ($<0,05$), a hipótese nula é rejeitada, demonstrando uma diferença estatisticamente significativa entre o resultado da primeira e segunda testagem. Desta

forma, as evidências estatísticas indicam que a proposta pedagógica construída neste estudo permite um incremento do Pensamento Computacional dos alunos.

5. Conclusões e Trabalhos Futuros

Este artigo apresentou uma proposta pedagógica para a construção do Pensamento Computacional através do Desenvolvimento de Games. De acordo com os resultados deste estudo, foi possível validar a presente metodologia para utilização em sala de aula na educação básica, uma vez que a avaliação do Pensamento Computacional foi significativamente superior após a realização do curso.

Conclui-se que, mesmo intervenções breves, como o curso utilizado nesta pesquisa, quando elaboradas com uma metodologia adequada, podem produzir efeitos relevantes para seus participantes. Desta forma, espera-se que esta pesquisa incentive práticas de ensino do Pensamento Computacional no Brasil.

Considera-se interessante a ampliação da amostra, com intuito de verificar se os resultados desta pesquisa serão confirmados em outros grupos de alunos, trabalho que já está sendo realizado pelos autores.

Com intuito de facilitar a reprodução da metodologia proposta, estuda-se a possibilidade de elaboração de um manual descrevendo de maneira mais completa o plano de aula aqui utilizado.

Para atingir um maior número de alunos, sugere-se como trabalhos futuros a tentativa de reprodução desta experiência utilizando um Ambiente Virtual de Aprendizagem, sendo necessário remodelar este curso para educação a distância.

Referências

AUSUBEL, David P., NOVAK, Joseph D., HANESIAN, Helen. Psicologia educacional. Tradução Eva Nick. Rio de Janeiro: Interamericana, 1980

BALANSKAT, A.; ENGELHARDT, K. Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe. , out. 2015. European Schoolnet. Disponível em: <http://www.eun.org/c/document_library/get_file?uuid=3596b121-941c-4296-a7600f4e4795d6fa&groupId=43887>. Acesso em: 1/12/2015.

BRACKMANN, C.; BARONE, D.; CASALI, A.; BOUCINHA, R.; MUNOZ-HERNANDEZ, S. Computational thinking: Panorama of the Americas. . p.1–6, 2016. IEEE. Disponível em: <<http://ieeexplore.ieee.org/document/7751839/>>. Acesso em: 21/5/2017.

CSTA. K–12 Computer Science Standards - Revised 2011 - The CSTA Standards Task Force. [s.l.]: Association for Computing Machinery, 2011.

FONSECA, J. J. S. Metodologia da pesquisa científica. Fortaleza: UEC, 2002. Apostila

GIL, A. C. Como elaborar projetos de pesquisa. 3. ed. São Paulo: Atlas, 2010.

KOLB, D. A. (1984). Experiential learning: Experience as the source of learning and development. Englewood Cliffs, NJ: Prentice-Hall

KOLOGESKI, A.; SILVA, C.; BARBOSA, D.; MATTOS, R.; MIORELLI, S. Desenvolvendo o Raciocínio Lógico e o Pensamento Computacional: Experiências no Contexto do Projeto Logicando. Revista Novas Tecnologias na Educação (RENOTE), v. 14, n. 2, 2016.

- PRENSKY, Mark. Digital game-based learning. New York: McGraw-Hill, 2001
- ROMÁN, M. (2015). Test de Pensamiento Computacional [Test en línea]. Recuperado de <http://goo.gl/IYEKMB>
- RUDIO, F. V. Introdução ao projeto de pesquisa científica. Petropolis: Vozes, 2007.
- VILLELA, Paulo R. C. Escassez de mão de obra em ti: uma perspectiva quantitativa. In: Cadernos temáticos do observatório: mercado de trabalho e formação de mão de obra em TI. SOTEX, 2013. Disponível em: <http://www.softex.br/wpcontent/uploads/2015/10/Cadernos_Tematicos_Mercado_d_e_trabalho.pdf> Acesso em: 05 de junho de 2016.
- WEISER, M. The computer for the 21 st century, Scientific America, v.265, nº 3, p. 6675, 1991.
- WING, J. M. 2006 Computational thinking. *Commun. ACM* 49, 33–35.