



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
CENTRO INTERDISCIPLINAR DE NOVAS TECNOLOGIAS NA EDUCAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA NA EDUCAÇÃO

CHRISTIAN PUHLMANN BRACKMANN

DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL
ATRAVÉS DE ATIVIDADES DESPLUGADAS NA EDUCAÇÃO BÁSICA

PORTO ALEGRE

2017

DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL
ATRAVÉS DE ATIVIDADES DESPLUGADAS NA EDUCAÇÃO BÁSICA

CHRISTIAN PUHLMANN BRACKMANN

Tese apresentada ao Programa de Pós-Graduação em Informática na Educação (PPGIE) do Centro Interdisciplinar de Novas Tecnologias na Educação (CINTED) da Universidade Federal do Rio Grande do Sul (UFRGS), como requisito para obtenção do título de Doutor em Informática na Educação.

Orientador: Dante Augusto Couto Barone

Co-orientadora: Ana Casali

Linha de pesquisa: Paradigmas para a Pesquisa sobre o Ensino Científico e Tecnológico

CIP - Catalogação na Publicação

BRACKMANN, CHRISTIAN PUHLMANN
DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL
ATRAVÉS DE ATIVIDADES DESPLUGADAS NA EDUCAÇÃO BÁSICA
/ CHRISTIAN PUHLMANN BRACKMANN. -- 2017.

226 f.

Orientador: Dante Augusto Couto Barone.

Coorientadora: Ana Casali.

Tese (Doutorado) -- Universidade Federal do Rio Grande do Sul, Centro de Estudos Interdisciplinares em Novas Tecnologias na Educação, Programa de PósGraduação em Informática na Educação, Porto Alegre, BRS, 2017.

1. Pensamento Computacional. 2. Educação Básica.
3. Computação nas Escolas. 4. Avaliação. I. Couto Barone, Dante Augusto, orient. II. Casali, Ana, coorient. III. Título.



SERVIÇO PÚBLICO FEDERAL
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
CENTRO INTERDISCIPLINAR DE NOVAS TECNOLOGIAS NA EDUCAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA NA EDUCAÇÃO

**ATA SOBRE A DEFESA DE TESE DE DOUTORADO
CHRISTIAN PUHLMANN BRACKMANN**

Às quatorze horas do dia vinte e dois de agosto de dois mil e dezessete, na sala 329 do PPGIE/CINTED, nesta Universidade, reuniu-se a Comissão de Avaliação, composta pelos Professores Doutores: Rosa Maria Vicari, Daltro José Nunes, André Luis Alice Raabe e Evandro Manara Miletto, para a análise da defesa de Tese de Doutorado intitulada “*Desenvolvimento do Pensamento Computacional através de Atividades Desplugadas na Educação Básica*”, do doutorando do Programa de Pós-Graduação em Informática na Educação Christian Puhlmann Brackmann, sob a orientação do Prof. Dr. Dante Augusto Couto Barone e coorientação da Profª. Drª. Ana Casali.

A Banca, reunida, após a apresentação e arguição, emite o parecer abaixo assinalado.

Considera a Tese aprovada

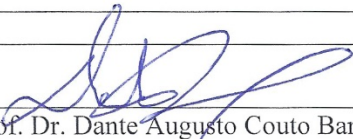
sem alterações;

e recomenda que sejam efetuadas as reformulações e atendidas as sugestões contidas nos pareceres individuais dos membros da Banca;

Considera a Tese reprovada.

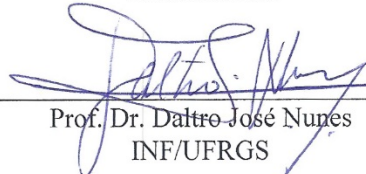
Considerações adicionais (a critério da Banca):

A banca concorda na aprovação da tese por sua pertinência e contribuição na área do pensamento computacional e sugere que as sugestões e recomendações feitas durante a arguição sejam contempladas no texto final, para homologação.



Prof. Dr. Dante Augusto Couto Barone
Presidente e Orientador

(videoconferência)
Profª. Drª. Ana Casali
Coorientadora

(videoconferência)
Profª. Drª. Rosa Maria Vicari
PPGIE/UFRGS


Prof. Dr. Daltro José Nunes
INF/UFRGS

(videoconferência)
Prof. Dr. André Luis Alice Raabe
UNIVALI


Prof. Dr. Evandro Manara Miletto
IFRS

“Pensamento Computacional é uma habilidade fundamental para todos, não apenas para cientistas da Computação. Além de aprender a ler, escrever e calcular, deveríamos adicionar Pensamento Computacional na capacidade analítica de cada criança”

- Jeannette Wing

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Profa. Jane Fraga Tutikian

Pró-Reitor de Pós-Graduação: Prof. Celso Giannetti Loureiro Chaves

Diretor do CINTED: Prof. Leandro Krug Wives

Coordenadora do PPGIE: Profa. Liane Margarida Rockenbach Tarouco

IN MEMORIAM

Rodrigo de Brum Chimainski, um amigo para todas as horas, sempre sorridente e companheiro.

Iedo, meu segundo pai, exemplo de pessoa que deixou um vazio e saudade imensa.

AGRADECIMENTOS

Agradeço a todas as pessoas que, de uma forma ou de outra, contribuíram para este trabalho, em especial meus pais, minha esposa e minha filha que, muitas vezes serviu como ponto de estímulo.

Minha sincera gratidão à Agência Executiva de Educação, Audiovisual e Cultura da Comissão Europeia (EACEA / Programa SMART²) que permitiu a execução de parte da pesquisa em solo Espanhol na Universidad Politécnica de Madrid (UPM), onde recebi pleno apoio da equipe composta pela professora Dr. Susana Muñoz Hernández e pessoas muito pacientes que foram essenciais nesse processo, como Julio González Cuñado e Ascención Gutiérrez de la Solana Hernández.

Não posso deixar de reconhecer e demonstrar minha gratidão a pessoas como Marcos Román-González, criador do Teste de Pensamento Computacional que também me recebeu na Espanha de maneira ímpar, junto com seus colegas Gregorio Robles, Jesús Moreno-León e Yucnary Torres. No Brasil, ao meu colega Rafael Boucinha que me apoiou em diversos pontos de minha pesquisa.

Por fim, obrigado às empresas Maurício de Souza Produções e Thinkfun Inc., que acreditaram na pesquisa e serviram como apoio, consulta e inspiração para a pesquisa.

RESUMO

Computadores impactam em quase todos os aspectos de nossas vidas, porém as escolas não conseguem acompanhar esse caminho sem volta. A simples utilização massiva de aparatos tecnológicos na sala de aula não garante a melhoria do ensino, porém pode ser o meio pelo qual os estudantes encontram alternativas para a solução de problemas complexos. O Pensamento Computacional é uma abordagem de ensino que usa diversas técnicas oriundas da Ciência da Computação e vem gerando um novo foco educacional no quesito inovação nas escolas mundiais como um conjunto de competências de solução de problemas que devem ser compreendidos por uma nova geração de estudantes em conjunto com as novas competências do século 21 (*i.e.*, pensamento crítico, colaboração, etc.). Até o momento, não há um consenso de metodologia de ensino e disponibilidade de material para atender as expectativas dos professores. Para auxiliar sanar essa incerteza, esta pesquisa tem como objetivo a verificação da possibilidade de desenvolver o Pensamento Computacional na Educação Básica utilizando exclusivamente atividades desplugadas (sem o uso de computadores) em estudantes da educação primária para que crianças em regiões/escolas onde não há computadores/dispositivos eletrônicos, Internet e até mesmo energia elétrica também possam se beneficiar desse método. Os resultados obtidos através de uma abordagem Quase-Experimental em escolas Espanholas e Brasileiras, apresentam dados estatísticos que apontam uma melhoria significativa no desempenho dos estudantes que tiveram atividades de Pensamento Computacional Desplugado em ambos os países.

Palavras-chave: Pensamento Computacional Desplugado, Computação na Escola, Avaliação, Educação Primária

ABSTRACT

Computational thinking is nowadays being widely adopted and investigated. Educators and researchers are using two main approaches to teach these skills in schools: with computer programming exercises, and with unplugged activities that do not require the use of digital devices or any kind of specific hardware. While the former is the mainstream approach, the latter is especially important for schools that do not count with proper technology resources, Internet connections or even electrical power. However, there is a lack of investigations that prove the effectiveness of the unplugged activities in the development of computational thinking skills, particularly in primary schools. This paper, which summarizes a quasi-experiment carried out in two primary schools in Spain and Brazil, tries to shed some light on this regard. The results show that students in the experimental groups, who took part in the unplugged activities, enhanced their computational thinking skills significantly more than their peers in the control groups who did not participate during the classes, proving that the unplugged approach is effective for the development of this ability.

Keywords: Computational Thinking Unplugged, Evaluation, Computers in Education, Primary School, Computational Thinking Test, Assessment

LISTA DE ABREVIATURAS E SIGLAS

AD – Alfabetização Digital

Art. – Artigo

BNCC – Base Nacional Curricular Comum

CC – Ciência da Computação

CD – Competências Digitais

CFE – Consejo Federal de Educación

CINTED – Centro Interdisciplinar de Novas Tecnologias na Educação

CP – Computational Thinking

CS – Computer Science

CSTA – Computer Science Teacher Association

DVD – Digital Video Disc

e.g. – exempli gratia (por exemplo)

i.e. – id est (isto é)

ICTs – Information and Communications Technology

LDB – Lei de Diretrizes e Bases da Educação Nacional

MEC – Ministério da Educação

NSF – National Science Foundation

NTIC – Novas Tecnologias da Informação e Comunicação

OECD – Organization for Economic Co-operation and Development

PC – Pensamento Computacional

PISA – Programme for International Student Assessment

PPGIE – Programa de Pós-Graduação em Informática na Educação

SMART² – SmartCities & SmartGrids for Sustainable Development Erasmus Mundus

TI – Tecnologia de Informação

TICs – Tecnologias da Informação e Comunicação

TPC – Teste de Pensamento Computacional

UFRGS – Universidade Federal do Rio Grande do Sul

LISTA DE FIGURAS

Figura 1: Visão do grupo de cientistas a partir do início da década de 1980	25
Figura 2: Movimento das empresas de maior valor de mercado nos EUA (em dólares).....	26
Figura 3: Nuvem de palavras gerada a partir da análise de 125 artigos	28
Figura 4: Relacionamento entre áreas e seus contextos	30
Figura 5: Representação estrutural das competências relacionadas	31
Figura 6: Os Quatro Pilares do Pensamento Computacional	33
Figura 7: Os quatro pilares do Pensamento Computacional segundo BBC	34
Figura 8: Partes da Bicicleta (Decomposição)	35
Figura 9: Similaridade entre raças de cachorros (Reconhecimento de Padrões).....	36
Figura 10: Padrão e replicações (Reconhecimento de Padrões).....	37
Figura 11: Sequência de Fibonacci.....	38
Figura 12: O mapa do metrô como exemplo de abstração do mundo real	39
Figura 13: Conta de soma "armada"	41
Figura 14: Exemplo de atividade de desenho.....	50
Figura 15: Baralho para ensino de números binários	51
Figura 16: Tabuleiro e cartas do jogo <i>Haathi Mera Saathi (My Elephant Friend)</i>	53
Figura 17: Tabuleiro de Programação do Jogo <i>Haathi Mera Saathi (My Elephant Friend)</i>	53
Figura 18: Jogo Robot Rally (edição 2015).....	54
Figura 19: Jogo Bits & Bytes	55
Figura 20: Jogo Robot Turtles	56
Figura 21: Execução de um programa no jogo Littlecodr.....	57
Figura 22: Jogo Giggle Chips (frente).....	58
Figura 23: Jogo Giggle Chips (verso)	58
Figura 24: Chocolate Fix	59
Figura 25: Jogo Circuit Maze	59
Figura 26: Jogo Laser Maze	60
Figura 27: Jogo Code Master (tabuleiros e peças)	60
Figura 28: Algumas soluções do jogo Code Master.....	61
Figura 29: Jogo Code Monkey Island.....	61
Figura 30: Cartas do jogo CodingIsGood.....	62
Figura 31: Tabuleiro e cartas do jogo CodingFarmer	63
Figura 32: Fichas do Personagem e Pistas	64
Figura 33: Exemplo de condicional na história de Sherlock Holmes.....	64
Figura 34: Exemplo de condicional utilizando pistas na história de Sherlock Holmes	65
Figura 35: Exemplo do livro <i>Lift-the-Flap Computers and Coding</i>	65
Figura 36: Exemplo do Guia Mangá de Banco de Dados	66
Figura 37: Peças do livro <i>CodyRoby</i>	67
Figura 38: Teste inicial e final de PC	75
Figura 39: Princípios da Computação (Ensino Fundamental).....	79
Figura 40: Estados que adotaram a Computação em seu currículo.....	84
Figura 41: Modelo de integração do Pensamento Computacional	91
Figura 42: Integração do Pensamento Computacional no currículo	92
Figura 43: Implicações políticas e práticas na introdução do PC na Educação Básica.....	95

Figura 44: Comparativo entre duas gerações de algoritmos	107
Figura 45: Histórico de pontuações do Brasil no PISA.....	109
Figura 46: Comparativo de pontuações do Brasil em relação à média mundial	109
Figura 47: Etapas do primeiro projeto piloto	114
Figura 48: Instrumento avaliativo incipiente.....	115
Figura 49: Exemplo I de questão do Teste de Pensamento Computacional.....	120
Figura 50: Exemplo II de questão do Teste de Pensamento Computacional	120
Figura 51: Exemplo III de questão do Teste de Pensamento Computacional.....	121
Figura 52: Etapas do segundo projeto piloto	128
Figura 53: Exemplo de atividade traduzida.....	132
Figura 54: Personagens da Turma da Mônica em Espanhol	132
Figura 55: Música “Um Elefante se Equilibrava” em Espanhol (europeu).....	132
Figura 56: Registro da entrega das cartas nas escolas	134
Figura 57: Escolas da região de Carabanchel.....	134
Figura 58: Diagrama das etapas do projeto na Espanha.....	135
Figura 59: Laboratório do Colégio República de Ecuador (aplicação do teste).....	135
Figura 60: Laboratório do Colégio Lope de Vega (aplicação do teste).....	136
Figura 61: Sala da aula do Colégio República de Ecuador (intervenções)	136
Figura 62: Sala da aula do Colégio Lope de Vega (intervenções)	137
Figura 63: Diagrama das etapas do projeto no Brasil.....	140
Figura 64: Laboratório do Colégio Marieta D’Ambrósio (aplicação do teste)	140
Figura 65: Laboratório do Colégio Paulo Lauda (aplicação do teste).....	141
Figura 66: Sala da aula do Colégio Marieta D’Ambrósio (intervenções).....	141
Figura 67: Sala da aula do Colégio Paulo Lauda (intervenções).....	142
Figura 68: Fluxo de testes (equivalência dos grupo).....	143
Figura 69: Fluxo de testes (mudança de desempenho significativa).....	144
Figura 70: Exemplo de exercício em Scratch (Desenhar um quadrado).....	153

LISTA DE TABELAS

Tabela 1: Instituições homologadas para aplicação das provas GCSE e GCE	90
Tabela 2: Objetivos da introdução do PC no currículo da Educação Básica	93
Tabela 3: Participantes da “Hora do Código”	97
Tabela 4: Quadro Comparativo do PC no Mundo.....	103
Tabela 5: Resultados do primeiro piloto.....	117
Tabela 6: Distribuição dos sujeitos participantes da pesquisa.....	128
Tabela 7: Resultado do segundo projeto piloto	130
Tabela 8: Teste de Normalidade do Piloto 2	130
Tabela 9: Teste T de Student - Duas amostras em par para médias (Grupo Experimental)...	130
Tabela 10: Distribuição dos sujeitos participantes da pesquisa na Espanha	138
Tabela 11: Total de matrículas no Ensino Fundamental na rede pública e privada	139
Tabela 12: Distribuição dos sujeitos participantes da pesquisa no Brasil	142
Tabela 13: Comparação de Perfil (todos sujeitos).....	144
Tabela 14: Tabela com resultados geral	145
Tabela 15: Comparação de Perfil dos quintos anos (ESP-CON-5 e ESP-EXP-5).....	148
Tabela 16: Resultados do TPC (ESP-CON-5 e ESP-EXP-5).....	148
Tabela 17: Comparação de Perfil (ESP-CON-6 e ESP-EXP-6).....	149
Tabela 18: Resultados do TPC (ESP-CON-6 e ESP-EXP-6).....	150
Tabela 19: Comparação de perfil dos quintos e sextos anos (ESP-CON e ESP-EXP)	151
Tabela 20: Resultados do TPC dos quintos e sextos anos (ESP-CON e ESP-EXP).....	151
Tabela 21: Comparação de perfil do quinto ano (BRA-CON-5 e BRA-EXP-5)	155
Tabela 22: Resultados do TPC do quinto ano (BRA-CON-5 e BRA-EXP-5).....	155
Tabela 23: Comparação de Perfil (BRA-CON-6 e BRA-EXP-6).....	157
Tabela 24: Resultados do TPC do sexto ano (BRA-CON-6 e BRA-EXP-6).....	157
Tabela 25: Comparação de perfil (BRA-CON e BRA-EXP).....	158
Tabela 26: Resultados do TPC (BRA-CON e BRA-EXP).....	159
Tabela 27: Comparação de perfil em pré-testes (ESP-EXP e BRA-EXP).....	160
Tabela 28: Comparação de perfil em pós-testes (ESP-EXP e BRA-EXP).....	161
Tabela 29: Comparativo de médias entre gêneros (todos grupos).....	162
Tabela 30: Comparativo de médias entre gêneros dos grupos experimentais (por país)	162

LISTA DE QUADROS

Quadro 1: Sugestões de inserção do PC nas disciplinas de Matemática e Ciências	48
Quadro 2: Sugestões de inserção do PC nas disciplinas de Estudos Sociais, Linguagens e Artes	48
Quadro 3: Proposta de dimensões, Categorias e Indicadores de Barcelos e Silva (2014)	69
Quadro 4: Variáveis de Evidência	70
Quadro 5: Variáveis mapeadas com o PC de acordo com Seiter e Foreman	71
Quadro 6: Medidas de “boa” abstração no PC segundo Wing	73
Quadro 7: Equivalência de disciplinas por estado	84
Quadro 8: Histórico de pontuações do Brasil no teste PISA.....	108
Quadro 9: Histórico da média mundial de pontuações no teste PISA.....	108
Quadro 10: Recursos ausentes em escolas do Ensino Fundamental	110
Quadro 11: Relação Idade e Nível Escolar na Espanha	113
Quadro 12: Relação Idade e Nível Escolar no Brasil	113
Quadro 13: Primeira atividade utilizada nas intervenções	115
Quadro 14: Segunda atividade utilizada nas intervenções	116
Quadro 15: Pilares do PC no Teste de Pensamento Computacional	118
Quadro 16: Terceira atividade utilizada nas intervenções.....	122
Quadro 17: Quarta atividade utilizada nas intervenções	123
Quadro 18: Quinta atividade utilizada nas intervenções	123
Quadro 19: Sexta atividade utilizada nas intervenções	124
Quadro 20: Sétima atividade utilizada nas intervenções	125
Quadro 21: Oitava atividade utilizada nas intervenções	125
Quadro 22: Nona atividade utilizada nas intervenções	126
Quadro 23: Décima atividade utilizada nas intervenções.....	127
Quadro 24: Relação das questões com os Quatro Pilares do PC.....	127
Quadro 25: Nome das amostras e conjunto de amostras (Espanha).....	147
Quadro 26: Nome das amostras e conjunto de amostras (Brasil).....	147

LISTA DE GRÁFICOS

Gráfico 1: Melhoria de desempenho da amostra total (pontuação).....	146
Gráfico 2: Gráfico comparativo dos resultados (total)	146
Gráfico 3: Gráfico comparativo dos resultados na Espanha (Quinto Ano).....	149
Gráfico 4: Gráfico comparativo dos resultados na Espanha (Sexto Ano).....	150
Gráfico 5: Gráfico comparativo dos resultados na Espanha (Quinto e Sexto Anos)	152
Gráfico 6: Gráfico comparativo dos resultados no Brasil (Quinto Ano).....	156
Gráfico 7: Gráfico comparativo dos resultados no Brasil (Sexto Ano).....	157
Gráfico 8: Gráfico comparativo dos resultados no Brasil (Quinto e Sexto Anos)	159

SUMÁRIO

INTRODUÇÃO	17
1 PENSAMENTO COMPUTACIONAL	23
1.1 DEFINIÇÃO.....	25
1.2 OS QUATRO PILARES	33
1.2.1 Decomposição	34
1.2.2 Reconhecimento de Padrões	35
1.2.3 Abstração	38
1.2.4 Algoritmos	40
1.3 BENEFÍCIOS	41
1.3.1 Empregos	42
1.3.2 Compreender o Mundo	42
1.3.3 Transversalidade em diferentes áreas	43
1.3.4 Alfabetização Digital.....	43
1.3.5 Produtividade.....	43
1.3.6 Programação ajuda no aprendizado de outras disciplinas	44
1.3.7 Inclusão de Minorias	44
1.3.8 Diminuição nas limitações físicas	45
1.3.9 Trabalhar em equipe	45
1.4 CONTRADIÇÕES	45
1.5 INTEGRAÇÃO NA EDUCAÇÃO BÁSICA	46
1.6 ABORDAGEM DESPLUGADA.....	50
1.7 AVALIAÇÃO	69
2 PANORAMA GLOBAL DA ADOÇÃO DO PC	77
2.1.1 Iniciativas Governamentais Globais	77
2.1.2 Iniciativa Privada e de Organizações Não-Governamentais	91
2.1.3 Iniciativas Brasileiras	99
2.1.4 Tabela Comparativa.....	103
3 MOTIVAÇÃO E ESTADO-DA-ARTE.....	105
3.1 EXPERIÊNCIA, EVASÃO DOS CURSOS NA ÁREA DA INFORMÁTICA E ALTA DEMANDA POR PROFISSIONAIS.....	105

3.2 PREPARAR OS ESTUDANTES PARA UM MUNDO QUE NECESSITA DE FLUÊNCIA DIGITAL.....	106
3.3 RESULTADOS DO BRASIL NO TESTE PISA.....	107
3.4 VIABILIDADE TÉCNICA, ECONÔMICA E CULTURAL.....	110
4 MATERIAIS E MÉTODOS.....	112
4.1 PROJETO PILOTO 1.....	114
4.1.1 Instrumento Avaliativo.....	114
4.1.2 Desenvolvimento de atividades.....	115
4.1.3 Resultados do primeiro piloto.....	117
4.2 PROJETO PILOTO 2.....	118
4.2.1 Instrumento Avaliativo.....	118
4.2.2 Desenvolvimento de atividades.....	121
4.2.3 Execução do segundo Piloto.....	128
4.2.4 Resultado do segundo piloto.....	129
4.3 APLICAÇÃO NA ESPANHA.....	131
4.3.1 Instrumento Avaliativo.....	131
4.3.2 Desenvolvimento de atividades.....	131
4.3.3 Execução da pesquisa.....	132
4.3.4 Participantes e Grupos.....	137
4.4 APLICAÇÃO NO BRASIL.....	138
4.4.1 Instrumento Avaliativo.....	138
4.4.2 Desenvolvimento de atividades.....	138
4.4.3 Execução da Pesquisa.....	139
4.4.4 Participantes e Grupos de Teste.....	142
5 DISCUSSÃO DOS RESULTADOS.....	143
5.1 ESPANHA.....	147
5.1.1 Quintos anos.....	147
5.1.2 Sextos anos.....	149
5.1.3 Resultado geral da Espanha (Quintos e Sextos Anos).....	151
5.1.4 Observações nas Aulas de Scratch.....	152
5.2 BRASIL.....	154
5.2.1 Quintos anos.....	154

5.2.2 Sextos anos	156
5.2.3 Resultado geral no Brasil (Quintos e Sextos Anos)	158
5.2.4 Observações nas Aulas de Scratch	160
5.3 COMPARATIVO ENTRE PAÍSES.....	160
5.4 COMPARATIVO ENTRE GÊNEROS	161
CONCLUSÃO E TRABALHOS FUTUROS.....	163
REFERÊNCIAS	168
ANEXOS	180
APÊNDICES	189

INTRODUÇÃO

Os conhecimentos em Computação são tão importantes para a vida na sociedade contemporânea quanto os conhecimentos básicos de Matemática, Filosofia, Física, dentre outras, assim como contar, abstrair, pensar, relacionar ou medir. Desta forma, torna-se fundamental tanto no presente quanto no futuro que todos os indivíduos tenham conhecimentos básicos de Computação.

Por um lado, o uso de tecnologia tem se tornado cada vez mais presente. Por exemplo, atualmente já é rotineiro ler um livro utilizando um leitor digital (*tablet*, Kindle ou similar); assinar documentos digitalmente; adquirir músicas, filmes e jogos virtualmente; falar ao telefone podendo-se visualizar a outra pessoa por vídeo e também enviar/receber arquivos dos mais diversos tipos durante a conversação; trabalhar colaborativamente no desenvolvimento de novas tecnologias ou novas pesquisas de maneira instantânea com outras pessoas ou grupos localizados em pontos distintos do mundo, sem limitações geográficas; dirigir veículos sem motorista; e, comprar objetos na Internet e imprimir-los em casa na impressora 3D. Entre os diversos fatores que têm auxiliado nestas mudanças podemos citar o aumento no número de pessoas que estão conectadas à Internet, a expansão da telefonia móvel para os mais diversos lugares do mundo, o aumento na geração e compartilhamento de dados e a melhoria contínua dos algoritmos, tornando-os cada vez mais hábeis e práticos. Outro elemento importante nesse movimento é o poder de processamento dos *microchips* que segue aumentando e diminuindo o consumo energético, sendo possível que em poucos anos se equipare em alguns aspectos ao poder de processamento do cérebro humano.

O impacto da Computação nas outras áreas do conhecimento também é cada vez maior e mais profundo. Problemas complexos de diferentes áreas da ciência estão agora sendo abordados com uma perspectiva computacional, uma vez que a Computação provê estratégias e artefatos para lidar com a complexidade, avançando na solução de problemas que há poucos anos não seriam possíveis.

Exemplos notórios podem ser encontrados na Biologia onde foi possível mapear o genoma humano, identificação de variações de enzimas, simulação da adaptação de seres vivos em diferentes ambientes, simulação da função de um micróbio que produz hidrogênio pela energia solar, criação de modelos para prever o fluxo de contaminantes do lençol freático, cálculo da pegada de carbono, entre outros. Na saúde, encontramos a Computação no desenvolvimento de medicamentos, cirurgias remotas e até mesmo na simulação de previsão de tempo de contaminação por uma doença em um certo ambiente. Na Química utiliza-se a

Computação para simular reações químicas sem a necessidade de colocar pesquisadores em risco de vida, como por exemplo uma simulação de uma chama de fogo que analisou 500 milhões de pontos em 120 mil incrementos de tempo, encontrando 11 tipos diferentes de moléculas, 21 reações químicas e a geração de 30 Terabytes de dados (o equivalente a aproximadamente 40 mil CDs ou 7 mil DVDs de uma camada) em uma fração de segundo (SANKARAN, 2007). Na Matemática é possível a geração de desenhos geométricos complexos, calcular o PI (π) com uma precisão antes nunca imaginada de 3,3 trilhões de casas decimais e encontrar o maior número primo até o momento de 22.338.618 dígitos (GIMPS, 2016). Na Arquitetura e Engenharia é possível projetar novos prédios, casas e outras estruturas através de softwares e verificar se suas estruturas são resistentes em condições normais e durante desastres naturais. A Computação também permite simular o funcionamento de um carro e aviões para treinar novos motoristas e pilotos, respectivamente. Na Física é possível simular o acelerador de partículas, permitido que se realizem testes com diversas configurações sem a necessidade de gastos com tempo e dinheiro. Enfim é difícil encontrar uma área do conhecimento que não esteja sendo impactada e até transformada pela abordagem computacional. Esta convergência da digitalização com o cotidiano, inclusive, encontra-se prevista na Lei de Diretrizes e Bases (LDB), em seu Artigo 32, onde afirma que, além desses conhecimentos, é objetivo da formação básica do cidadão “a compreensão do ambiente natural e social, do sistema político, da tecnologia, das artes e dos valores em que se fundamenta a sociedade” (SOUZA e SILVA, 1997, p. 53).

Esse conhecimento combinado com a quantia de dados existentes e as possibilidades atuais de processamento computacional podem ajudar a melhorar a vida das pessoas quando forem utilizados para a criação de soluções inovadoras. Por exemplo, dados sobre crimes que acontecem em diversos lugares de uma cidade podem ser analisados e processados para estimar onde os crimes podem acontecer, economizando assim recursos. Ou ainda, armazenar informações sobre alimentos (*e.g.* agricultura, pecuária, pesca) em um local único de forma a gerenciar o suprimento de comida de maneira global, podendo fazer, assim, uma melhor produção e logística de alimentos.

Computadores impactam em quase todos os aspectos de nossas vidas. Os primeiros a utilizar os computadores depois dos militares, ou seja, comercialmente, foram os contadores e os engenheiros, pois eram muito velozes e sua precisão era alta. Posteriormente, escritores, estudantes e jornalistas começaram a depender de processadores de texto para criar e modificar facilmente seus documentos (RILEY *et al.*, 2014, p. 2). Atualmente, praticamente qualquer serviço ou produto que utilizamos diariamente depende, direta ou indiretamente, de algum

processo ou decisão determinado por um chip de computador. O fato é que não temos limites para as infinitas possibilidades que os sistemas computacionais permitem. A humanidade evolui e por isso precisamos reconhecer sua importância e decidir por qual caminho seguir: estudar como utilizar os computadores da melhor forma possível, ou boicotar seu uso e sofrer as consequências de um ludita¹.

Em relatório emitido pelo Conselho de Assessoria em Ciência e Tecnologia do Presidente dos Estados Unidos da América, Barack Obama, em outubro de 2010, foi recomendado que fossem efetuados altos investimentos federais² em pesquisa e desenvolvimento em tecnologias (PCAST, 2010, p. vii). Essa recomendação não deve surpreender, já que investimentos anteriores permitiram invenções como a Internet que mudou o mundo, reforçando a liderança tecnológica mundial. O relatório também trata que até este momento, de maneira crescente e limites difíceis de prever, a vida moderna depende dessas tecnologias e tem diversas implicações, entre elas, que pessoas que possuírem domínio delas terão um grande poder em suas mãos e que países que incentivarem esse tipo de conhecimento terão enormes vantagens sobre os demais. Esse relatório não tem a intenção de incentivar que as salas de aula sejam simplesmente equipadas com dispositivos eletrônicos de última geração, pois pesquisas recentes identificaram que essa abordagem não é sinônimo de melhoria no desempenho de estudantes, conforme recente documento denominado “*Students Computers and Learning*”, da *Organization for Economic Co-operation and Development* (OECD, 2015). O relatório conclui (p. 17) que a tecnologia pode ampliar um ensino de alta qualidade, mas uma tecnologia de alta qualidade não pode substituir um ensino pobre. A mesma tendência tem ocorrido em outros países, onde a Computação está sendo inserida no currículo escolar gradualmente e, muitas vezes, de maneira obrigatória.

Pode-se afirmar que muitos jovens têm vasta experiência e bastante familiaridade na interação com novas tecnologias, mas têm pouca experiência para criar (coisas) com novas tecnologias e expressarem-se com as mesmas. É quase como se conseguissem ler, mas não conseguissem escrever com as novas tecnologias. Como fazer os jovens fluentes para que

¹ Termo utilizado para indicar trabalhadores britânicos entre os anos de 1811 e 1816, que eram contra a mecanização de produção e foram responsáveis pelas greves que também destruíram diversos equipamentos. Hoje em dia, utiliza-se esta palavra para se referir a pessoas que se opõem a inovações (OXFORD DICTIONARY, 2017).

² Em janeiro de 2016, foram disponibilizados 4 bilhões de dólares para incentivar o ensino da Computação (<https://goo.gl/bByrPg>). Um novo investimento ocorreu em setembro de 2017, injetando mais 200 milhões de dólares anuais no orçamento das escolas (<https://goo.gl/xCgroi>).

possam escrever com novas tecnologias? Isso quer dizer que eles precisam saber programar (RESNICK, 2012).

No momento que os estudantes aprendem a programar, estão também programando para aprender. Este aprendizado permite que eles aprendam muitas outras coisas e criem novas oportunidades de aprendizagem. Pode-se exemplificar isso, fazendo uma analogia entre a leitura e a escrita. Quando aprendemos a ler e escrever, criam-se novas oportunidades para que possamos aprender outras coisas. Quando aprendemos a ler, podemos então ler para aprender. O mesmo processo ocorre com a programação, ou seja, se aprendemos a programar, podemos programar para aprender ainda mais. Quando nos tornamos fluentes a ler e a escrever, não o fazemos para nos tornarmos escritores profissionais. Na realidade, poucas pessoas se tornam escritores profissionais, mas é útil para todos que aprendam a ler e escrever. Outro exemplo é o ensino da Matemática na escola, pois não se forma um matemático, mas um estudante que saiba fazer cálculos de uso rotineiro.

O mesmo processo acontece com a programação. A maioria dos estudantes quando crescerem não irão necessariamente se tornar programadores ou profissionais em Computação, mas deverão ter a capacidade de pensar de uma forma criativa, com pensamento estruturado e de trabalhar em colaboração, independentemente de sua profissão futura.

É necessário tratar da tecnologia não apenas como ferramenta de aprendizagem, haja visto que, além de ser fascinante recurso didático pedagógico de elevado impacto, também pode ser utilizada como uma forma de estruturar problemas e encontrar soluções para os mesmos, utilizando fundamentos da Computação (Pensamento Computacional). Para que funcionem como tal é preciso, no entanto, uma profunda mudança no paradigma de como é entendido todo o processo pedagógico. A primeira grande mudança de paradigma diz respeito à concepção sobre quais conteúdos devem ser ensinados e qual modelo adotar: a Computação como uma disciplina, ou ensinar Computação de maneira transversal.

Não faz sentido, diante da possibilidade de acesso *full-time* a dados e a informações, que se priorize a memorização, a repetição e a cópia pela cópia. Pelo contrário, o mais importante é ensinar a buscar e a selecionar a informação necessária, abstrair, decompor, reconhecer padrões e programar para que o aluno possa, de modo criativo e dinâmico, enfrentar os problemas propostos em determinada circunstância, através do pensamento crítico e uma metodologia para auxiliar no processo de resolução de problemas.

Dessa forma, a realização desta presente tese pretende contribuir para uma maior reflexão sobre a inclusão do Pensamento Computacional no Ensino Fundamental a partir do entendimento que atualmente ele é uma habilidade básica, assim como ler e escrever. Os

benefícios são diversos e incluem uma maior empregabilidade, uma melhor compreensão do mundo através da transversalidade em outras áreas, além de auxiliar na alfabetização digital, melhoria de produtividade, entre outros. Este estudo também compreende que a habilidade de pensar computacionalmente possibilitará uma maior competitividade internacional (competências do século XXI¹) e auxilia na organização do pensamento, harmonizando com as novas tecnologias existentes e as que virão, preparando-os para a dinamicidade das profissões atuais que demandam uma formação cada vez mais multidisciplinar, necessitando o aprendizado e aperfeiçoamento constante. Levando-se em consideração a situação socioeconômica brasileira, onde 48,8% das escolas não possuem um laboratório de informática e 5,5% sequer possuem energia elétrica (MEC/INEP, 2017), o ensino de conceitos da Computação através de atividades *off-line* (sem o uso de máquinas ou aparatos eletrônicos), também conhecido como “Desplugada” ou “*Unplugged*” é uma alternativa interessante para universalizar o acesso a este conhecimento.

Embora o uso de atividades desplugadas sejam aceitos no meio científico, estas atividades em geral são utilizadas como complementares às atividades realizadas com computadores. O efeito das atividades desplugadas no desenvolvimento do Pensamento Computacional não é plenamente conhecido, sendo possível questionar sua eficácia quando utilizada isoladamente. Diante disso, coloca-se o problema de pesquisa desta Tese: Qual a eficácia da abordagem de ensino desplugado para promover o desenvolvimento do Pensamento Computacional na Educação Básica?

Esta pesquisa propõe uma abordagem desplugada para o ensino do Pensamento Computacional em escolas da Educação Básica, o que inclui a criação de objetos de aprendizagem (quase inexistentes em português) e sua aplicação, além de verificar a eficácia dessa abordagem. Desta forma, o objetivo geral desta pesquisa, é verificar a possibilidade de desenvolver o Pensamento Computacional na Educação Básica utilizando exclusivamente atividades desplugadas, desdobrando-se através dos seguintes objetivos específicos ao fazer uso do “Tripé Educacional”, ou seja, aplicando a pesquisa, ensino e extensão:

1. Desenvolver objetos de aprendizagem de Pensamento Computacional para utilização desplugada;
2. Realizar intervenções em sala de aula com as atividades desplugadas desenvolvidas;

¹ As novas aptidões, no contexto do PC, definidas por P21 (2015) são: Criatividade e Inovação; Pensamento crítico; Solução de Problemas; Comunicação e Colaboração; Informação (procurar, avaliar gerenciar); Uso de mídias (construção, legalidade); TICs (ferramentas, equipamentos).

3. Avaliar o Pensamento Computacional dos estudantes que participaram da intervenção e do grupo controle;
4. Verificar o efeito das atividades propostas no desenvolvimento do Pensamento Computacional;
5. Comparar os resultados obtidos nas intervenções realizadas no Brasil e Espanha;
6. Comparar os efeitos das intervenções entre gêneros.

No primeiro capítulo deste documento será explanado o que é o Pensamento Computacional (PC) e como ele pode contribuir no processo de resolução de problemas com os conceitos herdados da Computação. Ainda neste capítulo, será feita uma revisão bibliográfica sobre o tema, explicando os Quatro Pilares que sustentam sua teoria e os benefícios e contradições que a introdução do PC trará para a sociedade, a educação e a indústria. Por fim, o capítulo fará uma explanação da importância da adoção do Pensamento Computacional na Educação Básica e exemplos de como isso pode ocorrer. Para completar o capítulo, são apresentadas pesquisas recentes com o estado da arte da abordagem desplugada e a avaliação dos estudantes.

O segundo capítulo dissertará a respeito de um panorama mundial do estado da arte do Pensamento Computacional nas escolas, incluindo iniciativas públicas e privadas, tendências, ações promovidas pela academia para promoção do PC e pesquisas recentes relacionadas ao ensino do PC nas instituições de ensino. O terceiro capítulo tratará da motivação do autor e o que levou a selecionar essa temática de pesquisa. No quarto capítulo trata da metodologia e materiais desenvolvidos que se utilizou para atingir o objetivo desta pesquisa e as suas referidas etapas. No quinto capítulo são apresentados os resultados estatísticos da pesquisa e, por fim, apresenta-se a conclusão desta tese com as percepções do autor relativo ao tema trabalhado, trabalhos futuros e as referências utilizadas.

Com a intenção de incentivar e facilitar a leitura deste documento, tentara-se fazer uso de elementos didáticos e uma linguagem acessível também a professores de escolas da educação básica, sempre que possível.

1 PENSAMENTO COMPUTACIONAL

Enquanto entusiastas estão cada vez mais animados devido à rapidez com que a tecnologia avança, possibilitando até mesmo tornar a máquina mais inteligente que o ser humano, muitos críticos encaram essa prospecção com receios. A lógica tradicional, desenvolvida por Aristóteles, foi criada originalmente para ajudar pessoas a pensar de forma mais efetiva, através do uso de silogismo, o qual é a base da Matemática e da Computação. Podemos considerar a lógica computacional uma versão muito mais poderosa, concisa e prática em relação a sua versão antecessora. A lógica foi aperfeiçoada posteriormente durante o século XIX, através da utilização de técnicas de símbolos lógicos, a qual foi iniciada por George Boole e Gottlob Frege e, posteriormente, por Bertrand Russell, Alfred North Whitehead, Kurt Gödel, entre outros (KURZWEIL, 1999).

Por fim, a Lógica Computacional teve seu auge na metade do século XX através da tentativa de mecanizar a verificação de cálculos matemáticos e executar operações de solução de problemas gerais. Nos dias atuais, a técnica de modelagem computacional é essencial para pesquisas nas diversas áreas do conhecimento, além de ser imprescindível entender como definir requisitos de software e os limites da Computação a todos (desde crianças até idosos, independentemente da área que atuam) que, de uma forma ou de outra, se relacionarão em um mundo onde a informação é armazenada, acessada e manipulada por softwares (RILEY e HUNT, 2014) (GUZDIAL, 2016).

Quando nos referimos à manipulação de dados, não se está referindo apenas a computadores (*desktops* ou *laptops*), *tablets*, *smartphones*, *smartwatches*, etc., mas também a dispositivos e equipamentos que são usados rotineiramente. Muitas vezes não se percebe que até mesmo o leite que é consumido em diversos alimentos passa por diversos processos gerenciados por máquinas. Até o produto chegar aos consumidores, é provável que tenha sido necessário passar por diversas etapas em que houve a necessidade do uso da Computação, tais como: o chip que é colocado na coleira da vaca que mantém o histórico e auxilia no controle da ração do animal, os equipamentos de automação industrial que ordenham, armazenam, purificam, esterilizam, pasteurizam, refrigeram, envasam e geram a logística necessária para que o produto chegue até o mercado e, por fim, o refrigerador que armazena o leite e é controlado por um software localizado em um microchip. É inegável que a Computação impacta nossas vidas em vários aspectos e evitá-la é impraticável, ou seja, temos que aprender a conviver com computadores e usá-los de forma cada vez mais satisfatória.

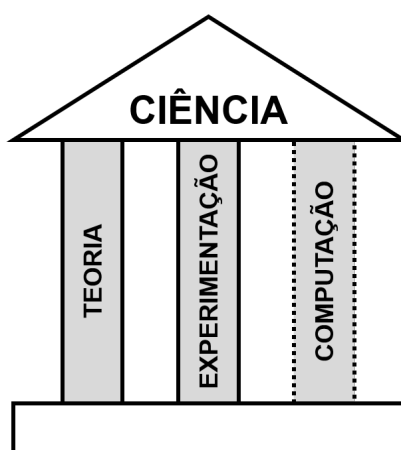
Há cerca de uma década, a comunidade científica identificou que futuras descobertas científicas necessitariam obrigatoriamente de cientistas com certo conhecimento na área da Computação. Essa percepção teve como consequência a criação de novas áreas, como a Inteligência Artificial, Física Computacional, Biologia Computacional, Genômica Computacional, Simulação Molecular, Bioquímica, Biofísica Computacional, entre outras. O pensamento de forma estruturada com os conceitos da Computação, gerou um interesse maior quando se percebeu que seu alcance vai além da comunidade científica, podendo ser utilizada em diversas profissões, como na engenharia, arquitetura, música, logística, possibilitando o estudo de objetos e a solução de problemas que até então eram incrivelmente pequenos, exageradamente grandes, muitos distantes, rápidos demais ou excessivamente complexos. A Computação traz diversos benefícios para a humanidade, porém, do que adianta possuímos essa ferramenta chamada computador, se não sabemos usá-la adequadamente?

O primeiro passo para que se possa aprender melhor a utilizar dispositivos computacionais, é compreender como codificar informações do mundo real em dados que possam ser compreendidos pelas máquinas e como relacionar dados de diversas fontes e formatos diferentes. A conversão destas informações em dados computáveis apresenta diversos desafios, tais como: armazenar uma imagem, textos, áudios, vídeos, digital de um dedo, entre outros.

Na década de 1940, John von Neumann profetizou que computadores não seriam apenas uma ferramenta para ajudar a ciência, mas também uma forma de fazer ciência. Entre as décadas de 1950 e 1960 surgiu o termo “Pensamento Algorítmico” que era compreendido como “orientação mental para formular problemas com conversões, com alguma entrada (*input*) para uma saída (*output*), utilizando uma forma algorítmica para executar as conversões” (DENNING, 2009). Algumas décadas depois, mais especificamente no ano de 1975, o ganhador do Prêmio Nobel de Física, Laureate Ken Wilson promoveu a ideia de que a simulação e a Computação eram formas de fazer ciência que não estavam disponíveis anteriormente. Foi através do uso das máquinas que ele conseguiu criar modelos computacionais que produziram uma compreensão nunca antes imaginada sobre a mudança de estados de materiais.

No início da década seguinte, Laureate se uniu a outros cientistas de destaque para tentar defender que a Computação seria a solução para os grandes desafios da ciência (DENNING, 2009). A partir dos levantamentos realizados pelos pesquisadores, definiu-se que a Computação seria o terceiro pilar da ciência, além dos tradicionais: teoria e experimentação (vide Figura 1). Nesse momento nasce o que seria posteriormente chamado de “Pensamento Computacional”.

Figura 1: Visão do grupo de cientistas a partir do início da década de 1980



Fonte: próprio autor

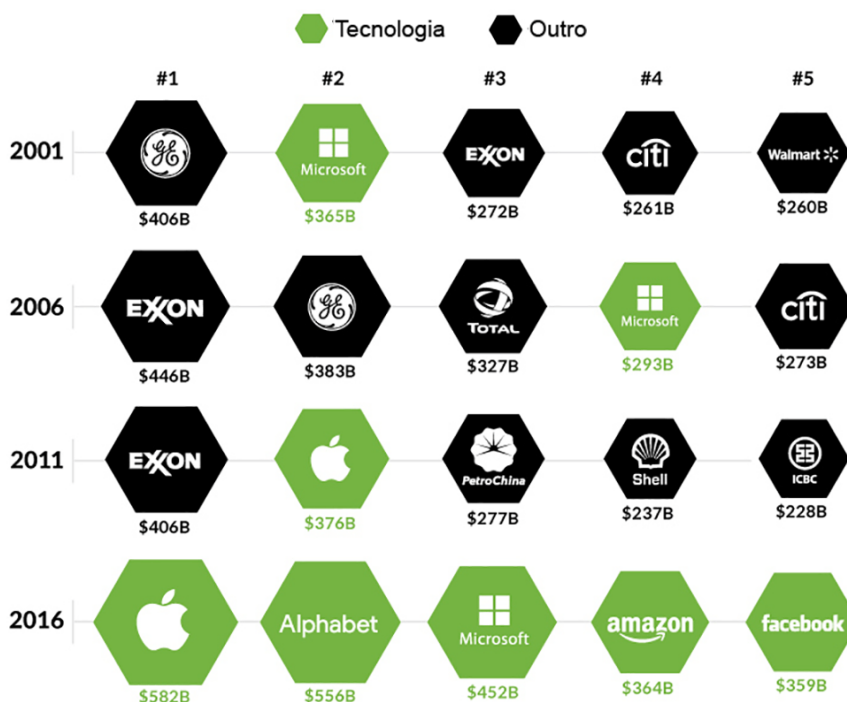
No ano de 2006, a diretora em pesquisas computacionais do *National Science Foundation* (NSF), Jeanette Wing popularizou o termo “Pensamento Computacional” através de um artigo publicado em uma revista muito influente no âmbito acadêmico da Computação (*Communications of the ACM*) onde ela argumentou que a maneira que os Cientistas da Computação pensam sobre o mundo é útil para outros contextos (WING, 2006). Ela não inventou o termo, mas definiu o que os Cientistas da Computação fazem e descreveu o que a Ciência da Computação poderia oferecer para as outras áreas leigas no assunto. Em recente levantamento (DESJARDINS, 2016) também é possível perceber que a indústria também teve a mesma percepção e os investimentos passaram das companhias energéticas para as de tecnologia. Esse movimento é ilustrado na Figura 2, onde constam as empresas com maior valor de mercado dos EUA a cada cinco anos. Note que no ano de 2001 apenas uma empresa de tecnologia se encontrava na lista. Já no ano de 2016 todas as cinco maiores empresas dos EUA são de tecnologia.

1.1 DEFINIÇÃO

O termo “Pensamento Computacional” jamais pode ser confundido com a simples aptidão de manusear aplicativos em dispositivos eletrônicos (Alfabetismo Digital) ou uma forma de pensar de forma mecânica, limitando a criatividade da mente humana. O termo Pensamento Computacional foi apontado e teve grande repercussão por Jeannette Wing em 2006. Mas lendo o artigo "*Twenty things to do with a computer*" de Seymour Papert e Cynthia Solomon, escrito no ano de 1971 (PAPERT e SOLOMON, 1972), pode-se perceber que as ideias do Pensamento Computacional já existiam, porém não tinham sido denominados com esse termo. No ano de 1980, Papert em seu livro intitulado "*Mindstorms: Children, Computers,*

And Powerful Ideas” (PAPERT, 1980, p. 182) utilizou o termo “Pensamento Computacional” na literatura. A obra trata da cultura dos computadores e o papel da tecnologia no ensino de crianças; porém naquele período não houve uma mobilização para a difusão de seus princípios.

Figura 2: Movimento das empresas de maior valor de mercado nos EUA (em dólares)



Fonte: Adaptado de DESJARDINS (2016)

Em 1981 também ocorre o surgimento de outra vertente de pesquisadores que tentava definir o que era uma “Máquina Nocial” (BOULAY *et al.*, 1981) como sendo “uma abstração do computador que se pode usar para pensar sobre o que um computador pode e vai fazer”. De acordo com Boulay, são propriedades gerais da máquina que se está aprendendo a controlar durante o processo de aprendizado de programação. A Máquina Nocial tem sido utilizada em estudos para fornecer uma orientação teórica para examinar como as pessoas pensam sobre a Computação e os equívocos que podem surgir, auxiliando na aprendizagem e comprovando que não é uma proposição hipotética. Caspersen *et al.* (2016) resumem uma Máquina Nocial como sendo um Mapa Mental da operação de uma máquina ou sistema e não de maneira cinestésica (*e.g.* movimentar-se, usar cartões, cortar, colar, desenhar, pintar, resolver enigmas, etc.), estimulando todos os sentidos (SANT’ANNA, 1995, p. 22). Acredita-se que devido à abordagem ser de cunho mais teórico e utilizado prioritariamente em investigações na academia, não teve a mesma disseminação e adoção comparado ao Pensamento Computacional.

Em 2006 o termo Pensamento Computacional (PC) voltou a ser utilizado e causou grande repercussão devido a um artigo de três páginas publicado pela pesquisadora Jeannette M. Wing sobre o que era o PC e como esta habilidade é essencial para qualquer pessoa, independentemente da área e não somente às pessoas que estão, de uma forma ou de outra, relacionadas com a Informática (WING, 2006).

Wing, em seus trabalhos, conceituou o termo “Pensamento Computacional” em mais de uma forma. Em seu primeiro artigo descreve como “a combinação do pensamento crítico com os fundamentos da Computação define uma metodologia para resolver problemas, denominada Pensamento Computacional” e “uma distinta forma de pensamentos com conceitos básicos da Ciência da Computação para resolver problemas, desenvolver sistemas e para entender o comportamento humano, habilidade fundamental para todos”. Em outra publicação, Wing (2007) explica a amplitude do Pensamento Computacional, combinando e complementando a forma de pensar na Matemática e na Engenharia, afirmando que de um lado, o Pensamento Computacional “baseia-se nos fundamentos da Matemática, porém é limitado pela física do equipamento em um nível inferior” e de outro lado, “utiliza a base da Engenharia desde a interação com o mundo real, porém pode-se construir mundos virtuais sem se preocupar com as limitações físicas”.

Em seu trabalho seguinte, Wing (2010) já define o termo como “processos de pensamento envolvidos na formulação de problemas e as suas soluções de modo que as mesmas são representadas de uma forma que pode ser eficazmente executada por um agente de processamento de informações”. Em seu artigo posterior, Wing (2014), a autora faz uma pequena alteração na definição anterior, afirmando que “são os processos de pensamento envolvidos na formulação de um problema e que expressam sua solução ou soluções eficazmente, de tal forma que uma máquina ou uma pessoa possa realizar”; ainda complementa como sendo uma “automação da abstração” e “o ato de pensar como um cientista da Computação”.

Bundy (2007) e Nunes (2011) definem o Pensamento Computacional como habilidades comumente utilizadas na criação de programas computacionais como uma metodologia para resolver problemas específicos nas mais diversas áreas. Google for Education (2015) faz uma definição similar como sendo “uma abordagem usada para solução de problemas utilizando o que se sabe sobre Computação”. Essa definição é muito próxima da fornecida pela Sociedade Real do Reino Unido (FURBER, 2012): “o processo de reconhecer aspectos da computação em um mundo que nos cerca e, aplicar ferramentas e técnicas da

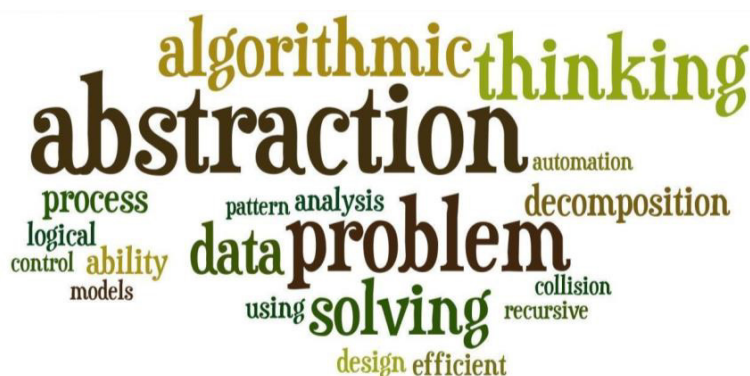
Ciência da Computação para entender e argumentar sobre sistemas e processos naturais e artificiais”.

Liukas (2015), coautora do currículo de Computação da Finlândia, faz uma abordagem bem acessível, definindo como “pensar nos problemas de forma que um computador consiga solucioná-los”. Liukas ainda complementa: “O Pensamento Computacional é executado por pessoas e não por computadores. Ele inclui o pensamento lógico, a habilidade de reconhecimento de padrões, raciocinar através de algoritmos, decompor e abstrair um problema”.

Já McMaster *et al.* (2010) compreendem o Pensamento Computacional como uma das práticas fundamentais da Ciência da Computação, porém o termo não é de uso exclusivo da Computação e não é adequado para retratar a totalidade do campo.

Em um artigo de uma pesquisa bibliográfica sistemática (KALELIOĞLU, 2017) foram analisados 125 artigos relativos ao PC selecionados de acordo com critérios pré-definidos e seis bibliotecas digitais na tentativa de definição do termo, onde foi possível a geração de uma nuvem de palavras com os termos mais utilizados nos trabalhos. Infelizmente, mesmo após o levantamento, ainda não foi possível encontrar um consenso de definição entre os diferentes autores. O resultado do processo de geração da nuvem de palavras encontra-se na Figura 3.

Figura 3: Nuvem de palavras gerada a partir da análise de 125 artigos



Fonte: KALELIOĞLU (2017)

No ano de 2011, a *International Society for Technology in Education* (ISTE) em conjunto com a *Computer Science Teachers Association* (CSTA) divulga a “definição operacional” do PC que passou por um processo de avaliação e aprovação de quase 700 professores de Ciência da Computação (CSTA/ISTE, 2011) a saber:

O Pensamento Computacional é um processo de resolução de problemas que inclui (mas não está limitado a) as seguintes características:

- Formulação de problemas de forma que nos permita usar um computador e outras ferramentas para nos ajudar a resolvê-los;
- Organização e análise lógica de dados;
- Representação de dados através de abstrações, como modelos e simulações;
- Automatização de soluções através do pensamento algorítmico (uma série de etapas ordenadas);
- Identificação, análise e implementação de possíveis soluções com o objetivo de alcançar a combinação mais eficiente e efetiva de etapas e recursos;
- Generalização e transferência deste processo de resolução de problemas para uma grande variedade de problemas.

Essas habilidades são apoiadas e reforçadas por uma série de qualidades ou atitudes que são dimensões essenciais do PC. Essas qualidades ou atitudes incluem:

- Confiança em lidar com a complexidade;
- Persistência ao trabalhar com problemas difíceis;
- Tolerância para ambiguidades;
- A capacidade de lidar com os problemas em aberto;
- A capacidade de se comunicar e trabalhar com outros para alcançar um objetivo ou solução em comum.

Em recente publicação da ISTE (2016), o Pensamento Computacional é novamente definido apenas como uma forma de “desenvolver e empregar estratégias para entender e resolver problemas de forma a aproveitar o poder dos métodos tecnológicos para desenvolver e testar soluções”.

Mesmo após diversos estudos e quase uma década de esforços para definir o Pensamento Computacional, ainda existem críticas que sugerem que não sabemos o que o Pensamento Computacional significa ou sua forma de medir (KURSHAN, 2016). Por esta razão este pesquisador, após a fusão de diversas fontes, propõe-se a seguinte definição para o termo: “O Pensamento Computacional é uma distinta capacidade criativa, crítica e estratégica humana de saber utilizar os fundamentos da Computação, nas mais diversas áreas do conhecimento, com a finalidade de identificar e resolver problemas, de maneira individual ou colaborativa, através de passos claros, de tal forma que uma pessoa ou uma máquina possam executá-los eficazmente”.

Outro questionamento que é comumente encontrado ao definir o termo é a fronteira entre o Pensamento Computacional e a Computação ou como a Pensamento Computacional interage com a Computação. Não foi encontrado uma explanação formal ou apuração clara dos limites ou intersecções entre eles. Desta forma, o pesquisador, por meio de listas de discussões com outros pesquisadores, realizou uma tentativa de registrar suas diferenças e semelhanças de maneira estruturada. O resultado do levantamento encontra-se na Figura 4.

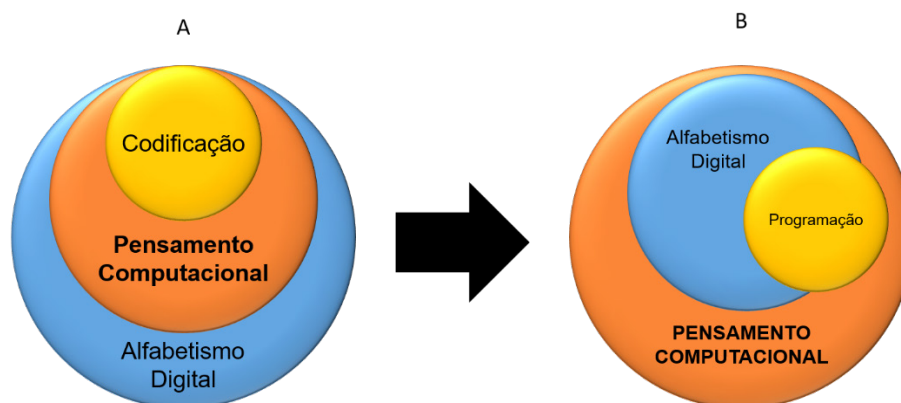
Figura 4: Relacionamento entre áreas e seus contextos



Fonte: próprio Autor

Já o Google for Education (JOHNSON, 2015), faz uma leitura diferente, onde abstrai diversos elementos da proposta apresentada pelo autor (vide Figura 5 A). Porém, após a revisão bibliográfica, o autor redesenhou o diagrama com uma nova proposta (vide Figura 5 B). Esse redesenho se deu devido à compreensão do autor de não haver necessidade de o estudante ser um alfabetizado digital para dominar o Pensamento Computacional. Pode-se exemplificar essas situações em tarefas cotidianas, como por exemplo, escrever uma receita de um bolo ou prato, definir o itinerário de uma viagem ou criar um mapa mental de como resolver uma conta armada. Artefatos eletrônicos encontrados nas residências também estão inclusos nessa situação, como por exemplo saber manusear uma televisão (*e.g.* programar a gravação de um filme, navegar no menu do aparelho), micro-ondas (*e.g.* programar o tempo de operação, informar o peso do alimento para o aparelho calcular o tempo de descongelamento), ar condicionado (*e.g.* ajustar o timer ou temporizador, definir o modo de funcionamento), entre outros. Em nenhum desses casos citados existe a necessidade de ser alfabetizado digitalmente, sendo assim, percebe-se que a programação não está diretamente enraizada no alfabetismo digital, porém assimilamos o Pensamento Computacional em nosso dia-a-dia sem perceber.

Figura 5: Representação estrutural das competências relacionadas



Fonte: adaptado de Google for Education (JOHNSON, 2015) / próprio autor

A partir de uma visão ampla, o Pensamento Computacional (PC) é uma habilidade que qualquer pessoa deveria saber, independentemente da área de conhecimento ou atividade profissional, assim como ler, escrever e calcular. Ao contrário do que o nome sugere, o PC não envolve apenas conceitos de Computação para solução de problemas em suas raízes, pois também agrega práticas de projetar sistemas, entender o comportamento humano e o pensamento crítico (WING, 2010) e contempla as seguintes características (WING, 2008) (ISTE; CSTA; NSF, 2011):

- Identificar, analisar e implementar as soluções possíveis com o objetivo de conseguir a combinação mais eficiente e eficaz de etapas e recursos;
- Reformular um problema de grande dificuldade para que se possa resolvê-lo (redução, incorporação, transformação ou simulação);
- Escolher a representação ou modelagem apropriada com aspectos importantes do problema para facilitar sua manipulação;
- Interpretar o código como dados e dados como código;
- Usar abstração e decomposição na solução de uma tarefa complexa;
- Avaliar a simplicidade e elegância de um sistema;
- Pensar de forma recursiva;
- Verificar o padrão, utilizando generalização da análise dimensional;
- Prevenir, detectar e recuperar das piores situações com a utilização de redundância, contenção de danos e correção de erros;
- Modularizar antecipadamente e pré-carregar necessidades dos usuários;
- Prevenir congestionamentos e impasses (*deadlocks*), além de evitar condições de corrida ao sincronizar reuniões;

- Utilizar a Inteligência Artificial para a resolução de problemas específicos ou complexos;
- Formular problemas de modo que seja possível usar o computador e outras ferramentas para ajudar a resolvê-los;
- Organizar e analisar dados de forma lógica;
- Automatizar soluções através do pensamento algorítmico;
- Generalizar e transferir esse processo de resolução de problemas para uma grande variedade de problemas.

Grover e Pea (2013), após ampla revisão bibliográfica, resumiram em nove elementos que o PC tende a atender para apoiar a aprendizagem dos alunos de forma interdisciplinar, bem como avaliar o seu desenvolvimento:

- Abstração e reconhecimento de padrões (incluindo modelos e simulações);
- Processamento sistemático da informação;
- Sistema de símbolos e representações;
- Noções de controle de fluxo em algoritmos;
- Decomposição de problemas estruturados (modularização);
- Pensamento iterativo, recursivo e paralelo;
- Lógica condicional;
- Eficiência e restrições de desempenho;
- Depuração e detecção de erro sistemático.

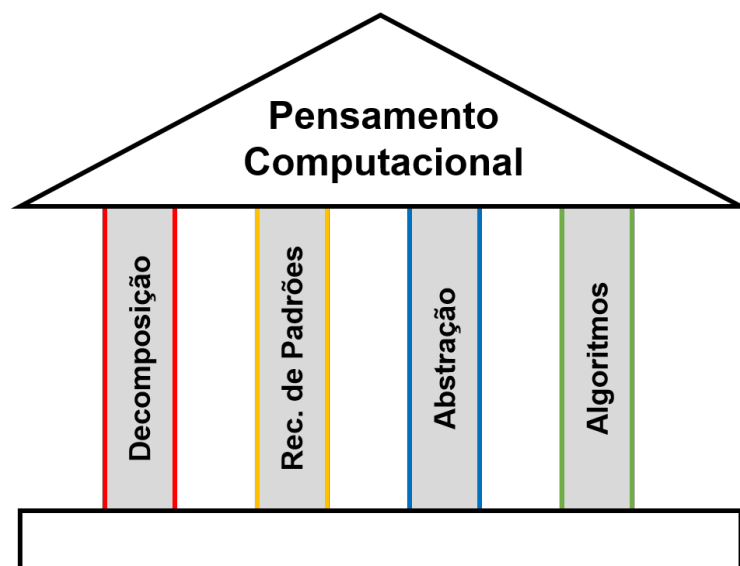
O Reino Unido está liderando a implantação da Computação no ensino fundamental, onde o Currículo Nacional, desde 2011, inclui a disciplina de Computação como obrigatória em todos os seus quatro níveis (*Key Stages – KS*), independentemente da idade dos estudantes. Pesquisas lideradas por Code.Org (2016), Liukas (2015) e BBC Learning (2015), mesclaram os elementos citados por Grover e Pea (2013) e o guia difundido por Computer at School (CSIZMADIA *et al.*, 2015) e resumiram nos chamados “Quatro Pilares do Pensamento Computacional”, sendo eles: Decomposição, Reconhecimento de Padrões, Abstração e Algoritmos. A seguir, os pilares serão melhor detalhados na seção seguinte.

1.2 OS QUATRO PILARES

O Pensamento Computacional envolve identificar um problema complexo e quebrá-lo em pedaços menores e mais fáceis de gerenciar (DECOMPOSIÇÃO). Cada um desses problemas menores pode ser analisado individualmente com maior profundidade, identificando problemas parecidos que já foram solucionados anteriormente (RECONHECIMENTO DE PADRÕES), focando apenas nos detalhes que são importantes, enquanto informações irrelevantes são ignoradas (ABSTRAÇÃO). Por último, passos ou regras simples podem ser criados para resolver cada um dos subproblemas encontrados (ALGORITMOS). Seguindo os passos ou regras utilizadas para criar um código, é possível também ser compreendido por sistemas computacionais e, conseqüentemente, utilizado na resolução de problemas complexos eficientemente, independentemente da carreira profissional que o estudante deseja seguir.

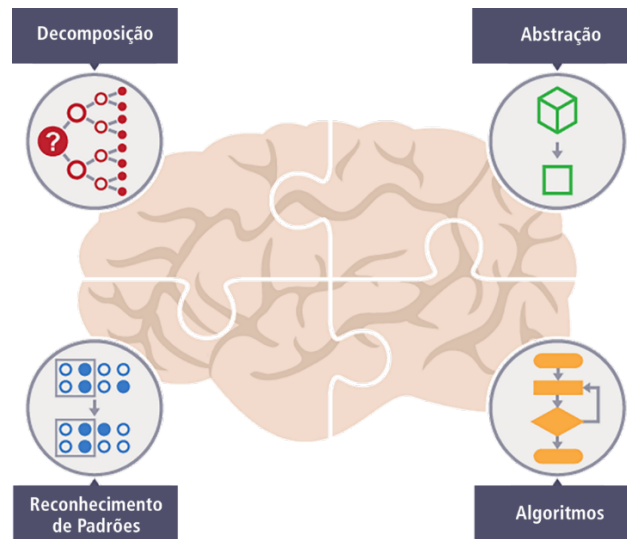
Como pode ser percebido, o Pensamento Computacional utiliza essas “quatro dimensões”, denominados aqui como “Quatro Pilares” (Decomposição, Reconhecimento de Padrões, Abstração e Algoritmos), para atingir o objetivo principal: a resolução de problemas. Os pilares que formam a base do PC podem ser visualizados na Figura 6 e Figura 7. Todos os Quatro Pilares têm grande importância e são interdependentes durante o processo de formulação de soluções computacionalmente viáveis. Cada um dos pilares será explanado na sequência.

Figura 6: Os Quatro Pilares do Pensamento Computacional



Fonte: próprio Autor

Figura 7: Os quatro pilares do Pensamento Computacional segundo BBC



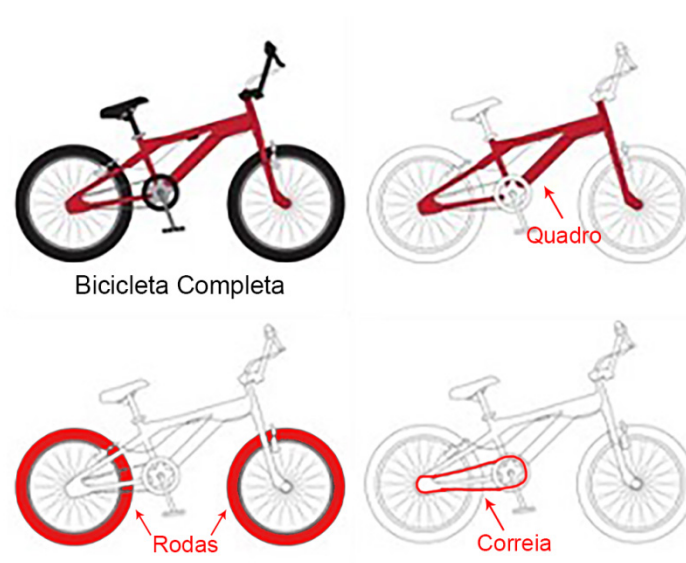
Fonte: adaptado de BBC Learning, (2015).

1.2.1 Decomposição

Liukas (2015) relata que a decomposição é um processo pelo qual os problemas são quebrados em partes menores. Ela exemplifica isto através da decomposição de refeições, receitas culinárias e as fases que compõem um jogo. Trata-se de quebrar um problema ou sistema complexo em partes menores, que são mais manejáveis e mais fáceis de entender. As partes em menor tamanho podem, então, serem examinadas e resolvidas, ou concebidas individualmente, uma vez que são mais fáceis de trabalhar.

Quando um problema não está decomposto, sua resolução é muito mais difícil. Ao lidar com muitos estágios diferentes ao mesmo tempo, torna-se mais difícil sua gestão. Uma forma de facilitar a solução é dividir em partes menores e resolvê-las, individualmente. Esta prática também aumenta a atenção aos detalhes. Por exemplo, para entender o funcionamento de uma bicicleta, é mais fácil através do desmembramento de suas partes. Na Figura 8 demonstra-se na seguinte ordem: a bicicleta completa e seus mecanismos de estrutura, tração e transferência de energia. Note que ao decompô-la em partes menores, é possível identificar de forma mais nítida cada uma das suas partes e funcionalidades.

Figura 8: Partes da Bicicleta (Decomposição)



Fonte: Produção do autor.

Quando a decomposição é aplicada a elementos físicos, como por exemplo a bicicleta, a manutenção torna-se mais fácil quando é possível modularizar suas partes. Caso contrário, se o item em questão fosse desenvolvido em uma única peça, seu reparo se tornaria muito difícil e a forma de consertá-lo seria trocando por outro. O mesmo acontece com o desenvolvimento de programas.

Liukas (2015) afirma que programadores utilizam frequentemente esta técnica para dividir um algoritmo em pedaços menores para facilitar sua compreensão e manutenção. Em um código-fonte, pode-se exemplificar a decomposição através de funções, procedimentos, objetos, módulos, entre outros. Esta técnica possibilita resolver problemas complexos de forma mais simples, facilita a compreensão de novas situações e possibilita projetar sistemas de grande porte (CAS, 2014) (CSIZMADIA *et al.*, 2015).

1.2.2 Reconhecimento de Padrões

Ao realizar a decomposição de um problema complexo, seguidamente se encontra padrões entre os subproblemas gerados. Padrões são similaridades ou características que alguns dos problemas compartilham e que podem ser explorados para que sejam solucionados de forma mais eficiente. Liukas (2015) define o Reconhecimento de Padrões como encontrar similaridades e padrões com o intuito de resolver problemas complexos de forma mais eficiente. Para isso, procura-se por elementos que sejam iguais ou muito similares em cada problema. Na

literatura, o Reconhecimento de Padrões também pode estar associado ao termo “Generalização”.

O Reconhecimento de Padrões é uma forma de resolver problemas rapidamente fazendo uso de soluções previamente definidas em outros problemas e com base em experiências anteriores. Os questionamentos “Esse problema é similar a um outro problema que já tenha resolvido?” ou “Como ele é diferente?” são importantes nesta etapa, pois ocorre a definição dos dados, processos e estratégias que serão utilizados para resolver o problema. Algoritmos que são responsáveis pela solução de algum problema específico podem ser adaptados para resolver uma variedade de problemas similares. Sempre que necessário, o algoritmo pode aplicar uma solução de forma generalizada.

Um exemplo prático da aplicação desta técnica é através da identificação de similaridades entre raças de cachorros, conforme pode ser visto na Figura 9. Note que os cachorros possuem olhos, rabo e pelos, porém suas características podem ser diferentes, como por exemplo, a cor dos olhos, comprimento do rabo e a cor dos pelos. No Pensamento Computacional, estas características são chamadas de Padrões. No momento em que se consegue um padrão de cachorro, pode-se descrever outros, simplesmente seguindo o padrão e alterando as características, como por exemplo: olhos azuis, rabo curto e pelagem cinza; outro já possui olhos pretos, rabo longo e pelos ruivos.

Figura 9: Similaridade entre raças de cachorros (Reconhecimento de Padrões)

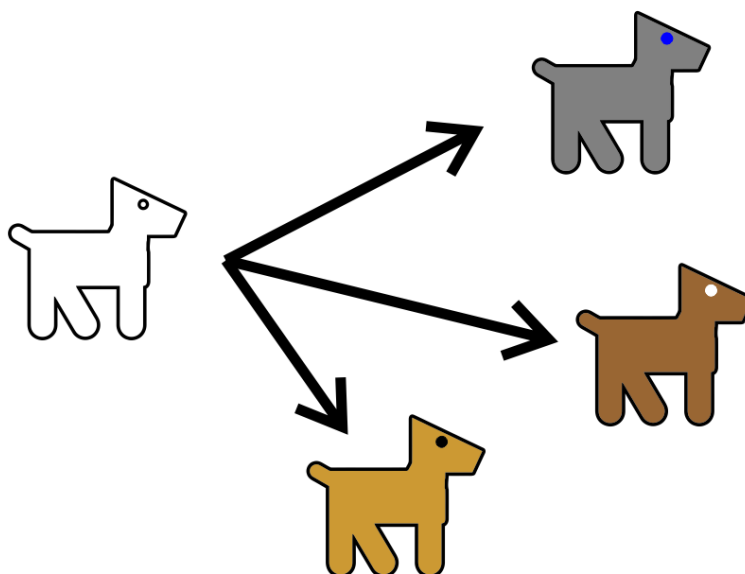


Fonte: adaptação do autor

Através do reconhecimento de padrões, é possível simplificar a solução de problemas e replicar esta solução em cada um dos subproblemas, caso haja semelhança. Quanto mais padrões se consegue encontrar, mais dinâmico e rápido a macro solução é encontrada.

Caso se quisesse desenhar um cachorro, pode-se fazer uma replicação do padrão identificado, como se pode ver na Figura 10, onde no lado esquerdo se encontra o desenho do padrão identificado e no lado direito constam as variações replicadas a partir de um molde genérico.

Figura 10: Padrão e replicações (Reconhecimento de Padrões)



Fonte: Produção do autor

Caso a etapa de reconhecimento de padrão não ocorra no processo de resolução de um problema, seria necessário retornar à etapa anterior e verificar novamente o formato de um cachorro, todas as vezes. Este retrabalho não seria impossível, porém desaceleraria a atividade, tornando a execução morosa e desnecessariamente repetitiva. Além de ineficiente, também aumentaria o possível erro humano e se não ocorresse esta análise, haveria uma possibilidade de se não perceber que todos os cachorros possuem rabo, olhos e pelos. Quando desenhado, o cachorro poderia não ter um formato característico (de cachorro) pelo fato de não ter sido identificado o padrão na etapa anterior; ou seja, o problema não foi resolvido da forma correta.

Outra forma de exemplificar o uso do reconhecimento de padrões se dá através da aplicação da Sequência de Fibonacci (vide Figura 11). A sequência de Fibonacci é uma sucessão de números que frequentemente é encontrado em fenômenos naturais (KNOTT, 2016). Note que para resolver o F_4 é possível reutilizar resultados já resolvidos nas etapas

anteriores e todos os cálculos podem ser representados através de uma fórmula que generaliza os demais (padrão).

Figura 11: Sequência de Fibonacci

$$\begin{array}{l}
 F_0 = 0 \\
 F_1 = 1 \\
 F_2 = F_1 + F_0 \\
 F_3 = F_2 + F_1 \\
 F_4 = F_3 + F_2 \\
 [\dots] \\
 F_n = F_{(n-1)} + F_{(n-2)}
 \end{array}$$

Fonte: Produção do autor

1.2.3 Abstração

Este pilar envolve a filtragem dos dados e sua classificação, essencialmente ignorando elementos que não são necessários para que se possa concentrar nos que são relevantes. Através desta técnica, consegue-se criar uma representação (ideia) do que está se tentando resolver. A competência essencial deste pilar é escolher o detalhe a ser ignorado para que o problema seja mais fácil de ser compreendido sem perder nenhuma informação que seja importante para tal (CAS, 2014).

De acordo com Wing (2006), é o conceito mais importante do Pensamento Computacional, pois o processo de abstrair é utilizado em diversos momentos, tais como:

- a) Na escrita do algoritmo e suas iterações;
- b) Na seleção dos dados importantes;
- c) Na escrita de uma pergunta;
- d) Na alteridade de um indivíduo em relação a um robô;
- e) Na compreensão e organização de módulos em um sistema.

Já Liukas (2015) define a abstração como um processo de separação de detalhes que não são necessários para poder se concentrar em coisas que são importantes. Ela ainda exemplifica o calendário como uma abstração do tempo e o mapa de metrô como sendo uma abstração do mundo real, pois informa apenas o essencial para o passageiro se locomover na cidade, excluindo informações que não são úteis para os mesmos, tais como altitude, posição

geográfica, entre outros. Em posse de um mapa desses, o passageiro consegue definir seu itinerário de forma clara. Um exemplo de mapa de metrô pode ser encontrado na Figura 12.

Figura 12: O mapa do metrô como exemplo de abstração do mundo real



Fonte: CPTM-SP

A Ciência da Computação já está presente em praticamente todos os aspectos do cotidiano. Seu impacto na sociedade é visto através da proliferação de computadores, sistemas de informação, editores de texto, planilhas, e todos os programas de aplicação que foram desenvolvidos para tornar os computadores mais fáceis de usar e as pessoas mais produtivas. Uma parte importante do campo lida com a forma de tornar a programação do software mais fácil e mais confiável. Mas, fundamentalmente, Ciência da Computação é uma ciência de abstração, cria o modelo certo para pensar sobre um problema e elaborar as técnicas mecanizáveis adequadas para resolvê-lo (AHO e ULLMAN, 1995). Os cientistas da Computação, devem criar abstrações de problemas do mundo real que podem ser compreendidas por usuários de computador e, ao mesmo tempo, podem ser representados e manipulados facilmente dentro de um sistema computacional.

Por exemplo, um algoritmo é uma abstração de um processo que recebe uma entrada, executa a sequência de passos e produz uma saída que satisfaça um objetivo específico. Projetar algoritmos eficientes, por natureza, envolve também a criação de tipos de dados abstratos. A abstração proporciona um poder para escalonamento e tratamento com a complexidade (WING, 2010).

A abstração pode ser exemplificada através de histórias infantis que envolvam atividades matemáticas. É necessário ocorrer a abstração das informações pertinentes da história para poder acompanhá-la e resolver a equação. Outra forma é utilizar momentos históricos para abstrair informações que não são óbvias e necessitam uma análise onde ocorre o reconhecimento de padrões, tais como: “Após estudar a história de grandes líderes mundiais, quais serão as características de um bom líder? ”.

1.2.4 Algoritmos

Considerado por Wing (2014) o elemento que agrega todos os demais, o algoritmo é um plano, uma estratégia ou um conjunto de instruções claras necessárias para a solução de um problema (CSIZMADIA *et al.*, 2015). Em um algoritmo, as instruções são descritas e ordenadas para que o seu objetivo seja atingido e podem ser escritas em formato de diagramas ou pseudocódigo (linguagem humana), para depois serem escritos códigos em uma linguagem de programação. Liukas (2015) define “Algoritmos” como “um conjunto de passos específicos usado para solucionar um problema” e ainda o diferencia do termo “Programa” como sendo “uma sequência de instruções precisas escritas em uma linguagem que computadores compreendam”.

É o que se pode chamar do núcleo principal, pois possui uma grande abrangência em diversos momentos das atividades propostas pelo Pensamento Computacional. É um conjunto de regras para a resolução de um problema, como a receita de um bolo; porém, diferentemente de uma simples receita de bolo, pode-se utilizar diversos fatores mais complexos. Existem algoritmos muito pequenos, que podem ser comparados a pequenos poemas. Outros algoritmos são maiores e precisam ser escritos como se fossem livros, ou então maiores ainda, necessitariam inevitavelmente serem escritos em diversos volumes de livros. Para entender melhor, é possível fazer questionamentos que possam facilitar a compreensão de como gerar e quais as limitações do mesmo, tais como: “É possível solucionar um problema utilizando algoritmos?”, ou “Qual a precisão que se necessita para solucionar um problema?”.

Algoritmos devem ser compreendidos como soluções prontas, pois já passaram pelo processo de decomposição, abstração e reconhecimento de padrões para sua formulação. Ao serem executados, seguirão os passos pré-definidos, ou seja, aplicar-se-á solução quantas vezes forem necessárias, não havendo a necessidade de criar um novo algoritmo para cada uma de suas execuções posteriores.

A propósito, pode-se exemplificar através de uma atividade que é ensinada nas escolas durante as aulas de Matemática: calcular uma soma com a sobreposição dos números (“conta armada”). Um exemplo de conta armada é apresentada na Figura 13, onde é possível identificar uma sequência de passos necessários para atingir o resultado (algoritmo). Se os alunos ou o computador seguirem as mesmas regras claras para resolver a conta, ambos teriam condições de determinar a soma de quaisquer números. No momento em que se define corretamente todos os passos do algoritmo nos sistemas computacionais, não há mais necessidade de preocupar-se com a resolução deste problema e pode-se, então, focar em elementos mais complexos do sistema. A principal característica do algoritmo é a possibilidade de automação das soluções.

Figura 13: Conta de soma "armada"

$$\begin{array}{r} 13 \\ +28 \\ \hline 41 \end{array}$$

Fonte: próprio autor

1.3 BENEFÍCIOS

Os diversos esforços em tornar o Pensamento Computacional uma habilidade básica para qualquer pessoa surgiu devido a diferentes razões. No momento que temos conhecimento dos efeitos esperados, pode-se definir o que é necessário para atingir os objetivos pretendidos e os métodos a serem adotados para alcançar as metas, assim como o que não deve ser feito. Destacam-se a seguir os principais argumentos a respeito dos benefícios que o Pensamento Computacional pode trazer à parte interessada (GUZDIAL, 2016).

1.3.1 Empregos

Há uma alta demanda de mão de obra qualificada com conhecimentos em programação. Através de uma carta aberta assinadas pelos funcionários do alto escalão da Microsoft, Google e Facebook foi citada a seguinte frase: “Nós precisamos de talentos, precisamos disso agora, e nós simplesmente não conseguimos achar o suficiente” (CITYOFNY, 2015).

De acordo com Guzdial (2016), o Escritório de Dados Estatísticos Trabalhista dos EUA estimou que o desenvolvimento de software é uma das categorias com maior crescimento, e um alto percentual (aprox. 50%) de profissões relacionadas à ciência, tecnologia, engenharia e matemática que, no futuro necessitarão de pessoas com conhecimento em Computação. Essa carência poderá ser abastecida se a programação for introduzida nas escolas. Evidências em pesquisas (GUZDIAL *et al.*, 2012) comprovam efetivo desenvolvimento da motivação para seguir carreira na área da Computação em estudantes que tiveram aulas de Computação no ensino médio. Outro levantamento do Ministério do Trabalho dos EUA (BUREAU OF LABOR STATISTICS, 2015), prevê um crescimento de 12 por cento de 2014 a 2024 na área da Tecnologia da Informação e Comunicação. Um percentual maior do que a média para todas as demais ocupações. A agência também projeta também a abertura de 488.500 vagas de trabalho, passando de 3.9 milhões para 4.4 milhões de empregos entre os anos de 2014 e 2024.

No Capítulo 3 (p. 105), como parte da motivação desta pesquisa, é também relacionada a alta demanda de profissionais com uma formação adequada na área.

1.3.2 Compreender o Mundo

As escolas ensinam química com a finalidade de que os estudantes consigam compreender o mundo em que vivem, ou seja, com uma diversidade de reações químicas em sua volta. Os estudantes também aprendem biologia, pois vivemos em um mundo vivo. Os sistemas inteligentes têm um grande impacto sobre nossas vidas, assim como a inteligência artificial que toma decisões que afetam a vida diretamente. Vive-se neste século um mundo computacional e a realidade da Computação muito provavelmente irá impactar muito mais na rotina dos estudantes do que a necessidade de relembrar a estrutura de um anel de benzeno ou um estágio de uma mitose. Para que isso ocorra, uma das principais iniciativas seria a readequar o currículo das escolas, como já vem ocorrendo em diversos países e continua sendo uma tendência mundial de ensino (competências do século XXI). Este tema será aprofundado no Capítulo 2 (p. 77).

1.3.3 Transversalidade em diferentes áreas

A aplicação de ideias de acordo com critérios computacionais facilitará o trabalho de outras áreas para que seja possível utilizar o poder computacional para a solução de problemas de forma muito mais rápida e até mesmo aumentando a escala do problema com demandas mais complexas.

Wing (2010) relata também que a utilização de métodos da Computação também pode ajudar na melhoria de processos da nossa vida diária (saber identificar informações importantes em grande quantidade de dados, saber encontrar a informação necessária ou até mesmo a posição dos alimentos e bebidas em um restaurante). Além disso, com um modelo estruturado de pensamento, é possível auxiliar no processo de aprender a aprender, *i.e.*, da mesma forma que ocorre o aprendizado da leitura: aprendemos a ler para que possamos ler para aprender.

1.3.4 Alfabetização Digital

Com o uso cada vez mais massivo de computadores na sociedade, o uso de dispositivos digitais será mais uma forma de criar, se expressar e outra forma de apropriação de novos conhecimentos dentro de sua área profissional, pois muitas delas utilizam a Computação para realizar modelagens e simulações para entender melhor o mundo e de uma maneira diferente. Outro ponto importante é que os estudantes aprendem que a internet não é uma rede totalmente segura e enxergam, na prática, como podem ser expostos às mais variadas ameaças, como se proteger e até mesmo criar soluções de segurança.

1.3.5 Produtividade

De acordo com McLuhan (1964), o homem deve utilizar as ferramentas como extensão do seu corpo para aumentar suas capacidades, assim como o martelo a força exercida por nossa mão e as bicicletas e os carros nossas pernas. Esses exemplos demonstram que as ferramentas fazem as mesmas ações que nossos membros exercem, porém de forma otimizada.

O alfabetismo (leitura e escrita) nos proporcionou uma nova forma de se expressar, pensar e compartilhar, o que promoveu avanços significantes. Com o computador e a internet, o sujeito encurta a distância e o tempo para disseminar o seu modo de se expressar, pesquisar, acessar, pensar, decidir e executar atividades, o que resulta no aumento de sua produtividade. É evidente que o computador se tornou um parceiro indispensável para a resolução de problemas.

Sabe-se que a maior parte das pessoas programará em algum nível (*e.g.* planilha eletrônica) parte da sua vida profissional no futuro (claro que muito menos do que um programador profissional). Para explicar esta afirmação, o autor faz a seguinte comparação: a pessoa não precisa desenvolver uma rede social para aprender a programar, assim como não é necessário você aprender a escrever um livro para conseguir escrever uma carta.

1.3.6 Programação ajuda no aprendizado de outras disciplinas

A literatura, através de pesquisas em educação computacional, sugere que a programação pode constituir um método para ensino de Matemática e Ciências (GUZDIAL, 2016, p. 48). A pesquisa realizada por Tew *et al.* (2008) afirma que após o aprofundamento de conhecimento de seus estudantes em conceitos da Computação, alunos que tinham dificuldades em outras disciplinas, começaram a correlacionar esses conceitos com os temas trabalhados nas outras disciplinas e tiveram um rendimento superior, inclusive comparável aos melhores alunos.

1.3.7 Inclusão de Minorias

Com a existência de projetos que incentivam o ingresso de profissionais do gênero feminino na área da Computação, é clara a predominância de homens na área (GUZDIAL, 2016, p. 48). No Google, por exemplo, apenas 30% são mulheres e 2% são da raça negra (GOOGLE FOR EDUCATION, 2015). Esta realidade também existe em instituições brasileiras.

De acordo com Guzdial, uma forma de incentivar estudantes a entrar no mercado de trabalho, ocorre através do domínio da Computação, o que gera um impacto significativo em sua carreira. Pelo fato de estudantes terem acesso a Computação desde a Educação Básica, essa aproximação é facilitada. Ela ainda cita o seguinte exemplo:

Se você não sabe ler, suas oportunidades de trabalho podem ser limitadas. Se você sabe trabalhar com números, você irá expandir oportunidades de trabalho. Se você sabe se expressar com a Computação, você tem a habilidade que tem alta demanda e você terá acesso aos empregos mais lucrativos na economia moderna (p.10).

Sabe-se que somente pessoas com um maior poder aquisitivo têm acesso à computadores e isso é considerado um problema de justiça social, existindo uma obrigação moral da sociedade de prover oportunidades para um grupo maior de estudantes, porém não é o que ocorre muitas vezes.

1.3.8 Diminuição nas limitações físicas

A Computação é capaz de expandir o horizonte dos estudantes através de diversas técnicas que podem desafiar a gravidade ou eliminar riscos (*e.g.* simulações), facilitando a capacidade de abstrair, criar e lidar com problemas variados e complexos, de forma crítica.

1.3.9 Trabalhar em equipe

Este benefício não é diretamente relacionado com a Computação, mas de acordo com Wing (2006), esta qualidade é essencial para que projetos tenham um bom resultado. Por ser um exercício de persistência, os estudantes, mesmo desenvolvendo individualmente seus projetos, são estimulados a compartilhar com os demais colegas seus erros e acertos e dessa forma trabalham juntos colaborativamente.

1.4 CONTRADIÇÕES

Tem-se que admitir que é difícil encontrar artigos, reportagens, blogs ou declarações que não sejam a favor da adoção do Pensamento Computacional nas escolas. De qualquer forma, após muita pesquisa, encontrou-se algumas opiniões que colocam em dúvida essa abordagem.

Enquanto a academia e a indústria tentam com grande esforço incentivar e comprovar os benefícios que o Pensamento Computacional traz, um grupo muito restrito de pessoas possui argumentos diferentes. Pappano (2017) relata que pesquisadores se preocupam com a tendência cultural atual de nos encontrarmos embriagados com a Computação e vê-la como um conhecimento supremo e melhor que o obtido em outros campos.

Já Setzer (2008) defende que a educação das crianças precisa ser mais humana e menos tecnológica, pois computadores são máquinas que simulam pensamentos restritos, impondo o raciocínio lógico simbólico que elas não possuem maturidade suficiente para entender. Além disso, Setzer complementa que crianças que dominam a tecnologia, não terão facilidade maior para entrar no mercado de trabalho, exemplificando através de pessoas que se encontram em postos importantes no mercado atualmente e não tiveram acesso a computadores quando jovens. A ausência da tecnologia não prejudicou o aprendizado e o avanço profissional deles. Setzer, ainda, sugere:

[...] eu poderia dar algumas recomendações adicionais para programadores. Por exemplo, seria interessante que eles abandonassem o costume corrente de programar

on-line, isto é, usando diretamente o computador. Isso significa voltar ao que se fazia quando não havia microcomputadores pessoais, ou terminais ligados a main frames: programava-se com lápis e papel, e só depois de ter o programa razoavelmente desenvolvido e bem pensado é que se o introduzia na máquina (SETZER, 1996).

Farag (2016), em seu blog, postou um texto denominado “*Please don’t learn to code*” com uma argumentação para que não seja ensinado programação para crianças. De acordo com sua teoria, vender a codificação como um bilhete para a salvação econômica para as massas é desonestidade, pois o PC é apenas mais uma “moda” do Vale do Silício e que passará em breve. A proliferação dos Bootcamps¹ e a divulgação massiva da Computação nos EUA, produz uma quantia relativamente grande de pessoas querendo abrir sua própria *Startup* ou desejando ser Engenheiro de Software sem uma formação adequada.

Farag, ainda em sua linha de raciocínio, alega que as escolas não deveriam ensinar a codificar e se concentrar apenas na solução de problemas. Além disso, ele entende que as linguagens de programação estão em um nível tão abstrato e fácil de usar que ignora a realidade da profissão de um Cientista da Computação, chamando inclusive esse último grupo de uma “Confraria de Programadores”. Por fim, entende-se a argumentação de Farag como sendo um desespero de uma inundação de programadores (mais jovens) que irão concorrer no mercado com ele, tornando-o ainda mais competitivo.

De qualquer forma, é compreensível a preocupação de Farag de que programadores com uma formação pobre, comecem a trabalhar de uma forma prematura, pois a demanda de profissionais pode ser tão alta que eles chegam a conseguir empregos, mas escrevem código de má qualidade, colocando em risco a infraestrutura baseada em software. Um recurso que a sociedade precisa que seja cada vez mais robusta, segura e fácil de usar.

Baseado nesses argumentos, entende-se que a melhor opção para que isso não ocorra é o ensino dos conceitos da Computação desde os primeiros anos da Educação Básica. Um aprofundamento do assunto é apresentado no capítulo seguinte.

1.5 INTEGRAÇÃO NA EDUCAÇÃO BÁSICA

A escola e a sociedade não devem ter entre si muros e muito menos uma natureza impermeável, havendo a necessidade de reinventam-se mutuamente, aprendendo a ultrapassar os desafios comuns. A sociedade espera da escola respostas equilibradas e relevantes no que

¹ Programa de ensino imersivo que foca nas habilidades mais relevantes de determinada área para atuar no mercado do século XXI.

diz respeito à preparação dos seus estudantes para a integração harmoniosa nas tarefas e funções sociais de cada um. Espera-se da escola propostas que permitam proporcionar a todos uma educação moderna e atualizada, incluindo propostas que permitam aos mesmos aprender a usar a tecnologia de forma inovadora e criativa, aprender a conhecer e a usar as tecnologias, apreender a programar, aprender a ser e estar informado, construir novo conhecimento com as tecnologias disponíveis e avaliar de forma crítica o papel das tecnologias na sociedade, na economia, cultura e estilos de vida (RAMOS e ESPADEIRO, 2014).

Na construção trabalhosa e constante dessa base de conhecimento, estamos hoje (e ontem) expostos a ventos de mudança que varrem e abanam as nossas convicções e nos obrigam a questionar permanentemente o que ensinar. A educação pública é avessa aos riscos e à inovação porque é preciso garantir que estamos fazendo o que é certo com crianças.

Entre esses ventos da mudança destacam-se a urgência de movimentos e iniciativas que exigem da escola e das instituições de ensino, não apenas uma mudança cosmética, nem uma forma de uma nova tecnologia ou aplicação, mas de algo mais profundo e duradouro: uma mudança de paradigma, através do ensino do “desenhar, criar e combinar, ao invés de navegar, conversar e interagir” (RESNICK, 2009).

A formação do professor também deve ser levada em consideração. A integração das TICs no currículo e na aprendizagem no campo da sua formação inicial é um desafio, pois durante o processo de tornar o pensamento computacional como uma competência fundamental para o século XXI, os professores também devem obrigatoriamente se aproximar dessas tecnologias e refletir dentro da sua área como a diversidade de conceitos, teorias, modelos e práticas podem ser aplicados dentro de sua disciplina de forma abrangente (utilizando os Quatro Pilares do Pensamento Computacional).

De acordo com estudos realizados por Barcelos e Silveira (2012), uma estratégia para inserção do Pensamento Computacional no Ensino Básico deve ocorrer através de disciplinas pré-existentes no atual currículo, como por exemplo a Matemática, incentivando seu uso também em outras disciplinas para poder atingir um público cada vez maior. CSTA/ISTE (2009) também realizaram um levantamento com diversas propostas de adoção de forma Pluri, Multi e Transdisciplinar. Outras propostas similares são relatadas por Barr e Stephenson (2011). As propostas combinadas encontram-se em Quadro 1 e Quadro 2.

Quadro 1: Sugestões de inserção do PC nas disciplinas de Matemática e Ciências

Conceitos de PC	MATEMÁTICA	CIÊNCIAS
Coleção de Dados	Encontrar uma fonte de dados de uma experiência, por exemplo: cara ou coroa, lançar dados	Coletar dados de um experimento
Análise de Dados	Contar a ocorrência de jogadas, lançamento de dados e análise de resultados	Analisar dados de um experimento
Representação de Dados	Utilizar gráfico de barras e de pizza para representação de dados. Usar conjuntos, listas, representações gráficas, etc. para a visualização de informações	Resumir dados de um experimento
Decomposição de Problemas	Aplicar ordem de operadores	Realizar uma classificação de espécies
Abstração	Usar variáveis na álgebra. Estudar funções de álgebra através de comparação em computadores.	Construir um modelo de uma entidade física
Algoritmos e Procedimentos	Realizar divisões longas, fatorar	Criar um procedimento experimental
Automação	Utilizar ferramentas como: Geometer, Sketch Pad, Star Logo, linhas de código em Python, etc.	Usar simulação de dados
Paralelismo	Resolução de sistemas lineares. Multiplicação de matrizes.	Realizar experimentos com diferentes parâmetros simultaneamente
Simulação	Desenhar uma função em um plano cartesiano e modificar os valores das variáveis	Simular os movimentos do Sistema solar

Fonte: adaptado de CSTA/ISTE (2009) e Barr e Stephenson (2011)

Quadro 2: Sugestões de inserção do PC nas disciplinas de Estudos Sociais, Linguagens e Artes

Conceitos de PC	ESTUDOS SOCIAIS	LINGUAGEM E ARTES
Coleção de Dados	Estudar estatísticas de guerras ou dados populacionais	Identificar padrões em diferentes tipos de frases
Análise de Dados	Identificar as tendências dos dados estatísticos	Representar padrões de diferentes tipos de frases
Representação de Dados	Resumir e representar tendências	Escrever um rascunho
Abstração	Resumir fatos. Deduzir conclusões dos fatos.	Uso de metáforas e analogias. Escrever uma história com diversas vertentes.
Algoritmos e Procedimentos	-	Escrever instruções
Automação	Usar planilhas eletrônicas	-
Paralelismo	-	Utilizar o corretor ortográfico
Simulação	Incentivar com jogos que utilizem bases históricas	Encenação de uma história

Fonte: adaptado de CSTA/ISTE (2009) e Barr e Stephenson (2011)

Outros diversos exemplos de integração do Pensamento Computacional na Educação Básica em outras disciplinas são amplamente descritos em livros como Bunce (2015), Burrett (2016), Patterson (2016) e Bird *et al.* (2017) através de lições que utilizam computadores; e Caldwell e Smith (2017) sem a necessidade de usar qualquer equipamento eletrônico. Wang (2016) também faz uma abordagem diferenciada para cursos superiores.

A integração do PC na Educação Básica também é analisada na pesquisa de Valente (2016), onde realiza um levantamento entre diferentes autores e define seis categorias de abordagens no ensino dos conceitos da Computação na Educação Básica, a saber:

- 1) Atividades sem o uso das tecnologias: utilizar abordagens lúdicas, truques de mágica e competições para mostrar às crianças o tipo de pensamento que é esperado por um cientista da Computação.
- 2) Programação em Scratch: linguagem de programação baseada em blocos visuais, projetados para facilitar a manipulação da mídia por programadores novatos.
- 3) Robótica pedagógica: utilização de aspectos/abordagens da robótica industrial em um contexto no qual as atividades de construção, automação e controle de dispositivos robóticos propiciam a aplicação concreta dos conceitos em um ambiente de ensino e de aprendizagem.
- 4) Produção de narrativas digitais: consiste no uso das TICs da produção de narrativas que tradicionalmente são orais ou impressas, também conhecidas como histórias digitais, narrativas interativas ou *digital storytelling*.
- 5) Criação de jogos: desenvolvimento de sistemas composto pela estética do visual e som, narrativa contando a história do jogo, mecânica de regras do jogo e a tecnologia usada para produzir um jogo eletrônico.
- 6) Uso de simulações: uso de softwares que criam um mundo-faz-de-conta para observar fenômenos que não sejam passíveis de serem desenvolvidas no mundo real.

Cada uma das abordagens possui uma característica diferente para se chegar ao objetivo em comum: o desenvolvimento do Pensamento Computacional. Note que todas as abordagens necessitam o uso de equipamentos e softwares específicos, exceto o primeiro. Uma abordagem sem o uso de tecnologias, também conhecido na literatura como “Pensamento Computacional Desplugado” ou “*Unplugged*”, vem a ser um dos focos principais desta pesquisa, levando-se em consideração sua facilidade da aplicação em diferentes realidades econômicas e sociais no Brasil, uma das principais motivações do projeto (vide Capítulo 3, p.

105). Para entender melhor o que é e como aplicar o Pensamento Computacional Desplugado, faz-se um levantamento mais aprofundado da abordagem e apresenta-se o estado-da-arte.

1.6 ABORDAGEM DESPLUGADA

Muitos tópicos importantes da Computação podem ser ensinados sem o uso de computadores. A abordagem desplugada introduz conceitos de hardware e software que impulsionam as tecnologias cotidianas a pessoas não-técnicas. Em vez de participar de uma aula expositiva, as atividades desplugadas ocorrem frequentemente através da aprendizagem cinestésica¹ (e.g. movimentar-se, usar cartões, recortar, dobrar, colar, desenhar, pintar, resolver enigmas, etc.) e os estudantes trabalham entre si para aprender conceitos da Computação. Trabalhar com objetos tangíveis do mundo real é um princípio central do construcionismo de Papert (Papert e Harel, 1991) (que se baseia no construtivismo). Assim, os princípios construtivistas sustentam as estratégias de usar abordagens mais cinestésicas e ativas no ensino da Computação em sala de aula.

Pode-se exemplificar essas atividades através das sugeridas por Bell (2015), onde é proposto um jogo, cujo objetivo é completar um mapa de piratas. Na verdade, trata-se de problema que pode ser caracterizado como um autômato finito e a atividade consiste em percorrer o campo do jogo, na tentativa de encontrar um caminho para uma ilha onde se encontra um tesouro. Outro exemplo é vestir um personagem com roupas para diferentes ocasiões (LIUKAS, 2015, p. 73), realizando uma referência às condições que são impostas dentro de um código (*if/else*). Na Figura 14 pode-se conhecer outra atividade que exercita uma maneira diferenciada de guardar informações que formam uma imagem, contando-se primeiramente a quantidade de caixas brancas, pretas, brancas, pretas e assim sucessivamente.

Figura 14: Exemplo de atividade de desenho

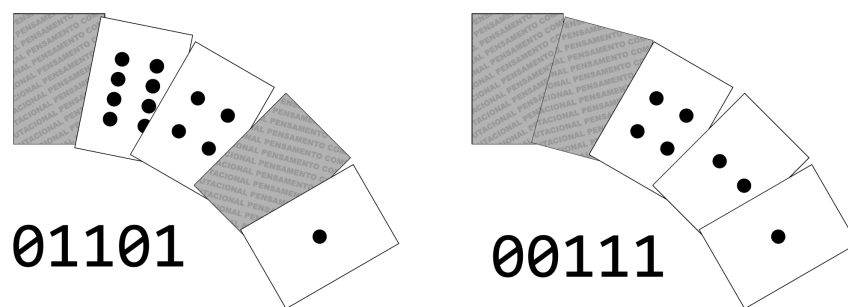
												4,1 (4 brancos, 1 preto)
												3, 3
												2, 5
												1, 7
												2, 5
												2,5
												2, 2, 1, 2 (2 brancos, 2 pretos, 1 branco, 2 pretos)
												2, 2, 1, 2

Fonte: próprio autor

¹ Percepção dos movimentos musculares, peso e posição dos membros, por meio de estímulos próprios (MICHAELIS MODERNO, 2017)

Na Figura 15 demonstra-se outra atividade para ensino de números binários. O jogo é composto de cinco cartas com 1, 2, 4, 8 e 16 pontos cada uma e na parte traseira sem nenhuma informação. As cartas são colocadas lado a lado, iniciando d'a esquerda para a direita com a carta de 16 pontos, depois a com 8 pontos e assim sucessivamente até a última carta à direita com apenas um ponto. Cada carta que esteja com sua face para cima, representa o número 1 binário e as cartas que estão viradas para baixo representam 0 (zero).

Figura 15: Baralho para ensino de números binários



Fonte: próprio autor

O surgimento do Pensamento Computacional Desplugado não é muito claro, pois a necessidade de abstração para a criação de qualquer software e hardware são essenciais. É importante constar também que o uso de exemplos físicos e materiais escolares são comuns para simular o comportamento de máquinas até os dias atuais em cursos de graduação. Quando se trata de salas de aulas da educação básica, os primeiros registros são encontrados a partir de 1997 quando Bell *et al.* (1997) lançaram um rascunho de um livro em formato digital denominado "*Computer Science Unplugged... Off-line activities and games for all ages*" para professores de todos os níveis escolares que queriam dar um diferencial a seus alunos com os seguintes objetivos (TAUB *et al.*, 2009):

- Aumentar o interesse dos estudantes em CC;
- Avaliar se os alunos perceberão a CC como algo mais desafiador, intelectualmente estimulante e cooperativo do que anteriormente;
- Conduzir os alunos para uma melhor compreensão do que é CC e evitar confundir-lo com a programação;
- Promover CC como uma carreira para mulheres.

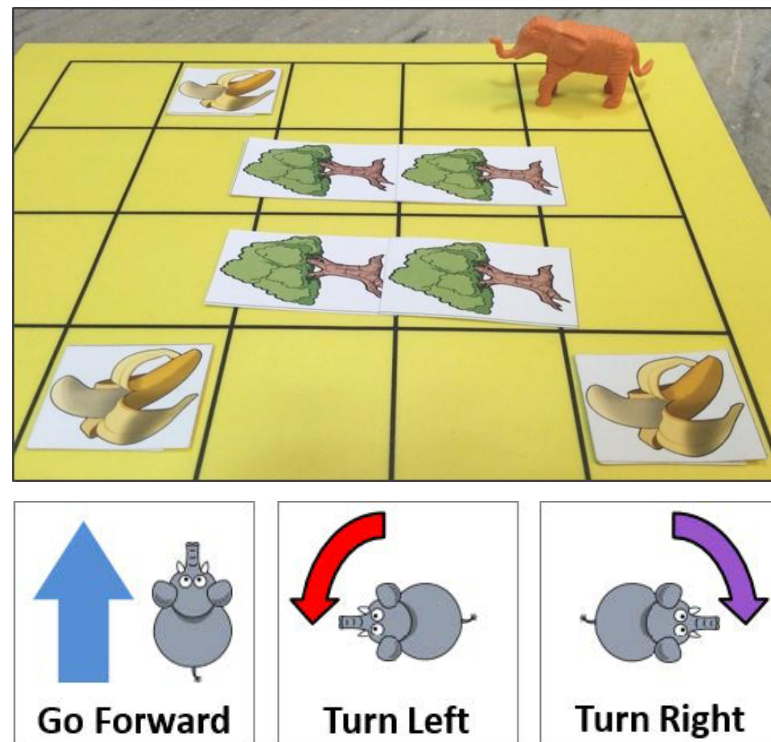
A ideia foi muito bem recebida pelos demais professores e a própria academia. Devido a qualidade do material publicado, a *Association for Computing Machinery* (ACM) recomendou que as atividades contidas no livro fizessem parte do currículo proposto pela *Computer Science Teachers Association* (CSTA) dos Estados Unidos da América (ASSOCIATION FOR COMPUTING MACHINERY, 2003). Até a publicação deste artigo, o livro *CS Unplugged* se encontra na versão 3.1 e pode ser acessado no site do projeto (BELL *et al.*, 2015). Outro exemplo de atividade desenvolvida por Bell *et al.* será apresentado no Capítulo 4 (p. 112).

Embora o uso de atividades de programação de computadores seja a principal abordagem para ensinar habilidades de PC em escolas, educadores e pesquisadores também estão fazendo uso da abordagem desplugada, como afirmado em uma revisão de literatura sistemática que estudou 125 artigos focados em PC (KALELIOGLU *et al.*, 2016). Conclusões semelhantes são encontradas em um levantamento a respeito da metodologia de ensino da Computação. Nessa pesquisa, 13% dos 357 professores participantes afirmam que eles usam atividades desplugadas em suas aulas de Computação (SENTANCE e CSIZMADIA, 2015).

No entanto, enquanto a eficácia da programação em computadores para promover o desenvolvimento de aptidões de PC está sendo amplamente investigada (LYE e KOH, 2014), o mesmo não ocorre integralmente com a abordagem desplugada.

A maioria das pesquisas que utilizam atividades desplugadas visa promover o interesse dos estudantes para a Computação. Pode-se destacar o trabalho de Unnikrishnan *et al.* (2016) que é executado em áreas rurais e desprivilegiadas da Índia. Devido ao elefante ser um animal bastante comum e respeitado na região, os pesquisadores propuseram uma atividade para crianças, envolvendo esse personagem. O jogo denominado “*Haathi Mera Saathi*” (Meu Amigo Elefante) é composto por um tabuleiro quadriculado, uma miniatura de elefante e fichas com desenhos de bananas e árvores. As crianças deveriam, então, usar cartas com instruções para que o elefante visitasse todos os quadrados onde houvesse uma banana. Na Figura 16 consta uma foto do tabuleiro com algumas cartas utilizadas no jogo.

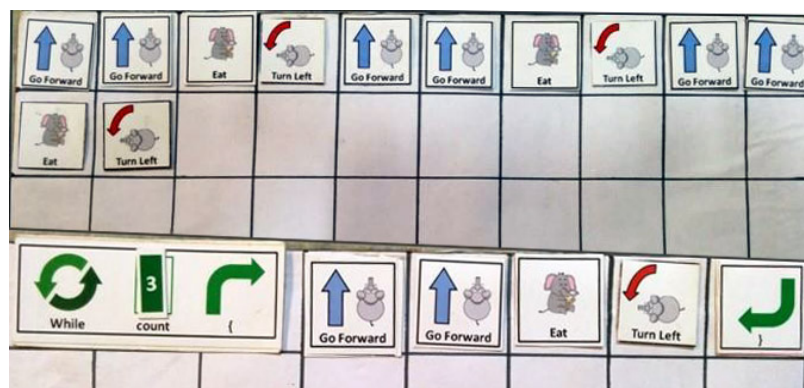
Figura 16: Tabuleiro e cartas do jogo *Haathi Mera Saathi (My Elephant Friend)*



Fonte: Unnikrishnan *et al.* (2016)

As cartas com as instruções, então, eram posicionadas em outro tabuleiro (Tabuleiro de Programação) com a sequência de passos para atingir o objetivo. Na Figura 17 é exemplificado a resolução do desafio proposto na Figura 16 de duas formas: com e sem a utilização de iterações (*loops*).

Figura 17: Tabuleiro de Programação do Jogo *Haathi Mera Saathi (My Elephant Friend)*



Fonte: Unnikrishnan *et al.* (2016)

Outros jogos similares que não necessitam o uso de computadores para exercitar o Pensamento Computacional já existem comercialmente no mercado desde 1994, como por exemplo o “Robo Rally”. O jogo é composto por um tabuleiro e cartas de instruções para os

robôs que são os personagens principais. Com as cartas é possível mover os personagens no tabuleiro de tal forma que eles consigam visitar todas as casas (*checkpoints*) indicadas. As regras do jogo incluem também cartas de atualização do robô, conceitos de variáveis, consumo de energia, nível de dano, perda de memória (reboot) e, principalmente a criação de estratégias para completar os objetivos do jogo e ainda causando danos nos robôs dos demais competidores. Tendo em vista que o jogo possui uma alta complexidade, ele é recomendado para crianças com doze anos ou mais de idade e suporta de 2 a 6 jogadores em sua última edição de 2015. Devido ao seu constante sucesso em vendas, diversas edições já foram lançadas. O tabuleiro e as demais peças que compõe o jogo encontram-se na Figura 18.

Figura 18: Jogo Robot Rally (edição 2015)



Fonte: Garfield e Hasbro Gaming (2015)

Tendo em vista o aumento da importância do Pensamento Computacional dado pelos governos e a consequente elevada demanda e oportunidades de negócios, outras iniciativas de jogos surgiram, porém com uma abordagem mais infantil. Faz-se a seguir um levantamento dos jogos de cunho computacional que são encontrados exclusivamente no mercado exterior e que não fazem uso de computadores:

- Bits & Bytes: de acordo Games4Learning (2014), é um jogo de cartas para ensinar lógica, resolução de problemas e pensamento crítico. Para iniciar, as cartas são colocadas de forma organizada e viradas para baixo em cima de uma mesa formando um tabuleiro, onde os jogadores deverão explorar o espaço, desvirando as cartas, controlando um dos personagens (Bit, Byte, Perl ou Data) com outras cartas de instruções. Existe uma possibilidade de utilizar funções (procedures), porém as regras

não são claras e recomenda-se precaução ao usá-la com crianças. O objetivo principal é chegar ao centro do tabuleiro, onde se localiza o planeta dos personagens, chamado de RAM. No percurso, ao desvirar as cartas, o jogador pode encontrar elementos que podem dificultar seu trajeto. Recomendado para crianças a partir de quatro anos.

Figura 19: Jogo Bits & Bytes



Fonte: Games4Learning (2014)

- Jogo Robot Turtles: de acordo com Shapiro e Thinkfun (2014), o jogo de tabuleiro que nasceu de um projeto Kickstarter, possui características muito próximas do Jogo Bits & Bytes, porém com um maior grau de maturidade. Por o tabuleiro ser uma peça única, foram adicionadas novas instruções, ou seja, a possibilidade de criação clara de funções (*procedures*) e um aumento expressivo na quantidade de cartas (224 cartas), acabando virtualmente com a limitação de passos necessários para se chegar ao centro do tabuleiro. Recomendado para crianças a partir de quatro anos. Vide a Figura 20 com um exemplo de uso do jogo.

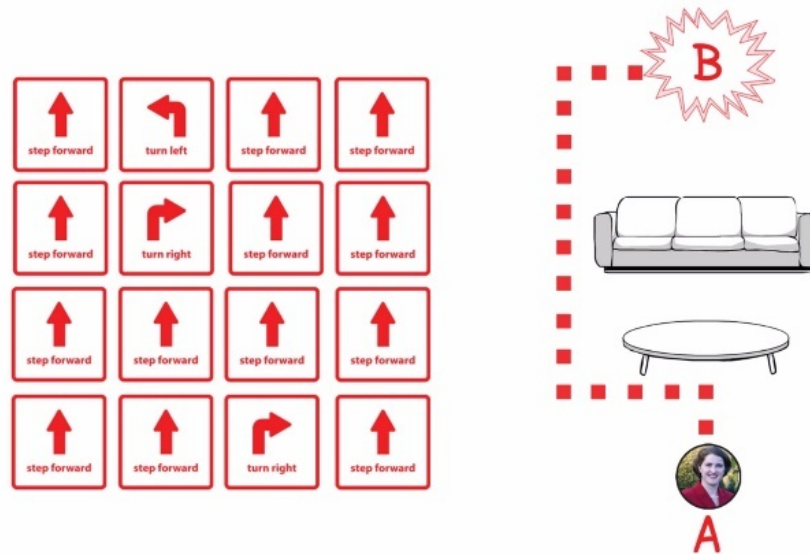
Figura 20: Jogo Robot Turtles



Fonte: Shapiro e Thinkfun (2014)

- Littlecodr: o jogo minimalista, desenvolvido pelos pesquisadores Greenhill e Slee (2015), é composto por 80 cartas com instruções e 20 contendo sugestões de aplicações ou desafios. O jogo de cartas não tem um uso exclusivo e pode ser utilizado em diversas situações genéricas, ou seja, não havendo a necessidade de um tabuleiro. De acordo com os criadores, o jogo apresenta às crianças de 4 a 8 anos conceitos de codificação, lógica, planejamento, pensamento sequencial, prototipação, solução de problemas, além de ensiná-las a contar, ler, seguir instruções e distinguir a esquerda e direita. Um exemplo de aplicação para tentar desviar de uma mesa e um sofá, encontra-se na Figura 21.

Figura 21: Execução de um programa no jogo Littledodr



Fonte: Greenhill e Slee (2015)

Percebe-se que as regras dos jogos recém relacionados têm um foco voltado para movimentos em um espaço pré-definido e com objetivos muito similares: utilização de cartas para representar as etapas necessárias para completar um objetivo. É evidente que o Pensamento Computacional não foca apenas no movimento de personagens em um tabuleiro através de combinações de instruções que o usuário define. Deve-se levar em consideração que existem outros jogos que trabalham outros conceitos essenciais para a Computação, direta e indiretamente. Além disso, alguns jogos podem exercitar o trabalho colaborativo, uma das aptidões do Pensamento Computacional. São alguns exemplos:

- Giggle Chips: jogo de cartas desenvolvido por Ellis e Charlotte (2017) que ajuda na alfabetização das crianças e a conhecer números binários. Ao invés de usar palavras com animais (A para Abelha, B para Boi, C para Cavalo, etc.), usam-se termos voltados para a área da tecnologia e Computação, tais como App, Blog, Coding, Digital, etc. Na parte traseira da carta há também outros dois jogos, sendo o primeiro de conversão de números binários e segundo de perguntas e respostas. O jogo é recomendado para crianças de cinco a nove anos e está disponível somente na língua inglesa. As partes frontal e traseira das cartas encontram-se na Figura 22 e Figura 23, respectivamente.

Figura 22: Jogo Giggle Chips (frente)



Fonte: Ellis e Charlotte (2017)

Figura 23: Jogo Giggle Chips (verso)



Fonte: Ellis e Charlotte (2017)

- Chocolate Fix: jogo desenvolvido por Engelberg (Engelberg e Thinkfun, 2012), composto por um tabuleiro com seis reentrâncias para encaixe de peças que lembram chocolates e um bloco com 40 desafios. Cada peça possui uma cor e um formato que não se repetem. Estes devem ser posicionados conforme pistas que constam no desafio, onde o jogador deve fazer uso de técnicas de reconhecimento de padrões e criar estratégias para que consiga posicionar todos os doces de uma forma que atenda a todos os padrões, os quais fazem parte do desafio. O jogo pode ser disputado individualmente ou com mais pessoas a partir de 8 anos de idade. Um exemplo de desafio é apresentado na Figura 24.

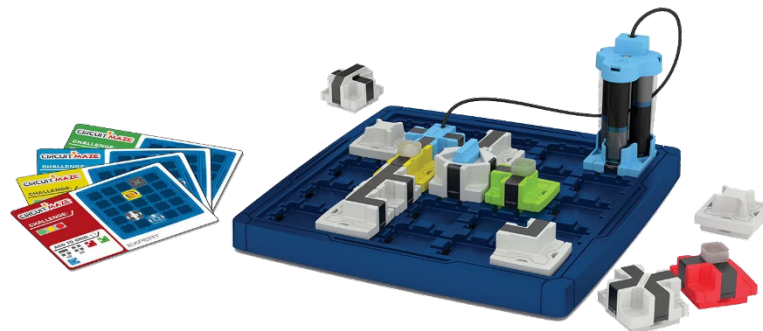
Figura 24: Chocolate Fix



Fonte: Engelberg; ThinkFun (2012)

- Circuit Maze e Laser Maze: o jogo Circuit Maze (YAKOS e THINKFUN, 2015) é composto por um tabuleiro com vinte e cinco reentrâncias para encaixe de peças que podem, ou não, conduzir eletricidade. Cada peça possui uma característica, tais como: condutor em linha reta, em curva, pontes, transversal, interruptor ou lâmpada. Com essas peças é possível simular circuitos (lógica booleana), criar estratégias para a resolução de 60 problemas propostos, além de exercitar atividades relacionadas às disciplinas de Ciências e Física (vide Figura 25). O jogo Laser Maze (HOOPER e THINKFUN, 2012) segue a mesma linha de raciocínio, porém utiliza um raio laser e espelhos para replicar a luz, a fim de atingir o alvo pré-definido no desafio. Ambos os jogos são recomendados para crianças a partir de 8 anos e pode ser jogado sozinho ou com mais pessoas.

Figura 25: Jogo Circuit Maze



Fonte: Yakos e Thinkfun (2015)

Figura 26: Jogo Laser Maze



Fonte: Hooper e Thinkfun (2012)

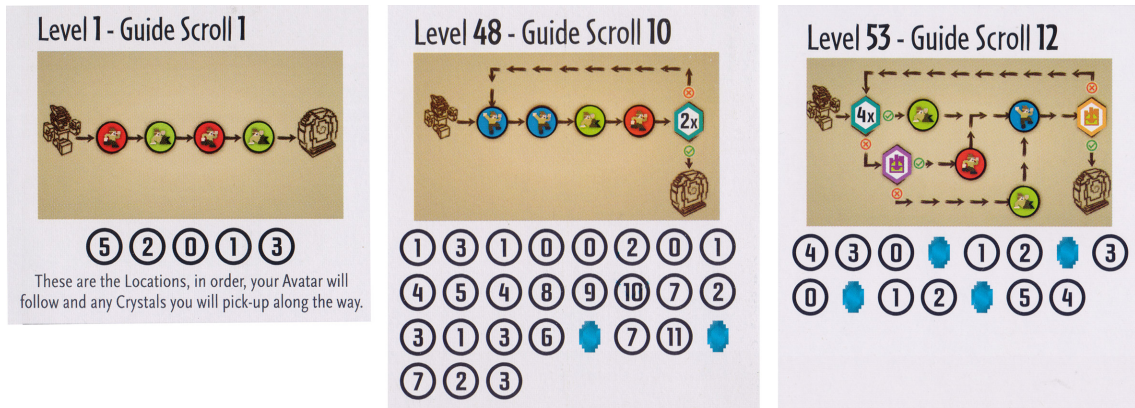
- Code Master: o jogo criado por Engelberg (ENGELBERG e THINKFUN, 2015) é composto por dois tabuleiros: o primeiro possui dez mapas que lembram autômatos que possuem ligações de diferentes cores entre os nodos e o segundo tabuleiro serve para o jogador registrar o trajeto que optou por seguir para chegar de um ponto a outro (vide Figura 27). Após o jogador escolher um dos 60 níveis de dificuldade, ele deve posicionar o personagem e o portal de saída nos nodos indicados, abrir o segundo tabuleiro na página referente ao desafio e utilizar a quantidade de fichas coloridas definidas no exercício. O objetivo do jogo é chegar até o portal utilizando todas as fichas coloridas. O grau de dificuldade aumenta no momento em que há a necessidade de coletar cristais em certos nodos do mapa. Este jogo foi projetado para ensinar diversas características computacionais, sendo algumas delas: abstração, resolução de problema, repetição (*looping*), reconhecimento de padrão, decomposição, condicionais e, sem esquecer o algoritmo. Algumas soluções selecionadas do jogo encontram-se na Figura 28.

Figura 27: Jogo Code Master (tabuleiros e peças)



Fonte: Engelberg e Thinkfun (2015)

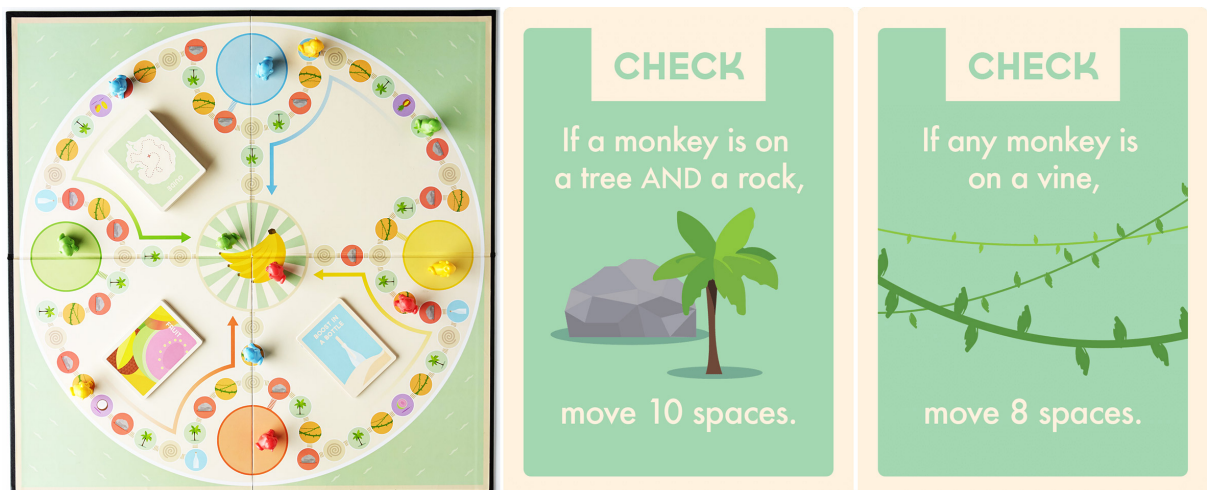
Figura 28: Algumas soluções do jogo Code Master



Fonte: Engelberg e Thinkfun (2015)

- Code Monkey Island: criado por Sidhu (2014), quando participou de campanha de KickStarter onde necessitava de 15 mil dólares para lançar o jogo e conseguiu arrecadar quase 40 mil dólares. Ele é composto de um tabuleiro, 80 cartas e 12 miniaturas de macacos. O jogo possui regras muito parecidas com o tradicional “Ludo”, porém sem o uso de dados. Os dados são substituídos por uma série de instruções e condições escritas nas cartas. De acordo com o criador, o jogo exercita declarações condicionais, repetições, operadores e lógica booleana, execução de tarefas e estrutura de dados. Ele foi desenvolvido para jogadores a partir de 8 anos. Uma imagem do tabuleiro e duas cartas encontra-se na Figura 29.

Figura 29: Jogo Code Monkey Island



Fonte: Sidhu (2014)

- CodingIsGood: jogo de cartas para dois jogadores com similaridades ao tradicional jogo “Super Trunfo” e desenvolvido por MathAndCoding (2017). Na parte frontal da carta,

é apresentado ao jogador um algoritmo em Python e na parte traseira a resposta que deve ser visualizada pelo oponente. O jogador deve, então, “executar” o código mentalmente e dizer em voz alta a resposta. Caso a resposta estiver correta, a carta é colocada em uma pilha, senão descartada. O jogador com a maior pilha é o vencedor da partida. Existem três níveis de dificuldade (básico, intermediário e avançado). Recomendado para crianças a partir de 10 anos.

Figura 30: Cartas do jogo CodingIsGood



Fonte: MathAndCoding (2017)

- CodingFarmer: desenvolvido por MathAndCoding (2015) e similar ao tradicional “Jogo da Escadinha” ou “Jogo da Trilha”, onde até quatro jogadores colocam uma peça no formato de um trator no início do caminho e, através de comandos em Inglês e na linguagem de programação Java que constam nas cartas, devem guiá-lo até o celeiro. Algumas casas (posições) no trajeto podem influenciar o código a ser executado. Recomendado para jogadores a partir de sete anos. O tabuleiro e algumas cartas encontram-se na Figura 31.

Figura 31: Tabuleiro e cartas do jogo CodingFarmer



Fonte: MathAndCoding (2015)

Outro formato de material que pode ser encontrado são os livros. Cada um deles, aqui citados, possui uma estratégia diferente de trabalhar os conceitos que envolvem o Pensamento Computacional. São eles:

- Ryan (1987) publicou um livro contando uma das histórias de Sherlock Holmes, um personagem britânico fictício que, através de sua experiência como detetive, tenta solucionar um crime que é narrado em suas páginas. Porém, esse livro não é uma peça ordinária, pois trabalha conceitos como: variáveis, constantes, números randômicos, condicionais, repetições, banco de dados, indexação, entre outros. Na Figura 32 é apresentada a ficha utilizada para que o leitor do livro possa apontar as características do seu personagem, tais como: nome, habilidades, equipamentos e dinheiro em posse, apontamentos que se façam necessários e uma relação de pistas que são encontradas durante a aventura. A experiência é muito similar a jogos de RPG, porém faz uso de instruções e condições bem claras em suas 372 seções textuais, como por exemplo: ir até uma página específica de acordo com a ação que o leitor deseja realizar (vide Figura 33), seguir até uma seção do livro, caso você tenha uma pista necessária (vide Figura 34), ou conseguir levantar um objeto pesado se o personagem possuir um preparo físico. A publicação do livro ocorreu no Brasil somente no ano de 1994 (Ryan, 1994).


Figura 32: Fichas do Personagem e Pistas

FICHA DO PERSONAGEM		
Nome: _____		
Habilidade	Bônus	Equipamento:
Preparo físico	_____	1)
Astúcia	_____	2)
Observação	_____	3)
Intuição	_____	4)
Comunicação	_____	5)
Erudição	_____	6)
Dinheiro: ___ pence		7)
___ xelins		8)
___ guinéus		9)
___ libras		10)
		11)
Apontamentos:		

FOLHA DE PISTAS	
<input type="checkbox"/> A	_____
<input type="checkbox"/> B	_____
<input type="checkbox"/> C	_____
<input type="checkbox"/> D	_____
<input type="checkbox"/> E	_____
<input type="checkbox"/> F	_____
<input type="checkbox"/> G	_____
<input type="checkbox"/> H	_____
<input type="checkbox"/> I	_____
<input type="checkbox"/> J	_____
<input type="checkbox"/> K	_____
<input type="checkbox"/> L	_____
<input type="checkbox"/> M	_____
<input type="checkbox"/> N	_____
<input type="checkbox"/> O	_____
<input type="checkbox"/> P	_____
<input type="checkbox"/> Q	_____
<input type="checkbox"/> R	_____
<input type="checkbox"/> S	_____
<input type="checkbox"/> T	_____
<input type="checkbox"/> U	_____
<input type="checkbox"/> V	_____
<input type="checkbox"/> W	_____
<input type="checkbox"/> X	_____
<input type="checkbox"/> Y	_____
<input type="checkbox"/> Z	_____

Fonte: Ryan (1994)

Figura 33: Exemplo de condicional na história de Sherlock Holmes



A carta datilografada, sem remetente, diz o seguinte:

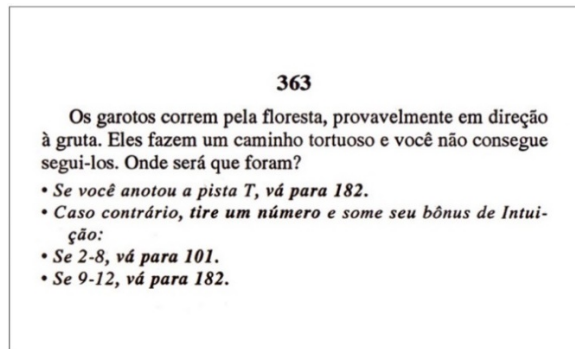
Para um garoto esperto:
Caia fora, conforme já lhe foi dito. O gato era curioso também. Seja um bom garoto e cuide de sua própria vida, assim pode ser que você se torne um cavalheiro.

No momento em que desce os degraus do prédio do correio, você vê o pai de Muskrat passando dentro de uma carruagem fechada de quatro rodas que pára mais adiante num cruzamento. O telegrama de Holmes reverbera na sua cabeça.

- Se você tentar se agarrar à parte de trás da carruagem de Rafferty, vá para 137.
- Caso contrário, vá para 250.

Fonte: Ryan (1994)

Figura 34: Exemplo de condicional utilizando pistas na história de Sherlock Holmes



Fonte: Ryan (1994)

- Lift-the-Flap Computers and Coding: livro escrito por Dickins e Nielsen (2015) e originalmente lançado na Inglaterra para ensinar crianças a partir de 7 anos conceitos da Computação. A publicação é composta por 16 páginas e através do artifício de abas ensina o que é um computador e o que há dentro dele, o significado de codificar, como os computadores “pensam”, como executar instruções nele, o que são as linguagens de programação, como criar um código eficiente, como a internet funciona e um breve histórico da Computação. Não há até o momento uma tradução do livro para o português. Um exemplo das atividades proposto pelo livro encontra-se na Figura 35.

Figura 35: Exemplo do livro Lift-the-Flap Computers and Coding



Fonte: Dickins e Nielsen (2015)

- Guia Mangá de Banco de Dados: o livro escrito por Takahashi *et al.* (2009) relata a história de como um grupo de jovens que organiza seu negócio com o auxílio de um banco de dados (BD), através de uma história em quadrinhos (mangá), tornando o aprendizado muito mais acessível para jovens. Além da história, a peça possui exercícios no final de cada capítulo e trabalha conteúdos relativos a BD, tais como: o porquê e como usá-lo, terminologias e modelos de BD, linguagens de BD e estrutura de sistemas de BD. Como o livro trabalha esses aspectos de uma maneira mais aprofundada, não é recomendado para crianças. Um exemplo da metodologia utilizada pelo livro encontra-se na Figura 36.

Figura 36: Exemplo do Guia Mangá de Banco de Dados

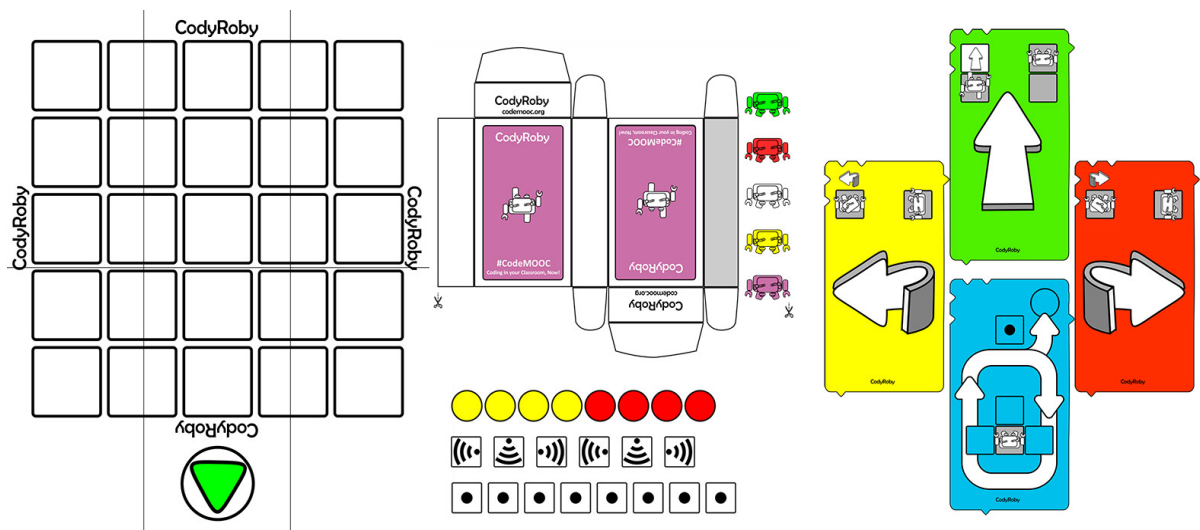


Fonte: Takahashi *et al.* (2009)

- *Hello Ruby – Adventures in Coding*: escrito por Liukas (2015), relata a história de uma pequena menina criativa que deve encontrar soluções para diversos problemas utilizando o PC. A história é contada até a metade do livro e, na seção seguinte, são propostas atividades utilizando os quatro pilares do PC para serem exercitados pela criança com a ajuda dos pais, baseados em elementos contidos no livro (*e.g.* personagens e exemplos). Algumas atividades desse livro foram adaptadas e utilizadas nesta pesquisa e são demonstradas no Capítulo 4. Não há, até o momento, uma tradução do livro para o português.
- *CodyRoby*: escrito por Bogliolo (2016), explica as regras de um jogo de tabuleiro onde o objetivo é controlar um robô com a utilização de três instruções básicas (andar para a frente, girar para a direita e girar para a esquerda) para atingir um objetivo. Sua

abordagem é similar aos jogos de tabuleiro Bits&Bytes e Robot Turtles que já foram citados anteriormente. O que diferencia dos demais jogos é a possibilidade de recortar os tabuleiros, cartas e demais peças necessárias na última seção do livro ou realizar o download sem nenhum custo no site¹ do livro.

Figura 37: Peças do livro CodyRoby



Fonte: Bogliolo (2016)

Enfim, encontramos no mercado diversas opções de aplicação dos fundamentos da Computação em um formato desplugado, porém não se pode deixar de citar também alguns jogos tradicionais que também possuem objetivos de exercitar as aptidões de resolução de problemas e estratégia: Xadrez, Go, Gamão, entre outros. Nota-se que os jogos com um foco em fundamentos da Computação tiveram um grande salto nos últimos três anos, comprovando a alta demanda desse tipo de aptidão.

Taub *et al.* (2012) levaram a abordagem desplugada para a sala de aula, mais especificamente as atividades propostas inicialmente por Bell *et al.* (1997) e após a aplicação, através do uso de questionários e entrevistas, investigou-se os efeitos dessas atividades na visão de alunos do ensino fundamental. Os resultados mostram que “embora a maior parte dos estudantes entenderem o que é a Computação, eles também compreendiam como se o computador fosse a essência da Computação e não como uma ferramenta, contrariamente à intenção das atividades”. Com objetivos e resultados semelhantes, outro projeto de Computação Desplugada foi implementado como parte de um programa de divulgação com duração de um ano para estudantes do ensino médio com o objetivo de "estimular a próxima geração de alunos

¹ <http://codemooc.org/codyroby/>

de graduação em pesquisa na área da Computação" (FEASTER *et al.*, 2011). Os resultados mostram que o programa não teve impacto percebido sobre os alunos, nem uma melhor compreensão do conteúdo envolvido ou sobre as suas atitudes em relação à Computação.

Resultados contrários, porém, são atingidos com um grupo de pesquisadores que visitou diversas turmas de quarto ano com o objetivo de aumentar o interesse dos estudantes na área da Computação com uso de atividades desplugadas. Os resultados, baseados em pré e pós-teste, mostram uma melhoria na confiança e interesse, tanto na área da Computação, como na de Matemática (LAMBERT e GUIFFRE, 2009).

O uso da abordagem desplugada para treinamento de professores também foi estudado por Curzon *et al.* (2014). A pesquisa se deu através de uma série de oficinas organizadas para explorar a eficácia dos métodos desplugados para introduzir os educadores aos tópicos da Computação. A avaliação, baseada em formulários, sugere que “as atividades desplugadas proporcionam uma sessão inspiradora e divertida para os professores e que eles também acham útil, interessante e melhora sua confiança”.

Na mesma linha, Curzon (2013) descreve como as atividades desplugadas incorporadas em histórias podem ser usadas para ensinar PC a professores. Especificamente, o artigo apresenta dois exemplos, o primeiro é baseado no problema de ajudar as pessoas com Síndrome do Encarceramento a se comunicar e a segunda baseada em truques mágicos. Após uma oficina de desenvolvimento profissional de 2 horas para professores, 100% dos participantes afirmaram que o evento lhes deu ideias úteis e viáveis para a sala de aula. Um resultado similar foi relatado por Faber *et al.* (2017), onde, após um total de 14.040 horas-aula de ensino de PC em 26 escolas concluíram que a abordagem desplugada é uma valiosa alternativa em relação à programação em computadores.

Como pode ser percebido, a maioria das pesquisas anteriores se concentra em medir o entusiasmo e o interesse dos participantes pela Computação, mas não há avaliação se os participantes desenvolvem suas habilidades de PC. A revisão da literatura também destaca que, há sim uma necessidade de mais pesquisas empíricas que forneçam mais evidências sobre a eficácia das atividades desplugadas no desenvolvimento das habilidades do PC, especialmente quando se trata das escolas primárias. Consequentemente, neste documento, tenta-se esclarecer esse assunto de maneira mais aprofundada, porém como medir essa alteração de aprendizagem? Para tentar responder a essa dúvida, se fará uma revisão bibliográfica a respeito do assunto na seção seguinte.

1.7 AVALIAÇÃO

A respeito da Avaliação do PC, é um ponto bastante discutido, porém sem consenso acadêmico. O National Research Council (2011, p. 5) destaca a falta de metodologia adequada para avaliação do Pensamento Computacional em escolas primárias, assim como pesquisas formais com relação à melhoria cognitiva de alunos sobre conceitos envolvidos no PC. Sem a devida avaliação, o Pensamento Computacional na sala de aula não terá probabilidade de seguir o caminho de sucesso em nenhum currículo (GROVER, 2013), ou seja, além da necessidade de se avaliar a eficácia de qualquer abordagem curricular ao integrar o PC, necessita-se definir atributos que permitam aos educadores avaliar o que os estudantes aprenderam. Até o momento, ainda são escassos os trabalhos e propostas neste domínio da avaliação das aprendizagens no quadro do PC. Quando encontrado, as pesquisas utilizam uma linguagem de programação (*e.g.* Python) e programação visual (*e.g.* Scratch), sem ênfase na Programação Desplugada (BOUCINHA *et al.*, 2017).

A avaliação de conceitos da Computação em escolas tem sido um desafio também para os educadores, conforme consta no relatório emitido pelo National Research Council (2011, p. 46). A literatura também cita a necessidade de criação de *frameworks*, métodos ou métricas para avaliação do desempenho dos estudantes quando envolvidos em atividades relacionadas ao Pensamento Computacional, porém não existe ainda um consenso a respeito (BARCELOS e SILVEIRA, 2012), (SEITER e FOREMAN, 2013) e (BRENNAN, 2011).

Barcelos e Silveira (2012, p. 16) propõem a avaliação dos estudantes baseado em três dimensões e quinze categorias para análise dos indicadores para ser utilizado em conjunto com atividades em um programa de programação visual denominado Scratch. A proposta completa encontra-se no Quadro 3.

Quadro 3: Proposta de dimensões, Categorias e Indicadores de Barcelos e Silva (2014)

TEMAS/DIMENSÕES	CATEGORIA DE ANÁLISE	INDICADORES
Conceitos Computacionais	Categorias de comandos: Movimento, Aparência, Som, Caneta, Controle, Sensores, Operações, Variáveis	Blocos de programação de cada categoria de comando presentes nos projetos Scratch
Práticas computacionais	Incremental e interativo; Teste e depuração; Reutilização e mistura; Abstração e modularização	Expressões nos testemunhos (discurso escrito e falado): por exemplo: “fazer experiências”, “tentativa e erro”, “descobrimo”

TEMAS/DIMENSÕES	CATEGORIA DE ANÁLISE	INDICADORES
Perspectivas computacionais	Expressar, Conectar, Questionar	Expressões nos testemunhos (discurso escrito e falado): “Criar diálogo”, “criar personagens”, “recriar um evento”, “criar um jogo”, “movimentar”, “escrever”, “expressar”, “comunicar”, “em conjunto”, “Colocar questões”, “explicar”, “ensinar”, “responder”, “fazer novas cores”, “demonstrar”, “interagir”, “colegas”

Fonte: Adaptado de Barcelos e Silva (2012)

Outra perspectiva de avaliação também foi apresentada por Seiter e Foreman (2013, p. 61), onde realiza uma avaliação mais simplificada, denominada como Variáveis de Evidências, através de uma generalização de treze indicadores sendo que cada indicador possui até três níveis de desempenho, todos voltados também para o uso na programação visual Scratch. Os três níveis de desempenho também podem ser encontrados em Rodriguez (2015) como “Insatisfatório”, “Parcialmente Proficiente” e “Proficiente”. A relação de indicadores e sua gradação encontra-se no Quadro 4. Neste trabalho também é proposta outra alternativa de avaliação (p. 63), levando-se em consideração cinco dimensões do Pensamento Computacional, o qual é apresentado no Quadro 5.

Quadro 4: Variáveis de Evidência

INDICADOR	GRADAÇÃO		
	1 – BÁSICO	2 – EM DESENVOLVIMENTO	3 – PROFICIENTE
Aparência	Falar, pensar	Próxima fantasia. Mostrar, esconder.	Trocar fantasia, modificando cor, tamanho, etc.
Som	Tocar som, nota	Tocar som vs. Tocar som até término da execução	-
Movimento	Mover, ir para um <i>Sprite</i> , apontar, girar	Ir até X, Y. Deslizar até X, Y	Definir e mudar X, Y
Variáveis	Variáveis Scratch (<i>Sprite</i> , <i>mouse</i> , ponteiro, responder, etc.	Nova variável (definir, mudar)	Nova lista
Sequência e Iteração	Sequência	Repetir sempre	Repetir enquanto condição. Repetir até.
Expressões Booleanas	Compreender seu funcionamento	<, =, >	AND, OR, NOT
Operadores	Matemática	Cadeia de Caracteres (strings) e Aleatoriedade (random)	Lista

	GRADAÇÃO		
INDICADOR	1 – BÁSICO	2 – EM DESENVOLVIMENTO	3 – PROFICIENTE
Condicionais	IF	IF... ELSE	IF... ELSE aninhado
Coordenação	Esperar	Transmitir quando receber	Espere até
Eventos da Interface do Usuário	Bandeira verde clicada	Tecla pressionada, <i>Sprite</i> clicado	Pergunte e espere
Paralelização	2 <i>scripts</i> iniciam no mesmo evento		
Localização Inicial	Definir propriedades de localização (X,Y) na bandeira verde		
Aspecto inicial	Definir propriedade de aparência (fantasia, visibilidade, etc.) na bandeira verde		

Fonte: Adaptado de Seiter e Foreman (2013)

Quadro 5: Variáveis mapeadas com o PC de acordo com Seiter e Foreman

	Procedimentos e Algoritmos	Decomposição de Problemas	Paralelização e Sincronização	Abstração	Representação de Dados
<u>BÁSICO</u>					
Aparência de Animação	Sequência	-	-	-	Propriedades de <i>Sprites</i>
Movimento de Animação	Sequência	-	-	-	Propriedades de <i>Sprites</i>
Diálogo	Sequência	-	-	-	-
Colisão	Condicional	-	-	-	-
Manutenção da Pontuação	Inicializar, depois incrementar	-	-	-	Nova variável inteira
Interação do Usuário	Trigger sequencial pela interface do usuário	-	-	-	-
<u>EM DESENVOLVIMENTO</u>					
Aparência de Animação	Inicializar e depois mudar			Estado Inicial	Definição de propriedades
Movimento de Animação	Inicializar e depois mudar			Estado inicial, Canvas 2D	Definição de propriedades

	Procedimentos e Algoritmos	Decomposição de Problemas	Paralelização e Sincronização	Abstração	Representação de Dados
Diálogo	Alternar entre falar e esperar		Diversos segmentos, bloqueio, sincronia de tempo		
Colisão	Teste de colisão através de <i>loop</i> infinito	Detecção modularizada de colisão	Diversos segmentos assíncronos	Estado de execução, evento contínuo	
Manutenção da Pontuação	Incrementar e depois testar				Nova variável inteira
Interação do Usuário	Mover relativo a localização do ponteiro				Variável do ponteiro
<u>PROFICIENTE</u>					
Aparência de Animação	Disparar a animação conforme necessidade	Animação modular da <i>Sprite</i>	Multi-segmentada, assíncrona, sincronizar no estado e evento	Estado do programa e eventos	Atribuição de propriedade
Movimento de Animação	Mover relativo a localização atual				Atribuição relativa de propriedade
Diálogo	Disparar áudio conforme necessidade	Modularizar o funcionamento do áudio	Multi-segmentada, assíncrona, sincronizar no estado e evento	Estado de execução do programa e eventos	
Colisão	Aguardar por colisão		Bloqueio multi-segmentado	Estado de execução, evento discreto	
Manutenção da Pontuação	Reação global de acordo com a pontuação	Modularizar troca de cenário	Multi-segmentada, assíncrona, sincronizar no estado e evento	Estado de execução do programa e eventos	Nova variável inteira
Interação do Usuário	Perguntar e depois usar o dado		Bloquear execução ao pressionar tecla		Guardar informação de variável

Fonte: adaptado de Seiter e Foreman (2013, p. 63)

Já Wing (2011, p. 4) defende a avaliação utilizando conceitos mais amplos da Computação ao verificar o desempenho dos estudantes em atividades relativas ao Pensamento Computacional. Os indicadores são divididos em três grupos e podem ser conhecidos no Quadro 6.

Quadro 6: Medidas de “boa” abstração no PC segundo Wing

Indicador	Questão / Definição
Eficiência	<ul style="list-style-type: none"> • Rapidez • Espaço de armazenamento • Consumo de energia
Exatidão	<ul style="list-style-type: none"> • Resposta correta • Processo correto
“-ilidades”	<ul style="list-style-type: none"> • Simplicidade e Elegância • Usabilidade • Modificabilidade • Manutenibilidade • Custo • Etc.

Fonte: adaptado de Wing (2011)

Berry (2013), além de apresentar sugestões a professores de planejamento de aulas, recursos que podem ser utilizados, metodologias de ensino, indica sugestões de como os professores podem avaliar o progresso dos estudantes ao realizar atividades de PC. O relatório relata pontos interessantes, tais como:

- É difícil para os professores avaliar o conhecimento e a compreensão dos alunos com base apenas nos resultados das tarefas práticas.
- Se os alunos trabalham de forma colaborativa, pode ser difícil identificar a contribuição de cada estudante.
- Se o ensino da Computação estiver incorporado em outras disciplinas ou assuntos abordados em sala de aula, muitas vezes é difícil separar o aprendizado de conceitos computacionais de outros assuntos.

O documento também aponta seis técnicas de avaliação que devem ser usadas em conjunto com critérios previamente definidos pelos professores, a saber:

- a) Auto Avaliação: os alunos devem manter um *blog* ou registro de vídeo de seu trabalho, incorporando um comentário reflexivo ao lado de exemplos do que eles fizeram.
- b) Avaliação em pares: os estudantes, trabalhando com um parceiro, devem revisar e ajudar a corrigir, algoritmos e programas, ou fornecer *feedback* crítico e construtivo ao seu parceiro. Esse método é o mesmo utilizado pelos desenvolvedores de software profissional.
- c) Questionamentos abertos: uma forma de avaliar aqueles conhecimentos de estudantes sobre os conceitos abrangidos pelo programa de estudo e pode não ser de imediato

aparente no trabalho que eles desenvolvem. Alguns exemplos de perguntas são: por que o Google colocou esse resultado no topo? Como o seu programa funciona? Por que isso não pode ser um site seguro?, etc.

- d) Discussão/Argumentação com colegas: mostra-se muito efetivo o uso pelos estudantes, de perguntas abertas entre eles, mantendo o foco no que aprenderam e não apenas no que fizeram. Os debates podem ocorrer na sala de aula ou online ou ultrapassar as paredes da escola e envolver estudantes de outras instituições.
- e) Definição de metas: as habilidades de gerenciamento de projetos, como planejamento, organização, motivação de outros e alocação de recursos, são de grande importância em projetos do mundo real e podem ser amplamente aplicadas na educação. O pilar “decomposição” é usado neste quesito para dividir os grandes problemas em tarefas menores.
- f) KWL (*Know, Want and Learned* – Sabe, Quer e Aprendido): tentar identificar o que os estudantes já sabem, o que eles querem e o que eles efetivamente aprenderam é uma técnica efetiva de identificar sua evolução.
- g) *Feedback* automático: utilizar páginas na internet que oferecem tutoriais interativos para aprender linguagens de programação, provendo feedback imediato de sucesso ou falha na resolução de um desafio.

Conforme comentado anteriormente, Lambert e Guiffre (2009) também realizaram uma tentativa de medir se houve um incremento no interesse pela área da Computação ou Matemática através de três sessões de 30 minutos, porém não foi verificado se houve um aumento/decremento nas habilidades relacionadas ao Pensamento Computacional. Já Faber *et al.* (2017) desenvolveu seis aulas para serem aplicadas em 26 escolas da Educação Básica na Holanda. Mesmo recebendo um feedback positivo de professores e estudantes a respeito das aulas planejadas, não houve uma métrica do grau de satisfação dos mesmos ou dados estatísticos da melhoria ou não do desempenho do PC desses estudantes que participaram da pesquisa.

Como se pode ver nas propostas anteriores, as avaliações de alunos ou professores eram realizadas prioritariamente de maneira qualitativa ou com escalas de baixa precisão, omitindo uma avaliação quantitativa. Neste sentido, encontraram-se estudos que investigam a eficiência de linguagens de programação (visual e código) com crianças, tais como, Román-González *et al.* (2015), Román-González *et al.* (2016), Grover e Basu (2017) e Franklin *et al.* (2017), porém carecem de abordagens desplugadas. Outras pesquisas tentam padronizar a

avaliação e o ensino de atividade de PC Desplugado, como por exemplo Nishida *et al.* (2009) onde apresenta uma proposta de padrão de design e Román-González *et al.* (2015) que propõe um padrão de teste de Pensamento Computacional (que será melhor aprofundado posteriormente).

Campos *et al.* (2014), realizaram uma das primeiras tentativas de medir, através de pré e pós-testes, uma melhoria do Pensamento Computacional em crianças. Os testes foram aplicados antes e após três aulas com atividades desenvolvidas originalmente por Bell *et al.* (1997). O teste era composto por quatro perguntas, sendo que duas delas são apresentadas na Figura 38. Os resultados demonstram que não houve uma melhoria estatisticamente significativa do PC dos alunos, porém permitiu uma percepção da necessidade de melhorias no instrumento. De qualquer forma, as atividades trabalhadas com as crianças foram muito gratificantes e incentivaram o grupo de pesquisa a realizar novas edições do projeto.

Figura 38: Teste inicial e final de PC

<p>1. Preencher o pontilhado com o elemento que complete cada sequência: ○ ● ○ ● 1 2 4 8</p>
<p>2. Observar como podemos usar símbolos para escrever as palavras abaixo: arara = ♠ △ ♠ △ ♠ moeda = ◇ ♥ ♣ √ ♠ morar = ◇ ♥ △ ♠ △ Usando os mesmos códigos, escrever a palavra: amor =</p>

Fonte: Campos *et al.* (2014)

Outra pesquisa que trata da avaliação do PC no ensino fundamental é relatada por Scaico *et al.* (2012), o qual descreve a realização de três sessões de PC Desplugado para estudantes de quinto e nono anos e logo após um teste relativo ao tema trabalhado, porém sem o uso de pré e pós-testes para verificar alterações no desempenho dos mesmos, ou seja, realizou apenas um teste no final da pesquisa. O trabalho, contudo, faz uma análise da compatibilidade das atividades com a idade das crianças.

A solução proposta por Rodriguez (2015), Rodriguez *et al.* (2016) e Rodriguez *et al.* (2017) procura avaliar estudantes com cinco atividades desplugadas utilizando uma adaptação dos níveis de pensamento de Bloom (Taxonomia de Bloom), ou seja, em três níveis (proficiente, parcialmente proficiente e insatisfatório). Ainda, Rodriguez utilizou um teste composto por perguntas abertas que não passou por um processo de validação e o aplicou em turmas de sétimo

ano do EF, onde foi possível verificar que alunos aprenderam habilidades do PC com atividades desplugadas.

Os trabalhos aqui citados são parte de um grande conjunto de pesquisas que tentam medir as habilidades do PC, porém não utilizam uma solução direta, de fácil aplicação e com um processo de validação formal para que se possa atingir um resultado mais preciso e confiável. Diferentemente da proposta de Román-González *et al.* (2015), seu teste de PC passou por um rigoroso processo de validação e que culminou em uma tese de doutorado (Román-González, 2016). Esse teste será utilizado por esta pesquisa, bem como aprofundado no Capítulo 4 (p. 112).

A revisão de estudos que fornecem evidências sobre a utilidade da Computação desplugada para desenvolver habilidades do PC mostra que ainda existe espaço para pesquisar essa área e existe também uma necessidade de mais pesquisas empíricas sobre esse assunto, especialmente quando se trata de seu uso nos primeiros anos do Ensino Fundamental. Conseqüentemente, com a investigação relatada neste artigo, tenta-se esclarecer um pouco mais a respeito do assunto.

No capítulo seguinte, far-se-á um panorama da integração do PC na Educação Básica, uma tendência mundial, a maneira de conscientizar e sensibilizar o leitor da importância do ensino dos fundamentos da Computação nas escolas brasileiras e que também serviu de inspiração e motivação para esta pesquisa.

2 PANORAMA GLOBAL DA ADOÇÃO DO PC

A adoção de noções de Computação em escolas na educação básica é, atualmente, uma preocupação em diversos países, onde a implantação ocorre de forma rigorosa. Reconhecidamente, cresce a ideia de que a disciplina de Computação é muito distinta das aulas de Informática e que o uso de habilidades da área da Computação possui benefícios educacionais (habilidades de reflexão e solução de problemas, compreensão de que o mundo está impregnado com a tecnologia digital) e econômicos (alta demanda de profissionais com boa formação).

O presente capítulo apresenta, de forma resumida, resultados de uma revisão sistemática da literatura a respeito do panorama de diversos países que já adotaram ou estão em processo de adoção do Pensamento Computacional na educação formal ou informal. Infelizmente, não é possível fazer uma comparação homogênea entre todos os países, pois os sistemas educacionais são distintos e em algumas situações há pouca documentação disponível, chegando, às vezes, a ser contraditórias.

Esse levantamento foi motivado pelo trabalho liderado pela Microsoft (JONES *et al.*, 2011), realizado em âmbito internacional, para averiguar a situação da introdução de disciplinas voltadas ao ensino e funcionamento de sistemas computacionais. Como os dados de Jones *et al.* (2011) se encontram defasados, realizou-se uma nova investigação para atualizar o estudo inicial.

Este capítulo está dividido em duas partes, sendo a primeira voltada para iniciativas governamentais, e a segunda para movimentos de instituições não-governamentais e comerciais.

2.1.1 Iniciativas Governamentais Globais

2.1.1.1 Alemanha

Cada um dos 16 estados federativos da Alemanha possui um departamento de educação, porém, até o ano de 2004 não havia uma padronização dos conteúdos (ementas/tópicos) específica para alunos entre o quinto e o décimo anos (*Realschule* ou *Sekundarstufe I*). A partir de então, as disciplinas de Computação não são mais obrigatórias, no entanto, podem ser utilizadas como “Crédito Extra”. A motivação do país para adoção da Computação nas escolas foi baseada em dois objetivos, sendo eles:

1) Permitir que os estudantes saibam lidar com a grande quantidade de artefatos e tecnologias da informação na vida cotidiana;

2) Promover o ensino mais avançado da Computação após o ensino básico (técnico e superior).

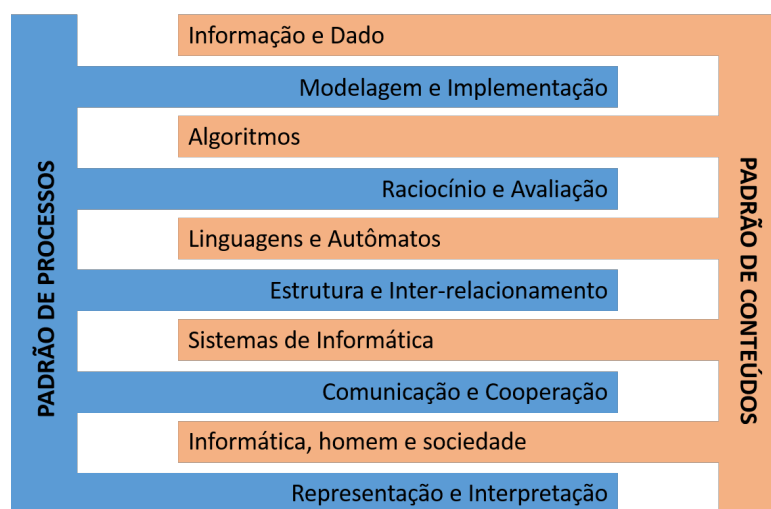
Todos os tópicos que fazem parte da integração do PC no currículo são integralmente especificados no documento chamado *Einheitliche Prüfungsanforderungen – Informatik* (em tradução livre, Normatização dos Requisitos para Ciência da Computação), norma publicada como referência para o Exame Unificado “*Abitur*” (*Einheitlichen Prüfungsanforderungen in der Abiturprüfung*) (EPA, 2004). Alguns dos tópicos que o estudante pode escolher são (BRINDA *et al.*, 2009):

- Orientação a Objetos (codificação);
- Modelagem entidade-relacionamento;
- Autômatos;
- Modelagem algorítmica;
- Interação homem-máquina;
- Privacidade;
- Segurança;
- Arquitetura de Computadores;
- Computabilidade;
- Eficiência;
- Questões sociais.

No ano de 2008, a Associação da Informática (*Gesellschaft für Informatik*) publicou um padrão para a segunda metade do ensino fundamental. O documento foi modelado dentro do padrão definido pelo NCTM¹ (*National Council of Teachers of Mathematics*) e organizado em Conteúdos e Processos. Os princípios encontram-se listados na Figura 39.

¹ <http://www.nctm.org/>

Figura 39: Princípios da Computação (Ensino Fundamental)



Fonte: Tradução de Brinda *et al.*, 2009.

É importante ressaltar que o padrão separa claramente o ensino de mídias digitais e TICs das aulas de Computação. Como o padrão foi criado de forma ascendente, ou seja, de forma colaborativa, ele vem sendo adotado em diversas cidades sem obstáculos.

O padrão alemão descreve as competências para os anos 8, 9 e 10, porém o foco maior é na modelagem e não no ensino de hardware e software. Um ponto forte do sistema de ensino alemão no processo de adoção da Computação nas escolas é a alta disponibilidade de professores com formação adequada para ministrar as aulas. O treinamento necessário para a regência de aulas no país exige dois cursos de nível superior, além de uma preparação pedagógica, teoria educacional focada na área em que atua e estágio de dois anos.

No ano de 2013, a obrigatoriedade da Computação voltou a ser pauta do governo (SCHMUNDT, 2013) e foi apresentado ao parlamento em 2015, porém ainda não houve uma definição oficial.

2.1.1.2 Argentina

Lançado em janeiro de 2013, a Fundação Sadosky publicou um manifesto denominado “CC-2016: Una propuesta para refundar la enseñanza de la computación en las escuelas Argentinas”, que tem como principal objetivo conscientizar a comunidade científica da importância de uma profunda mudança do ensino em nível fundamental e médio com a introdução dos princípios de Computação em sua estrutura (SADOSKY, 2013). O documento argumenta que o ensino da Computação é essencial para grandes oportunidades proporcionadas por essas tecnologias. Além disso, a entidade adverte a necessidade de os estudantes

desenvolverem habilidades e competências essenciais para a vida moderna (BRACKMANN *et al.*, 2016).

Em 12 de agosto de 2015 foi emitida a resolução de número 263/15 pelo Conselho Federal de Educação Argentino (CFE Argentina, 2015) movida pela necessidade estratégica de desenvolvimento econômico-social da nação, estabelecendo diversos elementos, dentre eles:

- Ensino da programação como parte do currículo ou como atividade extraclasse nas escolas durante os anos compulsórios;
- Criação da “Rede de Escolas que Programam” (REP), categorizadas conforme disponibilidade de professores com formação e oferta de aulas;
- Intensificar a propagação da REP em todas as jurisdições atendidas pelo Ministério da Educação até alcançar todas as instituições de ensino estatais, através de iniciativas de formação docente específica e durante o serviço;
- Criação do prêmio anual “Clementina”, para as produções de destaque na área de Informática e desenvolvido pelos alunos em escolas que fazem parte da REP;
- Criação do Repositório Nacional de Produções em Informática, tornando o acesso aberto a todos os demais membros da REP (escolas, classes docente e discente).

O documento foi aprovado, levando como base as resoluções 123/10, que tratam das políticas de inclusão digital educativa, 244/15 onde são também aprovadas intensificações no uso de TICs nas escolas para melhoria nos processos de ensino e aprendizagem e o artigo terceiro da Lei de Educação Nacional.

Na tentativa de formar os professores em exercício, a Universidade Nacional de Rosário (UNR), em conjunto com a Fundação Sadosky, desenvolveu um curso Lato-Sensu pioneiro denominado “Especialización Docente de Nivel Superior en Didáctica de las Ciencias de la Computación: Aprendizaje y Enseñanza del Pensamiento Computacional y la Programación en el Nivel Primario” (não publicado), composto por 400 horas aula, sendo 80 a distância e cobrindo temáticas como: introdução do PC nas escolas primárias, atividades de PC, programação, projetos integradores, organização de computadores, robótica, segurança e privacidade dos dados e avaliação (ISAÍAS, 2017).

2.1.1.3 Austrália

O Ministro da Educação, Christopher Pyne, publicou no dia 18 de setembro de 2015 uma grande reestruturação no currículo das escolas públicas (ACARA, 2015). O novo modelo coloca a programação como uma das principais competências e recebe 12 milhões de dólares australianos (aproximadamente 9.588.000 dólares estadunidenses) com o intuito de:

- Desenvolvimento de recursos curriculares para o ensino da matemática;
- Dar suporte para introdução da programação em todos os níveis de escolaridade;
- Criar uma escola piloto baseado na estratégia *P-TECH-like Schools*¹;
- Financiar Escolas de Verão para estudantes nas áreas de Ciência, Tecnologia, Engenharia e Matemática (*STEM – Science, Technology, Engineering and Mathematics*).

Nos quintos e sextos anos, os estudantes aprenderão a codificar (pensar computacionalmente) e, a partir do sétimo, os mesmos irão aprender a programar (escrever programas). Com a introdução dessas disciplinas no currículo, os alunos não serão mais obrigados a ter aulas de história e geografia, ou seja, tais disciplinas, até então essenciais e obrigatórias, passam a ser opcionais (gerando revolta por parte de alguns professores e pais de alunos) (DAVIS, 2015).

2.1.1.4 Coreia do Sul

Possui longo histórico de uso considerável de computadores em sala de aula desde a década de 1970 com um objetivo vocacional; em 1987 foram introduzidos na alfabetização digital e a partir de 1992 seu uso foi reforçado no ensino das TICs. No ano de 2004, 80% das escolas já possuíam aulas de Informática e 40% de codificação. Em 2007, ocorreu um consenso entre os educadores, de que os computadores não deveriam ser utilizados apenas para aulas de alfabetização digital, mas também deveriam ter um papel essencial na solução de diversos tipos de problemas do mundo real, aumentando a habilidade de raciocínio. Como resultado, as escolas mudaram o nome da disciplina “Computadores” para “Informática”, onde o foco principal são os princípios e conceitos da Computação (métodos e procedimentos para solução

¹ Escolas P-TECH são organizações sem fins lucrativos e possuem um currículo e metodologia de ensino inovadores, caminhando juntos com a indústria e demais instituições de ensino locais, focando na formação do aluno para o mercado de trabalho (<http://saf.org.au/p-tech-australia/>).

de problemas, programação com orientação a objetos, algoritmos simples e circuitos lógicos) desde o ensino fundamental como disciplina eletiva. Destacam-se, ainda, os conteúdos de ética na Informática e cibercrime (CHOI, 2015).

Em 2015 as escolas tiveram uma diminuição muito grande, tanto na oferta como na matrícula nas aulas de Informática, devido basicamente aos seguintes motivos:

- Diminuição da carga horária exigida pelas disciplinas eletivas;
- Ausência de uma regulamentação específica das aulas de Informática;
- Ausência de uma avaliação institucional relativa às aulas de Informática.

Atualmente, apenas 10% das escolas do ensino fundamental ofertam aulas de Informática e menos de 5% dos alunos possuem a oportunidade de participar delas. Para que essa realidade mude, o governo está redigindo um novo currículo que deve entrar em vigor em 2018.

2.1.1.5 Escócia

O ensino formal da Computação ocorre desde 1980, a partir do ensino fundamental, aproximadamente aos 14 anos. No princípio, as escolas trabalhavam com o desenvolvimento de software, porém, logo o foco mudou para um formato mais avançado de Computação e Sistemas de Informação. No último ano, o estudante tem um importante componente que trabalha basicamente dois elementos: o desenvolvimento de software e bancos de dados relacionais voltado para atividades práticas. O ensino desses dois elementos tem sido bastante debatido com universidades, pois não foi comprovada ainda sua real necessidade de ser ministrada antes do ingresso a nível de educação superior.

Após a troca de currículo, ocorreu no ano de 2004 uma pesquisa denominada “*Computer Science: What do pupils think?*”, onde chegou-se à conclusão de que os estudantes achavam as aulas de Computação muito tediosas, pois o novo currículo incentivava o ensino de aplicativos de escritório, o que gerou uma diminuição de interesse dos alunos pela disciplina.

Em 2011, houve o lançamento do “*Curriculum for Excellence*” (ALBA, 2009), um documento que rege o sistema de ensino do país dos 3 aos 18 anos e apresenta os efeitos esperados (*outcomes*) pelo novo modelo dividido em duas categorias: 1) TICs para melhoria da aprendizagem e 2) Ciências de Computação contextualizada no desenvolvimento de competências tecnológicas e conhecimentos.

2.1.1.6 Estados Unidos da América

Em 2014, Barack Obama, tornou-se o primeiro presidente a escrever uma linha de código, e emitiu um amplo plano de ação para expandir a Ciência da Computação nas salas de aula do país (The White House, 2016). Assinado no dia 10 de dezembro de 2015, a Lei Federal “*Every Student Succeeds Act*” (ESSA), é responsável pelas políticas públicas na área da educação do país. Nesse documento são detalhadas, desde a forma como ocorrem os financiamentos, até a maneira que as escolas são avaliadas. A lei também coloca a Ciência de Computação em condições de igualdade com outras disciplinas acadêmicas, tais como Matemática, Geografia, História, Inglês e Ciências. O documento não define como a implantação deve ocorrer, porém incentiva sua adoção e permite a obtenção de recursos para tal (âmbito federal e estadual). Mesmo após a assinatura da lei, de acordo com Code.Org (2016) e Guzdial (2014), o país não possuía ainda uma legislação específica para o ensino da Ciência da Computação como obrigatória, porém existem iniciativas estaduais que permitem a concretização do ensino formal, substituindo disciplinas de Matemática, Ciências, Língua Estrangeira e outras, por Computação. Uma visualização gráfica dos estados participantes deste movimento, encontra-se na Figura 40 destacado em azul. Essas substituições variam de estado para estado e podem ser conhecidos no Quadro 7.

Os estados participantes seguem o currículo proposto pela *Computer Science Teacher Association* (CSTA) denominado “*A Model Curriculum for K–12 Computer Science*” (CSTA/ISTE, 2011), onde realiza aconselhamentos, devidamente estruturados, no ensino da Computação nas escolas, desde o jardim de infância até o último ano do ensino médio, além de exemplificar exercícios que podem ser realizados em sala de aula.

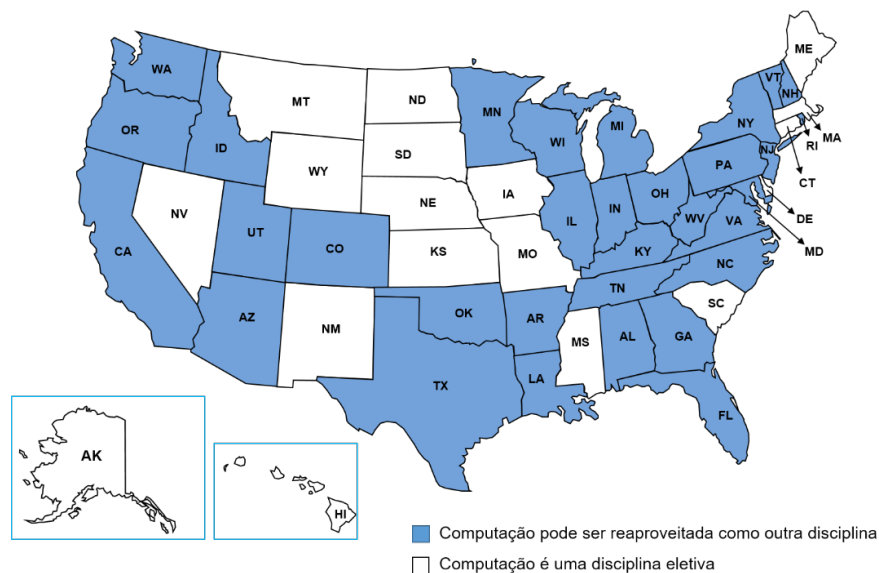
O país teve um grande impacto a respeito da compreensão sobre Ciência da Computação a partir do lançamento do projeto Code.Org (2015), atraindo a atenção dos estudantes, pais e escolas (vide página 95). Em recente pesquisa encomendada pelo Google e executado pela empresa Gallup-Google (2016), entre diversos dados coletados, destacam-se neste trabalho: 1) 90% dos pais querem que seus filhos tenham aulas de programação nas escolas; 2) 50% dos pais acreditam que a Ciência da Computação é uma competência tão importante quanto ler, escrever e fazer cálculos.

Através de propostas de leis lideradas pelo Senador Casey (2013) e Deputado Brooks (2013), com novo pedido de apreciação (Casey, 2015), o governo estadunidense liberou em 2016 o montante de 4 bilhões de dólares em recursos para os estados e 100 milhões de dólares diretamente para os distritos (The White House, 2016):

- 1) Aumentar o acesso a disciplinas de Ciência da Computação
- 2) Treinar professores
- 3) Expandir o acesso a materiais de instrução de alta qualidade
- 4) Construir parcerias regionais efetivas.

Em resumo, o recurso tem objetivo específico de incentivar a oferta de disciplinas de Computação, além de formar professores e equipar as escolas para atender a demanda da área, tendo em vista que disciplinas ainda não são obrigatórias. Segundo Cuny (2017), algumas cidades ou distritos (*e.g.* New York City, San Francisco, Broward County (FL), Rhode Island, Virginia, Chicago) tomaram medidas mais drásticas e tornaram a CC obrigatória em suas escolas.

Figura 40: Estados que adotaram a Computação em seu currículo



Fonte: produção do autor, adaptado de CODE.ORG, 2016

Quadro 7: Equivalência de disciplinas por estado

Estado	Atendem os requisitos de Ciência da Computação
Alabama	Matemática
Arizona	Matemática
Arkansas	Matemática
Califórnia	Matemática, Ciências
Distrito de Colúmbia	Matemática
Flórida	Matemática, Ciências
Geórgia (Pioneiro)	Ciências
Idaho	Matemática, Ciências
Illinois	Matemática

Estado	Atendem os requisitos de Ciência da Computação
Indiana	Matemática
Kentucky	Matemática
Maryland	Matemática
Michigan	Matemática
Minnesota	Matemática
Nova Iorque	Matemática
Carolina do Norte	Matemática
Ohio	Matemática
Oklahoma	Matemática
Oregon	Matemática
Rhode Island	Matemática, Ciências
Tennessee	Matemática
Texas	Matemática, Outro
Utah	Matemática, Ciências
Vermont	Matemática
Virgínia	Matemática, Ciências
Washington	Matemática, Ciências
Wisconsin	Matemática

Fonte: CODE.ORG, 2016

Atualmente, os EUA estão no processo de criação da próxima geração do currículo-referência para as disciplinas da área das Ciências, denominado *Next Generation Science Standards*¹ (NGSS). A nova proposta possui como inspiração o sistema de ensino de países como Finlândia, Canadá, Hungria, Singapura, Coreia do Sul, China e Japão, tendo em vista sua alta pontuação nas áreas de Ciência e Matemática.

2.1.1.7 Estônia

O país que originou o software Skype é considerado um dos países mais dependentes de internet no mundo e sua ascensão digital ocorre alguns anos depois de sua independência, ou seja, no início da década de 90 (MANSEL, 2013).

O projeto de introdução à programação no currículo foi proposto pelo então presidente da nação, Toomas Hendrik Ilves, que também já foi embaixador do país nos Estados Unidos da América. Durante seu período no exterior, teve a oportunidade de conhecer diversos casos nas indústrias onde as máquinas substituíam a mão-de-obra de milhares de trabalhadores sem perder sua eficácia. Essa nova realidade fez com que ele repensasse a força de trabalho do país, informatizando tudo que possível e de todos os modos para melhorar o tamanho funcional da nação que beirava 1,4 milhão de cidadãos (MUFFET, 2014).

¹ <https://www.nextgenscience.org/>

No final da década de 90, com a ajuda do projeto Tiger's Heap, todas as escolas do país já possuíam internet. O projeto também é coadjuvante no ensino de programação para crianças a partir dos 7 anos, levando-se em consideração a seguinte lógica: “se você aprende regras gramaticais aos 7 ou 8 anos, o que difere em relação às regras de programação?”.

Hoje, o ensino da programação foi deliberado a nível nacional e abrange desde a educação fundamental, ensino médio e formação técnica como uma disciplina facultativa. As escolas utilizam uma plataforma chamada ProgeTiiger, onde disponibiliza diversas ferramentas e suporte a professores para incentivar o ensino da programação nas escolas. Um exemplo é a ferramenta chamada Oppematerjalid¹, onde dispõe de uma série de sugestões de uso de atividades e tecnologias de acordo com o nível escolar, área, linguagem de programação e sistema operacional.

2.1.1.8 França

Em fevereiro de 2015, o presidente Francois Hollande, juntamente com o Ministro da Educação, Benoît Hamon, publicaram um anúncio de que escolas iriam gradualmente receber aulas de programação a partir de setembro do mesmo ano como atividades extracurriculares. O ministro argumenta que é de extrema importância os alunos saberem falar francês, contar, calcular, compor e decompor números na matemática, porém a escola não pode ignorar a importância da intervenção da digitalização em todas as disciplinas, pois o aprendizado da lógica facilita o manuseio de conceitos. Benoît ainda complementa: “A dúvida não é mais se devemos ou não ensinar Computação e programação aos alunos, mas como, qual o propósito e em que nível ele deve ser introduzido no currículo” (JOHNSON, 2015).

De acordo com o Conselho Superior Curricular, tem-se como objetivo repensar os métodos de ensino e avaliação, além de modernizar o currículo. Para isso, foram definidos os primeiros requisitos para o ensino no país. São eles (FLEURY e NEVEUX, 2014):

- Fundamentos das Linguagens de Programação;
- Desenvolvimento de aplicativos com a utilização de algoritmos simplificados.

Estudos demonstram que os franceses possuem grande medo do erro, incitando culpa e penalização. A Computação também é utilizada neste contexto na tentativa de mudar esta realidade, pois estudantes podem simplesmente apagar todo o código e reiniciar o algoritmo na tentativa de encontrar outra estratégia. Essa abordagem também tem como objetivo promover a interatividade na sala de aula, estimular a participação de todos os estudantes, além daqueles

¹ <http://www.progetiiger.ee/oppematerjalid>

que não têm acesso à tecnologia, permitindo que os estudantes superem dificuldades e inibições, inclusive sociais.

A intenção de encorajar os estudantes para a carreira na Computação é uma tentativa de torná-los cidadãos do mundo o qual, a cada dia, está mais conectado. Essas mudanças no currículo não têm a intenção de torná-los todos programadores, porém facilitar a detecção de talentos e vocações técnicas em um setor de competição global.

2.1.1.9 Finlândia

De autoria de Mykkänen e Liukas (2015), o guia “Koodiaapinen 2016” (em tradução livre, Abecedário do Código) tinha como objetivo convencer o governo da Finlândia a implantar o ensino da Computação nas escolas, desde o ensino fundamental. Seu conteúdo é composto por um passo-a-passo, leve e direto, de como ensinar nas escolas o Pensamento Computacional. Desde sua publicação, esse guia é a principal referência para o legislativo na criação de leis para sua adoção, no currículo que entrou em vigor em 2016 como disciplina obrigatória desde a educação primária.

Mykkänen (WEINBERG, 2015) relata ainda que “as pessoas precisam dominar minimamente a linguagem do computador. “Quem não investir nisso vai ficar para trás”. Ele ainda complementa com dados: “se seu país não acelerar, acumulará um déficit de 17.000 programadores em quatro anos”.

O guia apelidado de “Koodi 2016” ainda propõe as atividades trabalhadas de acordo com o ano escolar:

- Anos 1 a 2: Ensino dos fundamentos da Computação através de atividades lúdicas;
- Anos 3 a 6: Desenvolver atividades no computador utilizando programação visual e aprender a controlá-lo sem medo de errar;
- Anos 7 a 9: Familiarizar os alunos com uma autêntica linguagem de programação.

Alguns fatos relevantes a respeito da implantação do novo currículo incluem (KOODIAAPINEN, 2016):

- Para que um aluno de sexta série tenha um bom rendimento na disciplina de Matemática, o educando deve saber criar um programa simples utilizando uma linguagem de desenvolvimento visual (*i.e.* Scratch);

- Uma formação gratuita para professores é realizada por um MOOC¹ (*Massive Open Online Course*) e que também dispõe de uma biblioteca aberta com sugestão de conteúdos e atividades. A primeira edição do MOOC em 2015 formou 511 professores e no ano de 2016 houve 2000 inscrições;
- A introdução do Pensamento Computacional nas escolas é uma reforma proposta pelo psicólogo educacional e pesquisador de design Tarmo Toikkanen da Aalto University, porém construído basicamente com os esforços de voluntários e poucos recursos.

2.1.1.10 Grécia

No ensino fundamental, crianças têm aulas de Computação desde a terceira série (oito anos de idade) em disciplinas denominadas “Desenvolvimento de Aplicativos em Ambientes de Programação”. Além de conceitos básicos de Informática, tais como: o que é um computador, partes que o compõem, seu uso na rotina diária e como usá-lo. Além de atividades *off-line*, crianças também usam computadores para aprender através de jogos (*e.g.* Logo) e aplicativos simples de comunicação pela internet (BALANSKAT e ENGELHARDT, 2014).

A partir dos 10 anos de idade os estudantes já trabalham com o desenvolvimento de aplicativos simples para a resolução de problemas e o aluno pode decidir entre sete áreas de interesse:

- Aprofundamento da alfabetização digital;
- Processamento de texto, imagens e arquivos;
- Uso e criação de gráficos;
- Programação de computadores;
- Busca de informações utilizando a internet e apresentação de dados;
- Comunicação pela internet;
- Computadores na vida diária.

É importante salientar que, quando essa adoção foi implantada, não houve preparo de professores de Informática ou criação de material didático; porém, de um modo geral, os professores já possuíam um preparo avançado em Computação (*e.g.* Engenheiro de Software e Cientista da Computação), podendo tornar as aulas mais desafiadoras, ou ainda, muitas vezes,

¹ É um tipo de curso aberto ofertado por meio de ambientes virtuais de aprendizagem, ferramentas da Web 2.0 ou redes sociais que visam oferecer para um grande número de alunos.

deixando a desejar na metodologia de ensino devido a exíguos conhecimentos pedagógicos em sua formação. Sendo assim, os estudantes dependiam da intimidade e destreza do professor com os computadores.

Na segunda metade do ensino fundamental, as aulas de Informática são obrigatórias desde 1993, compreendendo dois níveis de idade:

- Entre 12 e 14 anos: baseadas em quatro áreas de ensino:
 - Aprofundamento em hardware de computadores;
 - Uso avançado de interface de usuário (GUI) em sistemas operacionais;
 - Busca, processamento e apresentação de informações, utilizando a internet e aplicativos de escritório;
 - Uso de computadores para a vida profissional ou de uso rotineiro.
- Na faixa dos 15 anos: focadas em pensamento algorítmico, programação e amplo uso da internet para apresentações e projetos.

Durante essa etapa, os estudantes têm material didático, mas os professores podem criar seu próprio material, desde que atenda aos requisitos do currículo.

No ensino médio (entre 15 e 18 anos de idade), os estudantes trabalham as mesmas áreas do ensino fundamental na disciplina denominada “Tecnologia”, porém de forma mais aprofundada. E, no último ano do ensino médio, aqueles que optarem por uma linha tecnológica, os conceitos de Ciência da Computação tornam-se obrigatórios, com ênfase em Pensamento Algorítmico, Princípios da Ciência da Computação e Conceitos de Programação. Todos os temas possuem material didático próprio, porém o professor pode criar seus próprios conteúdos.

2.1.1.11 Reino Unido

O Currículo Nacional do Reino Unido requer que cada estudante estude Tecnologias da Informação e Comunicação (TICs) e é regido pelo Currículo Nacional para TICs. Esse documento possui uma ampla especificação de alto nível, porém, quando colocado em prática, pouco é ensinado além dos pacotes de escritório (processadores de texto, planilhas eletrônicas, editores de apresentação, etc.).

Em 2011 houve uma grande revisão do Currículo Nacional, onde a disciplina de Computação foi considerada obrigatória em todos os quatro níveis (*Key Stages – KS*), independentemente da idade dos alunos. Após concluído os KS, os alunos são preparados para as provas GCSE (*General Certificate of Secondary Education*), onde podem comprovar suas competências ocupacionais em diversas áreas, ou seja, certificando apenas o conhecimento e

compreensão. Em uma etapa posterior, caso o estudante opte pela continuidade dos estudos no ensino superior, é necessário se preparar por mais dois anos e realizar as provas GCE (*General Certificate of Education*), também conhecidas como *AS-A Levels*. Após concluídas essas provas, o aluno terá uma certificação necessária para entrar em uma universidade. Tanto no GCSE e GCE, ambos possuem Computação como opção de certificação nas cinco instituições homologadas para aplicação das provas (SCC, 2015) (TEACHING JOBS LONDON, 2013).

Tabela 1: Instituições homologadas para aplicação das provas GCSE e GCE

	CCSE	GCE
Assessment and Qualifications Alliance (AQA) ¹	✓	✓
Council for the Curriculum, Examinations & Assessment (CCEA) ²	✓	✓
Edexcel (Edexcel - London Examinations) ³	✓	✓
Oxford, Cambridge and RSA Examinations (OCR) ⁴	✓	✓
Welsh Joint Education Committee (WJEC) ⁵	✓	✓

Fonte: Produção do autor.

Rossi (2015) relata em seu trabalho que os pais são favoráveis à introdução da programação nas escolas. Uma pesquisa recente aponta que 60% dos pais e 75% dos alunos preferem aulas de Python ao invés do idioma Francês no ensino Fundamental em um universo de 1000 crianças (5-11 anos de idade), 1000 crianças (11-16 anos de idade) e 1000 pais. Um dado curioso dessa pesquisa é o fato de as crianças preferirem as aulas de programação, por estas serem “mais fáceis” em relação ao Francês.

¹ <http://www.aqa.org.uk/>

² <http://www.ccea.org.uk/>

³ <http://www.edexcel.com/>

⁴ <http://www.ocr.org.uk/>

⁵ <http://www.wjec.co.uk/>

2.1.2 Iniciativa Privada e de Organizações Não-Governamentais

2.1.2.1 European Schoolnet

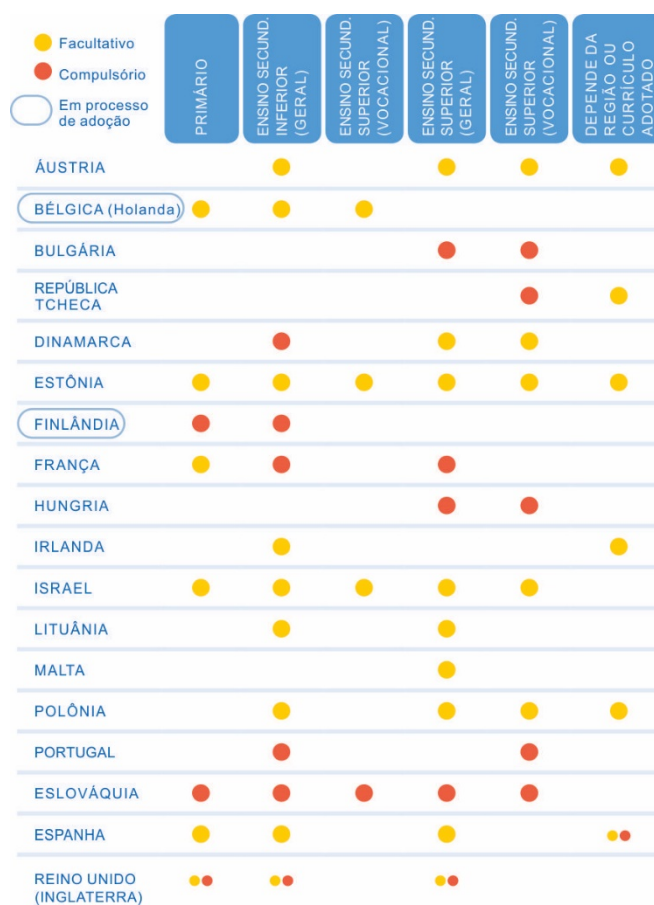
Rede de troca de experiências, sem fins lucrativos, entre 31 Ministérios da Educação da Europa, com sede em Bruxelas, Bélgica, e tem como foco principal a inovação do ensino e aprendizagem na área de TICs. Entre seus objetivos estão: criar políticas para regulamentar o ensino de Computação na Europa e identificar evidências escaláveis e com possibilidade de transferência de práticas e prioridades emergentes relacionadas à educação no continente.

A Figura 41 demonstra um levantamento realizado pela Organização nos países-membros. A Figura 42 apresenta a situação da integração do Pensamento Computacional no currículo desses países. Os que estão em processo de adoção, encontram-se circulados. A cor vermelha representa a obrigatoriedade, e a amarela como opcional.



Fonte: Adaptado de Balanskat e Engelhardt (2015)

Figura 42: Integração do Pensamento Computacional no currículo



Fonte: adaptado de Balanskat e Engelhardt, 2015

2.1.2.2 European Comission (Europa)

Na publicação do departamento de ciência e conhecimento da Comissão Europeia, Joint Research Centre (JRC), é relatado um levantamento em 13 países (Áustria, Rep. Tcheca, Dinamarca, Finlândia, França, Grécia, Hungria, Itália, Lituânia, Polónia, Portugal, Suíça e Turquia) com o intuito de fornecer apoio científico baseado em evidências para o processo de formulação de políticas na Europa (Bocconi *et al.*, 2016). O documento inclui duas apurações bastante pertinentes, sendo elas: a motivação dos países em introduzir o PC em seu currículo e as implicações políticas e práticas na introdução do PC na Educação Básica.

Em relação ao primeiro item, os países possuem diferentes intenções ao incluir o PC no seu currículo. Como se pode ver na Tabela 2, todos os países entendem que o PC promove habilidades de pensamento lógico e resolução de problemas. Uma quantidade menor tenta atrair mais estudantes para a área da CC ou simplesmente tem a intenção que seus estudantes saibam

programar e codificar¹. Poucos países têm uma visão mais abrangente ao acompanhar a empregabilidade dos estudantes.

Tabela 2: Objetivos da introdução do PC no currículo da Educação Básica

	Áustria	Rep. Tcheca	Dinamarca	Finlândia	França	Grécia	Hungria	Itália	Lituânia	Polónia	Portugal	Suíça	Turquia
Promover habilidades de pensamento lógico	●	●	●	●	●	●	●	●	●	●	●	●	●
Promover habilidades de resolução de problemas	●	●	●	●	●	●	●	●	●	●	●	●	●
Promover outras habilidades chaves		●		●		●		●	●	●	●	●	●
Atrair mais estudantes para a Ciência da Computação				●	●				●	●			●
Promover habilidades de programação e codificação				●	●				●	●	●	●	●
Promover empregabilidade no setor de TICs				●	●								●

● = já adotado ● = em processo de adoção Fonte: adaptado de (Bocconi *et al.*, 2016)

Em relação ao segundo item, o relatório conclui através de um gráfico, sugestões que podem servir de inspiração a outros países no processo de inserção do PC no currículo. A proposta é dividida em quatro dimensões, sendo elas: a abrangência da integração, a sistemática de sua implantação, a consolidação do entendimento e as políticas de apoio. A seguir, cada uma das dimensões será melhor detalhada:

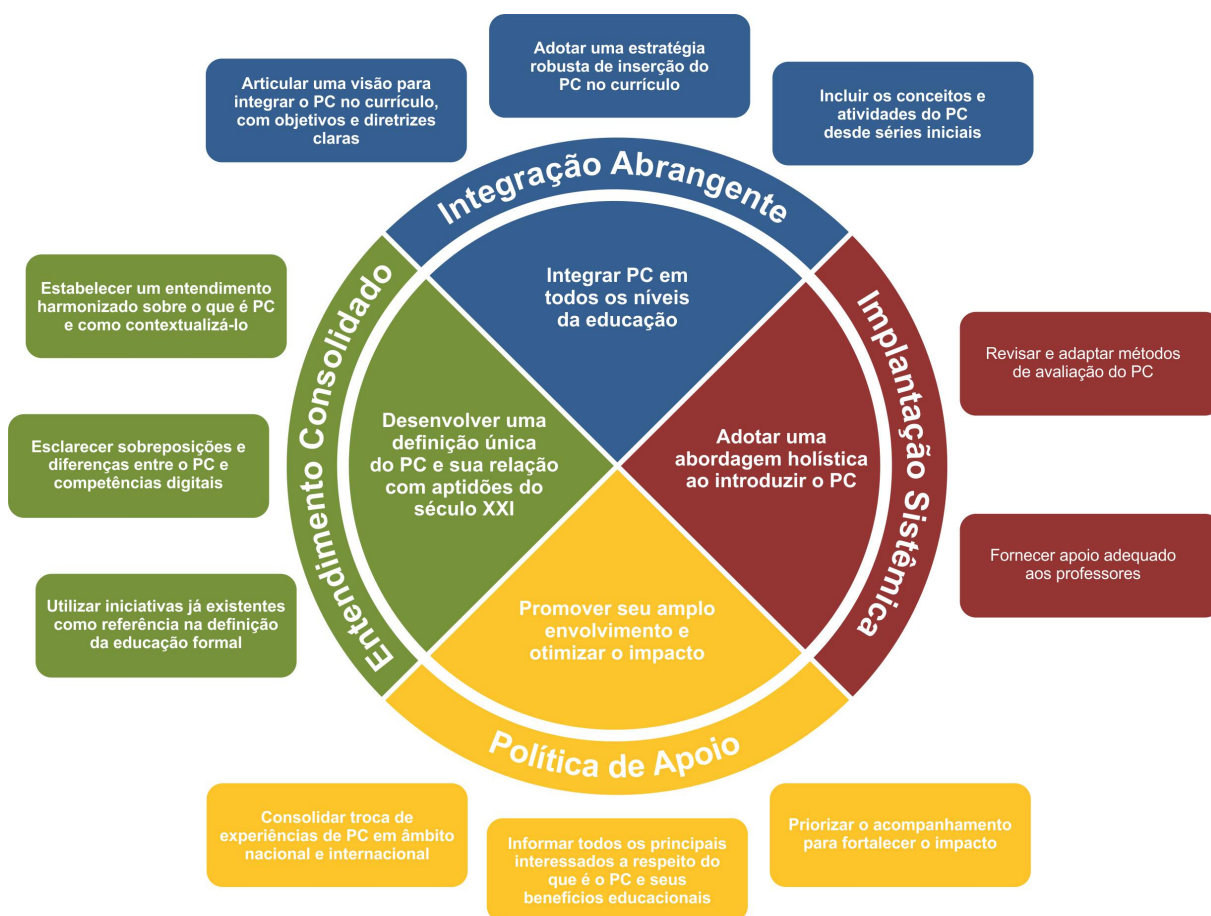
- **Integração Abrangente:** definir de maneira clara a integração do PC no currículo é crucial. Uma estratégia recomendada é iniciar a implantação através de objetivos específicos e concretos, incluir as partes interessadas no processo e tornar o PC obrigatório nas escolas gradativamente. Para que a adoção seja robusta, é necessário um planejamento cuidadoso e selecionar práticas pedagógicas compatíveis, incluindo

¹ “Programação é um conjunto de técnicas e práticas voltadas para solucionar problema computável (solúvel por meio computacional), incluindo técnicas para modelagem formal de um problema e técnicas de projeto de algoritmos”. “Codificação é sinônimo de "geração do código fonte". A codificação é escrita, utilizando uma linguagem de programação, das instruções que o computador deve realizar para alcançar um resultado“ (CONSTANTINO, 2009)

ferramentas de ensino e estratégias de avaliação em todos os níveis. Crianças devem ser incentivadas o quanto antes para aprender conceitos do PC.

- **Implantação Sistêmica:** a literatura é clara no tocante à importância dos métodos de avaliação do PC através da produção de artefatos (*e.g.* jogos ou modelos) que os estudantes desenvolvem, testes com múltiplas alternativas, rubricas (critérios avaliativos) ou alteração de códigos para atingir um objetivo específico. Não há ainda um consenso da melhor alternativa de avaliar os estudantes, sendo necessária o desenvolvimento de novos métodos. Para que isso ocorra também é necessária uma formação continuada e constante de professores e encontrar novas abordagens pedagógicas.
- **Entendimento Consolidado:** a pluralidade do significado do termo pode causar confusão, porém esse problema é solucionado com a inclusão de especialistas e parcerias com outros países no momento em que o Ministério da Educação cria sua própria definição, sempre levando em consideração os conceitos já existentes no currículo, leis e pelas próprias escolas. No momento em que o termo se encontra consolidado, se torna mais fácil assinalar onde é a fronteira entre o PC e o ensino das TICs, assim como, identificar conteúdos em comum. Por fim, recomenda-se realizar parcerias com iniciativas da educação informal, pois com a experiência adquirida e as lições aprendidas por essas instituições podem fornecer contribuições valiosas para a integração do PC na educação formal, desde que sejam adaptadas aos contextos específicos.
- **Política de Apoio:** a troca de experiência com políticos de outros países, representantes de iniciativas pré-existentes, pesquisadores e outras partes interessadas (*e.g.* indústria) podem fornecer informações extremamente valiosas, especialmente porque as razões para integrar o PC no currículo são semelhantes em todos os países. É imprescindível que, no momento de implantação do novo currículo, as partes interessadas (principalmente os professores) sejam informadas de maneira clara e objetiva a resposta às seguintes perguntas: “O que é o PC?”, “O que requer/demanda?” e “Quais são os benefícios educacionais?”. Dada a necessidade de adotar uma abordagem holística para a integração efetiva do PC na Educação Básica, deve monitorar e analisar constantemente os resultados das ações implantadas para que sejam comparados às metas políticas e estratégicas.

Figura 43: Implicações políticas e práticas na introdução do PC na Educação Básica



Fonte: adaptado de Bocconi *et al.*, 2016

2.1.2.3 Code.org (EUA/Global)

Lançado no ano de 2013 nos Estados Unidos da América, o Code.Org é uma organização sem fins lucrativos que se dedica na expansão do acesso à Ciência da Computação, incluindo minorias. Tem como visão, possibilitar aos estudantes o acesso à Ciência da Computação, além de acreditarem que a referida disciplina deveria fazer parte do currículo escolar, juntamente com outras, como por exemplo, biologia, química e álgebra.

Tem como principais objetivos:

- Aumentar a diversidade na CC;
- Inspirar estudantes;
- Criar cursos voltados para a CC;
- Levar a CC para as salas de aula;
- Formar professores;
- Mudar o currículo em distritos escolares;
- Auxiliar na mudança de leis estaduais para inclusão da CC no currículo;

- Possibilitar que estudantes de todo o mundo tenham acesso ao material.

A ONG Code.Org teve uma aceitação muito grande nos EUA, chamando a atenção de grandes empresas na área de TI, recebendo muito apoio das mesmas em prol da promoção de seus propósitos. Algumas das empresas parceiras são:

- Amazon;
- America Airlines;
- Apple;
- Association for Computer Machinery (ACM);
- Computer Science Teachers Association (CSTA);
- Disney;
- Dropbox;
- Facebook;
- GitHub;
- Google;
- Khan Academy;
- Microsoft;
- Rovio;
- Salesforce;
- Zendesk.

Uma das iniciativas mais importantes do Code.Org é a “*Hour of Code*” (Hora do Código). A Hora do Código é um movimento global que atinge dezenas de milhões de estudantes em mais de 180 países. Qualquer pessoa, independentemente do lugar, pode organizar um evento da Hora do Código e anunciar no site da organização. Tutoriais de uma hora estão disponíveis em mais de 40 idiomas. Não é exigida qualquer experiência dos participantes do evento que devem ter no mínimo 4 anos. O crescimento do evento encontra-se disponível na Tabela 3.

Tabela 3: Participantes da “Hora do Código”

Ano	Público atingido
2013	20 milhões
2014	90 milhões
2015	195 milhões
2016	308 milhões

Fonte: CODE.ORG (2015)

O principal objetivo da campanha é que dezenas de milhões de estudantes participem da Hora do Código no período pré-estabelecido pela ONG. No ano de 2016, a semana selecionada foi de 5 a 11 de dezembro, quando também é celebrada a Semana da Educação em Ciência da Computação. A semana é determinada como foco na promoção do evento, porém as atividades encontram-se disponíveis permanentemente no site. No Brasil, o principal parceiro da Hora do Código é o Programae.org.br.

2.1.2.4 Programae.org.br (Brasil)

Iniciativa brasileira fundada em 2002 pelo empresário Jorge Paulo Lemann, a Fundação Lemann é uma organização familiar, sem fins lucrativos, promotora dessa iniciativa. A Fundação Lemann desenvolve e apoia projetos inovadores em educação, pela realização de pesquisas para embasar políticas públicas no setor, oferecendo formação para profissionais da educação e aprimorando lideranças em diversas áreas. As iniciativas e ações buscam contribuir para que o Brasil apresente, até 2018, soluções inovadoras, de alta qualidade, no cotidiano da educação de 30 milhões de pessoas, na capacitação de 200 mil professores, a fim de garantir o aprendizado de todos os seus alunos e, ainda, 65 líderes promovendo e acelerando transformações sociais de alto impacto, além de um padrão digno e de grandes expectativas do que é esperado na educação e aprendizagem de todos os alunos.

O Programae.org.br é um movimento que tem como prioridade aproximar a programação do cotidiano de jovens de todo o Brasil e foi criado devido a tecnologia ter um alto poder transformador. De acordo com a ONG, usá-la para a Educação pode fazer a diferença para muitas pessoas, através de um portal prático e agregador de ideias, soluções e dicas de gente experiente e inspiradora. É parceira do Code.Org.

2.1.2.5 Supergeeks.com.br (Brasil)

Primeira escola de programação e robótica para crianças (a partir de sete anos) e adolescentes no Brasil. As aulas baseiam-se na criação de games, aplicativos, robôs, sistemas, empreendedorismo e língua inglesa. A ideia de abertura da escola surgiu durante o período em que um casal brasileiro morava nos Estados Unidos, mais especificamente no Vale do Silício, onde notou-se que empresas e políticos estadunidenses estavam se mobilizando para ensinar Ciência da Computação para crianças e adolescentes. Hoje, a SuperGeeks está se expandindo por todo o Brasil através de franquias e unidades próprias. No final do ano de 2016 a rede já possuía 31 escolas em todo o Brasil. A SuperGeeks utiliza quatro metodologias de ensino, sendo elas:

- *Game Learning*: utilização de *games* para promover o aprendizado de conceitos de programação e Ciência da Computação. Alguns desses games são da própria escola, outros são títulos disponíveis comercialmente;
- *Gamification*: utilização de mecanismos de jogos, como pontos, recompensas e desafios de forma lúdica, para engajar os alunos a aprender. Os alunos recebem ou perdem pontuações como se estivessem em um jogo. A cada tarefa entregue, ganham-se pontos, vidas ou créditos. Se o aluno não entrega alguma tarefa ou apresenta atitudes de inquietação ou indisciplina durante as aulas, perde pontuação ou vida dentro do jogo, ou seja, utiliza os mesmos mecanismos de *games* dentro das salas de aula e durante o curso;
- *Empreendedorismo*: desde a primeira fase, os alunos aprendem a respeito do mercado de *games*, como projetá-los da melhor forma, quais os melhores canais de distribuição e divulgação. A partir da Fase 3, os alunos serão incentivados a criar suas próprias *startups*, sozinhos ou em grupo, e lançar seus produtos no mercado; seja ele um *game*, um aplicativo, uma aplicação web, um hardware, ou qualquer outro tipo de produto tecnológico. Com o passar das fases, os alunos terão contato com conceitos de marketing, vendas, direito, contábil, MVP, RH, Canvas, entre outros conceitos extremamente importantes para empreendedores e futuros empresários.
- *Storytelling*: histórias inseridas em animações, livros e quadrinhos para que os alunos absorvam este conteúdo de forma mais prazerosa. As histórias de aventuras estão sendo inseridas no conteúdo. Assim, além do aluno aprender durante a aula, ou estudando pela apostila em casa, ele irá rever o conteúdo ao ler, assistir ou escutar essas histórias.

Após o pioneirismo da SuperGeeks e a alta demanda por escolas do mesmo gênero, surgiram diversas outras iniciativas similares, tais como: Buddys¹, Ctrl+Play², DragonByte³, Escolapixel⁴, Happy Code⁵, Konfidegeek⁶ e Madcode⁷.

2.1.3 Iniciativas Brasileiras

No Brasil, até o momento da elaboração deste documento, as políticas educacionais relacionadas à tecnologia estão restritas à abordagem de letramento e inclusão digital. Nenhum documento oficial menciona a introdução do ensino de Fundamentos de Computação na Educação Básica, porém existem diversas motivações para que isso ocorra. Essas, estão listadas e detalhadas no Anexo A.

Em 2015 iniciou-se a construção da Base Nacional Curricular Comum (BNCC)⁸ que define os conhecimentos essenciais aos quais todos os estudantes brasileiros têm o direito de acesso, bem como se apropriar durante sua trajetória na Educação Básica (BNCC, 2015). Em pleno processo de renovação e o aprimoramento da Educação Básica no Brasil através da BNCC, a SBC⁹ (Sociedade Brasileira de Computação) está ativamente engajada na introdução do Pensamento Computacional nesse projeto nacional e de grande repercussão, através de ações com as demais sociedades e institutos na tentativa de unir forças, propondo novas diretrizes para alinhamento dos currículos de Licenciatura em Computação e da BNCC e também na elaboração de materiais de divulgação e conscientização. O documento com os argumentos utilizados junto ao MEC encontram-se no Anexo A e a proposta das competências a serem trabalhadas com os estudantes encontra-se detalhada nos Anexo B e Anexo C.

¹ <http://www.buddys.com.br/>

² <http://www.ctrlplay.com.br/>

³ <http://www.dragonbyte.com.br/>

⁴ <http://escolapixel.com.br/>

⁵ <http://www.happycode.com.br/>

⁶ <http://konfidegeeks.com.br/>

⁷ <http://www.madcode.com.br/>

⁸ A Base Nacional Comum Curricular (BNCC) é um instrumento de gestão pedagógica e tem como objetivo definir os conhecimentos essenciais aos quais todos os estudantes brasileiros têm o direito de ter acesso e se apropriar durante sua trajetória na Educação Básica (pública ou particular). A Base é parte de uma nova proposta de Currículo e orienta a formulação do projeto Político-Pedagógico das escolas, permitindo maior articulação deste, sendo também uma importante conquista social.

⁹ A Sociedade Brasileira de Computação (SBC) é uma Sociedade Científica sem fins lucrativos que reúne estudantes, professores, profissionais, pesquisadores e entusiastas da área de Computação e Informática de todo o Brasil. A SBC tem como função fomentar o acesso à informação e cultura por meio da informática, promover a inclusão digital, incentivar a pesquisa e o ensino em computação no Brasil, e contribuir para a formação do profissional da computação com responsabilidade social.

A versão inicial da BNCC não faz referência à área de Computação, mas apresenta Tecnologias Digitais como tema integrador. Ou seja, ainda não há no Brasil o reconhecimento da importância dada aos conhecimentos ligados à Computação da forma como ocorre em outros países. Nesse sentido, a Sociedade Brasileira de Computação (SBC)¹ se articulou para solicitar modificações no teor da BNCC visando considerar a Computação como uma área de conhecimento (SBC, 2016).

A terceira e última versão da Base Nacional Comum Curricular (BNCC) para o ensino infantil e fundamental foi apresentada no dia 06 de abril de 2017 pelo Ministério da Educação (MEC). Este documento inédito no Brasil é considerado fundamental para reduzir desigualdades na educação. O documento define, em linhas gerais, do que os alunos das 190 mil escolas do país devem aprender a cada ano.

Entre diversas propostas, a BNCC determina que, ao longo da educação básica, os estudantes devem desenvolver “Dez Competências Gerais”, tanto cognitivas quanto sócioemocionais, que incluem o exercício da curiosidade intelectual, o uso das tecnologias digitais de comunicação e a valorização da diversidade dos indivíduos. Pode-se destacar três competências que seguem a linha do Pensamento Computacional, sendo elas (MEC, 2017):

- Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e inventar soluções com base nos conhecimentos das diferentes áreas;
- Utilizar conhecimentos das linguagens verbal (oral e escrita) ou verbo-visual (como Libras), corporal, multimodal, artística, matemática, científica, tecnológica e digital para expressar-se e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e, com eles, produzir sentidos que levem ao entendimento mútuo;
- Utilizar tecnologias digitais de comunicação e informação de forma crítica, significativa, reflexiva e ética nas diversas práticas do cotidiano (incluindo as escolares) ao se comunicar, acessar e disseminar informações, produzir conhecimentos e resolver problemas.

Além disso, a BNCC prevê o uso de conceitos do Pensamento Computacional nas disciplinas de Matemática para auxiliar no processo de resolução de um problema, conforme (BNCC, 2017):

A aprendizagem de Álgebra pode contribuir para o desenvolvimento do Pensamento Computacional dos alunos, tendo em vista que eles precisam ser capazes de traduzir

¹A Sociedade Brasileira de Computação (SBC) é uma sociedade científica que incentiva atividades de ensino, pesquisa e desenvolvimento em Computação no Brasil e permanece atenta às políticas governamentais que afetam as atividades de Computação no Brasil, no sentido de assegurar a emancipação tecnológica do país (SBC, 2015).

uma situação dada em outras linguagens, como transformar situações-problema, apresentadas em língua materna, em fórmulas, tabelas e gráficos e vice-versa. Associado ao Pensamento Computacional, cumpre salientar a importância dos algoritmos e de seus fluxogramas, que podem ser objetos de estudo nas aulas de Matemática. Um algoritmo é uma sequência finita de procedimentos que permite resolver um determinado problema. Assim, o algoritmo é a decomposição de um procedimento complexo em suas partes mais simples, relacionando-as e ordenando-as, e pode ser representado graficamente por um fluxograma. A linguagem algorítmica tem pontos em comum com a linguagem algébrica, sobretudo em relação ao conceito de variável. Outra habilidade relativa à álgebra que mantém estreita relação com o Pensamento Computacional é a identificação de padrões para se estabelecer generalizações, propriedades e algoritmos (p. 227).

O documento final da BNCC, foi entregue ao Conselho Nacional de Educação (CNE), onde será analisado e preparado um parecer técnico a ser submetido à votação no conselho. O resultado ainda não é conhecido.

Houve também uma iniciativa do Ensino Superior para que o Pensamento Computacional seja inserido nas escolas brasileiras. Em recente relatório publicado pela CAPES, demonstra-se a necessidade de debater a respeito do assunto e incentivar sua adoção (Navaux *et al.*, 2016):

A Ciência da Computação tem entre seus objetos de estudo a ‘resolução de problemas’. Estes podem ser de qualquer natureza e situados, por exemplo, no contexto de Administração, Antropologia, Biologia, Direito, Educação, Engenharia, Física, Matemática, Medicina, Veterinária, Zoologia e Zootecnia.

O processo cognitivo usado pelos seres humanos para resolver problemas por meio de algoritmos é chamado de Pensamento (Raciocínio) Computacional ou Pensamento Algorítmico. Esse processo, ao lado do raciocínio lógico e matemático, habilita os estudantes a compreender, analisar, especificar e organizar a solução de problemas, a partir do desenvolvimento de habilidades como abstração, refinamento, modularização, recursão e metacognição. O aprimoramento destas habilidades cognitivas tem impacto direto sobre a forma como os indivíduos constroem relações com o mundo.

O domínio da Ciência da Computação e das Tecnologias da Informação é estratégico para o desenvolvimento social e econômico mundial. Esse domínio fundamenta-se em um fluxo contínuo de aprendizado, disseminação e evolução do conhecimento e tecnologias subjacentes, com diversos atores, como estudantes, professores, gestores, escolas, outras instituições de ensino e pesquisa, governo, indústria e associações científicas.

Desta forma, entendemos que o ensino da Ciência da Computação deva ser estimulado desde o ensino fundamental, a exemplo de outras ciências. Estas são questões muito importantes para que no futuro tenhamos cidadãos qualificados capazes de responder aos grandes desafios que se apresentam à humanidade (p. 8).

Em contrapartida, um corpo sólido de pesquisas e projetos, envolvendo o ensino de Computação na Educação Básica, é realizado no Brasil desde a década de 1980. As iniciativas

são muitas e bastante diversificadas. Em meados da década de 1980, Papert (PAPERT e HAREL, 1991) iniciam o uso da linguagem LOGO em escolas em todo mundo. No Brasil até o ano de 1996 muitos projetos foram realizados com programação dessa linguagem (VALENTE, 1996). O uso de robótica educacional que iniciou timidamente com kits de empresas, como a Lego, hoje está amplamente disseminado em muitas escolas e instituições educacionais, utilizando, inclusive, alternativas de baixo custo, envolvendo por vezes a reciclagem de componentes eletrônicos.

Diversas iniciativas de introdução ao Pensamento Computacional têm sido realizadas nos últimos anos, envolvendo pesquisadores de escolas e instituições de ensino superior (público e privado) em diferentes níveis da educação escolar, como por exemplo: Barcelos e Silveira (2012), França e Amaral (2013), Ribeiro *et al.* (2013), Andrade *et al.* (2013), Viel *et al.* (2014), Campos *et al.* (2014), Kologeski *et al.* (2016), Scaico *et al.* (2012), Vieira *et al.* (2013) e Barreto (2013). Outras tentativas de inserção do PC também ocorrem por parcerias entre as prefeituras e universidade (Meirelles, 2016) ou do Ministério da Educação diretamente com o município ou governo estadual através da doação de 27 milhões de reais para acelerar o processo de adoção de aulas de programação na grade curricular em 242 da cidade de São Paulo (SILVA, 2017) (DÓRIA, 2017).

O tema do Pensamento Computacional tem também se tornado foco de muitos trabalhos de mestrado e doutorado, cujos resultados são geralmente divulgados em conferências como o *Workshop* sobre Educação em Computação conhecido pela sigla WEI durante o Congresso Anual da Sociedade Brasileira de Computação (CSBC) e o *Workshop* de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg) que ocorre durante o Congresso Brasileiro de Informática na Educação (CBIE).

Outras iniciativas merecedoras de registro são os CodeClubs, organizados por voluntários para levar atividades de programação para escolas, a criação de ambientes de programação em português como Portugol Studio (NOSCHANG *et al.*, 2014) e iniciativas de incentivo ao ensino de programação em larga escala como o Programaê que tem cada dia alcançado mais adeptos.

Existem também esforços feitos pela Sociedade Brasileira de Computação (SBC) para a disseminação do Pensamento Computacional na Educação Básica no Brasil. Um exemplo é a Olimpíada Brasileira de Informática (OBI)¹, "uma competição organizada nos moldes das outras olimpíadas científicas brasileiras, como Matemática, Física e Astronomia. O objetivo da

¹ <http://olimpiada.ic.unicamp.br/>

OBI é despertar nos alunos o interesse por uma ciência importante na formação básica hoje em dia (no caso, Ciência da Computação), através de uma atividade que envolve desafio, engenhosidade e uma saudável dose de competição".

2.1.4 Tabela Comparativa

Após a revisão bibliográfica sistemática, organizou-se uma tabela comparativa entre diversos países. O levantamento dos dados coletados encontra-se na Tabela 4.

Tabela 4: Quadro Comparativo do PC no Mundo

País	Ano de adoção	Fundamental	Médio	Técnico/Vocacional	Depende da região ou currículo adotado	Possui disciplina específica	Modo de Integração
Alemanha	2004	F	F				N
Argentina	2015	F	F		Sim	Varia	NR
Austrália	2015	C					N
Áustria	2009	F	F	F	Sim	Sim	N
Bélgica / Holanda	Varia	F	F				R
Bulgária	2006		C	C		Sim	N
Coréia do Sul	2007	F	F			Sim	N
Dinamarca	2014	C	F	F		Varia	N
Escócia	1987	F	F		Sim	Varia	R
Eslováquia	1990	C	C	C		Sim	NE
Espanha	2015	F	F		Sim	Varia	NR
Estados Unidos	2015	F	F		Sim	Varia	NR
Estônia	1996	F	F	F	Sim	Varia	NR
Finlândia	2016	C				Não	NRE
França	2016	FC	C		Sim	Não	N
Grécia	1993	C	C			Sim	N
Hungria	1995		C	C		Sim	N
Irlanda	2014	F			Sim	Sim	NE

País	Ano de adoção	Fundamental	Médio	Técnico/Vocacional	Depende da região ou currículo adotado	Possui disciplina específica	Modo de Integração
Israel	1976	F	F	F		Sim	N
Lituânia	1986	F	F			Varia	NE
Malta	1997		F			Sim	N
Polônia	1985	F	F	F	Sim	Sim	N
Portugal	2012	C		C		Sim	N
Reino Unido	2014	FC	FC		Sim	Sim	N
Rep. Tcheca	1990		C	F		Sim	E

F = facultativo / C = compulsório / N=nacional, R=regional, E=escolar

Fonte: próprio autor

Este capítulo teve a intenção de situar o leitor do quanto à implantação do Pensamento Computacional em diferentes países e instituições. O capítulo seguinte dissertará a respeito das motivações desta pesquisa.

3 MOTIVAÇÃO E ESTADO-DA-ARTE

Após uma ampla e extensa revisão bibliográfica a respeito do que é o Pensamento Computacional, os conceitos envolvidos, benefícios e contradições na implantação, alternativas de como o PC pode ser integrado na Educação Básica, como realizar a avaliação do PC em estudantes, conhecer os países que já possuem o PC na educação formal e informal, este capítulo vem à luz deste levantamento dissertar o que motivou o pesquisador a realizar esta investigação.

Este trabalho teve a motivação de quatro macro elementos, sendo eles: a experiência, evasão dos cursos na área da informática e alta demanda por profissionais, a preparação dos estudantes para um mundo que necessita de fluência digital, os resultados do Brasil no teste pisa e a viabilidade técnica, econômica e cultural. Esses itens serão melhor detalhados a seguir.

3.1 EXPERIÊNCIA, EVASÃO DOS CURSOS NA ÁREA DA INFORMÁTICA E ALTA DEMANDA POR PROFISSIONAIS

O autor atua na docência em disciplinas na área de programação, principalmente nas de Algoritmos e Estrutura de Dados, onde é facilmente percebida a dificuldade que os estudantes têm de codificar. Essa dificuldade reflete nos índices de reprovação e evasão das disciplinas ou até mesmo do curso, conforme estudos apontados por Rodrigues *et al.* (2015). A evasão é um fenômeno presente no ensino superior brasileiro e que causa reflexos negativos em vários sentidos, seja para os estudantes que abandonam o curso, reduzindo suas chances de crescimento pessoal e profissional, seja para as instituições, que deixam de cumprir sua função social de educar, por um lado, e de fomentar o mercado de trabalho, por outro. Dessa forma, fica estabelecida a importância deste fator motivacional e sua relação com o atual cenário e realidade, sendo necessário que a evasão seja vista no contexto educacional como um problema a ser combatido.

Dessa forma, a realização deste trabalho pretende contribuir para uma maior reflexão empírica sobre a inclusão dos conceitos envolvidos na Computação, desde o Ensino Fundamental, para que ocorra uma possível diminuição das barreiras que os estudantes encontram ao adentrar no ensino superior. Com a diminuição dessas dificuldades, acredita-se que também ocorra uma diminuição da evasão a longo prazo e, conseqüentemente, o aumento no número de egressos para atender a alta demanda de profissionais necessários em pesquisas científicas e demais áreas.

Além disso, é necessário levar em consideração que, quando os estudantes chegarem no ensino superior, é provável que se terá também programadores com uma melhor formação, pois os obstáculos iniciais já foram sanados e o estudante pode se concentrar na sofisticação de seus conhecimentos. A demanda por profissionais que desenvolvam uma solução robusta, segura e fácil de usar tem se tornado cada vez mais frequente, tendo em vista que empresas não podem colocar em risco sua infraestrutura baseada em software.

Para que crianças tenham um interesse maior pela Computação, é possível perceber uma mudança de comportamento do comércio, pois vem ocorrendo uma demanda maior a procura por jogos e livros que ensinam e exercitam o PC. Essa demanda surge, tanto pelo lado da indústria que necessita de profissionais com uma formação sólida, quanto pelas escolas que ainda não possuem um material didático conciso e específico, também pelos pais que desejam seus filhos bem preparados para o mercado altamente competitivo a ser enfrentado, ou os próprios estudantes que acham a programação muito divertida e acaba se tornando mais um *hobby* do que um estudo. Algumas dessas peças foram relatadas no capítulo 1.6.

3.2 PREPARAR OS ESTUDANTES PARA UM MUNDO QUE NECESSITA DE FLUÊNCIA DIGITAL

Vivemos em tempos em que a criatividade do homem faz a diferença, em que a nova economia mundial não se baseia apenas em recursos naturais e matérias-primas, mas em conhecimento, fluxos de informação e habilidades em usá-los, sem esquecer, claro, que a situação é ideal em países como o Brasil, com o potencial de combinar os dois fatores acima mencionados. A Ciência da Computação oferece não apenas softwares úteis e artefatos de hardware para essa competição, mas também uma maneira diferenciada de pensar, e que todos, independentemente da área, podem se beneficiar ao pensar computacionalmente e descobrir novas ciências através da análise de uma quantidade gigantesca de dados ou fazer questionamentos que nunca foram cogitados ou ousados devido a sua escalabilidade, facilmente atendida pela Computação. Os benefícios são diversos e já foram citados no capítulo 1.3 (p. 41).

O ensino dos fundamentos da Computação na Educação Básica (Pensamento Computacional) beneficia o desenvolvimento de habilidades e competências essenciais para a vida moderna, independente da área em que atuará. Vale salientar que essa proposta coaduna com ações em diversos países que já possuem disciplinas de Computação em seu currículo. Se crianças que se encontram no primeiro ano do ensino fundamental já aprendem as regras

fonéticas e gramaticais ou identificam sequências recursivas de números naturais, o que difere em relação às regras de programação? (Brasil/MEC, 2017, p. 235). Assim como a Geografia que nos faz compreender o mundo em que vivemos, a História que nos ensina as nossas origens e como chegamos a ser o que somos e a Matemática que nos explica a lógica dos números, a Computação com os sistemas inteligentes e a inteligência artificial tem grande impacto sobre nossas vidas e tomam decisões que afetam a vida de todas as pessoas. Sob esse ponto de vista, não seria relevante a inclusão de algoritmos no currículo da escola?

A linguagem de programação que será utilizada em sala de aula é uma decisão importante no ensino dos fundamentos da Computação. Conforme Downey (2017), as linguagens de programação evoluíram de tal forma que estão mais expressivas, legíveis, concisas, precisas e executáveis, tornando o processo de ensino-aprendizagem muito mais fluido e natural. Uma comparação de algoritmo em pseudocódigo utilizado comumente em livros didáticos (A) e moderno (B) encontra-se na Figura 44. Essa facilidade possibilita que estudantes possam ter uma maior afinidade com a codificação, independentemente da área que venha a estudar no futuro.

Figura 44: Comparativo entre duas gerações de algoritmos

A	B
<pre> BFS(<i>G</i>, <i>s</i>) 1 for each vertex <i>u</i> ∈ <i>V</i>[<i>G</i>] − {<i>s</i>} 2 do <i>color</i>[<i>u</i>] ← WHITE 3 <i>d</i>[<i>u</i>] ← ∞ 4 π[<i>u</i>] ← NIL 5 <i>color</i>[<i>s</i>] ← GRAY 6 <i>d</i>[<i>s</i>] ← 0 7 π[<i>s</i>] ← NIL 8 <i>Q</i> ← {<i>s</i>} 9 while <i>Q</i> ≠ ∅ 10 do <i>u</i> ← head[<i>Q</i>] 11 for each <i>v</i> ∈ Adj[<i>u</i>] 12 do if <i>color</i>[<i>v</i>] = WHITE 13 then <i>color</i>[<i>v</i>] ← GRAY 14 <i>d</i>[<i>v</i>] ← <i>d</i>[<i>u</i>] + 1 15 π[<i>v</i>] ← <i>u</i> 16 ENQUEUE(<i>Q</i>, <i>v</i>) 17 DEQUEUE(<i>Q</i>) 18 <i>color</i>[<i>u</i>] ← BLACK </pre>	<pre> def bfs(<i>G</i>, start): dist = {start: 0} tree = DiGraph() queue = deque([start]) while queue: node = queue.popleft() for child in G.neighbors(node): if child not in dist: dist[child] = dist[node] + 1 tree.add_edge(child, node) queue.append(child) return dist, tree </pre>

Fonte: Adaptado de Downey (2017)

3.3 RESULTADOS DO BRASIL NO TESTE PISA

O PISA (*Programme for International Student Assessment*) é um estudo mundial realizado pela OCDE (*Organization for Economic Co-operation and Development*) em 70 países que mede o desempenho escolar de estudantes na faixa de 15 anos de idade nas áreas de matemática, ciências e leitura. O teste ocorre a cada três anos e tem como objetivo fornecer dados comparáveis, permitindo que os países melhorem suas políticas e resultados

educacionais. Dados do PISA do ano de 2015 (mais recente), nos demonstra dados preocupantes em relação a outros países, pois o Brasil ficou na 63ª posição em ciências, 59ª em leitura e na 66ª colocação em matemática (OECD, 2017).

Como se pode perceber no Quadro 8, as três áreas tiveram uma melhora significativa, sendo que na última edição houve uma estabilidade na área de leitura. Embora tenha havido uma elevação na pontuação de 396 pontos em 2000, para 407 pontos em 2015, esta diferença não representa uma mudança estatisticamente significativa. O mesmo caso ocorre na área da ciência que passou de 390 para 401 pontos. Já na área de matemática, houve um aumento significativo de 21 pontos na média dos alunos entre 2003 a 2015, porém, ao mesmo tempo, houve um declínio de 11 pontos se compararmos a média de 2012 à média de 2015.

Um gráfico com o histórico de pontuações dos alunos brasileiros na prova encontra-se na Figura 45, e uma relação da pontuação brasileira com a média mundial encontra-se na Figura 46. Lembrando que os dados utilizados na confecção dos gráficos encontram-se no Quadro 8 e no Quadro 9, respectivamente.

Quadro 8: Histórico de pontuações do Brasil no teste PISA

	Matemática	Leitura	Ciências
2000	*	396	*
2003	356	403	*
2006	370	393	390
2009	386	412	405
2012	389	407	402
2015	377	407	401

Fonte: adaptado de OECD (2017)

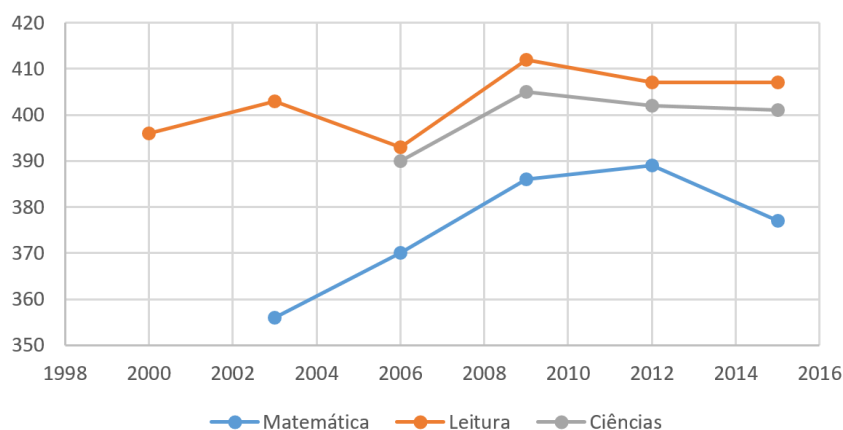
Quadro 9: Histórico da média mundial de pontuações no teste PISA

	Matemática	Leitura	Ciências
2000	*	493	*
2003	499	494	*
2006	494	489	498
2009	495	493	501
2012	494	496	501
2015	490	493	493

Fonte: adaptado pelo próprio autor de OECD (2017)

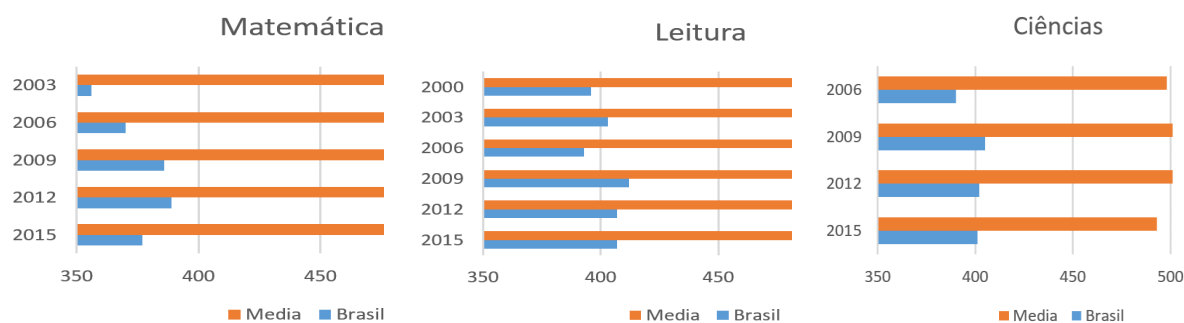
* Dados inexistentes

Figura 45: Histórico de pontuações do Brasil no PISA



Fonte: próprio autor com dados de OECD (2017)

Figura 46: Comparativo de pontuações do Brasil em relação à média mundial



Fonte: próprio autor com dados de OECD (2017)

Uma das alternativas para melhoria do desempenho no Brasil seria a inclusão do PC no currículo da educação formal, tendo em vista pesquisas que compararam o desempenho dos estudantes em outras disciplinas após adoção do PC em sala de aula. Clements (2002) e Papert (1980) relatam em suas pesquisas fortes evidências de melhoria significativa em áreas como matemática, ciência, arte e linguística. Em pesquisa mais recente (DORAN *et al.*, 2012), os mesmos resultados são encontrados nas áreas de matemática e linguagem e também são relatados por Wing (2014), onde ainda acrescenta as áreas de biologia e ciência. Moreno-León *et al.* (2017) também identificou melhorias nas áreas de matemática e ciências sociais.

Por fim, é importante citar que a prova PISA já está em processo de desenvolvimento através de um teste que também medirá a capacidade dos estudantes de resolver problemas e trabalhar colaborativamente (características do PC). A intenção é que este tipo de questão já esteja presente na próxima prova que ocorrerá no ano de 2019. A última versão do documento com a estrutura e exemplo de questões foi publicado em abril de 2017 (PISA, 2017) e está em fase de rascunho.

Levando-se em consideração a classificação do Brasil, em comparação a outros países que já adotaram o PC em seu currículo, pesquisas comprovando a eficácia do PC no ensino de outras áreas e a necessidade iminente de exercitar atividades que envolvam a resolução de problema e trabalho colaborativo, o autor entende que o PC pode, sim, auxiliar na melhoria de desempenho dos estudantes e o conseqüente aumento na posição mundial.

3.4 VIABILIDADE TÉCNICA, ECONÔMICA E CULTURAL

As escolas brasileiras que ofertam o Ensino Fundamental (EF) esbarram em uma dura realidade e diferente do ideal. O censo escolar mais recente (MEC/INEP, 2017) apresenta dados chocantes no EF, como por exemplo a ausência de bibliotecas ou sala de leituras em 53,6% delas. Levando-se em consideração o total de 27.931.210 matrículas no EF, estima-se que 14.971.129 das crianças não têm acesso a esse serviço tão importante na sua formação cultural. Outros indicadores pertinentes a essa pesquisa encontram-se no Quadro 10.

Quadro 10: Recursos ausentes em escolas do Ensino Fundamental

Recurso Ausente	%	Relativo ao Total de Matrículas
Internet	44,2%	12.345.595
Laboratório de informática	48,8%	13.630.430
Energia elétrica	5,5%	1.536.217

Fonte: próprio autor com dados de MEC/INEP (2017)

Os dados apresentados no Quadro 10 mostram os recursos que são essenciais para que se faça um bom uso dos equipamentos informáticos nas escolas. Note que esses dados não contemplam situações como: mau funcionamento e velocidade do *link* da internet, quantidade de computadores, número de máquinas que se encontram danificadas ou desativadas por algum motivo, instabilidade ou paralisação temporária do serviço da rede elétrica por motivos de força maior ou, ainda, por algum problema burocrático. Essas são situações que são facilmente encontradas ao visitar as escolas. Levando-se em consideração essas intempéries, acredita-se que os indicadores possam ser ainda maiores.

Em um movimento contrário, a maioria das escolas no mundo, onde o foco é informatizá-las, algumas na Austrália estão proibem alunos de trazer laptops para a escola e requerem que eles façam as tarefas à mão até o primeiro ano da educação básica (BITA, 2016). De acordo com os diretores das escolas, o equipamento é um meio de distração aos estudantes e o trabalho manual os ensina a serem mais sociáveis. Além disso, de acordo com Bita, essas escolas que seguem a mesma filosofia são frequentemente vistas no topo de listas de concursos

nacionais de alfabetização e aritmética. É possível encontrar também em escolas na Finlândia a mesma linha de raciocínio, ou seja, ensinando o Pensamento Computacional sem o uso de equipamentos, uma abordagem compatível com a realidade brasileira. De acordo com Deruy (2017), alunos aprendem o funcionamento de um *loop* (repetição) nas aulas de educação física, criando danças e sequências de passos, ou nas aulas de arte através do tricô. Essas atividades utilizam conceitos que os estudantes são familiarizados naquele país.

Para que as crianças sejam motivadas com atividades de Pensamento Computacional Desplugado, uma alternativa é usar elementos que sejam significativos para elas (*e.g.* desenhos, cultura local, etc.). Um exemplo é relatado por Unnikrishnan *et al.* (2016), onde jogos de PC em tabuleiros usaram animais e frutas típicas da Índia com sucesso, aproximando a criança da sua rotina. Essa estratégia se demonstra interessante para a realidade brasileira, devido a diferentes culturas que são encontradas em uma nação de proporções continentais. Essa tentativa de aproximação da cultura brasileira com jogos que ensinam lógica também é encontrada em Mello e Mello (2011) no denominado “Jogos Boole”, onde elementos do cotidiano das crianças são utilizados como uma forma de motivação.

Enfim, a presente pesquisa teve diversas motivações, tais como: as evasões dos cursos superiores que não atendem a alta demanda do mercado, uma formação condizente com a revolução digital em que se encontra, o baixo desempenho dos estudantes brasileiros no teste PISA e a viabilidade técnica, econômica, cultural e geográfica. Sendo assim, propôs-se a seguinte pergunta de pesquisa: “Levando-se em consideração a necessidade de inclusão dos fundamentos da Computação na Educação Básica em conjunto com a realidade brasileira, a abordagem desplugada é uma maneira eficiente de ensinar Pensamento Computacional para formar alunos com uma maior fluência digital e seguir uma tendência mundial?”.

No capítulo seguinte fará-se uma explanação a respeito da metodologia utilizada para se buscar uma resposta para esse questionamento.

4 MATERIAIS E MÉTODOS

Essa seção apresenta a metodologia e materiais utilizados para responder a questão e objetivos da pesquisa, incluindo cada uma das etapas percorridas durante a investigação que passou por dois projetos pilotos, uma aplicação na Espanha e outra no Brasil durante os anos de 2016 e 2017. As duas etapas iniciais aconteceram, efetivamente, com a intenção de colocar à prova, tanto as atividades, como também a maneira que a avaliação ocorria. Desta forma seria possível identificar pequenos ajustes, melhorias e retoques para que as demais aplicações pudessem ocorrer com lisura. Pelo fato de os dois procedimentos iniciais terem sido considerados piloto, os resultados qualitativos e quantitativos serão relatados, cujos dados gerados serviram apenas como uma referência às etapas posteriores.

Quanto à forma de abordagem, esta pesquisa é categorizada como quali-quant, pois utiliza elementos qualitativos e quantitativos. Qualitativa, por não se preocupar com representatividade numérica, mas, sim, com a compreensão, além de descrever e explicar a maneira de pensar e agir dos estudantes. Quantitativa por focar na objetividade e comprovar uma teoria através de dados coletados com o auxílio de instrumentos padronizados e neutros. A utilização conjunta da pesquisa qualitativa e quantitativa permite compilar mais informações do que se poderia conseguir isoladamente (GERHARDT e SILVEIRA, 2009). Tendo em vista que esta pesquisa tem o objetivo de gerar conhecimentos para aplicação prática, dirigidos à solução de problemas específicos, entende-se que sua natureza é aplicada e com objetivos exploratórios. Já em relação aos procedimentos, optou-se por uma abordagem quase-experimental, pois caracteriza-se por sua execução com grupos de comparação (experimental e controle), não necessitar longos períodos de observação e tomada de dados e pelo fato da pesquisa não permitir uma seleção aleatória da amostra (SHADISH *et al.*, 2001). É importante fazer constar também que este trabalho envolve os três elementos do tripé educacional: pesquisa, extensão e ensino.

As crianças que participaram da pesquisa foram escolhidas aleatoriamente pela gestão da escola e gentilmente cedidas para participarem das atividades de pesquisa. A pesquisa foi aprovada e encontra-se protocolada no Comitê de Ética (CEP/UFRGS) sob CAE 57968016.8.0000.5347. O termo de consentimento livre e esclarecido é apresentado no Apêndice O. A respeito da faixa etária das crianças, existem raros artigos, dos quais se destaca o trabalho de Moreno-León *et al.* (2016), que realizou avaliações quali-quantitativas em turmas de segundo e sexto anos. A pesquisa concluiu que as crianças do segundo ano não apresentaram uma diferença expressiva no aprendizado, diferentemente das do sexto ano, que revelaram um

aumento significativo nas notas dos testes aplicados. Outra pesquisa (SCHIFF e VAKIL, 2015) também comprovou que crianças de sexto ano superaram as de terceiro ao tentar resolver o problema das Torres de Hanoi, uma atividade que envolve principalmente as aptidões cognitivas de planejamento, lógica, abstração, recursividade, modelagem mental e autocorreção de erros. Franklin *et al.* (2017) também realizou uma análise de 123 estudantes com idades entre 9 e 12 anos e identificou que a partir do quarto ano as crianças já conseguem programar efetivamente com uma linguagem de programação baseada em blocos, porém as que cursam o quarto ano têm uma dificuldade maior de iniciação em programação. Essa dificuldade também ocorre no quinto ano, mas com intensidade menor.

Levando-se em consideração os achados dos pesquisadores, optou-se por trabalhar com turmas de quinto e sexto anos, ou seja, frequentando o último ano do EF I e o primeiro ano do EF II, respectivamente. Crianças desses níveis escolares possuem entre 10 e 11 anos de idade. A relação de idade e nível escolar Espanhol e Brasileiro são apresentados no Quadro 11 e Quadro 12, respectivamente. Como se pode perceber, as crianças permanecem um tempo equivalente na Educação Básica, porém recebe uma nomenclatura distinta.

Quadro 11: Relação Idade e Nível Escolar na Espanha

EDUCAÇÃO BÁSICA – ESPANHA												
	Primária						Secundária				Bacchirelato	
Ano	1	2	3	4	5	6	1	2	3	4	1	2
Idade aprox.	6	7	8	9	10	11	12	13	14	15	16	17

Fonte: Adaptado de Ministerio de Educación, Cultura y Desporte (2017)

Quadro 12: Relação Idade e Nível Escolar no Brasil

EDUCAÇÃO BÁSICA – BRASIL												
	Ensino Fundamental I						Ensino Fundamental II			Ensino Médio		
Ano	1	2	3	4	5	6	7	8	9	1	2	3
Idade aprox.	6	7	8	9	10	11	12	13	14	15	16	17

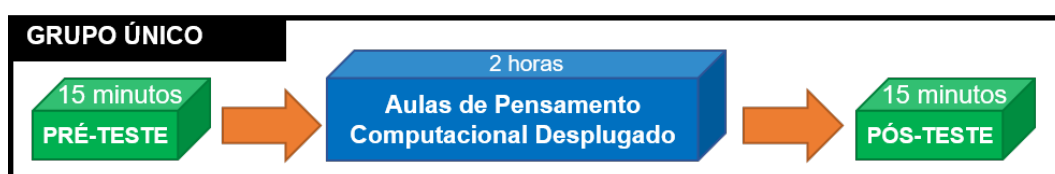
Fonte: Adaptado de Lei 11.274, de 06 de fevereiro de 2006

Após definidos as características e os sujeitos da pesquisa, relata-se as etapas que compreendem toda a investigação. Foram empregadas mais de uma ferramenta avaliativa, as quais serão melhor descritas na etapa relativa à sua utilização.

4.1 PROJETO PILOTO 1

A versão embrionária do projeto ocorreu na sexta série da Escola Básica Estadual Dr. Paulo Devanier Lauda na cidade de Santa Maria, RS. Como se tratava de uma intervenção inicial e com objetivo exploratório, selecionaram-se duas questões como ferramenta de avaliação e duas atividades que abrangessem a maior quantidade de pilares do PC possível, *i.e.*, ainda, que fossem avaliadas e praticadas as aptidões de abstração, decomposição, reconhecimento de padrão e algorítmica.

Figura 47: Etapas do primeiro projeto piloto



Fonte: próprio autor

Todo o processo de pré-avaliação, atividades e pós-ocorreram em apenas um turno de aula, devido a quantidade pequena de questões e exercícios envolvidos e não havia um grupo de controle. O grupo experimental foi composto por 16 estudantes, sendo que nenhum dos deles teve contato com aulas formais de programação. Esse último pré-requisito foi utilizado em todas as fases desta pesquisa.

4.1.1 Instrumento Avaliativo

O instrumento de avaliação foi modelado com a intenção de incluir os quatro pilares do Pensamento Computacional em duas questões. A primeira questão tinha como foco principal abstrair as informações importantes de uma “conta armada”, reconhecer o padrão que se repete (duas colunas de números), decompondo o problema em passos simples de serem reproduzidos através de um algoritmo. A segunda questão trata de uma decomposição simples de números. O teste foi aplicado individualmente e em papel (desplugado), antes e após as atividades. O mesmo pode ser conhecido na Figura 48.

Figura 48: Instrumento avaliativo incipiente


<p>Explique os passos que você utiliza para solucionar a conta:</p> $\begin{array}{r} 1 \ 2 \\ + 3 \ 4 \\ \hline 4 \ 6 \end{array}$	<ol style="list-style-type: none"> 1. _____ 2. _____ 3. _____ 4. _____ 5. _____ 6. _____ 7. _____ 8. _____ 9. _____ 10. _____
<p>Decomponha os números:</p>	<p>125: _____ centos, _____ dezenas e _____ unidades</p> <p>311: _____ centos, _____ dezenas e _____ unidades</p>

Fonte: próprio autor

4.1.2 Desenvolvimento de atividades

Foram desenvolvidas duas atividades inspiradas em materiais criados por BBC Learning (2015), sendo a primeira focada na decomposição de problemas corriqueiros e a segunda na tentativa de encontrar trajetórias em um tabuleiro entre diferentes pontos (personagens). Um melhor detalhamento de cada atividade é apresentado nos Quadros 13 e 14 e foram entregues em papel (desplugado) para cada criança. As versões integrais das atividades estão localizadas no apêndice indicado em cada quadro.

Quadro 13: Primeira atividade utilizada nas intervenções

Atividade 1: Decomposição (vide Apêndice C, p. 208)	
<p>Questão-Exemplo</p>	<p style="text-align: center;">Plantar uma árvore</p> <div style="display: flex; justify-content: space-around; align-items: center;">  <div style="border-left: 1px dashed black; padding-left: 10px;"> <ol style="list-style-type: none"> 1. _____ 2. _____ 3. _____ 4. _____ 5. _____ 6. _____ 7. _____ 8. _____ </div> </div>
DESCRIÇÃO	
<p>Material necessário:</p> <ul style="list-style-type: none"> • uma folha com imagens de atividades cotidianas diversas • um lápis e uma borracha <p>Objetivo: exercitar prioritariamente os pilares de Abstração, Decomposição e Algoritmos através da criação de uma lista de instruções necessárias para atingir seis objetivos comuns do cotidiano.</p>	

Instruções:

- Entregar uma folha para cada estudante
- Pedir aos estudantes para escrever nas linhas laterais das situações os passos necessários para sua conclusão, decompondo um problema grande em diversos menores.

Após os estudantes terminarem, faz-se a correção oral, inserindo alguns possíveis equívocos, como por exemplo: colocar uma semente na terra antes de cavar um buraco, esquecer de tapar o buraco, etc.

Fonte: próprio autor

Quadro 14: Segunda atividade utilizada nas intervenções

Atividade 2: Mapa da Turma da Mônica (vide Apêndice D, p. 209)																																								
Questão-Exemplo																																								
	<table border="1"> <tr> <td rowspan="2">Cebolinha - Árvore</td> <td>A</td> <td>↑</td> <td>→</td> <td>→</td> <td>→</td> <td>→</td> <td>→</td> <td>→</td> <td>→</td> <td>→</td> <td>↓</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>B</td> <td>↑</td> <td>8x</td> <td>→</td> <td>↓</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	Cebolinha - Árvore	A	↑	→	→	→	→	→	→	→	→	↓									B	↑	8x	→	↓														
	Cebolinha - Árvore		A	↑	→	→	→	→	→	→	→	→	↓																											
		B	↑	8x	→	↓																																		
Mônica - Magali	A	↑	↑	↑	→	→	→	→	→	↓	↓	↓	↓	←	←																									
	B	3x	↑	5x	→	4x	↓	2x	←																															

DESCRIÇÃO

Material necessário:

- Um tabuleiro com os personagens da Turma da Mônica
- Uma folha de resposta

Objetivo: exercitar prioritariamente os pilares de Reconhecimento de Padrão e Algoritmos através da busca por trajetos entre dois pontos (personagens) e aprender uma forma de escrever resumidamente os mesmos comandos.

Instruções:

- Entregar uma folha para cada estudante
- O objetivo é encontrar o menor caminho entre o ponto inicial (personagem 1) e o ponto final (personagem 2) descrito no lado esquerdo.
- Registrar a rota escolhida através de flechas (instruções), indicando como o personagem deve se deslocar pelo tabuleiro, na linha indicada como “A”;
- Após finalizados todos os trajetos “A”, os estudantes devem então abreviar suas instruções com o uso de multiplicadores (2x, 3x, 4x, etc.) na linha “B” de cada trajeto. Por exemplo:
 $\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\leftarrow$ pode ser compactado como $5x\rightarrow 7x\uparrow\leftarrow$

O personagem não pode sobrepor a árvore durante o caminho. O rio não pode ser atravessado em qualquer ponto, neste caso deve-se usar a ponte.

Fonte: próprio autor

Como as atividades utilizadas por esta pesquisa adotam personagens que estão sob direitos autorais criados pelo desenhista Maurício de Souza, o autor solicitou autorização de uso de imagem da Turma da Mônica. Após um ano e meio de negociações, a permissão foi emitida e encontra-se no Anexo D, onde especifica que o material tem o objetivo de “servir de apoio, consulta e inspirações para alunos e professores de escolas públicas e particulares de ensino no Brasil e exterior”.

4.1.3 Resultados do primeiro piloto

Conforme já comentado, o tamanho da amostra foi de 16 sujeitos da sexta série. Foram contabilizados apenas a quantidade de estudantes que responderam as questões corretamente. Um comparativo simplificado da mudança de desempenho dos estudantes encontra-se na Tabela 5.

Tabela 5: Resultados do primeiro piloto

	Acertos	Estudantes
Antes das atividades	7	43,8%
Após as atividades	11	68,8%

Fonte: próprio autor

Após o recolhimento do material, correção e análise, pode-se perceber que houve uma melhora na quantidade de estudantes que acertaram as questões, passando de 7 para 11 acertos, uma melhora de 63,4%. Como essa etapa da pesquisa não tinha a intenção de gerar dados estatísticos, o teste não contemplou um grupo de controle e verificou-se a necessidade de buscar uma ferramenta de avaliação que contemplasse de maneira mais ampla e que mensurasse de forma mais objetiva todos os conceitos envolvidos pelo Pensamento Computacional, além de obter um reconhecimento científico.

A etapa exploratória também oportunizou ao pesquisador entender melhor a realidade das escolas e estudantes brasileiros, além de viabilizar a melhoria contínua das atividades desenvolvidas ou adaptadas. Mesmo que as crianças tivessem um alto nível de motivação e uma professora incentivada com o projeto, perceberam-se algumas deficiências. Um exemplo é a dificuldade de correção dos exercícios, onde foram encontrados graves erros de português e frases desconexas. Durante as atividades notou-se também uma notável dificuldade em entender as regras dos exercícios e como resolvê-los, mesmo após o pesquisador, juntamente com o professor da classe resolverem os problemas no quadro. No final, ao recolher o material, o pesquisador observou pacientemente o interesse dos alunos em pintar os personagens que constavam nas atividades.

Com os resultados analisados, identificou-se que o ensino do Pensamento Computacional poderia ser uma possibilidade válida. Porém, seria necessário utilizar uma ferramenta avaliativa validada e a criação de novas atividades que pudessem exercitar todos os pilares do PC e de maneira mais extensa. Esses dois quesitos foram atendidos no segundo projeto piloto.

4.2 PROJETO PILOTO 2

A segunda etapa da pesquisa também ocorreu no mesmo ano e escola da etapa anterior, porém em uma turma distinta que não foi submetida a intervenção. Levando-se em consideração uma boa aceitação por parte dos alunos, as atividades e avaliação permaneceram no formato desplugado (papel).

4.2.1 Instrumento Avaliativo

O Teste de Pensamento Computacional utilizado na pesquisa foi escolhido devido à sua abordagem quantitativa e aptitudinal e porque já sofreu um rigoroso processo de validação, que comprovou a validade do conteúdo (ROMÁN-GONZÁLEZ, 2015), a validade dos critérios (ROMÁN-GONZÁLEZ *et al.*, 2017) e a validade convergente (ROMÁN-GONZÁLES *et al.*, 2017).

O referido teste, composto por 28 questões, tenta identificar a habilidade de formação e solução de problemas, baseando-se nos conceitos fundamentais da Computação, além de utilizar sintaxes lógicas usadas nas linguagens de programação e inclui conceitos dos quatro pilares do Pensamento Computacional: abstração, decomposição, reconhecimento de padrões e algoritmos, foco desta pesquisa. Uma relação dos pilares com as questões é apresentada no Quadro 15. De um modo geral, os estudos psicométricos do teste mostram ser confiáveis ($\alpha \approx .80$) e compatíveis para avaliação do nível de PC em estudantes de 10 a 16 anos de idade.

Quadro 15: Pilares do PC no Teste de Pensamento Computacional

Questão	Abstração	Decomposição	Rec. de Padrões	Algoritmo
01	X			X
02	X			X
03	X			X
04		X	X	X
05		X	X	X
06		X	X	X
07	X	X		X
08			X	X
09			X	X
10		X	X	X
11	X	X	X	X
12	X	X	X	X
13	X	X		X
14	X		X	X
15	X	X	X	X
16				X

Questão	Abstração	Decomposição	Rec. de Padrões	Algoritmo
17			X	X
18			X	X
19				X
20			X	X
21	X	X		X
22	X	X	X	X
23	X	X	X	X
24				X
25	X	X	X	X
26	X	X	X	X
27	X	X	X	X
28	X	X	X	X

Fonte: próprio autor, tradução

O instrumento é constituído por questões de múltipla escolha, sendo que cada questão possui quatro alternativas de resposta e somente uma é válida. O teste se divide em, basicamente, três partes, a primeira utiliza flechas para deslocar o personagem, a segunda parte faz uma movimentação relativa à posição e direção do personagem utilizando blocos e a última utiliza um lápis para fazer desenhos também com o uso de blocos. É importante constar que as questões possuem compatibilidade com uma abordagem desplugada. A seguir se apresenta um exemplo de cada uma das partes.

O primeiro exemplo é apresentado na Figura 49, onde se pergunta quais são os comandos que levam o 'Pac-Man' até o fantasma pelo caminho indicado. Ou seja, levar 'Pac-Man' exatamente à caixa em que o fantasma está (sem passar, nem parar), e seguindo estritamente o caminho marcado em amarelo (sem sair e sem tocar nas paredes, representadas pelos quadrados laranja). A alternativa correta é a letra B.

Figura 49: Exemplo I de questão do Teste de Pensamento Computacional

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?

Alternativa A	→ → ↓
Alternativa B	→ → ↑
Alternativa C	→ ↑ ↑
Alternativa D	→ ↓ ↓

Fonte: traduzido pelo próprio autor

O segundo exemplo consta na Figura 50, em que novamente se pergunta quais são os comandos que levam o 'Pac-Man' até o fantasma pelo caminho assinalado. Mas neste caso, as opções de resposta, ao invés de flechas, são blocos que encaixam uns nos outros. A alternativa correta neste exemplo é a C.

Figura 50: Exemplo II de questão do Teste de Pensamento Computacional

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?


Alternativa A	avance vire à esquerda ↶ avance avance	Alternativa B	avance vire à direita ↷ avance avance
Alternativa C	avance avance vire à esquerda ↶ avance	Alternativa D	avance avance vire à direita ↷ avance

Fonte: traduzido pelo próprio autor

No terceiro exemplo pergunta-se quais comandos deve seguir o artista para desenhar a figura que aparece na tela, ou seja, como mover o lápis para esboçar a figura. O comando

avançar empurra o lápis, desenhando uma linha. A seta cinza indica a direção do primeiro movimento do lápis. A alternativa correta neste exemplo é A.

Figura 51: Exemplo III de questão do Teste de Pensamento Computacional

<p>Qual sequência o artista deve seguir para desenhara figura abaixo? O lado menor mede 50 pixels e o maior mede 100 pixels.</p> 	<p>Alternativa A ✓</p> <pre> avance por 50 pixels vire à esquerda por 90 graus avance por 100 pixels </pre>	<p>Alternativa B</p> <pre> avance por 50 pixels vire à direita por 90 graus avance por 100 pixels </pre>
	<p>Alternativa C</p> <pre> avance por 100 pixels vire à esquerda por 90 graus avance por 50 pixels </pre>	<p>Alternativa D</p> <pre> avance por 100 pixels vire à direita por 90 graus avance por 50 pixels </pre>

Fonte: traduzido pelo próprio autor

A versão original do teste foi desenvolvida no idioma Espanhol (Europeu) e traduzido pelos alunos Rafael Marimon Boucinha e Christian Puhmann Brackmann para o Português (Brasil) sob autorização do criador (Prof. Dr. Marcos Román-Gonzáles) que, também autorizou que seu teste fosse utilizado nesta pesquisa.

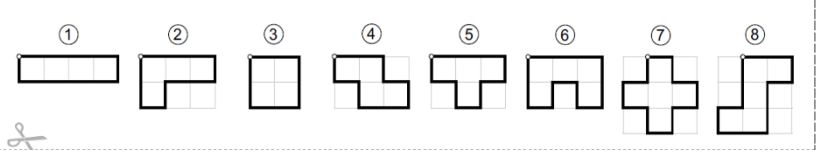
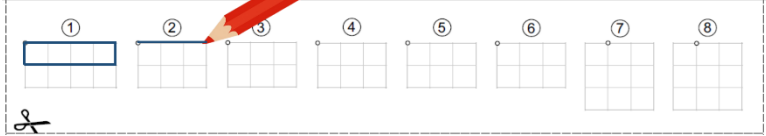
A aplicação do teste pode ocorrer em qualquer navegador (*e.g.* Chrome, Firefox, Edge) e pode ser acessado virtualmente de qualquer dispositivo (*e.g.* computador, tablet, *smartphone*), porém tendo em vista que esta pesquisa tem a intenção de trabalhar de maneira desplugada, optou-se pela adoção de uma prova em papel. Todas as questões no navegador foram convertidas para um documento impresso (Apêndice A, p. 190) e também foi desenvolvida uma Folha de Respostas (Apêndice B, p. 207) para que os alunos pudessem marcar a opção correta, sem que houvesse a necessidade de imprimir uma nova prova para cada sujeito.

4.2.2 Desenvolvimento de atividades

Nessa nova fase da pesquisa, desenvolveu-se uma quantidade superior de atividades para que se pudesse atingir de maneira mais abrangente e extensa os pilares do PC. Nos quadros 16, 17, 18, 19, 20, 21, 22 e 23 são descritas cada uma das questões criadas ou adaptadas. As atividades foram criadas pelo autor e outras tiveram como inspiração os trabalhos de Liukas

(2015) e Engelberg e Thinkfun (2015). As atividades propostas por Liukas possuem uma licença Creative Commons BY-NC-SA¹, o que permite sua reutilização e adaptação. O mesmo não acontece com a atividade adaptada do jogo CodeMaster, comercializada pela Thinkfun. O autor, após contato com a empresa, recebeu autorização por escrito do presidente da instituição pelo uso, adaptação, remixagem e criação de trabalhos derivados do original para fins educacionais e de pesquisa. O documento na íntegra encontra-se em Anexo E.

Quadro 16: Terceira atividade utilizada nas intervenções

Atividade 3: Tetris - Instruções (Apêndice E, p. 210)	
Questão-Exemplo	<p>ALUNO A (Controle): </p> <p>ALUNO B (Impressora): </p>
DESCRIÇÃO	
<p>Material necessário:</p> <ul style="list-style-type: none"> • Filetes de papel contendo oito imagens desenhadas sobre caixas quadriculadas • Filetes de papel contendo oito caixas quadriculadas (sem desenho) <p>Objetivo: exercitar os pilares de abstração, decomposição e algoritmos através de utilização de instruções específicas para desenhar uma série de figuras. Com esta atividade, a criança entende melhor que um algoritmo deve ser livre de erros para que o resultado seja o desejado. Caso ocorra um problema de programação ou as instruções não sejam descritas corretamente, ocorrerão erros e o objetivo não será atingido.</p> <p>Instruções:</p> <ul style="list-style-type: none"> • Convidar os alunos para formarem duplas com seus colegas; • Entregar um filete de papel para cada dupla. As crianças não podem enxergar o papel do outro; • A criança que receber o papelete com os desenhos (controle), deverá, então, cortar o filete de papel no meio e instruir o segundo (impressora) como desenhar a figura, utilizando apenas seis comandos. • Os comandos permitidos são: <ul style="list-style-type: none"> ○ Início: baixar o lápis e posicioná-lo no ponto superior esquerdo ○ Direita: movimentar o lápis para a direita ○ Esquerda: movimentar o lápis para a esquerda ○ Baixo: movimentar o lápis para baixo ○ Cima: movimentar o lápis para cima ○ Fim: levantar o lápis e finalizar o desenho • No fim, o estudante B (impressora) deve possuir as mesmas figuras que o A (controle). 	

Fonte: próprio autor

¹ Permite remixar, adaptar e criar a partir do trabalho original, para fins não comerciais, desde que atribua ao novo trabalho o devido crédito e que licencie as novas criações sob termos idênticos (CC Attribution-NonCommercial-ShareAlike).

Quadro 17: Quarta atividade utilizada nas intervenções

Atividade 4: Controle Remoto (Apêndice F, p. 211)	
Questão-Exemplo	
DESCRIÇÃO	
<p>Material necessário:</p> <ul style="list-style-type: none"> • Uma folha contendo o controle remoto e cartões de programas para serem recortados <p>Objetivo: exercitar os pilares de abstração, decomposição e algoritmos através da utilização de botões de eventos. A atividade deve ser realizada em dupla, sendo que uma das crianças fica com o controle e a outra executa as ações que constam no cartão de programação.</p> <p>Instruções:</p> <ul style="list-style-type: none"> • Entregar uma folha para cada dupla de estudantes; • Todas as peças devem ser recortadas (um controle e oito cartões de programas); • Cada criança deve ficar com quatro cartões de programas e preenchê-los com ações que sejam simples de serem executadas (e.g. pular, bater palmas, girar); • Após as cartas serem preenchidas, uma delas fica com o controle remoto e insere o cartão na fenda traseira e “instala” o programa; • O outro membro da dupla, então, por sua vez, deve seguir os comandos que constam no programa (cartão); • Após alguns instantes, a dupla troca de posição e o outro colega passa a executar os seus programas, inserindo novamente na fenda e instalando os comandos; • Caso os estudantes não tenham preenchido todos os cartões, encoraje-os a criar novas instruções. 	

Fonte: próprio autor

Quadro 18: Quinta atividade utilizada nas intervenções

Atividade 5: Os Elefantes (Apêndice G, p. 212)	
Questão-Exemplo	<p style="text-align: center;">OS ELEFANTES</p> <p style="text-align: center;">$X =$ <input style="width: 40px; height: 20px;" type="text"/></p> <div style="text-align: center;"> </div> <p>X elefante(s) se equilibrava(m) Em cima da teia de uma aranha E como via(m) que não caía(m) Foram chamar outro elefante</p> <p>SE X=3 OU X=6: PAUSA $X \leftarrow X + 1$ SE X<10, REPETIR ESTROFE</p>

DESCRIÇÃO
<p>Material necessário:</p> <ul style="list-style-type: none"> • Um filete de papel com a letra da música “Os Elefantes”; • Um filete de papel com o algoritmo da letra da música “Os Elefantes”. <p>Objetivo: exercitar os pilares de abstração, reconhecimento de padrões e algoritmos através da conversão de uma música em um pseudocódigo. Na referida música são trabalhados os conceitos de repetição, variáveis e condicionais.</p> <p>Instruções:</p> <ul style="list-style-type: none"> • Entregar uma folha para cada estudante com a letra, caso não conheçam. Caso seja necessário, usar o vídeo disponível em https://youtu.be/w80dCI1a1y4 • Cantar a música completa e pedir aos alunos identificar os elementos que se repetem na canção. • Propor a escrita de um algoritmo no caderno ou quadro, utilizando os conceitos de repetição e introduzindo as variáveis e condicionais. • Caso não se encontre um consenso na escrita do algoritmo, entregar o segundo filete de papel com uma alternativa de “letra algorítmica” e cantar novamente, refazendo os passos do algoritmo.

Fonte: próprio autor

Quadro 19: Sexta atividade utilizada nas intervenções

Atividade 6: Bugs (vide Apêndice H, p. 213)	
Questão-Exemplo	<pre> graph LR inicio([início]) --> A[colocar os pratos sobre a mesa] A --> B[colocar os talheres] B --> C[trazer o bolo] C --> D[colocar a toalha] D --> fim([fim]) </pre>
DESCRIÇÃO	
<p>Material necessário:</p> <ul style="list-style-type: none"> • Uma folha de papel com diversas situações cotidianas no formato de diagramas ou lista de instruções necessárias para concluir uma atividade. <p>Objetivo: exercitar os pilares de abstração, decomposição e algoritmos através do reconhecimento de equívocos na composição dos diagramas e relação de ações.</p> <p>Instruções:</p> <ul style="list-style-type: none"> • Entregar uma folha para cada estudante • Discutir cada um dos exemplos, tentando encontrar o problema em cada situação; • Tentar executar cada um dos exemplos até que todos os algoritmos estejam corrigidos. 	

Fonte: próprio autor

Quadro 20: Sétima atividade utilizada nas intervenções

Atividade 7: Boneca de papel (vide Apêndice I, p. 214)	
Questão-Exemplo	
DESCRIÇÃO	
<p>Material necessário:</p> <ul style="list-style-type: none"> • Uma folha-resposta (blocos de condicionais); • Uma folha contendo peças de roupas. <p>Objetivo: exercitar os pilares de decomposição, reconhecimento de padrão e algoritmos através da definição de roupas que devem ser utilizadas em diferentes situações.</p> <p>Instruções:</p> <ul style="list-style-type: none"> • Entregar uma folha-resposta e outra contendo as peças de roupas. • Cada aluno deve recortar as roupas e colar nos espaços indicados, conforme a situação. • Não existe uma única resposta correta, tendo em vista que se pode usar diferentes roupas para cada ocasião, porém existem certas roupas que não devem ser utilizadas em algumas situações (e.g. usar tênis para entrar na piscina, usar capa de chuva em dia de calor, etc.) 	

Fonte: próprio autor

Quadro 21: Oitava atividade utilizada nas intervenções

Atividade 8: Cupcakes (vide Apêndice J, p. 215)	
Questão-Exemplo	<p>forma + massa + cobertura = cupcake</p> <p>Se está pronto, utilize simplesmente um traço (-).</p> <p>A</p> <p>_____</p> <p>_____</p> <p>Cobertura</p> <p>Cobertura</p> <p>Massa + Cobertura</p> <p>Massa + Cobertura</p> <p>D</p> <p>P1 _____</p> <p>P1 _____</p> <p>P1 _____</p> <p>P1 _____</p> <p>P1 _____</p> <p>P1 _____</p> <p>P1: _____</p> <p>Massa + Cobertura</p> <p>E</p> <p>2X P2 _____</p> <p>_____</p> <p>_____</p> <p>P2: _____</p> <p>Massa + Cobertura</p> <p>Cobertura</p> <p>_____</p>

DESCRIÇÃO
<p>Material necessário:</p> <ul style="list-style-type: none"> • Uma folha com as instruções e exercícios. <p>Objetivo: exercitar os pilares de abstração, reconhecimento de padrão e algoritmos através da criação de uma série de comandos que auxiliam na fabricação de bolinhos. São exercitados também a criação de funções e reaproveitamento de código.</p> <p>Instruções:</p> <ul style="list-style-type: none"> • Entregar uma folha para cada estudante; • Explicar que o processo de confeitar um cupcake é composto por uma forma, massa e cobertura; • Pedir aos alunos escreverem abaixo os passos que estão faltando para completar o processo; • Perguntar se seria possível simplificar o processo, criando um conjunto de comandos; • As funções são exemplificadas no lado direito e são chamadas de P1 e P2. Essas funções são globais e podem ser reutilizadas em todos os demais exercícios da folha.

Fonte: próprio autor

Quadro 22: Nona atividade utilizada nas intervenções

Atividade 9: Tetris – Repetição (vide Apêndice K, p. 216)	
QUESTÃO-EXEMPLO	
DESCRIÇÃO	
<p>Material necessário:</p> <ul style="list-style-type: none"> • Um papel quadriculado com os desenhos propostos para cada estudante. <p>Objetivo: exercitar os pilares de abstração, decomposição, reconhecimento de padrão e algoritmos através da compreensão do uso de repetições para desenhar figuras.</p> <p>Instruções:</p> <ul style="list-style-type: none"> • Entregar uma folha para cada estudante ou utilizar o caderno; • Explicar aos estudantes a diferença entre utilizar as setas de uma perspectiva global (atividades “Mapa da Turma da Mônica” e “Tetris - Instruções”) e de uma perspectiva do objeto ou da pessoa (atividade “Controle Remoto”); • Diferentemente da atividade “Tetris - Instruções”, o estudante precisa usar as instruções baseadas na perspectiva da direção e posição da seta que consta na figura sendo trabalhada, ou seja, só pode utilizar os comandos: “para frente”, “gire à direita” e “gire para a esquerda”. <p>Quando o estudante identificar um padrão de setas, tentar utilizar o maior número de multiplicadores possíveis, conforme exemplo destacado.</p>	

Fonte: próprio autor

Quadro 23: Décima atividade utilizada nas intervenções

Atividade 10: Autômatos da Mônica (Apêndice L, p. 217)

Questão-Exemplo

DESCRIÇÃO

Material necessário:

- Duas folhas compostas por quatro mapas e dezesseis desafios (atividades)
- Um filete de papel com quadrados coloridos, um círculo com o rosto dos personagens e uma porta.

Objetivo: exercitar os pilares de abstração, decomposição, reconhecimento de padrão e algoritmos através da resolução de autômatos finitos determinísticos representados por um diagrama similar ao de transição de estados.

Instruções:

- Entregar uma folha para cada estudante;
- Pedir para que recortem os quadrados e usem para colar no local indicado conforme a solução (rota) encontrada;
- Para iniciar um desafio, deve-se posicionar a peça do Cebolinha no nodo (número) do mapa indicado no lado direito. Em seguida, utilizando todas as cores que constam do lado esquerdo da rota, encontrar o caminho que utilize todas as cores indicadas.
- Nenhum quadrado deve ficar vazio.

Fonte: próprio autor

Após relacionar e descrever as atividades que foram criadas ou adaptadas, criou-se um quadro com os pilares do PC trabalhados em cada uma das atividades. Tentou-se identificar os pilares com maior ênfase, embora de forma indireta ou em menor grau outro pilar possa ter sido exercitado.

Quadro 24: Relação das questões com os Quatro Pilares do PC

Atividade	Nome da Atividade	ABSTRAÇÃO	DECOMPOSIÇÃO	REC. DE PADRÕES	ALGORITMO
01	Decomposição	X	X	X	X
02	Mapa da Turma da Mônica	X		X	X
03	Tetris - Instruções	X	X		X
04	Controle Remoto	X	X		X
05	Os Elefantes	X		X	X
06	Bugs	X	X		X
07	Boneca de Papel		X	X	X

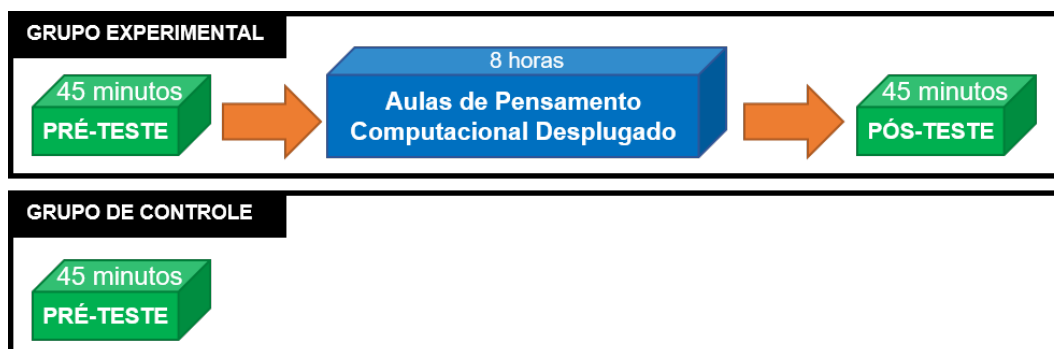
Atividade	Nome da Atividade	ABSTRAÇÃO	DECOMPOSIÇÃO	REC. DE PADRÕES	ALGORITMO
08	Cupcakes	X		X	X
09	Tetris – Repetição	X	X	X	X
10	Autômatos da Mônica	X	X	X	X

Fonte: próprio autor

4.2.3 Execução do segundo Piloto

Na aplicação do segundo piloto, foi adotado um pré-teste em uma outra turma (grupo de controle) com a intenção de verificar a equivalência de PC entre as duas turmas, ou seja, o grupo de controle tinha um objetivo exploratório e ainda não estatístico. Na Figura 52 é demonstrado um diagrama com as etapas que contemplaram o segundo projeto piloto, que teve um total de dois dias de duração. No primeiro dia ocorreu o pré-teste, juntamente com quatro horas de aula de PC. No dia seguinte, os estudantes tiveram mais quatro horas de aula sucedido pelo pós-teste. Salienta-se que o pré- e pós-testes são idênticos.

Figura 52: Etapas do segundo projeto piloto



Fonte: próprio autor

A quantidade de crianças (sujeitos) que participaram dessa etapa da pesquisa foi de 34, sendo 19 do grupo experimental e 15 de controle. A distribuição de sujeitos encontra-se na Tabela 6.

Tabela 6: Distribuição dos sujeitos participantes da pesquisa

Grupo	Meninos	Meninas	Total
Experimental	14	5	19
Controle	9	6	15
Total:	23	11	34

Fonte: próprio autor

4.2.4 Resultado do segundo piloto

Após as atividades serem aplicadas aos estudantes, notou-se grande entusiasmo por parte deles e da professora. As crianças se demonstraram muito interessadas pela temática das intervenções e a professora, por sua vez, pela oportunidade de seus alunos conhecerem uma nova maneira de pensar. Infelizmente, não foi possível a aplicação de todas as atividades dentro do período proposto inicialmente (oito horas) devido a complicações durante as aulas (*e.g.* mau comportamento, recreio, intervalo de lanche, avisos da escola, atrasos e antecipação de saída de estudantes).

Em relação ao resultado do Teste de Pensamento Computacional, é necessário explicar que a pontuação é calculada de acordo com a quantidade de questões respondidas corretamente, lembrando que o teste é composto de 28 questões, ou seja, a nota máxima. O teste foi aplicado tanto no Grupo Experimental como no de Grupo Controle, porém em momentos distintos. Devido à falta de estudantes no dia do pré-teste, pós-teste ou ausência durante as intervenções, somente 10 alunos foram considerados nos cálculos estatísticos no grupo experimental. Já no grupo de controle, foi possível a utilização de 14 sujeitos por se tratar de uma aplicação única.

A média aritmética simples das amostras (Experimental e Controle) já revelam indícios da existência de uma diferença positiva entre as pontuações dos alunos na avaliação tanto anterior como posterior às intervenções. Apresenta-se na Tabela 7 os resultados dos testes, onde se percebe um aumento da média total da pontuação entre o pré e pós-teste, ou seja, passando de 10,40 para 12,90 pontos. Além disso, nota-se também que a turma de controle possui uma pontuação muito similar do grupo experimental, atingindo um dos objetivos exploratórios desse projeto piloto.

Para verificar se essa melhoria de desempenho no grupo experimental é significativa, faz-se necessário a aplicação do Teste T de Student que utiliza conceitos estatísticos para a identificação de hipóteses válidas, além de possuir um alto grau de precisão para amostras que estejam normalizadas (NORMANDO *et al.*, 2010). Porém, antes, é necessário realizar um estudo da curva de distribuição dos dados por meio do teste de Shapiro-Wilk (SHAPIRO e WILK, 1965), onde verificou-se que os três grupos apresentaram uma distribuição normal (vide Tabela 8), pois os valores de p (p-valores) foram todos superiores a 0,05 (utilizando um grau de confiança de 95%) e a distribuição dos dados ocorreu de maneira uniforme. Isso permite o uso do Teste T para verificar se houve ou não uma melhoria estatisticamente significativa do desempenho dos estudantes.

Tabela 7: Resultado do segundo projeto piloto

Grupo	Teste PC	Média	N (Sujeitos)	Desvio Padrão
Intervenção	Pré-teste	10,40	10	±4,17
	Pós-teste	12,90	10	±3,38
Controle	Único	9,79	14	±2,72

Fonte: próprio autor

Tabela 8: Teste de Normalidade do Piloto 2

	Experimental (pré-teste)	Experimental (pós-teste)	Controle
N	10	10	14
Shapiro-Wilk W	0,9369	0,9646	0,9138
p(normal)	0,5186	0,837	0,1791

Fonte: próprio autor

Com o resultado alcançado após a verificação de uma distribuição normal, executou-se o Teste T de Student com os dados dos dois testes de Pensamento Computacional (pré-teste e pós-teste) do grupo experimental. Como pode ser visto na Tabela 9, comprovou-se a rejeição da hipótese nula, isto é, o grau de significância demonstrado pelo método apresentou um valor de 0,008 nas duas extremidades. Em outras palavras, a mudança de pontuação entre o pré e pós-teste foram significativos. Como não havia um pré-teste no grupo de controle, não foi possível identificar se essa mudança de pontuação ocorreu devido às intervenções. É necessário também ter cautela com os resultados, pois a amostra de estudantes foi muito pequena.

Tabela 9: Teste T de Student - Duas amostras em par para médias (Grupo Experimental)

	Pré-Teste	Pós-Teste	p-valor
Média	10,40	12,90	0,008
Variância	17,37	11,43	

Fonte: próprio autor

Ao final dessa etapa, foi possível apontar conclusões pertinentes para a continuidade da pesquisa, tendo em vista que os dados comprovam a eficácia da abordagem desplugada no ensino do Pensamento Computacional. Algumas alterações necessárias foram identificadas, tais como: acréscimo de 25% na duração das aulas para contemplar todas as atividades, pequenos ajustes nas mesmas para tornar mais clara sua compreensão e a aplicação de um pós-teste também no grupo de controle para que fosse possível atender a todos os quesitos estatísticos e regular o experimento através de uma comparação paralela ou pela comparação com outro padrão (SHADISH *et al.*, 2001).

Com um objetivo exploratório, o final da pesquisa teve seu tempo dilatado por mais quatro horas de atividades, utilizando Scratch com exercícios que envolviam conceitos já trabalhados de maneira desplugada (em sala de aula). Essa nova etapa permitiu ao pesquisador identificar qualitativamente possíveis alterações no comportamento das crianças após as aulas de PC desplugado.

Esses aperfeiçoamentos foram então incorporados ao projeto e utilizados na etapa seguinte, que ocorreu na Espanha, onde o autor do Teste de Pensamento Computacional, Prof. Dr. Marcos Román-González reside e realiza suas pesquisas na área, juntamente com seus grupos de pesquisa.

4.3 APLICAÇÃO NA ESPANHA

Esta etapa de pesquisa do autor ocorreu durante o período conhecido popularmente como “sanduíche” na cidade de Madrid, Espanha, e será descrito na sequência.

4.3.1 Instrumento Avaliativo

O instrumento avaliativo utilizado naquele país foi o mesmo do segundo piloto do Brasil, porém em Espanhol (europeu), o idioma nativo do teste. Por uma conveniência de ter os dados tabulados de maneira mais acessível e por recomendação do próprio criador do teste de PC, os testes foram realizados nos computadores da escola e não mais em papel. O teste foi desenvolvido utilizando a ferramenta *Google Forms* e encontra-se disponível online¹. As respostas de todos os estudantes foram registradas na *Google Cloud* para que, posteriormente, pudessem ser visualizadas, recuperadas, convertidas, tabuladas e analisadas estatisticamente.

4.3.2 Desenvolvimento de atividades

As atividades utilizadas na Espanha não sofreram uma alteração significativa em relação ao seu conteúdo, porém foi necessária a tradução e adaptação de todo o material a ser usado com os estudantes, como por exemplo: atividades específicas de tradução de textos (vide Figura 53), atividades em que se fez necessário a identificação dos personagens desconhecidos aos espanhóis e que constavam no exercício (vide Figura 54) e a busca da versão da música que

¹ Teste de Pensamiento Computacional: <http://tinyurl.com/escuelas>

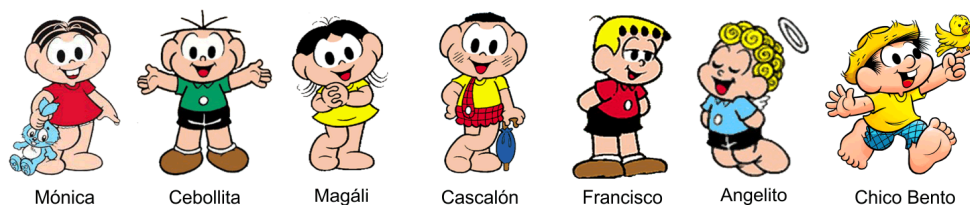
se encontrava no idioma português¹ para o espanhol² (vide Figura 55). Todas as atividades de versão e alterações foram revisadas, posteriormente, por um espanhol nativo e os devidos ajustes foram efetuados.

Figura 53: Exemplo de atividade traduzida



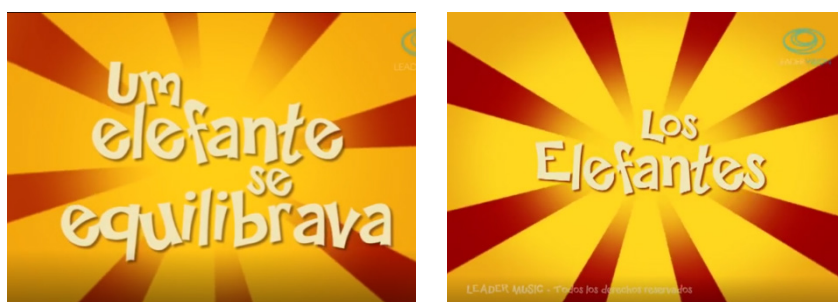
Fonte: adaptado pelo próprio autor de Maurício de Souza Produções®

Figura 54: Personagens da Turma da Mônica em Espanhol



Fonte: adaptado pelo próprio autor de Maurício de Souza Produções®

Figura 55: Música “Um Elefante se Equilibrava” em Espanhol (europeu)



Fonte: próprio autor / O Reino das Crianças e Leader Music

4.3.3 Execução da pesquisa

Após chegada no país em junho de 2016, realizou-se um levantamento de possíveis regiões na cidade de Madri, para a execução do projeto. Na tentativa de manter uma realidade mais próxima do Brasil, buscou-se um bairro com uma maior carência/vulnerabilidade social

¹ Os Elefantes: <https://youtu.be/w80dCIIa1y4>

² Los Elefantes: <https://youtu.be/J-HIg8hIXak>

na cidade. Além disso, outros fatores foram levados em consideração, entre eles a quantidade suficiente de escolas e o fácil acesso com transporte público. Após reunião com a supervisora, Profa. Dra. Susana Muñoz Hernández, da Universidade Politécnica de Madrid, responsável pela orientação da pesquisa em execução naquele país (durante o período ‘sanduíche’), definiu-se a região de Carabanchel para dar continuidade ao projeto.

O passo seguinte teve como objetivo mapear as escolas dentro da região através do PortalEscolar¹ do distrito de Madrid, onde foram localizadas 39 escolas que atendem turmas de quinto e sexto anos. Em seguida foram enviadas mensagens de e-mails para todas as escolas com uma carta de apresentação, onde era solicitada a possibilidade de ocorrer intervenções nas turmas, além de descrever a motivação do projeto, etapas, perguntas frequentes e como contatar os pesquisadores (vide Apêndice M). Após uma espera de duas semanas e nenhum retorno das escolas, enviou-se novamente a mensagem reforçando o pedido. Passadas mais duas semanas, não houve ainda manifestação alguma por parte das escolas.

Para agilizar o processo de recrutamento e sensibilização das escolas, optou-se por uma estratégia mais individual, onde foram preparadas 39 cartas com o mesmo conteúdo das mensagens de e-mail para cada uma das escolas. Também, na tentativa de otimizar a entrega das cartas, elaborou-se um mapa, com auxílio do *Google Maps*, onde se identificou as escolas e um roteiro de entrega das correspondências. Em uma semana, as cartas haviam sido entregues na maioria das escolas pessoalmente pelo pesquisador. Na Figura 57 é apresentado o mapa com as escolas do bairro de Carabanchel, onde os ponteiros azuis representam as escolas não visitadas e os círculos verdes (“ticks”) que referem-se às escolas onde a carta foi entregue. O registro fotográfico das visitas às escolas também foi realizado e se encontra na Figura 56.

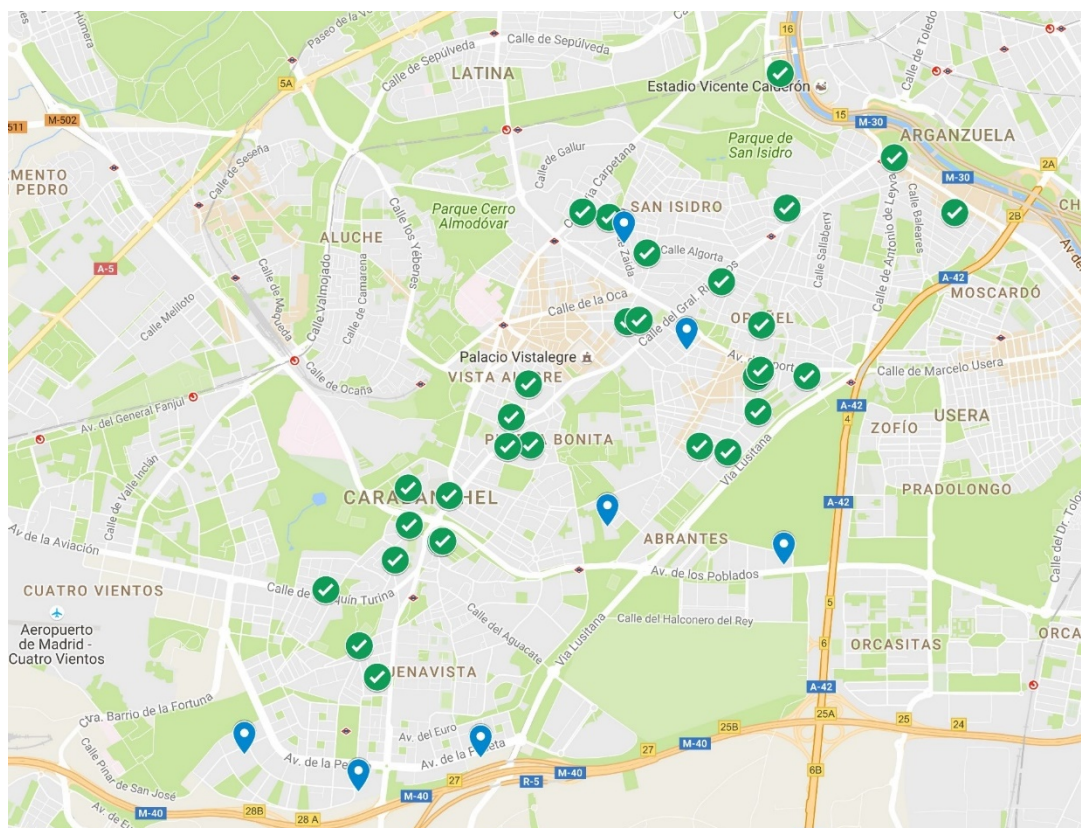
¹ http://www.madrid.org/wpad_pub/run/j/Inicio.icm

Figura 56: Registro da entrega das cartas nas escolas



Fonte: próprio autor

Figura 57: Escolas da região de Carabanchel

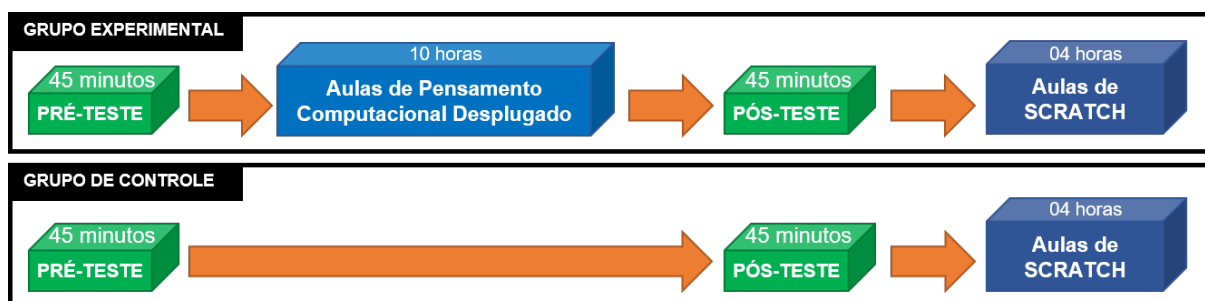


Fonte: próprio autor

Na semana seguinte, o pesquisador recebeu manifestações de três escolas, sendo que duas delas gostariam de participar do projeto, mas a quantidade de horas necessárias para a execução do projeto tornava-se inviável. O CEIP (*Colegio de Educación Infantil y Primaria*) *República de Ecuador* foi o único colégio que aderiu integralmente ao projeto. O *CEIP Lope de Vega* foi a segunda escola que participou do projeto, porém o contato foi gentilmente facilitado pelo Prof. Dr. Marcos Román-González, onde já haviam ocorrido outros projetos.

Tendo como base os dois pilotos realizados no Brasil e com as sofisticações do projeto, definiram-se as etapas para execução do mesmo, mostradas em um diagrama na Figura 58. Note que foram adicionados o pós-teste no grupo de controle e quatro horas de Scratch em ambas as turmas. As intervenções nas escolas ocorriam semanalmente, sendo que cada intervenção tinha duração de duas horas de aula, sendo assim, todo o processo levava cerca de oito semanas. Encontra-se no Apêndice P um planejamento detalhado das atividades trabalhadas com os estudantes durante cada uma das semanas. A quantidade de semanas poderia variar devido a algum feriado ou outras atividades do colégio. As etapas serão descritas com mais detalhes na sequência.

Figura 58: Diagrama das etapas do projeto na Espanha



Fonte: próprio autor

A aplicação do Teste de Pensamento Computacional ocorria na primeira semana das intervenções e após a última (testes idênticos), geralmente na sexta semana, em ambos os grupos (controle e experimental). A aplicação do teste acontecia no laboratório de informática das próprias escolas. Duas fotos demonstram a aplicação do pré-teste, quais sejam, no Colégio República de Ecuador, Figura 59 e no Colégio Lope de Vega, na Figura 60.

Figura 59: Laboratório do Colégio República de Ecuador (aplicação do teste)



Fonte: próprio autor

Figura 60: Laboratório do Colégio Lope de Vega (aplicação do teste)



Fonte: próprio autor

Após o Teste de Pensamento Computacional, iniciava-se a etapa das intervenções em sala de aula, sem o uso de computadores ou qualquer equipamento eletrônico ou projetor. As atividades utilizadas nesta etapa já foram citadas nos capítulos 4.1.2 e 4.2.2 . Para fins de ilustração, encontram-se nas Figuras 61 e 62, fotos das intervenções nos Colégios República de Ecuador e Lope de Vega, respectivamente.

Figura 61: Sala da aula do Colégio República de Ecuador (intervenções)



Fonte: próprio autor

Figura 62: Sala da aula do Colégio Lope de Vega (intervenções)



Fonte: próprio autor

Após a conclusão das intervenções, os alunos dos dois grupos eram novamente direcionados ao laboratório de informática da escola para a realização do pós-teste de Pensamento Computacional. Na semana seguinte, ocorriam as intervenções com a linguagem Scratch nos laboratórios de informática da escola, onde foram trabalhados conceitos já exercitados de maneira desplugada. Uma avaliação qualitativa será detalhada no capítulo 5.

A seguir serão detalhados os estudantes participantes da pesquisa, bem como a divisão dos grupos de controle e experimental.

4.3.4 Participantes e Grupos

Cada uma das escolas que participaram da pesquisa possuía duas turmas do mesmo ano (pré-requisito para aplicação das tarefas) e as turmas experimentais e de controle foram escolhidas aleatoriamente. Um melhor detalhamento da distribuição dos sujeitos que participaram efetivamente da pesquisa encontra-se na Tabela 10, onde se totaliza uma amostra total de 73 crianças, sendo 35 meninos e 38 meninas. Com a intenção de haver imparcialidade, optou-se por identificar as escolas pelos nomes “Escola A” e “Escola B”.

Tabela 10: Distribuição dos sujeitos participantes da pesquisa na Espanha

	Ano	Idade	Grupo	Sexo		Total
				Masculino	Feminino	
Escola A	5º	10-11 anos	Controle	10	13	23
			Experimental	11	8	19
Escola B	6º	11-12 anos	Controle	6	8	14
			Experimental	9	7	16
			Total	36	36	72

Fonte: próprio autor

Os resultados desta etapa da pesquisa estão também descritos em Brackmann *et al.* (2017) e serão melhor discutidos no capítulo 5.

4.4 APLICAÇÃO NO BRASIL

Nesta seção será apresentada a etapa da pesquisa no Brasil.

4.4.1 Instrumento Avaliativo

O instrumento avaliativo utilizado foi o mesmo da Espanha, porém na versão em Português (Brasil). Como o teste já tinha sido integralmente traduzido, o pesquisador converteu a versão existente (em papel) para uma versão que pudesse ser executada no navegador. Deu-se continuidade nesta plataforma devido a conveniência de ter os dados tabulados de maneira mais acessível e por recomendação do próprio criador do teste de PC. A nova versão do teste foi também desenvolvida utilizando a ferramenta *Google Forms* e encontra-se disponível online¹. Os registros de respostas dos alunos também foram guardados na *Google Cloud* para que, posteriormente, pudessem ser visualizadas, recuperadas, convertidas, tabuladas e analisadas estatisticamente.

4.4.2 Desenvolvimento de atividades

Para que os resultados da pesquisa trouxessem dados sólidos para serem comparados posteriormente, optou-se por não alterar nenhuma das atividades já citadas nos capítulos 4.1.2 e 4.2.2. Porém, foram adicionados dois novos exercícios de caráter artístico para que os estudantes pudessem se distrair durante a aplicação do Teste de Pensamento Computacional, tendo em vista que os colégios não tinham uma quantidade suficiente de máquinas e as crianças

¹ Teste de Pensamento Computacional: <http://tinyurl.com/escuelabr/>

ficavam ociosas enquanto esperavam a sua vez de utilizar o computador. As atividades foram adaptadas de Liukas (2015) e Bell *et al.* (2015) e estão disponíveis em Apêndice N.

4.4.3 Execução da Pesquisa

O pesquisador buscou no Brasil instituições de ensino da rede pública para aplicar a investigação, tendo em vista que 84% da população realiza matrículas no Ensino Fundamental em escolas municipais, estaduais e federais. A quantidade de matrículas encontra-se na Tabela 11. Levando-se em consideração a vulnerabilidade social dos estudantes, foram selecionadas duas escolas, sendo elas: Escola Estadual de Ensino Fundamental Marieta de Ambrósio e a Escola Básica Estadual Dr. Paulo Devanier Lauda, ambas na cidade de Santa Maria, RS.

Tabela 11: Total de matrículas no Ensino Fundamental na rede pública e privada

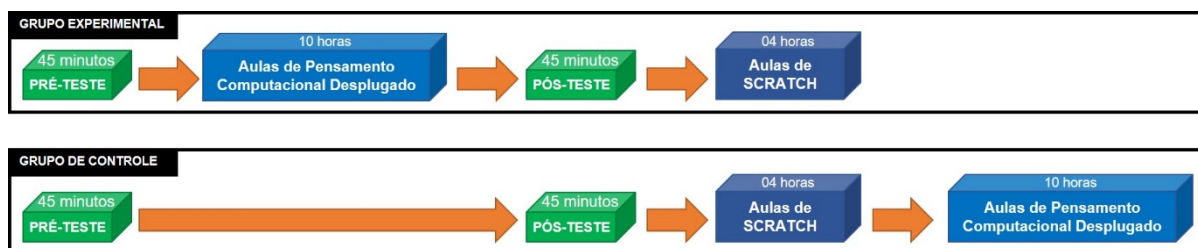
Rede	Matrículas	%
Pública	23.360.973	84%
Privada	4.570.237	16%
Total	27.931.210	100%

Fonte: MEC/INEP (2017, p. 9)

O pesquisador tomou o cuidado de manter o mesmo rito praticado na Espanha, ou seja, no primeiro momento ocorriam os pré-testes em ambos os grupos, seguidos de intervenções semanais com duas horas de aula cada e, a seguir, um pós-teste. Com o objetivo exploratório, foram ofertadas aulas da linguagem de programação visual Scratch para os grupos de controle e experimental.

Devido a regulamentação do comitê de ética, foram também ofertadas às escolas, aulas de Pensamento Computacional para os alunos do grupo de controle após o término de todas as etapas que fizessem parte da investigação principal, ou seja, haveria a necessidade de oportunizar a mesma experiência para ambos os grupos. Para atender esse quesito, adicionou-se uma etapa com aulas de Pensamento Computacional Desplugado como opcional (vide Figura 63) e que foi prontamente aceito por ambas as escolas.

Figura 63: Diagrama das etapas do projeto no Brasil



Fonte: próprio autor

Apresenta-se nas Figuras 64 e 65 a aplicação do Teste de Pensamento Computacional nas escolas Marieta D'Ambrósio e Paulo Devanier Lauda, respectivamente. Nas Figuras 66 e 67 também são demonstradas as configurações das salas de aula onde ocorreram as intervenções.

Figura 64: Laboratório do Colégio Marieta D'Ambrósio (aplicação do teste)



Fonte: próprio autor

Figura 65: Laboratório do Colégio Paulo Lauda (aplicação do teste)



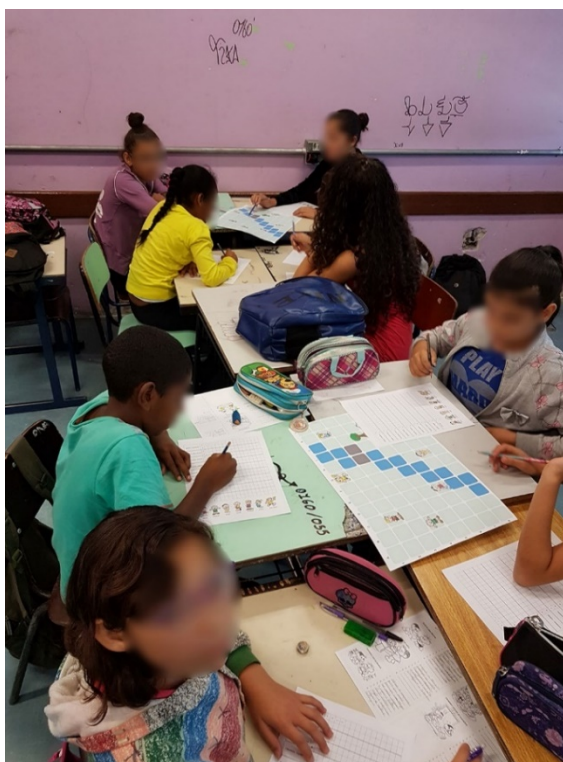
Fonte: próprio autor

Figura 66: Sala da aula do Colégio Marieta D'Ambrósio (intervencões)



Fonte: próprio autor

Figura 67: Sala da aula do Colégio Paulo Lauda (intervenções)



Fonte: próprio autor

4.4.4 Participantes e Grupos de Teste

Da mesma forma que ocorreu na Espanha, cada uma das escolas que participou da pesquisa também possuía duas turmas do mesmo ano e as turmas experimentais e de controle foram escolhidas aleatoriamente. Um melhor detalhamento da distribuição dos sujeitos que participaram efetivamente da pesquisa encontra-se na Tabela 12, onde se totaliza uma amostra total de 63 crianças, sendo 35 meninos e 28 meninas. Da mesma forma, com a intenção de haver imparcialidade, optou-se por identificar as escolas pelos nomes “Escola A” e “Escola B”.

Tabela 12: Distribuição dos sujeitos participantes da pesquisa no Brasil

	Ano	Idade Média	Grupo	Sexo		Total
				Masculino	Feminino	
Escola A	5º	10,32	Controle	7	3	10
			Experimental	7	8	15
Escola B	6º	12,21	Controle	13	6	19
			Experimental	8	11	19
			Total	35	28	63

Fonte: próprio autor

Os resultados da pesquisa obtidos no Brasil serão apresentados e discutidos no Capítulo 5.

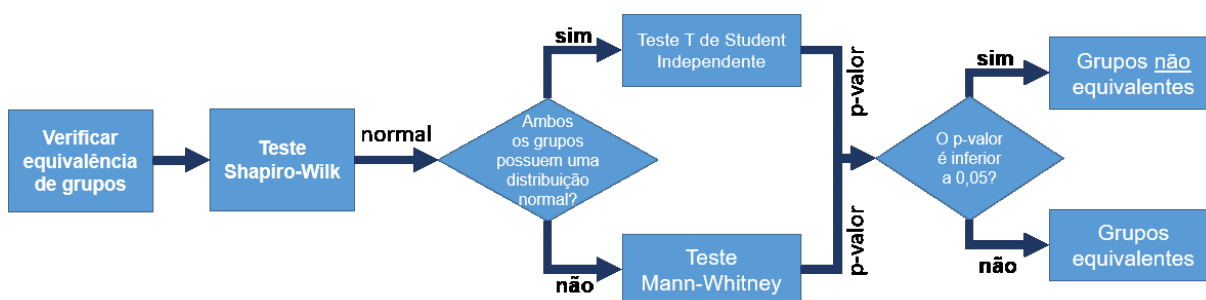
5 DISCUSSÃO DOS RESULTADOS

Esta investigação foi composta por quatro etapas, sendo que duas delas tinham o objetivo exploratório e não foram utilizadas para a análise estatística desta investigação (pilotos). Já os experimentos que ocorreram posteriormente na Espanha e Brasil, serão utilizados e analisados neste capítulo.

Partindo de uma perspectiva geral, os dados estatísticos comprovam que as atividades trabalhadas com os alunos tiveram um efeito positivo em relação ao Pensamento Computacional. Para se chegar a essa conclusão, fez-se um cruzamento de todos os participantes dos grupos de controle e experimental na Espanha e no Brasil. A pesquisa completa teve participação de 135 crianças, sendo 66 no grupo de controle e 69 no grupo experimental. A análise de dados entre os grupos de controle e experimental ocorreram em duas fases de maneira uniformizada, sendo elas: a equivalência dos grupos e a mudança de desempenho significativo. Essas duas etapas de análise também serão utilizadas nos demais agrupamentos desta pesquisa e serão melhor descritas a seguir.

O teste da equivalência dos grupos (Controle e Experimental) se faz necessário para se certificar que os grupos em questão possuem um mesmo perfil para que possam ser comparados estatisticamente. Caso os grupos tiverem perfis distintos, o teste da mudança de desempenho pode apresentar conclusões equivocadas, pois os grupos não seguem uma uniformidade. A verificação da equivalência se dá através do uso de dois tipos de testes, dependendo da distribuição normal dos dados dos grupos que pode ser verificada com o uso do teste Shapiro-Wilk (CAÇÃO, 2010). Um fluxograma para ilustrar e facilitar a compreensão do processo de verificação de uniformidade dos grupos é apresentado na Figura 68. Para realizar os cálculos desta pesquisa, foram utilizados os softwares ActionStat (uma interface gráfica da linguagem R) e o PAST (HAMMER, 2001).

Figura 68: Fluxo de testes (equivalência dos grupo)



Fonte: próprio autor

Neste caso, para verificar se os grupos de controle e experimental são equivalentes, fez-se um teste de Shapiro-Wilk dos pré-testes para comprovar a normal dos grupos e em seguida o Teste T de Student Independente para calcular o p-valor (probabilidade de significância) entre as duas amostras. É importante lembrar que o Teste de Pensamento Computacional (TPC) utilizado na pesquisa é composto de 28 questões, dessa forma, a pontuação máxima a ser atingida pelos estudantes é de 28 pontos.

Após análise dos dados que se encontram na Tabela 14, destaca-se um p-valor de 0,354 que é um número superior ao nível nominal ($\alpha=0,05$). Sendo assim, não se evidencia a rejeição da hipótese de igualdade dos grupos de pré-testes, comprovando a equivalência dos grupos.

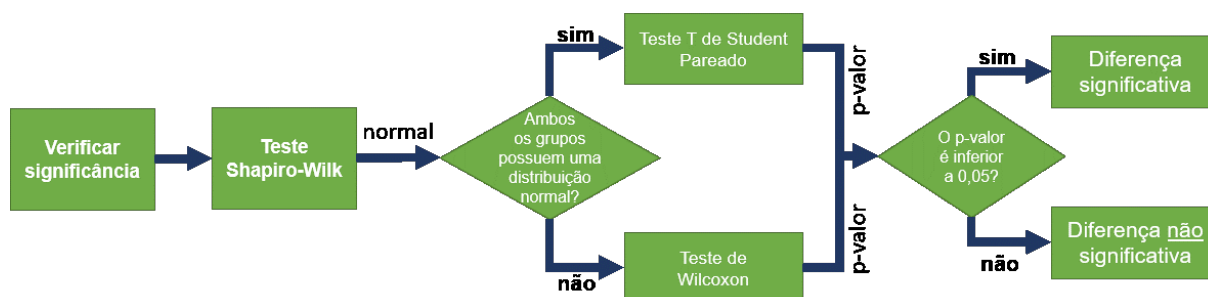
Tabela 13: Comparação de Perfil (todos sujeitos)

PRÉ-TESTE			
	N	Média (Desvio Padrão)	p-valor
Desempenho Controle	69	9,95 ($\pm 3,52$)	0,354
Desempenho Experimental	66	10,49 ($\pm 3,197$)	

Fonte: próprio autor

A etapa seguinte realiza a verificação da relevância da mudança de desempenho e tem como objetivo comprovar se ocorreu uma mudança na média da pontuação estatisticamente significativa das amostras estudadas. Para os grupos que apresentam uma distribuição normal dos dados, utilizou-se o Teste T de Student Pareado devido a sua maior sensibilidade ao testar as diferenças para a comparação de médias. Nos demais casos, utiliza-se o teste não-paramétrico Wilcoxon que realiza uma comparação de rankings (posição dos números) e confronta as medianas e o intervalo interquartilico para verificar se a alteração das médias foi significativa. Um fluxograma, com a finalidade de facilitar a compreensão do leitor, é apresentado na Figura 69.

Figura 69: Fluxo de testes (mudança de desempenho significativa)



Fonte: próprio autor

Levando-se em consideração toda a amostra de sujeitos da pesquisa (vide Tabela 14), evidencia-se que o grupo de controle, composto por 66 estudantes, atingiu uma média amostral de 9,95 com desvio padrão de 3,52 (para mais ou para menos) e no pós-teste apresenta média amostral de 10,39 com desvio padrão de 3,49. No entanto, o grupo experimental, composto por 69 estudantes, evidenciou média amostral de 10,49 com desvio padrão de 3,20 (para mais ou para menos) e no pós-teste apresenta média amostral de 12,61 com desvio padrão de 3,87. As demonstrações gráficas que apresentam a melhoria significativa dos grupos experimentais encontram-se em Gráfico 1 e Gráfico 2.

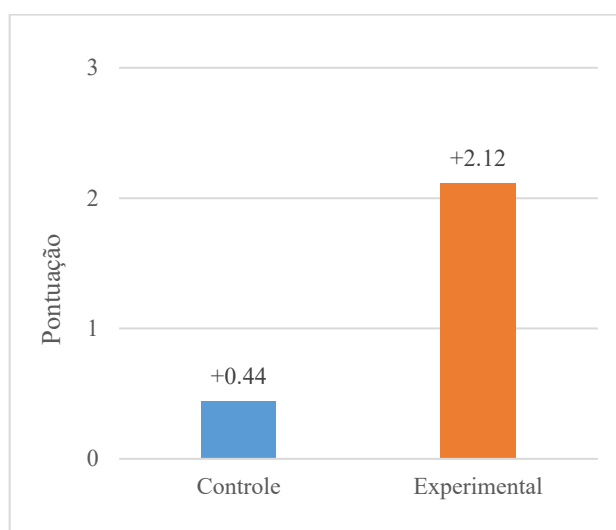
Tabela 14: Tabela com resultados geral

CONTROLE					
	N	Pré-Teste Média (Desvio Padrão)	Pós-Teste Média (Desvio Padrão)	p-valor	Alteração de Desempenho (%)
Desempenho	66	9,95 ($\pm 3,52$)	10,39 ($\pm 3,49$)	0,318	+0,44 (4,41%)
EXPERIMENTAL					
	N	Pré-Teste Média (Desvio Padrão)	Pós-Teste Média (Desvio Padrão)	p-valor	Alteração de Desempenho (%)
Desempenho	69	10,49 ($\pm 3,20$)	12,61 ($\pm 3,87$)	< 0,001	+2,12 (20,17%)

Fonte: próprio autor

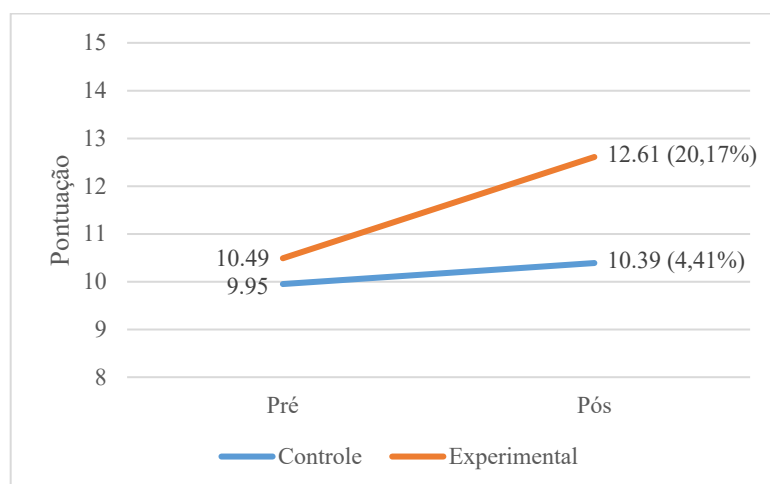
Levando-se em consideração um p-valor (0,318) superior ao nível nominal ($\alpha = 0,05$) no grupo de controle, não se evidenciou a rejeição da hipótese de igualdade do grupo para o Teste T pareado, comprovando que não houve melhoria de desempenho significativo. O contrário ocorreu no grupo experimental, onde foi verificado um p-valor inferior ao nominal (< 0,001), indicando diferença significativa entre o pré e pós-testes, evidenciando a rejeição da hipótese nula. Sendo assim, a partir das médias e dos p-valores calculados, comprova-se que houve uma melhoria de desempenho significativa apenas no grupo experimental, comprovando a eficácia das aulas de Pensamento Computacional Desplugado.

Gráfico 1: Melhoria de desempenho da amostra total (pontuação)



Fonte: próprio autor

Gráfico 2: Gráfico comparativo dos resultados (total)



Fonte: próprio autor

Nas seções seguintes serão detalhados os resultados da pesquisa na Espanha e no Brasil de maneira individual. Na sequência também se tentará realizar uma comparação entre os dois países. Com a intenção de facilitar a compreensão dos dados, será adotada a seguinte nomenclatura para identificar as amostras: os três primeiros caracteres identificam o país (ESP ou BRA), seguido de outros três caracteres que apontam se o grupo é de controle ou experimental (CON ou EXP) e, por fim, caso aplicável, o ano escolar (5 ou 6). Caso necessário, ainda é adicionada a informação se a amostra é relativa ao pré-teste ou pós-teste somente. Uma relação de todas as amostras encontram-se nos Quadros 25 e 26.

Quadro 25: Nome das amostras e conjunto de amostras (Espanha)

ESPAÑA	ESP-EXP	ESP-EXP-5	ESP-EXP-5 (pré)
			ESP-EXP-5 (pós)
		ESP-EXP-6	ESP-EXP-6 (pré)
			ESP-EXP-6 (pós)
	ESP-CON	ESP-CON-5	ESP-CON-5 (pré)
			ESP-CON-5 (pós)
		ESP-CON-6	ESP-CON-6 (pré)
			ESP-CON-6 (pós)

Fonte: próprio autor

Quadro 26: Nome das amostras e conjunto de amostras (Brasil)

BRASIL	BRA-EXP	BRA-EXP-5	BRA-EXP-5 (pré)
			BRA-EXP-5 (pós)
		BRA-EXP-6	BRA-EXP-6 (pré)
			BRA-EXP-6 (pós)
	BRA-CON	BRA-CON-5	BRA-CON-5 (pré)
			BRA-CON-5 (pós)
		BRA-CON-6	BRA-CON-6 (pré)
			BRA-CON-6 (pós)

Fonte: próprio autor

5.1 ESPANHA

Apresenta-se neste capítulo os dados obtidos em intervenções ocorridas na Espanha nos quintos e sextos anos da Educação Primária.

5.1.1 Quintos anos

Utilizou-se o mesmo fluxo de análise e aplicação dos dados para verificar se as aulas tiveram algum efeito nos grupos experimentais. Após análise dos dados apresentados na Tabela 15, o teste da equivalência dos grupos (Controle e Experimental) dos quintos anos, através de um Teste T Student Independente, apresentou um p-valor (0,080) superior ao nível nominal

($\alpha=0,05$). Sendo assim, não se evidencia a rejeição da hipótese de igualdade dos grupos de pré-testes, comprovando a equivalência dos grupos.

Tabela 15: Comparação de Perfil dos quintos anos (ESP-CON-5 e ESP-EXP-5)

PRÉ-TESTE			
	N	Média (dp)	p-valor
Desempenho Controle	23	9,70 ($\pm 3,15$)	0,080
Desempenho Experimental	19	11,42 ($\pm 3,04$)	

Fonte: próprio autor

Após a confirmação de equivalência dos grupos, fez-se um Teste T Student Pareado entre os grupos de controle e experimental (vide Tabela 16). No grupo de controle, composto por 23 estudantes, atingiu-se uma média amostral de 9,70 com desvio padrão de 3,15 (para mais ou para menos) e, no pós-teste, apresentou média amostral de 10,30 com desvio padrão de 3,31. Entanto, o grupo experimental, composto por 19 estudantes, evidenciou média amostral de 11,42 com desvio padrão de 3,04 (para mais ou para menos) e no pós-teste apresentou média amostral de 13,68 com desvio padrão de 3,13.

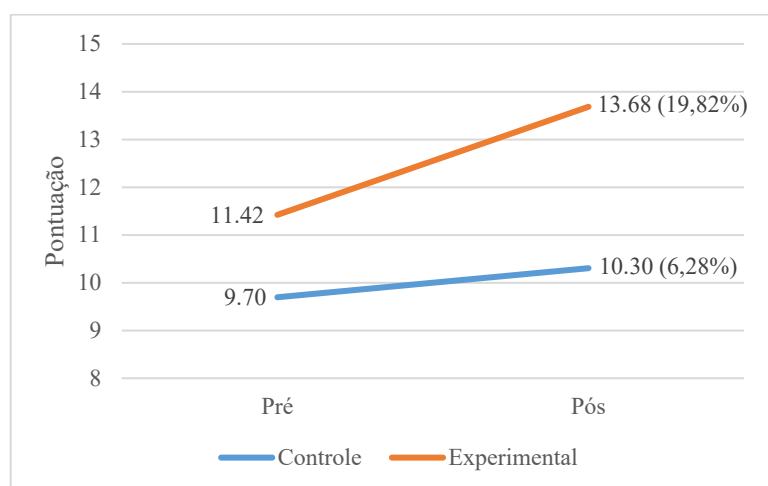
Tabela 16: Resultados do TPC (ESP-CON-5 e ESP-EXP-5)

CONTROLE					
	N	Pré-Teste Média (Desvio Padrão)	Pós-Teste Média (Desvio Padrão)	p-valor	Alteração de Desempenho (%)
Desempenho	23	9,70 ($\pm 3,15$)	10,30 ($\pm 3,31$)	0,370	+ 0,61 (6,28%)
EXPERIMENTAL					
	N	Pré-Teste Média (Desvio Padrão)	Pós-Teste Média (Desvio Padrão)	p-valor	Alteração de Desempenho (%)
Desempenho	19	11,42 ($\pm 3,04$)	13,68 ($\pm 3,13$)	0,005	+ 2,26 (19,82%)

Fonte: próprio autor

Levando-se em consideração um p-valor (0,370) superior ao nível nominal ($\alpha = 0,05$) no grupo de controle, não se evidenciou a rejeição da hipótese de igualdade do grupo para o Teste T pareado, comprovando não ter havido melhoria de desempenho significativo. O contrário ocorreu no grupo experimental, onde foi verificado um p-valor inferior ao nominal (0,005), indicando diferença significativa entre o pré e pós teste, evidenciando a rejeição da hipótese nula. Sendo assim, a partir das médias (vide Gráfico 3) e dos p-valores calculados, comprova-se que houve uma melhoria de desempenho significativa apenas no grupo experimental, confirmando a eficácia das aulas de Pensamento Computacional Desplugado no quinto ano espanhol.

Gráfico 3: Gráfico comparativo dos resultados na Espanha (Quinto Ano)



Fonte: próprio autor

5.1.2 Sextos anos

Através do mesmo processo de análise dos dados, realizou-se uma verificação se as aulas tiveram algum efeito nos grupos experimentais do sexto ano do Ensino Fundamental na Espanha. Após análise dos dados apresentados na Tabela 17, o teste de equivalência dos grupos (Controle e Experimental), através de um Teste T Student Independente, apresentou um p-valor (0,807) superior ao nível nominal ($\alpha=0,05$), ou seja, uma diferença não significativa. Sendo assim, não se evidencia a rejeição da hipótese de igualdade dos grupos de pré-testes, comprovando a equivalência dos grupos.

Tabela 17: Comparação de Perfil (ESP-CON-6 e ESP-EXP-6)

PRÉ-TESTE			
	N	Média (dp)	p-valor
Desempenho Controle	14	11,21 ($\pm 3,33$)	0,807
Desempenho Experimental	16	11,50 ($\pm 3,01$)	

Fonte: próprio autor

Após a confirmação de equivalência dos grupos, fez-se um Teste T Student Pareado entre o grupo de controle e experimental (vide Tabela 18). No grupo de controle, composto por 14 estudantes, atingiu-se uma média amostral de 11,21 com desvio padrão de 3,33 (para mais ou para menos) e no pós-teste apresentou média amostral de 11,71 com desvio padrão de 4,07. No entanto, o grupo experimental, composto por 16 estudantes, observou-se uma média amostral de 11,50 com desvio padrão de 3,01 (para mais ou para menos) e no pós-teste apresenta média amostral de 14,00 com desvio padrão de 2,97.

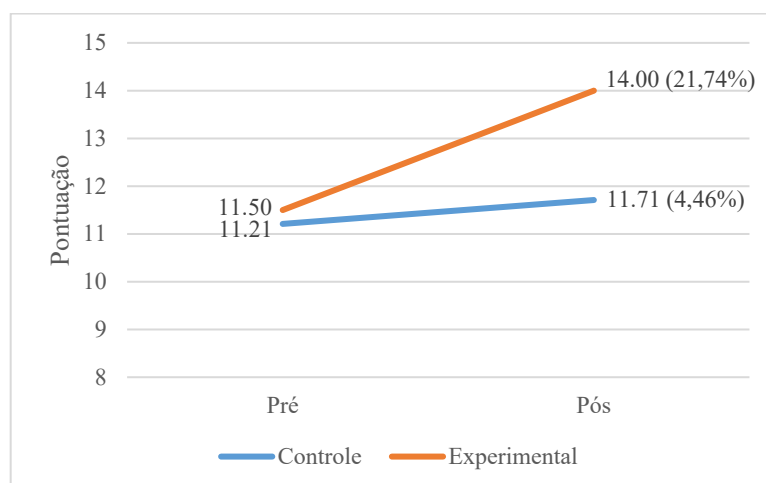
Tabela 18: Resultados do TPC (ESP-CON-6 e ESP-EXP-6)

CONTROLE					
	N	Pré-Teste Média (Desvio Padrão)	Pós-Teste Média (Desvio Padrão)	p-valor	Alteração de Desempenho (%)
Desempenho	14	11,21 ($\pm 3,33$)	11,71 ($\pm 4,07$)	0,538	+ 0,50 (4,46%)
EXPERIMENTAL					
	N	Pré-Teste Média (Desvio Padrão)	Pós-Teste Média (Desvio Padrão)	p-valor	Alteração de Desempenho (%)
Desempenho	16	11,50 ($\pm 3,01$)	14,00 ($\pm 2,97$)	0,016	+ 2,50 (21,74%)

Fonte: próprio autor

Levando-se em consideração um p-valor (0,538) superior ao nível nominal ($\alpha = 0,05$) no grupo de controle, o mesmo não evidenciou a rejeição da hipótese de igualdade do grupo para o Teste T pareado, comprovando que não houve melhoria de desempenho significativo. O contrário ocorreu no grupo experimental, onde foi verificado um p-valor inferior ao nominal (0,016), indicando diferença significativa entre os pré e pós-testes, acusando a rejeição da hipótese nula. Sendo assim, a partir das médias (vide Gráfico 4) e dos p-valores calculados, comprova-se ter havido uma melhoria de desempenho significativa apenas no grupo experimental, o que notabilizou a eficácia das aulas de Pensamento Computacional Desplugado também no sexto ano espanhol.

Gráfico 4: Gráfico comparativo dos resultados na Espanha (Sexto Ano)



Fonte: próprio autor

5.1.3 Resultado geral da Espanha (Quintos e Sextos Anos)

Por fim, realizou-se um paralelo com os dados obtidos de todas as amostras geradas durante as intervenções na Espanha, utilizando o mesmo fluxo de análise, conforme apontado nas Figuras 68 e 69, a fim de verificar se os estudantes dos grupos experimentais tiveram uma melhoria de desempenho significativo posterior às aulas de Pensamento Computacional Desplugado. Após análise dos dados apresentados na Tabela 19, o teste de equivalência dos grupos (Controle e Experimental) dos quintos e sextos anos, através de um Teste T Student Independente, apresentou um p-valor (0,116) superior ao nível nominal ($\alpha=0,05$). Dessa forma, não se evidencia a rejeição da hipótese de igualdade dos grupos de pré-testes, comprovando a equivalência dos grupos.

Tabela 19: Comparação de perfil dos quintos e sextos anos (ESP-CON e ESP-EXP)

PRÉ-TESTE			
	N	Média (dp)	p-valor
Desempenho Controle	37	10,27 ($\pm 3,26$)	0,116
Desempenho Experimental	35	11,48 ($\pm 2,98$)	

Fonte: próprio autor

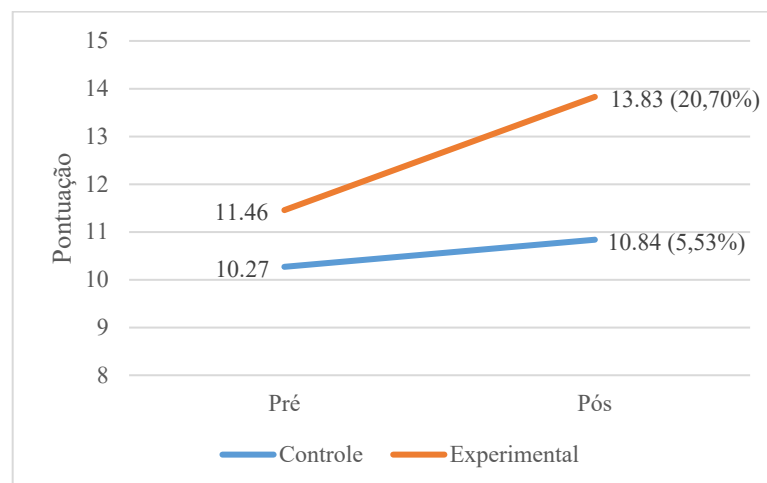
Após a validação da equivalência dos grupos, fez-se um Teste T Student Pareado entre os grupos de controle e experimental (vide Tabela 20). No grupo de controle, composto por 37 estudantes, atingiu-se uma média amostral de 10,27 com desvio padrão de 3,26 (para mais ou para menos) e no pós-teste apresentou média amostral de 10,84 com desvio padrão de 3,62. Entretanto, o grupo experimental, composto por 35 estudantes, observou-se uma média amostral de 11,46 com desvio padrão de 2,98 (para mais ou para menos) e, no pós-teste, apresentou média amostral de 13,83 com desvio padrão de 3,01.

Tabela 20: Resultados do TPC dos quintos e sextos anos (ESP-CON e ESP-EXP)

CONTROLE					
	N	Pré-Teste Média (Desvio Padrão)	Pós-Teste Média (Desvio Padrão)	p-valor	Alteração de Desempenho (%)
Desempenho	37	10,27 ($\pm 3,26$)	10,84 ($\pm 3,62$)	0,267	+ 0,57 (5,53%)
EXPERIMENTAL					
	N	Pré-Teste Média (Desvio Padrão)	Pós-Teste Média (Desvio Padrão)	p-valor	Alteração de Desempenho (%)
Desempenho	35	11,46 ($\pm 2,98$)	13,83 ($\pm 3,01$)	< 0,001	+ 2,37 (20,70%)

Fonte: próprio autor

Gráfico 5: Gráfico comparativo dos resultados na Espanha (Quinto e Sexto Anos)



Fonte: próprio autor

Levando-se em consideração um p-valor (0,267) superior ao nível nominal ($\alpha = 0,05$) no grupo de controle, não se evidenciou a rejeição da hipótese de igualdade do grupo para o Teste T pareado, comprovando que não houve melhoria de desempenho significativo. O contrário ocorreu no grupo experimental, onde foi verificado um p-valor inferior ao nominal ($< 0,001$), indicando diferença altamente significativa entre o pré- e pós-testes, caracterizando a rejeição da hipótese nula. Assim, a partir das médias (vide Gráfico 5) e dos p-valores calculados, comprova-se que houve uma melhoria de desempenho significativo apenas no grupo experimental, confirmando a eficácia das aulas de Pensamento Computacional Desplugado também nas amostras da Espanha.

5.1.4 Observações nas Aulas de Scratch

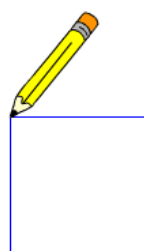
As aulas de Scratch ocorreram nos laboratórios de informática da escola depois da aplicação do pós-teste de Pensamento Computacional. Instalou-se o software Scratch 2¹ *build* 446 nos computadores para conceder as aulas “plugadas” aos alunos das escolas, levando-se em consideração a estabilidade e pelo fato de não haver necessidade de uma conexão com a internet. Infelizmente, devido ao tempo, poucos conceitos puderam ser exercitados com os estudantes durante as aulas de Scratch, assim como devido a quantidade de computadores os alunos precisaram sentar em duplas, dificultando a análise qualitativa do pesquisador.

Nessas aulas, foram trabalhadas atividades que tinham alguma ligação com os exercícios desplugados vistos anteriormente em sala de aula sem o uso de computadores. Como

¹ <https://scratch.mit.edu/download>

por exemplo, o uso das setas direcionais para movimentar um personagem no palco do Scratch ou desenhar objetos simples através de uma sequência de comandos (vide a versão desplugada em Quadro 22 e a versão em Scratch na Figura 70). Na Figura 70 também constam as possibilidades encontradas durante a atividade, sendo que a resposta que melhor contempla os conteúdos do PC encontra-se à direita e uma gradual diminuição de qualidade à sua esquerda.

Figura 70: Exemplo de exercício em Scratch (Desenhar um quadrado)



Fonte: próprio autor

Durante as sessões de Scratch, foi possível observar que:

- No grupo experimental foi percebida uma maior facilidade na solução dos exercícios propostos e ao responderem questionamentos proferidos pelo pesquisador. Acredita-se que isso ocorreu devido ao fato de ser possível relacionar os blocos aos conteúdos e estratégias trabalhados anteriormente em sala de aula;
- No grupo de controle, necessitou-se de um período maior por atividade, pois era necessário explicar aos estudantes a funcionalidade de cada bloco (*e.g.* “*si/si no*”, “*repetir*”, “*por siempre*”, “*eventos*”) antes de cada sessão. Essas dificuldades também impossibilitaram a conclusão de todo o plano de aula;
- Os estudantes do grupo experimental se demonstraram bastante interessados no Scratch, pois após a primeira aula começaram a praticá-lo em casa;
- O grupo de controle se demonstrou mais motivado em iniciar as aulas de Scratch, pois era o seu primeiro contato com o Pensamento Computacional. Ao mesmo tempo, a professora do grupo percebeu que seus alunos se apresentavam bastante esgotados no final da aula, pois a carga de trabalho era superior ao grupo experimental;
- Percebeu-se que os alunos dos grupos experimentais eram mais focados e agiam de maneira mais racional ao resolver as atividades. Já no grupo de

controle, a ansiedade tinha um efeito negativo e sobrepunha muitas vezes o raciocínio;

- Duas estudantes dos grupos de controle se demonstraram desinteressadas nas aulas, pois não conseguiam entender o significado de cada bloco e como organizá-los de uma maneira lógica. Pelo fato de não terem uma compreensão muito clara de como resolver as atividades, os alunos que perguntavam se faltava muito tempo para acabar a aula ou que pediam para sair da sala por algum motivo (*e.g.* ir ao banheiro, falar com outro professor) eram em maior quantidade. O efeito contrário ocorreu no grupo experimental, pois os alunos colocaram em prática o que já tinham trabalhado em sala de aula. Não se registrou nenhum pedido de sair da sala e no final os alunos não queriam que a aula acabasse;
- Observou-se que os estudantes dos grupos de controle exercitaram com mais ênfase a criatividade quando resolviam os problemas propostos. Já os grupos experimentais tentavam entender melhor o que estavam fazendo e, algumas vezes, já era possível perceber que alteravam seus scripts para otimizar a execução dos seus aplicativos;
- A atividade desplugada Boneca de Papel (vide Capítulo 4.2.2) teve um papel importante na compreensão dos blocos do Scratch, pois a atividade utiliza uma estrutura muito semelhante ao software.

No capítulo seguinte apresentam-se os resultados e observações obtidos no Brasil.

5.2 BRASIL

Conforme relatado no Capítulo 4, após realizar a pesquisa na Espanha, a mesma sequência de testes e intervenções ocorreram no Brasil, onde esta pesquisa teve continuidade. A seguir, serão apresentados os resultados provenientes dos quintos e sextos anos do Ensino Fundamental no Brasil.

5.2.1 Quintos anos

Para que fosse possível verificar a equivalência de amostras entre as turmas do quinto ano, com o objetivo de observar se houve uma melhoria de desempenho significativo nos grupos experimentais, realizou-se o mesmo fluxo de cálculos utilizados na seção anterior. Após

evidenciar uma distribuição não normal da pontuação do TPC, foi necessário a utilização do teste não paramétrico aplicado para duas amostras independentes (Mann-Whitney), a fim de que fosse comprovada a equivalência das duas turmas do quinto ano (controle e experimental). Como regra de decisão, caso o p-valor fosse inferior ao nível nominal estabelecido ($\alpha=0,05$), então rejeita-se a hipótese nula de igualdade de medianas, comprovando a diferença significativa entre os grupos. Após análise dos dados apresentados na Tabela 21, observou-se um p-valor (0,340) superior ao nível nominal ($\alpha=0,05$). Sendo assim, não se evidencia a rejeição da hipótese de igualdade dos grupos de pré-testes, comprovando a equivalência dos grupos.

Tabela 21: Comparação de perfil do quinto ano (BRA-CON-5 e BRA-EXP-5)

PRÉ-TESTE				
	N	Média de Postos	Média	p-valor
Desempenho Controle	10	4,5	9,30	0,340
Desempenho Experimental	15	8,5	9,93	

Fonte: próprio autor

Após a confirmação de equivalência dos grupos, levando-se em consideração a não normalidade dos dados, aplicou-se o teste não paramétrico (Wilcoxon) para verificar se houve uma melhoria de desempenho significativa no grupo experimental. No grupo de controle, composto por 10 estudantes, observou-se uma mediana de 8 com intervalo interquartil variando entre 6,75 e 11,25 e no pós-teste, identificou-se uma mediana de 11 com intervalo interquartil entre 7,5 e 12,25. No entanto, o grupo experimental, composto por 15 estudantes, constatou-se uma mediana de 9 com intervalo interquartil entre 8 e 11 e, no pós-teste, identificou-se uma mediana de 10 com intervalo interquartil entre 9 e 14. Os dados organizados encontram-se na Tabela 22.

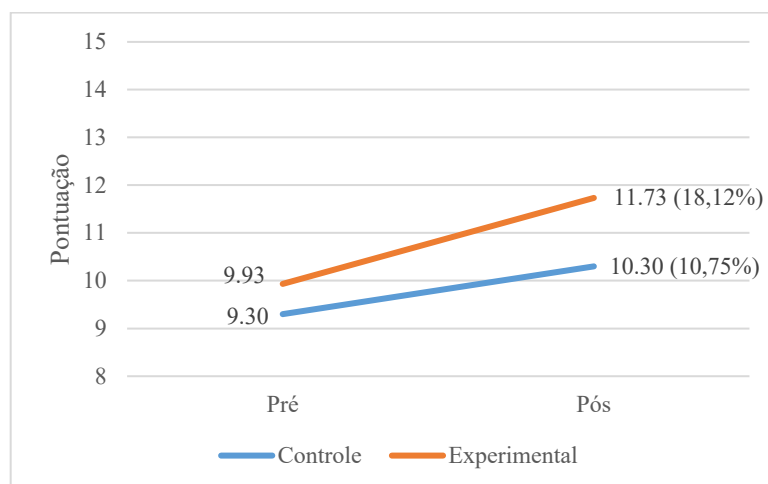
Tabela 22: Resultados do TPC do quinto ano (BRA-CON-5 e BRA-EXP-5)

CONTROLE							
	N	Pré-Teste Mediana (IQ)	Pós-Teste Mediana (IQ)	Média Pré-teste	Média Pós-teste	p-valor	Alteração de Desempenho (%)
Desempenho	10	8 (6,75-11,25)	11 (7,5-12,25)	9,30	10,30	0,484	+ 1,00 (10,75%)
EXPERIMENTAL							
	N	Pré-Teste Mediana (IQ)	Pós-Teste Mediana (IQ)	Média Pré-teste	Média Pós-teste	p-valor	Alteração de Desempenho (%)
Desempenho	15	9 (8-11)	10 (9-14)	9,93	11,73	0,013	+ 1,80 (18,12%)

Fonte: próprio autor

Levando-se em consideração um p-valor (0,484) superior ao nível nominal ($\alpha = 0,05$) no grupo de controle, não se evidenciou a rejeição da hipótese de igualdade do grupo para o Teste de Wilcoxon, com isso, sem êxito na melhoria de desempenho significativo. O contrário ocorreu no grupo experimental, onde foi verificado um p-valor inferior ao nominal (0,013), indicando diferença significativa entre os pré e pós-testes, evidenciando a rejeição da hipótese nula. Dessa forma, a partir das médias (vide Gráfico 6) e dos p-valores calculados, comprova-se que houve uma melhoria de desempenho significativa apenas no grupo experimental, confirmando a eficácia das aulas de Pensamento Computacional Desplugado também no quinto ano brasileiro.

Gráfico 6: Gráfico comparativo dos resultados no Brasil (Quinto Ano)



Fonte: próprio autor

5.2.2 Sextos anos

Para verificar a equivalência dos grupos dos sextos anos (controle e experimental), os mesmos testes foram realizados. Levando-se em consideração uma distribuição normal nas amostras de pré-teste, aplicou-se um Teste T de Student Independente. Após análise dos dados apresentados na Tabela 23, encontrou-se um p-valor (0,656) superior ao nível nominal ($\alpha = 0,05$), ou seja, uma diferença não significativa. Consequentemente, não se evidenciou a rejeição da hipótese de igualdade dos grupos de pré-testes, comprovando a equivalência dos grupos.

Tabela 23: Comparação de Perfil (BRA-CON-6 e BRA-EXP-6)

PRÉ-TESTE			
	N	Média (dp)	p-valor
Desempenho Controle	19	9,68 ($\pm 4,07$)	0,656
Desempenho Experimental	19	9,16 ($\pm 3,10$)	

Fonte: próprio autor

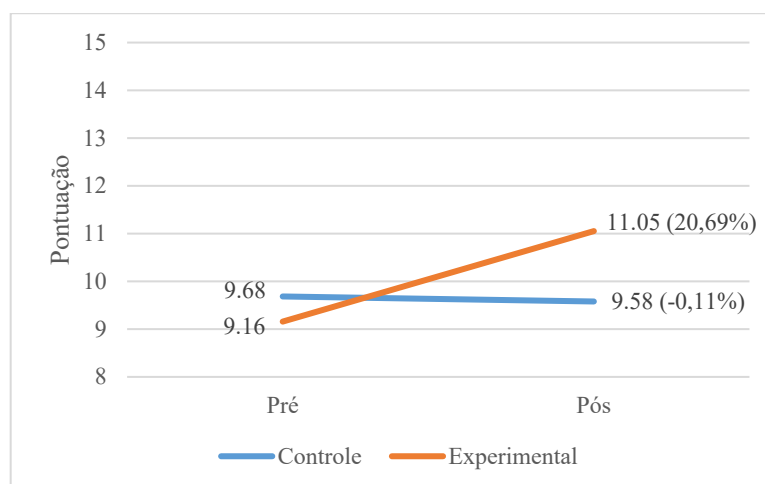
Após a confirmação de equivalência dos grupos, levando-se em consideração a normalidade dos dados, aplicou-se o Teste T de Student Pareado para verificar se houve uma melhoria de desempenho significativa no grupo experimental. No grupo de controle, composto por 19 estudantes, atingiu-se uma média amostral de 9,68 com desvio padrão de 4,07 (para mais ou para menos) e no pós-teste apresentou média amostral de 9,58 com desvio padrão de 3,45. No entanto, o grupo experimental, também composto por 19 estudantes, observou-se uma média amostral de 9,16 com desvio padrão de 3,01 (para mais ou para menos) no pré-teste e no pós-teste apresenta média amostral de 11,05 com desvio padrão de 4,54. Os dados organizados encontram-se na Tabela 24.

Tabela 24: Resultados do TPC do sexto ano (BRA-CON-6 e BRA-EXP-6)

CONTROLE					
	N	Pré-Teste Média (Desvio Padrão)	Pós-Teste Média (Desvio Padrão)	p-valor	Alteração de Desempenho (%)
Desempenho	19	9,68 ($\pm 4,07$)	9,58 ($\pm 3,45$)	0,916	- 0,11 (1,09%)
EXPERIMENTAL					
	N	Pré-Teste Média (Desvio Padrão)	Pós-Teste Média (Desvio Padrão)	p-valor	Alteração de Desempenho (%)
Desempenho	19	9,16 ($\pm 3,10$)	11,05 ($\pm 4,54$)	0,020	+ 1,89 (20,69%)

Fonte: próprio autor

Gráfico 7: Gráfico comparativo dos resultados no Brasil (Sexto Ano)



Fonte: próprio autor

Levando-se em consideração um p-valor (0,916) superior ao nível nominal ($\alpha = 0,05$) no grupo de controle, não se evidenciou a rejeição da hipótese de igualdade do grupo para o Teste T de Student, comprovando que não houve melhoria de desempenho significativo. O contrário ocorreu no grupo experimental, onde foi verificado um p-valor inferior ao nominal (0,020), o que indica diferença significativa entre os pré- e pós- testes, bem como evidencia a rejeição da hipótese nula. Sendo assim, a partir das médias (vide Gráfico 7) e dos p-valores calculados, comprova-se que houve uma melhoria de desempenho significativa apenas no grupo experimental, comprovando a eficácia das aulas de Pensamento Computacional Desplugado também no sexto ano brasileiro.

5.2.3 Resultado geral no Brasil (Quintos e Sextos Anos)

Por fim, realizou-se um paralelo com os dados de todas as amostras geradas durante as intervenções no Brasil, utilizando o mesmo fluxo de análise dos mesmos, o que mostra as Figuras 68 e 69, a fim de verificar se os estudantes dos grupos experimentais tiveram uma melhoria de desempenho significativo após as aulas de Pensamento Computacional Desplugado. A análise posterior dos dados apresentados na Tabela 25, o teste da equivalência dos grupos (Controle e Experimental) dos quintos e sextos anos, através de um Teste T Student Independente, revelou um p-valor (0,953) superior ao nível nominal ($\alpha = 0,05$). Sendo assim, não se evidencia a rejeição da hipótese de igualdade das amostras de pré-testes, comprovando a equivalência dos grupos.

Tabela 25: Comparação de perfil (BRA-CON e BRA-EXP)

PRÉ-TESTE			
	N	Média (dp)	p-valor
Desempenho Controle	29	9,55 ($\pm 2,98$)	0,953
Desempenho Experimental	34	9,50 ($\pm 3,15$)	

Fonte: próprio autor

Com a confirmação de equivalência dos grupos, levando-se em consideração a não normalidade dos dados, aplicou-se o teste não-paramétrico independente (Wilcoxon) para verificar se houve uma melhoria de desempenho significativa no grupo experimental. No grupo de controle, composto por 29 estudantes, observou-se uma mediana de 10 com intervalo interquartil variando entre 7 e 10 e, no pós-teste, identificou-se uma mediana de 11 com intervalo interquartil entre 7 e 12,5. Entretanto, o grupo experimental, composto por 34 estudantes, observou-se uma mediana de 9 com intervalo interquartil entre 7,75 e 11, e no

pós-teste, identificou-se uma mediana de 10 com intervalo interquartilico entre 8 e 14,25. Os dados organizados encontram-se na Tabela 26.

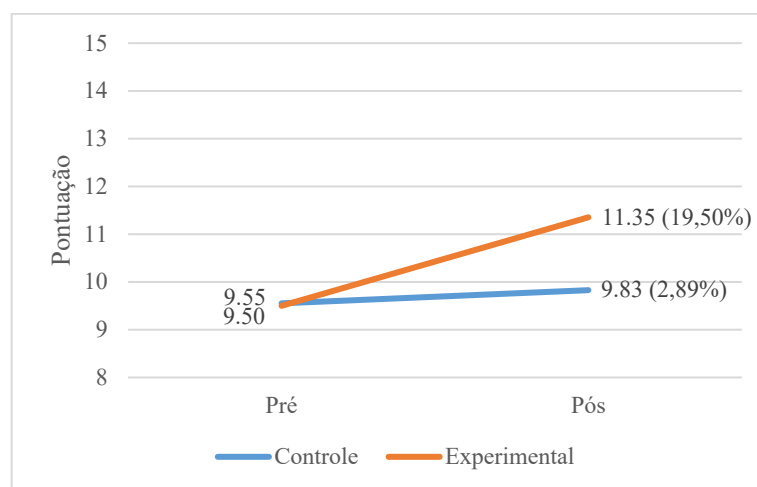
Tabela 26: Resultados do TPC (BRA-CON e BRA-EXP)

CONTROLE							
	N	Pré-Teste Mediana (IQ)	Pós-Teste Mediana (IQ)	Média Pré-teste	Média Pós-teste	p-valor	Alteração de Desempenho (%)
Desempenho	29	10 (7-11)	11 (7-12,5)	9,55	9,83	0,491	+ 0,28 (2,89%)
EXPERIMENTAL							
	N	Pré-Teste Mediana (IQ)	Pós-Teste Mediana (IQ)	Média Pré-teste	Média Pós-teste	p-valor	Alteração de Desempenho (%)
Desempenho	34	9 (7,75-11)	10 (8-14,25)	9,50	11,35	0,001	+ 1,85 (19,50%)

Fonte: próprio autor

Levando-se em consideração um p-valor (0,491) superior ao nível nominal ($\alpha = 0,05$) no grupo de controle, não evidenciamos a rejeição da hipótese de igualdade do grupo para o Teste T pareado, o que comprovou não ter havido melhoria de desempenho significativo. O contrário ocorreu no grupo experimental, onde foi verificado um p-valor inferior ao nominal (0,001), indicando diferença significativa entre o pré e pós teste, evidenciando a rejeição da hipótese nula. Destarte, a partir das médias (vide Gráfico 8) e dos p-valores calculados, confirma-se que houve uma melhoria de desempenho significativa apenas no grupo experimental, comprovando também a eficácia das aulas de Pensamento Computacional Desplugado em estudantes brasileiros.

Gráfico 8: Gráfico comparativo dos resultados no Brasil (Quinto e Sexto Anos)



Fonte: próprio autor

5.2.4 Observações nas Aulas de Scratch

Seguindo as mesmas etapas da Espanha, instalou-se o software Scratch em netbooks em uma das escolas e, devido a questões técnicas, na outra escola se optou por executá-lo no navegador. Infelizmente, não foi possível registrar devidamente o rendimento dos alunos devido a diversos fatores, como por exemplo:

- Ansiedade altíssima dos estudantes em utilizar os computadores, pois aulas no laboratório não são recorrentes. Alguns estudantes afirmaram que era a primeira vez que entravam no laboratório de informática;
- Estudantes não sabiam sequer ligar ou digitar no computador;
- Problemas de comportamento entre os pares;
- Dificuldades de funcionamento dos computadores.

Esses fatores impossibilitaram uma sequência lógica das atividades e conteúdos a serem trabalhados com os estudantes, tornando uma avaliação qualitativa inviável e com resultados inconclusivos.

No capítulo seguinte faz-se uma tentativa de cruzar os dados observados entre a Espanha e o Brasil.

5.3 COMPARATIVO ENTRE PAÍSES

Para que seja possível realizar uma análise entre os dois países, primeiramente é necessário verificar se as amostras ESP-EXP e BRA-EXP são grupos equivalentes, a fim de que permitam, posteriormente, analisar se houve de fato uma melhoria de desempenho mais significativa em um dos países. A amostra da Espanha apresentou uma média de pontuação de 11,45 e o Brasil 9,5. Ambos os grupos apresentaram uma distribuição normal, com aplicação do Teste T de Student Independente. Os dados utilizados e o resultado do Teste T de Student para verificar a uniformidade das duas amostras é apresentado na Tabela 27.

Tabela 27: Comparação de perfil em pré-testes (ESP-EXP e BRA-EXP)

EXPERIMENTAL – PRÉ-TESTE			
	N	Média (dp)	p-valor
ESP	35	11,45 (±2,98)	0,010
BRA	34	9,5 (±3,14)	

Fonte: próprio autor

Levando-se em consideração um p-valor (0,010) inferior ao nível nominal ($\alpha=0,05$) das amostras, evidenciou-se a rejeição da hipótese de igualdade do grupo para o Teste T Independente, confirmando que há uma diferença significativa entre os estudantes espanhóis e brasileiros. Em assim sendo, torna-se estatisticamente inviável a realização de um teste comparativo entre os estudantes dos dois países antes das intervenções. Com a intenção de verificar se os estudantes de um dos países tiveram uma melhoria de desempenho mais acentuada, realizou-se outro teste de comparação de perfil nos mesmos grupos, porém baseado nos resultados dos pós-testes. Os dados utilizados nessa análise encontram-se na Tabela 28.

Tabela 28: Comparação de perfil em pós-testes (ESP-EXP e BRA-EXP)

EXPERIMENTAL – PÓS-TESTE				
	N	Média de Postos	Média	p-valor
ESP	35	21,25	13,83	0,004
BRA	34	13,75	11,35	

Fonte: próprio autor

Levando-se em consideração um p-valor (0,004) inferior ao nível nominal ($\alpha=0,05$) das amostras, evidenciamos a rejeição da hipótese de igualdade do grupo para o Teste Wilcoxon, comprovando que há uma diferença significativa entre os estudantes espanhóis e brasileiros. Dessa forma, torna-se também estatisticamente inviável a realização de um teste comparativo entre os estudantes dos dois países após as intervenções. Essa constatação demonstra que os dois países possuem um perfil diferente e as intervenções tiveram um efeito esperado em ambos, sendo assim, pode-se especular que há indícios de que as intervenções são eficazes em diferentes perfis de estudantes.

5.4 COMPARATIVO ENTRE GÊNEROS

No primeiro momento, tentou-se encontrar uma melhoria de desempenho entre as crianças de diferentes gêneros, utilizando a amostra de todos os participantes da pesquisa (Espanha e Brasil). Como todas as amostras apresentaram uma distribuição normal dos dados, fez-se uso do Teste T de Student para analisar o p-valor. Como se pode perceber na Tabela 29, os estudantes do grupo de controle não revelaram uma melhoria significativa de desempenho, o que já era esperado. Já os estudantes do grupo experimental apresentaram um p-valor inferior ao nível nominal ($\alpha=0,05$), evidenciando uma melhoria significativa em ambos os grupos, no entanto, constatou-se um p-valor inferior no grupo dos meninos, ou seja, uma melhoria mais significativa neste gênero.

Tabela 29: Comparativo de médias entre gêneros (todos grupos)

Grupo	Gênero	N	Pré	Pós	p-valor	Alteração de Desempenho
Controle	Masculino	36	10,47	10,86	0,472	3,72%
	Feminino	30	9,33	10,03	0,272	7,50%
Experimental	Masculino	32	10,94	13,06	0,0001	19,38%
	Feminino	37	10,11	12,22	0,0006	20,87%

Fonte: próprio autor

Na tentativa de se identificar uma diferença significativa entre os dois gêneros de maneira alternativa, realizou-se também uma análise de gênero em cada um dos países individualmente. Para isso, analisaram-se apenas os grupos experimentais, tendo em vista a observação de mudança de desempenho apenas nessas amostras. Os dados utilizados/calculados encontram-se na Tabela 30.

Tabela 30: Comparativo de médias entre gêneros dos grupos experimentais (por país)

País	Gênero	N	Pré	Pós	p-valor	Alteração de Desempenho
Espanha	Masculino	17	12,70	15,29	0,001	20,39%
	Feminino	18	10,28	12,44	0,039	21,01%
Brasil	Masculino	15	8,93	10,53	0,051	17,92%
	Feminino	19	9,95	12,00	0,009	20,60%

Fonte: próprio autor

Após tabulação e análise dos dados, pôde-se observar que, na Espanha, houve uma melhoria mais significativa no gênero masculino, ao contrário do Brasil, em que o gênero feminino revelou maior destaque com relação à melhoria de desempenho mais significativa.

No capítulo seguinte, se apresenta as conclusões da pesquisa, desdobramentos do projeto e trabalhos futuros.

CONCLUSÃO E TRABALHOS FUTUROS

Computadores impactam em quase todos os aspectos de nossas vidas, qualquer serviço ou produto que utilizamos diariamente depende, direta ou indiretamente, de algum processo ou decisão determinada por um *chip* de computador. O fato é que não temos limites para as infinitas possibilidades que os sistemas computacionais permitem.

Para que as pessoas possam utilizar a Computação em todo seu potencial, todas as gerações (de crianças a idosos) deveriam ter acesso aos conceitos aqui discutidos; sobretudo, é vital conscientizar os legisladores, a fim de distinguir a Computação (físico e lógico) da Informática (TICs), para não ocorrer uma má compreensão dos termos e, sensibilizá-los da importância dos princípios da Computação, que devem ser ensinados desde o Ensino Básico. A literatura sugere que a introdução da Computação na Educação Básica traz diversos benefícios, tais como: o atendimento a uma alta demanda de mão-de-obra qualificada com conhecimentos em programação; o estímulo para que os estudantes não aprendam somente a trabalhar de maneira individual, porém também colaborativamente; a possibilidade de usá-la como extensão do corpo humano para aumentar suas capacidades, independente da sua área de atuação profissional, possibilitando uma escala de problemas mais complexos; e, o de tornar as TICs não mais um fim, porém um meio pelo qual se explora as mais diferentes áreas do conhecimento, impactando significativamente no desenvolvimento de habilidades e competências essenciais para a vida moderna.

Em um segundo momento, o autor entende que se torna necessário o desenvolvimento de materiais e abordagens no ensino do Pensamento Computacional, para que as autoridades deem suporte aos futuros professores, assim como também possam os gestores escolares disponibilizar uma maior compreensão e apoio pelo tema e, imbuídos no mesmo intuito, aderir à proposta de inserir o PC no currículo escolar. Curzon *et al.* (2014) demonstram que professores têm um altíssimo grau de interesse em entender mais sobre o PC e como aplicá-lo em sala de aula. Os resultados da pesquisa de Curzon, com a participação de 126 professores em um *workshop*, demonstraram uma melhora significativa na motivação, utilidade e, ainda, apontaram ser interessante, inspirador e divertido, além de aumentar sua confiança a respeito do tema.

Em um terceiro momento, deve ocorrer uma formação de professores para que atuem como multiplicadores do saber, em parceria com as Instituições de Ensino Superior (IFES). Certos países, como os Estados Unidos da América e a Alemanha, encontraram a solução quanto à falta de professores para lecionar a disciplina de Computação nas aulas de Matemática,

podendo esse exemplo servir de modelo ao Brasil. Não podemos esquecer que a alta demanda por profissionais da área da Computação reduziu a quantidade de professores especializados em potencial e esses fatores conspiram a favor de professores pouco valorizados e de baixa qualificação. Indiscutivelmente, existe um número grande de professores de escolas que necessitam de treinamento para essa nova realidade, porém ainda se desconhece como formá-los (COSTA, 2017). Atualmente no Brasil, no e-MEC (2017), é possível encontrar cursos de licenciatura em Computação; Ciência da Computação; Computação e Informática; e Interdisciplinar em Matemática e Computação e suas Tecnologias em 47 instituições de ensino superior (Público e Privado), ofertando 11.426 vagas em 73 cursos (Presenciais e EAD). Farias *et al.* (2015) descreve em sua pesquisa que apenas 13,3% dos formandos tiveram contato com o Pensamento Computacional durante o curso. Moreno-León *et al.* (2017) conclui que há uma necessidade muito grande da contratação de professores da área da Informática na Educação para comporem o corpo docente de cursos que são puramente na área da Educação.

Na etapa seguinte, propõe-se a oferta de disciplinas voltadas para a Computação de forma opcional, possibilitando o acesso dos estudantes aos conceitos do Pensamento Computacional para que, futuramente, essas disciplinas façam parte do currículo obrigatório. O Brasil, em pleno processo de formação da BNCC, está em um momento muito oportuno para discutir esse assunto. Seria um ponto desfavorável para nosso país deixar passar essa chance e um prejuízo incalculável para nosso futuro, onde os meios de produção são cada vez mais digitais e as oportunidades de emprego exigem como requisitos um novo conjunto de aptidões.

O estímulo através de competições, prêmios e outras formas de destaque também é muito incentivado por outros governos, pois permite a identificação de novos talentos e inspira outros estudantes a se tornarem profissionais em qualquer área, porém expandindo ainda mais suas possibilidades com o auxílio da Computação na solução de problemas. Ao mesmo tempo que os estudantes avançam em direção ao futuro, eles, na sua adolescência, não estão esclarecidos para saber que rumo seguir, com o que vão trabalhar ou qual carreira seguir, porém o Pensamento Computacional pode contribuir na sua formação, independentemente da área do conhecimento que atuará em sua vida profissional.

Enquanto alguns países possuem em suas escolas praticamente um computador por aluno (*e.g.* Austrália e Nova Zelândia), outros países se destacam pela qualidade no ensino, mesmo compartilhando um único computador com até quatro estudantes (*e.g.* Suécia e Portugal) (OECD, 2015), comprovando que o uso massivo de dispositivos tecnológicos em sala de aula não garante uma melhor educação.

Baseado nesse princípio e a situação socioeconômica brasileira, optou-se pelo formato “desplugado” como uma alternativa para que todo o país, independentemente da situação da escola, possa usufruir dos benefícios que o Pensamento Computacional viabiliza. A presente pesquisa desenvolveu materiais e apontou forte evidências que o ensino do Pensamento Computacional através de atividades que não necessitam de máquinas é possível. Os resultados sugerem que todos os grupos experimentais tiveram uma melhoria do Pensamento Computacional significativa, independente do país, anos escolares investigados ou gênero. O mesmo não ocorreu no grupo de controle, comprovando o efeito positivo das intervenções nos grupos experimentais.

Porém, considera-se importante salientar que atividades desplugadas não devem ser entendidas como uma solução completa de ensino. A abordagem desplugada não atende todos os fundamentos da Computação ou não proporciona uma prática plena. Por esse motivo recomenda-se que seja feito uso dela, sim, para a introdução do Pensamento Computacional pelo fato de “o trabalho com tecnologias digitais ter algo especial que não pode ser explorado por outros tipos de atividades” (NATIONAL US RESEARCH COUNCIL e COMMITTEE FOR THE WORKSHOPS ON COMPUTATIONAL THINKING, 2010). A mesma recomendação pode ser encontrada em Grover e Pea (2013) ou Aggarwal *et al.* (2017) em que foi realizado uma comparação entre o ensino do Pensamento Computacional com e sem o uso de computadores para a criação de um jogo. As atividades desplugadas, a partir de um certo ponto, podem não ser tão eficazes quanto esperado, por manter os alunos distantes de experiências com as tecnologias digitais. Isso poderá acarretar uma visão distorcida, como por exemplo, do que é a Computação, ou até mesmo distanciá-los dela, tendo em vista que programar no computador é uma experiência fundamental e praticamente única. O acesso às máquinas pelas crianças possibilita pôr em prática aquilo que foi trabalhado no formato desplugado, ampliando ainda mais o seu horizonte, a intimidade com uma linguagem de programação e possibilita solucionar problemas ainda mais complexos.

A presente investigação, conforme previsto em seus objetivos, utilizou-se do tripé educacional, integrando em uma única iniciativa, a pesquisa, o ensino e a extensão através do desenvolvimento de objetos de aprendizagem, a realização de intervenções em sala de aula, avaliação do Pensamento Computacional dos estudantes participantes, verificação do efeito das atividades propostas e a comparação dos resultados obtidos em ambos países e entre gêneros. A execução do projeto também possibilitou ao autor experiências e conquistas inesperadas quando esta pesquisa ainda era uma proposta e se desconhecia as proporções que iria tomar. Como desdobramentos desta investigação, pode-se exemplificar: a abrangência dos três

elementos do tripé educacional (pesquisa, ensino e extensão), a geração de material didático em parceria com a indústria (Maurício de Souza Produções e Thinkfun)¹, o desenvolvimento do site² “Pensamento Computacional Brasil” para divulgação de notícias, artigos e materiais produzidos, um grupo em rede social com a participação de pesquisadores de renome nacional e estrangeiro³, atividades de divulgação da importância do Pensamento Computacional em escolas através de palestras e formação de professores a nível local, regional e internacional⁴ (público e privado), a oportunidade de auxiliar a Sociedade Brasileira de Computação na sensibilização do Governo Federal para inclusão do Pensamento Computacional na Base Nacional Curricular Comum⁵, reportagem em jornal, a publicação de artigos em âmbito nacional e internacional e capítulo de livro para divulgação da revisão bibliográfica e dos resultados obtidos, ter sido aceito como membro da CSTA (*Computer Science Teacher Association*) mesmo não estando situado nos EUA, participação como membro da organização de eventos na área da Informática na Educação e Pensamento Computacional, o desenvolvimento de apostilas para crianças dos quartos e quintos anos escolares na Espanha em parceria com professores espanhóis, sem esquecer das oportunidades de vivenciar todo o processo de aplicação da pesquisa e um contato presencial do criador do Teste de Pensamento Computacional aqui utilizado.

Pesquisas que têm a intenção de propor a integração da Computação no currículo da educação formal, aperfeiçoar sua metodologia de ensino e apontam alternativas na formação de professores não podem cessar. Como trabalhos futuros, deseja-se responder questionamentos que surgiram durante a execução desta pesquisa, sendo eles:

- O material desplugado terá o mesmo efeito em estudantes de escolas particulares?
- A partir de que idade ou ano escolar deve-se introduzir o Pensamento Computacional?
- Como desenvolver e medir o Pensamento Computacional em crianças nos anos iniciais?
- Em que momento a abordagem desplugada começa a perder seu efeito desejado e a utilização de máquinas passa a ser recomendada?
- Como as atividades desplugadas podem auxiliar no aprendizado de linguagens visuais?

¹ Vide Apêndice C até Apêndice L

² Site Pensamento Computacional Brasil: www.computacional.com.br

³ Grupo Facebook: www.facebook.com/groups/pcomputacional/

⁴ Seminário Anual eMadrid: <https://goo.gl/xeZeE1>

⁵ Vide Anexo A

- O Pensamento Computacional está devidamente inserido nos currículos dos cursos de licenciatura na área da Computação para atender todas as demandas (Educação Infantil, Educação Básica, Cursos Técnicos e Superiores)?

Saber programar passa a ser algo fundamental daqui para a frente e, por isso toda criança matriculada em alguma escola (pública ou privada), deveria ter o direito de aprender a programar. A questão agora não é se devemos desenvolver o Pensamento Computacional de crianças nas escolas e, sim, como e a partir de que idade/ano. É um caminho sem volta.

REFERÊNCIAS

- ACARA. Australian Curriculum. , 2015. Australian Curriculum Assessment and Reporting Authority. Disponível em: <<http://www.australiancurriculum.edu.au/technologies/digital-technologies/curriculum/f-10?layout=1>>. .
- AGGARWAL, A.; GARDNER-MCCUNE, C.; TOURETZKY, D. S. Evaluating the Effect of Using Physical Manipulatives to Foster Computational Thinking in Elementary School. . p.9–14, 2017. ACM Press. Disponível em: <<http://dl.acm.org/citation.cfm?doid=3017680.3017791>>. Acesso em: 25/6/2017.
- AHO, A. V.; ULLMAN, J. D. **Foundations of computer science**. C ed ed. New York: Computer Science Press, 1995.
- ALBA, F. **Education Scotland - Technologies: Experiences and outcomes**. 2009.
- ANDRADE, D.; CARVALHO, T.; SILVEIRA, J.; et al. Proposta de Atividades para o Desenvolvimento do Pensamento Computacional no Ensino Fundamental. II Congresso Brasileiro de Informática na Educação (CBIE 2013) / XIX Workshop de Informática na Escola (WEIT 2013). **Anais...** . p.169, 2013.
- ASSOCIATION FOR COMPUTING MACHINERY. **A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee**. New York: ACM, 2003.
- 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-a760-0f4e4795d6fa&groupId=43887>. Acesso em: 1/12/2015.
- BARCELOS, T. S.; SILVEIRA, I. F. Teaching Computational Thinking in initial series An analysis of the confluence among mathematics and Computer Sciences in elementary education and its implications for higher education. . p.1–8, 2012. IEEE. Disponível em: <<http://ieeexplore.ieee.org/document/6427135/>>. Acesso em: 29/6/2017.
- BARR, V.; STEPHENSON, C. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? **ACM Inroads**, v. 2, n. 1, p. 48, 2011.
- BARRETO, R. **Computação Desplugada: Ensinando Computação sem Computadores**. 2013.
- BBC LEARNING, B. What is computational thinking? , 2015. Disponível em: <<http://www.bbc.co.uk/education/guides/zp92mp3/revision>>. Acesso em: 23/11/2015.
- BELL, T.; WITTEN, I. H.; FELLOWS, M. **Computer Science Unplugged... - Off-line activities and games for all ages (draft)**. 1º ed. 1997.
- BELL, T.; WITTEN, I. H.; FELLOWS, M. Computer Science Unplugged. , 2015. Disponível em: <http://csunplugged.org/wp-content/uploads/2015/03/CSUnplugged_OS_2015_v3.1.pdf>. Acesso em: 1/1/2015.
- BERRY, M. **Computing in the national curriculum - A guide for primary teachers**. 2013.

BIRD, J.; CALDWELL, H.; MAYNE, P. **Lessons in teaching computing in primary schools**. 2nd edition, revised and updated edition ed. London: Learning Matters, 2017.

BITA, N. Computers in class “a scandalous waste”: Sydney Grammar head. Disponível em: <<http://www.theaustralian.com.au/national-affairs/education/computers-in-class-a-scandalous-waste-sydney-grammar-head/news-story/b6de07e63157c98db9974cedd6daa503>>. Acesso em: 26/3/2016.

BLOOMBERG. Conheça o setor que tem mais vagas que profissionais no Brasil. Disponível em: <<http://www.infomoney.com.br/carreira/emprego/noticia/4447259/conheca-setor-que-tem-mais-vagas-que-profissionais-brasil>>. Acesso em: 8/12/2015.

BNCC. Base Nacional Curricular Comum. , 2015. Disponível em: <<http://basenacionalcomum.mec.gov.br/>>. Acesso em: 20/1/2016.

BOCCONI, S.; CHIOCCARIELLO, A.; DETTORI, G.; FERRARI, A.; ENGELHARDT, K. **Developing Computational Thinking in Compulsory Education - Implications for policy and practice**. Luxembourg: European Union, 2016.

BOGLIOLO, A. **Coding in your classroom, now!** 1º ed. Firenze: Giunti, 2016.

BOUCINHA, R.; BRACKMANN, C.; BARONE, D. A. C.; CASALI, A. Construção do Pensamento Computacional através do Desenvolvimento de Games. **Revista Novas Tecnologias na Educação**, v. 15, n. 1, 2017. Disponível em: <<http://seer.ufrgs.br/index.php/renote/article/view/75146>>. .

BOULAY, B. DU; O'SHEA, T.; MONK, J. The black box inside the glass box: presenting computing concepts to novices. **International Journal of Man-Machine Studies**, v. 14, n. 3, p. 237–249, 1981.

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.

BRACKMANN, C.; ROMÁN-GONZÁLES, M.; ROBLES, G.; et al. Development of Computational Thinking Skills through Unplugged Activities in Primary School. WIPSCCE 2017. **Anais...** . p.7, 2017. Nijmegen, The Netherlands: ACM Press.

BRASIL/MEC. Base Nacional Curricular Comum. , 5. abr. 2017. Disponível em: <<http://basenacionalcomum.mec.gov.br/images/BNCCpublicacao.pdf>>. Acesso em: 9/7/2017.

BRENNAN, K. Scratch Curriculum Guide Draft. **ScratchEd**, nov. 2011. Disponível em: <<http://scratched.gse.harvard.edu/resources/scratch-curriculum-guide-draft>>. .

BRINDA, T.; PUHLMANN, H.; SCHULTE, C. **Bridging ICT and CS – Educational Standards for Computer Science in Lower Secondary Education**. New York, NY: ACM, 2009.

BROOKS, S. H.R.2536 - 113th Congress (2013-2014): Computer Science Education Act of 2013. Disponível em: <<https://www.congress.gov/bill/113th-congress/house-bill/2536>>. Acesso em: 28/11/2015.

BUNCE, S. **100 ideas for primary teachers**. 2015.

BUNDY, A. Computational Thinking is Pervasive. **Journal of Scientific and Practical Computing**, v. 1, p. 67–69, 2007.

BUREAU OF LABOR STATISTICS. Computer and Information Technology Occupations. Disponível em: <<https://www.bls.gov/ooh/computer-and-information-technology/home.htm>>.

BURRETT, M. **Teaching primary computing**. London Oxford New York New Delhi Sydney: Bloomsbury, 2016.

CAÇÃO, R. Testes Estatísticos: Testes Paramétricos e Não Paramétricos. , 11. mar. 2010. Disponível em: <<https://www.slideshare.net/rosariocacao/testes-parametricos-e-nao-parametricos-3396639>>. Acesso em: 21/6/2017.

CALDWELL, H.; SMITH, N. **Teaching computing unplugged in primary schools: exploring primary computing through practical activities away from the computer**. 2017.

CAMPOS, G.; CAVALHEIRO, S.; FOSS, L.; et al. Organização de Informações via Pensamento Computacional: Relato de Atividade Aplicada no Ensino Fundamental. **20o. Workshop de Informática na Escola (WIE 2014)**, Congresso Brasileiro de Informática na Educação - CBIE 2014., 2014.

CAS, C. AT S. Developing Computational Thinking. **Teaching London Computing**, 2014. Disponível em: <<http://teachinglondoncomputing.org/resources/developing-computational-thinking/>>. Acesso em: 1/1/2016.

CASEY, R. Computer Science Education and Jobs Act of 2013 (2013 - S. 1407). Disponível em: <<https://www.govtrack.us/congress/bills/113/s1407>>. Acesso em: 28/11/2015.

CASEY, R. S.671 - 114th Congress (2015-2016): Computer Science Education and Jobs Act of 2015. Disponível em: <<https://www.congress.gov/bill/114th-congress/senate-bill/671>>. Acesso em: 28/11/2015.

CASPERSEN, M. E.; FISLER, K.; VAHRENHOLD, J.; HERBSTTRITT, M. **Assessing Learning In Introductory Computer Science (Dagstuhl Seminar 16072)**. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany, 2016.

CFE ARGENTINA. **Resolución CFE N° 263/15**. 2015.

CHOI, J.; AN, S.; LEE, Y. Computing Education in Korea - Current Issues and Endeavors. **ACM Transactions on Computing Education**, v. 15, n. 2, p. 1–22, 2015.

CITYOFNY. An Open Letter from the Nation’s Tech and Business Leaders. Disponível em: <<https://medium.com/cityofny/an-open-letter-from-the-nation-s-tech-and-business-leaders-ca9894a4d125>>.

CLEMENTS, D. H. Computers in Early Childhood Mathematics. **Contemporary issues in early childhood**, v. 3, n. 2, p. 160–181, 2002.

CODE.ORG. **Instructor Handbook - Code Studio Lesson Plans for Courses One, Two, and Three**. CODE.ORG, 2015.

CODE.ORG. Where computer science counts. Disponível em: <<https://code.org/action>>. Acesso em: 4/12/2015.

CONSTANTINO, A. A. Programação x Codificação. Engenharia de Algoritmo e Software de Automação, 25. jul. 2009. Disponível em: <<http://engenhariadealgoritmo.blogspot.com.br/2009/07/programacao-x-codificacao.html>>. Acesso em: 1/1/2016. CSIZMADIA, A.; CURZON, P.; DORLING, M.; et al. Computational

thinking - A guide for teachers. , 2015. Computing At School (CAS). Disponível em: <<http://community.computingatschool.org.uk/files/6695/original.pdf>>. Acesso em: 1/1/2016.

CSTA. K-12 Computer Science Standards - Revised 2011 - The CSTA Standards Task Force. Association for Computing Machinery, 2011.

CSTA/ISTE. Computational Thinking Across the Curriculum. , 2009. Disponível em: <<http://www.csta.acm.org/Curriculum/sub/CurrFiles/CTExamplesTable.pdf>>. Acesso em: 1/1/2016.

CSTA/ISTEA. Computational Thinking: leadership toolkit. , 2011. Disponível em: <<https://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershipToolkit-SP-vF.pdf>>. .

CSTA/ISTEB. Operational Definition of Computational Thinking for K–12 Education. Disponível em: <<http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf>>. .

CUNY, J. Computer Science Education for Everyone: A Groundswell of Support. Disponível em: <<http://www.infosys.org/infosys-foundation-usa/media/blog/Pages/groundswell-support.aspx>>. .

CURZON, P. cs4fn and computational thinking unplugged. . p.47–50, 2013. ACM Press. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2532748.2611263>>. Acesso em: 20/5/2017.

CURZON, P.; MCOWAN, P. W.; PLANT, N.; MEAGHER, L. R. Introducing teachers to computational thinking using unplugged storytelling. . p.89–92, 2014. ACM Press. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2670757.2670767>>. Acesso em: 20/5/2017.

DAVIS, J. Australia Forgets that Code is Cultural: Replaces History and Geography with Computer Science. , 8. out. 2015. Disponível em: <<https://thesocietypages.org/cyborgology/2015/10/08/australia-forgets-that-code-is-cultural-replaces-history-and-geography-with-computer-science/>>. Acesso em: 6/6/2017.

DENNING, P. J. The profession of IT Beyond computational thinking. **Communications of the ACM**, v. 52, n. 6, p. 28, 2009.

DERUY, E. In Finland, Kids Learn Computer Science Without Computers. , 24. fev. 2017. Disponível em: <<https://www.theatlantic.com/education/archive/2017/02/teaching-computer-science-without-computers/517548/>>. Acesso em: 10/3/2017.

DESJARDINS, J. The Largest Companies by Market Cap Over 15 Years. Disponível em: <<http://www.visualcapitalist.com/chart-largest-companies-market-cap-15-years/>>. .

DICKINS, R.; NIELSEN, S. **Computers and coding: with over 100 flaps to lift.** London: Usborne Publishing, 2015.

DORAN, K.; BOYCE, A.; FINKELSTEIN, S.; BARNES, T. Outreach for improved student performance: a game design and development curriculum. . p.209, 2012. ACM Press. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2325296.2325348>>. Acesso em: 4/12/2015.

DÓRIA, J. Prefeito João Dória. , 10. mar. 2017. Disponível em: <<https://www.facebook.com/jdoriajr/videos/1371324489591025/>>. Acesso em: 8/7/2017.

DOWNEY, A. Programming as a Way of Thinking. **Scientific American**, 26. abr. 2017. Disponível em: <https://blogs.scientificamerican.com/guest-blog/programming-as-a-way-of-thinking/?WT.mc_id=SA_WR_20170503>. .

ELLIS, J.; CHARLOTTE, M. Giggle Chips. , 2017. Disponível em: <<http://gigglechips.bigcartel.com/>>. .

EMEC. e-MEC de Instituições e Cursos de Educação Superior. Disponível em: <<http://emec.mec.gov.br/>>. Acesso em: 20/6/2017.

ENGELBERG, M.; THINKFUN. Chocolate Fix. , 2012. Thinkfun. Disponível em: <<http://www.thinkfun.com/products/chocolate-fix/>>. .

ENGELBERG, M.; THINKFUN. Code Master Programming Logic Game. , 2015. Thinkfun. Disponível em: <<http://www.thinkfun.com/products/code-master/>>. .

EPA. Einheitliche Prüfungsanforderungen – Informatik. , 2004. Disponível em: <http://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01-EPA-Informatik.pdf>. Acesso em: 10/11/2015.

FABER, H.; WIERDSMA, M.; DOORNBOS, R.; VEN, J. S. VAN DER; VETTE, K. DE. Teaching Computational Thinking to Primary School Students via Unplugged Programming Lessons. **Journal of the European Teacher Education Network**, v. 12, 2017.

FARAG, B. Please don't learn to code. Disponível em: <<https://techcrunch.com/2016/05/10/please-dont-learn-to-code/>>. Acesso em: 22/6/2017.

FARIAS, A.; ANDRADE, W.; ALENCAR, R. Pensamento Computacional em Sala de Aula: Desafios, Possibilidades e a Formação Docente. . p.1226, 2015. Disponível em: <<http://br-ie.org/pub/index.php/wcbie/article/view/6262>>. Acesso em: 29/7/2017.

FEASTER, Y.; SEGARS, L.; WAHBA, S. K.; HALLSTROM, J. O. Teaching CS unplugged in the high school (with limited success). . p.248, 2011. ACM Press. Disponível em: <<http://portal.acm.org/citation.cfm?doid=1999747.1999817>>. Acesso em: 20/5/2017.

FLEURY, A.; NEVEUX, C. Hamon : “Le code informatique à l'école dès septembre”. Disponível em: <<http://www.lejdd.fr/Societe/Hamon-Le-code-informatique-a-l-ecole-des-septembre-675912>>. Acesso em: 28/11/2015.

FRANÇA, R. S. DE; SILVA, W. C. DA; AMARAL, H. J. C. DO. Despertando o interesse pela ciência da computação: Práticas na educação básica. VIII International Conference on Engineering and Computer Education. **Anais...** . p.282–286, 2013.

FRANKLIN, D.; SKIFSTAD, G.; ROLOCK, R.; et al. Using Upper-Elementary Student Performance to Understand Conceptual Sequencing in a Blocks-based Curriculum. . p.231–236, 2017. ACM Press. Disponível em: <<http://dl.acm.org/citation.cfm?doid=3017680.3017760>>. Acesso em: 16/4/2017.

FURBER, S. **Shut down or restart? - The way forward for computing in UK schools**. London, England: The Royal Society, 2012.

GALLUP-GOOGLE. **Searching for Computer Science: Access and Barriers in U.S. K-12 Education**. 2016.

GAMES4LEARNING. Bits & Bites. , 2014. Imago. Disponível em: <<http://bitsandbytes.cards/>>. .

GARFIELD, R.; HASBRO GAMING. Robot Rally. , 2015. Hasbro. Disponível em: <<https://www.hasbro.com/en-us/product/richard-garfield-s-robo-rally-avalon-hill-game:6A791573-5056-9047-F529-BF676C121371>>. Acesso em: 27/6/2017.

GERHARDT, T.; SILVEIRA, D. **Métodos de pesquisa**. 1º ed. Editora da UFRGS, 2009.

GIMPS. Great Internet Mersenne Prime Search. , 7. jan. 2016. Disponível em: <<http://www.mersenne.org/>>. .

GOOGLE FOR EDUCATION. What is Computational Thinking? **Computational Thinking for Educators**, 2015. Disponível em: <<https://computationalthinkingcourse.withgoogle.com/unit?lesson=8&unit=1>>. Acesso em: 30/10/2015.

GREENHILL, A.; SLEE, N. Littlecodr. , 2015. Littlecodr Games Inc. Disponível em: <<http://littlecodr.com/>>. .

GROFF, R.; LOPES, S. Déficit de Profissionais de TI chega a 100 mil. **Correio do Povo**, p. 12, 18. jul. 2016. Porto Alegre, RS, Brasil.

GROVER, S.; BASU, S. Measuring Student Learning in Introductory Block-Based Programming: Examining Misconceptions of Loops, Variables, and Boolean Logic. . p.267–272, 2017. ACM Press. Disponível em: <<http://dl.acm.org/citation.cfm?doid=3017680.3017723>>. Acesso em: 12/7/2017.

GROVER, S.; PEA, R. Computational Thinking in K-12: A Review of the State of the Field. **Educational Researcher**, v. 42, n. 1, p. 38–43, 2013.

GUZDIAL, M. Why the U.S. is not ready for mandatory CS education. **Communications of the ACM**, v. 57, n. 8, p. 8–9, 2014.

GUZDIAL, M. **Learner-Centered Design of Computing Education - Research on Computing for Everyone**. Morgan & Claypool, 2016.

GUZDIAL, M.; ERICSON, B. J.; MCKLIN, T.; ENGELMAN, S. A statewide survey on computing education pathways and influences: factors in broadening participation in computing. . p.143, 2012. ACM Press. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2361276.2361304>>. Acesso em: 24/1/2016.

HAMMER, Ø.; HARPER, D. A. T.; RYAN, P. D. Past: Paleontological Statistics Software Package for Education and Data Analysis. **Palaeontologia Electronica**, v. 4, n. 1, p. 9, 2001.

HOOPER, L.; THINKFUN. Laser Maze. , 2012. Thinkfun. Disponível em: <<http://www.thinkfun.com/products/laser-maze/>>. .

ISAÍAS, M. Las maestras de primaria invitadas a formarse en programación. **Jornal “La Capital”**, 24. jun. 2017. Argentina. Disponível em: <<http://www.lacapital.com.ar/educacion/las-maestras-primaria-invitadas-formarse-programacion-n1421812.html>>. Acesso em: 24/6/2017.

ISTE. ISTE Standards for Students. Disponível em: <<http://www.iste.org/standards/standards/for-students#startstandards>>. .

JOHNSON, P. France to offer programming in elementary school. Disponível em: <<http://www.itworld.com/article/2696639/application-management/france-to-offer-programming-in-elementary-school.html>>. Acesso em: 27/11/2015.

JONES, S. P. International comparisons. , 2011. Microsoft / Computing at School. Disponível em: <<http://www.computingschool.org.uk/index.php?id=documents>>. Acesso em: 23/2/2015.

KALELIOGLU, F.; GÜLBAHAR, Y.; KUKUL, V. A Framework for Computational Thinking Based on a Systematic Research Review. **Baltic Journal of Modern Computing**, v. 4, n. 3, p. 583–596, 2016.

KNOTT, R. Fibonacci Numbers and Nature. Disponível em: <<http://www.maths.surrey.ac.uk/hosted-sites/R.Knott/Fibonacci/fibnat.html>>. .

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.

KOODIAAPINEN. Teacher's Guide to encode the school. Disponível em: <<http://koodiaapinen.fi/en/>>. Acesso em: 5/7/2017.

KURSHAN, B. Thawing from a Long Winter in Computer Science Education. **Forbes**, p. 2, fev. 2016.

KURZWEIL, R. **The age of spiritual machines: when computers exceed human intelligence**. New York: Viking, 1999.

LAMBERT, L.; GUIFFRE, H. Computer science outreach in an elementary school. **Journal of Computing Sciences in Colleges**, v. 24, n. 3, p. 118–124, 2009.

LIUKAS, L. **Hello Ruby: adventures in coding**. Feiwei & Friends, 2015.

LYE, S. Y.; KOH, J. H. L. Review on teaching and learning of computational thinking through programming: What is next for K-12? **Computers in Human Behavior**, v. 41, p. 51–61, 2014.

MANSEL, T. La receta de Estonia para convertirse en una potencia tecnológica. , 15. maio 2013. Disponível em: <http://www.bbc.com/mundo/noticias/2013/05/130515_tecnologia_cuna_ninos_informatica_estonia_aa>. Acesso em: 7/5/2017.

MATHANDCODING. Coding Farmers. , 2015. MathAndCoding. Disponível em: <<http://codingfarmers.net/>>. .

MATHANDCODING. CodingIsGood. , 2017. MathAndCoding. Disponível em: <<http://www.mathandcoding.org/>>. .

MCLUHAN, M. **Understanding media: the extensions of man**. 1st MIT Press ed ed. Cambridge, Mass: MIT Press, 1994.

MCMMASTER, K.; RAGUE, B.; ANDERSON, N. Integrating Mathematical Thinking, Abstract Thinking, and Computational Thinking. **40th ASEE/IEEE Frontiers in Education Conference**, out. 2010. Acesso em: 21/12/2015.

MEC. Base Nacional Comum Curricular - Estudo Comparativo entre a Versão 2 e a Versão Final. , 12. abr. 2017. Disponível em: <http://basenacionalcomum.mec.gov.br/images/BNCC_Comparativo.pdf>. Acesso em: 8/7/2017.

MEC/INEP. Anuário Brasileiro da Educação Básica / Todos pela Educação. , maio 2017. Editora Moderna. Disponível em: <<https://www.todospelaeducacao.org.br/biblioteca/1545/anuario-brasileiro-da-educacao-basica-2016/>>. .

MEIRELLES, L. Escolas do Município podem ter aulas de Ciência da Computação. , 14. out. 2016. Prefeitura Municipal de Pelotas, . Disponível em: <<http://www.pelotas.rs.gov.br/noticias/detalhe.php?controle=MjAxNi0xMC0xNA%3D%3D&codnoticia=43147>>. Acesso em: 8/7/2017.

MELLO, P. M.; MELLO, D. A. Jogos Boole. Disponível em: <<http://www.jogosboole.com.br/>>. Acesso em: 9/1/2017.

MINISTERIO DE EDUCACIÓN, CULTURA Y DEPORTE. Educación Primaria. Disponível em: <<https://www.mecd.gob.es/educacion-mecd/areas-educacion/estudiantes/educacion-primaria.html>>. Acesso em: 1/1/2017.

MORENO-LEÓN, J.; ROBLES, G.; ROMÁN-GONZÁLES, M. Programar para aprender en Educación Primaria y Secundaria: ¿qué indica la evidencia empírica sobre este enfoque? **Revista de Investigación en Docencia Universitaria de la Informática**, v. 10, n. 2, p. 45–51, 2017.

MUFFET, T. Computer coding taught in Estonian primary schools. , 8. jan. 2014. Estonia. Disponível em: <<http://www.bbc.com/news/av/education-25648769/computer-coding-taught-in-estonian-primary-schools>>. Acesso em: 5/7/2017.

MYKKÄNEN, J.; LIUKAS, L. **Koodi 2016**. 1º ed. Finlândia: Lönnberg Print, 2014.

NATIONAL RESEARCH COUNCIL (U.S.) (ORG.). **Report of a workshop of pedagogical aspects of computational thinking**. Washington, D.C: National Academies Press, 2011.

NATIONAL RESEARCH COUNCIL (U.S.); COMMITTEE FOR THE WORKSHOPS ON COMPUTATIONAL THINKING. **Report of a workshop on the scope and nature of computational thinking**. Washington, D.C.: National Academies Press, 2010.

NAVAUX, P. O. A.; CÁCERES, E. N.; ZORZO, A. F. **Documento de Área: Ciência da Computação**. CAPES, 2016.

NISHIDA, T.; KANEMUNE, S.; IDOSAKA, Y.; et al. A CS Unplugged Design Pattern. . p.231–235, 2009. New York, NY: ACM. Disponível em: <<http://dx.doi.org/10.1145/1508865>>. Acesso em: 18/5/2017.

NORMANDO, D.; TJÄDERHANE, L.; QUINTÃO, C. C. A. A escolha do teste estatístico – um tutorial em forma de apresentação em PowerPoint. **Dental Press J. Orthod**, v. 15, n. 1, p. 101–106, 2010.

NOSCHANG, L. F.; PELZ, F.; JESUS, E.; RAABE, A. Portugol Studio: Uma IDE para Iniciantes em Programação. , 2014. Brasília, DF.

NUNES, D. J. Ciência da Computação na Educação Básica. **ADUFRGS - Sindical**, 6. jun. 2011. Disponível em: <<http://www.adufrgs.org.br/artigos/ciencia-da-computacao-na-educacao-basica/>>. Acesso em: 23/11/2015.

OECD. **Students Computers and Learning**. OECD, 2015.

OECD. PISA (Programme for International Student Assessment). Disponível em: <<http://www.oecd.org/pisa/>>. Acesso em: 9/1/2017.

P21. Partnership for 21st Century Learning. [S.l.], [s.d.]. Disponível em: <<http://www.p21.org/>>. Acesso em: 24 nov. 2015.

PAPERT, S. **Mindstorms: Children, Computers, And Powerful Ideas**. Basic Books, 1980.

PAPERT, S.; HAREL, I. **Constructionism: research reports and essays, 1985-1990**. Norwood, N.J: Ablex Pub. Corp, 1991.

PAPERT, S.; SOLOMON, C. Twenty things to do with a Computer. **Educational Technology Magazine**, 1972. Disponível em: <<http://www.stager.org/articles/twentythings.pdf>>. .

PAPPANO, L. Learning to Think Like a Computer. **New York Times**, 4. abr. 2017. Disponível em: <<https://www.nytimes.com/2017/04/04/education/edlife/teaching-students-computer-code.html>>. .

PATTERSON, S. **Programming in the primary grades: beyond the hour of code**. Lanham, Maryland: Rowman & Littlefield, 2016.

PCAST, P. C. OF A. ON S. AND T. Report To The President and Congress - Designing A Digital Future: Federally Funded Research and Development in Networking and Information Technology. , dez. 2010. Disponível em: <<https://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast-nitrd-report-2010.pdf>>.

PISA. PISA 2015 COLLABORATIVE PROBLEM SOLVING FRAMEWORK. , abr. 2017. Disponível em: <<https://www.oecd.org/pisa/pisaproducts/Draft%20PISA%202015%20Collaborative%20Problem%20Solving%20Framework%20.pdf>>. .

RAMOS, J. L.; ESPADEIRO, R. G. Os Futuros Professores e os professores do futuro. Os desafios da introdução ao Pensamento Computacional na escola, no currículo e na aprendizagem. **Revista Educação, Formação & Tecnologias**, v. 7, p. 4–25, 2014.

RESNICK, M. **Mitch Resnick: Vamos ensinar as crianças a programar**. 2012.

RIBEIRO, L.; NUNES, D. J.; CRUZ, M. K. DA; MATOS, E. DE S. Computational Thinking: Possibilities and Challenges. 2nd Workshop-School on Theoretical Computer Science. **Anais...** , 2013.

RILEY, D. D.; HUNT, K. A. **Computational thinking for the modern problem solver**. Boca Raton, FL: CRC Press, 2014.

RODRIGUES, F.; BRACKMANN, C. P.; BARONE, D. A. C. Estudo da Evasão no Curso de Ciência da Computação da UFRGS. **Revista Brasileira de Informática na Educação**, v. 23, n. 01, p. 97, 2015.

RODRIGUEZ, B. **Assessing Computational Thinking in Computer Science Unplugged Activities**. Master Dissertation, Golden, Colorado, EUA: Colorado School of Mines, 2015.

RODRIGUEZ, B.; RADER, C.; CAMP, T. Using Student Performance to Assess CS Unplugged Activities in a Classroom Environment. . p.95–100, 2016. ACM Press. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2899415.2899465>>. Acesso em: 6/7/2017.

RODRIGUEZ, B.; STEPHEN, K.; RADER, C.; CAMP, T. Assessing Computational Thinking in CS Unplugged Activities. Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. **Anais...** . p.501–506, 2017. Seattle, Washington, USA: ACM.

ROMÁN-GONZÁLES, M.; MORENO-LEÓN, J.; ROBLES, G. Complementary Tools for Computational Thinking Assessment. , 2017. Hong Kong: In press.

ROMÁN-GONZÁLES, M.; PÉREZ, J. C.; ROBLES, G.; MORENO-LEÓN, J. Full validation of the Computational Thinking Test (CTt). , 2017.

ROMÁN-GONZÁLES, M.; PÉREZ-GONZÁLES, J.-C.; JIMÉNEZ-FERNÁNDEZ, C. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. **Computers in Human Behavior**, v. 72, p. 678–691, 2017.

ROMÁN-GONZÁLES, M. Computational Thinking Test: Design Guidelines and Content Validation. Proceedings of the 7th Annual International Conference on Education and New Learning Technologies (EDULEARN 2015). **Anais...** . p.2436–2444, 2015. Barcelona, Spain: IATED.

ROMÁN-GONZÁLES, M. **Códigoalfabetización y Pensamiento Computacional en Educación Primaria y Secundaria: Validación de un Instrumento y Evaluación de Programas**, fev. 2016. Madrid, Spain: Universidad Nacional de Educación a Distancia.

ROMÁN-GONZÁLES, M.; PÉREZ, J. C.; CARMEN JIMÉNEZ-FERNÁNDEZ. Test de Pensamiento Computacional: diseño y psicometría general. , 2015. Disponível em: <<https://doi.org/10.13140/RG.2.1.3056.5521>>. Acesso em: 16/4/2017.

ROSSI, B. Python overtakes French as the most popular language taught in primary schools | Information Age. Disponível em: <<http://www.information-age.com/it-management/skills-training-and-leadership/123460073/python-overtakes-french-most-popular-language-taught-primary-schools>>. Acesso em: 24/11/2015.

RYAN, P. **The Black River emerald**. New York: Berkley Books, 1987.

RYAN, P. **A Esmeralda do Rio Negro**. New York, EUA: Ediouro, 1994.

SADOSKY, F. Una propuesta para refundar la enseñanza de la computación en las escuelas Argentinas. , jan. 2013. Disponível em: <<http://www.fundacionsadosky.org.ar/wp-content/uploads/2014/06/cc-2016.pdf>>. Acesso em: 10/6/2015.

SANKARAN, R. Burning questions. **US Department of Energy**, abr. 2007. Disponível em: <<http://ascr-discovery.science.doe.gov/2007/04/burning-questions/>>. Acesso em: 27/7/2017.

SANT'ANNA, I. M. **Recursos de Ensino e Oficina Pedagógica - Processos Didáticos na Construção do Conhecimento**. 1º ed. Porto Alegre, RS, Brasil: Edições Caravela, 1995.

SCAICO, P.; MYCHELLINE, H.; CUNHA, F.; ALENCAR, Y. Um Relato de Experiências de Estagiários da Licenciatura em Computação com o Ensino de Computação para Crianças. **Revista Novas Tecnologias na Educação (RENOTE)**, v. 10, n. 3, 2012.

SCHIFF, R.; VAKIL, E. Age differences in cognitive skill learning, retention and transfer: The case of the Tower of Hanoi Puzzle. **Learning and Individual Differences**, v. 39, p. 164–171, 2015.

SCHMUNDT, H. Should IT Classes Be Required? **Spiegel**, 16. maio 2013. Disponível em: <<http://www.spiegel.de/international/germany/experts-in-germany-divided-on-computer-science-in-school-curriculum-a-899979.html>>. Acesso em: 6/7/2017.

SEITER, L.; FOREMAN, B. Modeling the learning progressions of computational thinking of primary grade students. . p.59, 2013. ACM Press. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2493394.2493403>>. Acesso em: 21/1/2016.

- SENTANCE, S.; CSIZMADIA, A. Teachers' perspectives on successful strategies for teaching Computing in school. IFIP TCS 2015. **Anais...**, 2015. Disponível em: <<http://community.computingatschool.org.uk/files/6769/original.pdf>>. .
- SETZER, V. W. An antidote to computer-thinking. Netfuture - Technology and Human Responsibility for the Future, v. n/a, n. 22, 1996. Disponível em: <http://www.netfuture.org/1996/Jun2096_22.html>. .
- SETZER, V. W. Roda Viva Com Professor Valdemar Setzer. , 1. dez. 2008. Canal Cultura. Disponível em: <https://www.youtube.com/watch?v=tNLb4Sw_vkk&t=12s>. .
- SHADISH, W. R.; COOK, T. D.; CAMPBELL, D. T. **Experimental and quasi-experimental designs for generalized causal inference**. 2nd ed. Boston: Houghton Mifflin, 2001.
- SHAPIRO, D.; THINKFUN. Robot Turtles. , 2014. Disponível em: <<http://www.robotturtles.com/>>. .
- SHAPIRO, S. S.; WILK, M. B. An Analysis of Variance Test for Normality (Complete Samples). **Biometrika**, v. 52, n. 3/4, p. 591–611, 1965.
- SIDHU, R. Code Monkey Island. , 2014. Code Monkey Games, LLC. Disponível em: <<https://www.kickstarter.com/projects/rajsidhu/code-monkey-island-making-programming-childs-play>>. .
- SILVA, C. Prefeito de São Paulo quer aulas de programação nas escolas. , 10. mar. 2017. Disponível em: <<http://codigofonte.uol.com.br/noticias/prefeito-de-sao-paulo-quer-aulas-de-programacao-nas-escolas>>. .
- SOFTEX. Relatório Anual SOFTEX. , 2015. Disponível em: <http://www.softex.br/wp-content/uploads/2015/04/Relatorio_Anuar_2014.pdf>. Acesso em: 23/11/2015.
- SOUZA, P. N. P. DE; SILVA, E. B. DA. **Como entender e aplicar a nova LDB: LEI no. 9.394/96**. Sao Paulo, Brazil: Cengage Learning, 1997.
- TAKAHASHI, M.; AZUMA, S.; TREND-PRO CO., LTD. **Guia Mangá de Banco de Dados**. 1º ed. Tokyo, Japão: Novatec, 2009.
- TAUB, R.; ARMONI, M.; BEN-ARI, M. CS Unplugged and Middle-School Students? Views, Attitudes, and Intentions Regarding CS. **ACM Transactions on Computing Education**, v. 12, n. 2, p. 1–29, 2012.
- TAUB, R.; BEN-ARI, M.; ARMONI, M. The effect of CS unplugged on middle-school students' views of CS. . p.99, 2009. ACM Press. Disponível em: <<http://portal.acm.org/citation.cfm?doid=1562877.1562912>>. Acesso em: 20/5/2017.
- TEW, A. E.; DORN, B.; LEAHY, W. D.; GUZDIAL, M. Context as Support for Learning Computer Organization. **Journal on Educational Resources in Computing**, v. 8, n. 3, p. 1–18, 2008.
- THE WHITE HOUSE. President Obama Announces Computer Science For All Initiative. Disponível em: <<https://obamawhitehouse.archives.gov/the-press-office/2016/01/30/fact-sheet-president-obama-announces-computer-science-all-initiative-0>>. Acesso em: 7/5/2017.
- UNNIKRISHNAN, R.; AMRITA, N.; MUIR, A.; RAO, B. Of Elephants and Nested Loops: How to Introduce Computing to Youth in Rural India. . p.137–146, 2016. ACM Press.

Disponível em: <<http://dl.acm.org/citation.cfm?doid=2930674.2930678>>. Acesso em: 1/5/2017.

VALENTE, J. A. Integração do Pensamento Computacional no Currículo da Educação Básica: Diferentes Estratégias Usadas e Questões de Formação de Professores e Avaliação do Aluno. **Revista e-Curriculum**, v. 14, n. 3, 2016.

VIEIRA, A.; PASSOS, O.; BARRETO, R. Um Relato de Experiência do Uso da Técnica Computação Desplugada. Anais do XXI WEI. **Anais...**, 2013. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2013/0031.pdf>>. .

VIEL, F.; RAABE, A.; ZEFERINO, C. Introdução à Programação e à Implementação de Processadores por Estudantes do Ensino Médio. WORKSHOP DE INFORMÁTICA NA ESCOLA (WIE). **Anais...** . v. 20, 2014.

WANG, P. S. **From computing to computational thinking**. Boca Raton, FL: CRC Press, Taylor & Francis Group, 2016.

WEINBERG, M. Voando para o Futuro. **Veja**, 91., , n. 2431, 2015. Disponível em: <<http://veja.abril.com.br/acervo/home.aspx>>. Acesso em: 30/6/2015.

WING, J. Computational Thinking. , 2007. Carnegie Mellon University. Disponível em: <http://www.cs.cmu.edu/afs/cs/usr/wing/www/Computational_Thinking.pdf>. Acesso em: 24/11/2015.

WING, J. **Computational Thinking with Jeannette Wing**. Columbia Journalism School, 2014.

WING, J. M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33, 2006.

WING, J. M. Computational thinking and thinking about computing. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 366, n. 1881, p. 3717–3725, 2008.

WING, J. M. Computational Thinking: What and Why? , 17. out. 2010. Disponível em: <<http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>>. Acesso em: 23/11/2015.

WING, J. M. Computational Thinking Benefits Society. **Social Issues in Computing**, 2014. Disponível em: <<http://socialissues.cs.toronto.edu/2014/01/computational-thinking/>>. Acesso em: 24/11/2015.

WING, J. M. Computational Thinking for Everyone. . Apresentação, . Acesso em: 18/12/2015.

YAKOS, D.; THINKFUN. Circuit Maze. , 2015. Thinkfun. Disponível em: <<http://www.thinkfun.com/products/circuit-maze/>>. .

OBRAS CONSULTADAS

FURASTÉ, Pedro Augusto. Normas Técnicas para o Trabalho Científico: Explicitação das Normas da ABNT. 17ª ed. Porto Alegre: Dáctilo Plus, 2015.

ANEXOS

Anexo A: ENSINO DO PENSAMENTO COMPUTACIONAL NO ENSINO BÁSICO

O objetivo desta proposta é iniciar um debate na comunidade científica e educacional sobre a necessidade de mudanças fundamentais na Educação Básica sobre vários temas relacionados a Computação (não confundir com Tecnologias da Informação e Comunicação), convencido de que eles são elementos-chave para o nosso país poder aproveitar as enormes oportunidades oferecidas por estes temas em tempos modernos.

CONSIDERANDO QUE:

- 1) vivemos em tempos em que a criatividade do homem faz a diferença; em que a nova economia mundial não se baseia apenas em recursos naturais e matérias-primas, mas em conhecimento, fluxos de informação e as habilidades em usá-los, sem esquecer, claro, que a situação é ideal para países como o Brasil, com o potencial de combinar os dois fatores acima mencionados;
- 2) a Ciência da Computação oferece não apenas softwares úteis e artefatos de hardware, mas também uma maneira diferenciada de pensar; e que todos, independentemente da área, podem se beneficiar ao pensar computacionalmente e descobrir novas ciências através da análise de uma quantidade gigantesca de dados ou fazer questionamentos que nunca foram cogitados ou ousados devido a sua escalabilidade, facilmente atendida pela Computação;
- 3) os jovens têm muita experiência e considerada familiaridade na interação com novas tecnologias, mas têm pouca experiência em criá-las e expressarem-se com elas; é quase como se conseguissem ler, porém não conseguissem escrever com essas novas tecnologias. Quando aprendemos a ler, podemos então ler para aprender. É o que ocorre com a solução de uma maneira sistematizada (algorítmica), procedimento também chamado de Codificação. Se aprendemos a codificar, conseqüentemente podemos codificar para aprender, fato esse que propicia uma oportunidade de se aprender uma gama de muitas outras coisas. Vale lembrar que saber codificar será tão importante quanto saber ler no próximo século;
- 4) muitas pessoas estão fascinadas por novas tecnologias, com as quais lidamos diariamente, e o potencial do estudo da Ciência da Computação ainda não é claro para a maioria que faz apenas o consumo de aparelhos e aplicativos computacionais;
- 5) a adoção de noções de Computação em escolas de Educação Básica é uma preocupação em diversos países, onde a implantação ocorre de forma rigorosa e possui benefícios educacionais (habilidades de reflexão e solução de problemas, compreensão que o mundo está impregnado com a tecnologia digital) e econômicos (alta demanda de profissionais com boa formação);
- 6) o ensino dos Fundamentos da Computação no Ensino Básico (Pensamento Computacional) beneficia o desenvolvimento de habilidades e competências essenciais para a vida moderna, independente da área em que atuará, vale salientar que esta proposta coaduna

com ações em diversos países como Alemanha, Argentina, Austrália, Coréia do Sul, Escócia, França, Estados Unidos da América, Finlândia, Grécia, Índia, Israel, Japão, Nova Zelândia, Reino Unido e outros, os quais já possuem disciplinas de Computação em seu currículo (vide Anexo A);

7) existe forte evidência científica que comprova que crianças e adolescentes que aprendem a programar, melhoram seu desempenho em outras áreas disciplinares, entre elas a Matemática e Línguas Estrangeiras (CLEMENTS, 2002) (DORAN, 2012) (PAPERT, 1980) (WING, 2014);

8) as habilidades trabalhadas na Computação podem ser usadas em diversas áreas e são conhecimentos e técnicas importantes para aumentar as chances de acelerar o desenvolvimento do país, mantendo sua competitividade, apoiar a descoberta científica em outras áreas e potencializar suas capacidades de inovar e criar novas tecnologias;

9) a programação é parte de uma disciplina acadêmica mais ampla da Computação que inclui os saberes necessários para poder formular soluções efetivas e sistemáticas de diversos tipos de problemas, não havendo obrigatoriedade de utilização de computadores ou equipamentos específicos;

10) o setor de Tecnologia da Informação (TI) está constantemente em busca de profissionais qualificados, onde pesquisas apontam um déficit atual de 40 mil profissionais, salientando que, segundo a Associação Brasileira de Empresas de Tecnologia da Informação e Comunicação (Brasscom), o Brasil é o 5º maior centro de TI do mundo, com um saldo negativo estimado de 408 mil profissionais até o ano de 2022.

PROPÕE-SE:

1) Introduzir na Base Nacional Comum Curricular (BNCC) os Conceitos Fundamentais da Ciência da Computação a todos os estudantes do Ensino Fundamental de forma obrigatória, assim como as disciplinas de Matemática, Ciência, etc. A relação das competências, de acordo com cada ano escolar, encontra-se nos Anexos C e D;

2) Permitir que estudantes do Ensino Médio possam, de forma facultativa, aprofundar seus estudos nos fundamentos da Ciência da Computação com o objetivo de prepará-los para o mercado de trabalho ou ensino superior.

09 de dezembro de 2015.

Colaboraram na elaboração desta análise os seguintes membros da Sociedade Brasileira de Computação: Christian Puhlmann Brackmann (Instituto Federal Farroupilha/UFRGS), Avelino Francisco Zorzo (PUCRS)

Anexo B: PROPOSTA DAS COMPETÊNCIAS POR ANO ESCOLAR (COMPUTAÇÃO)

NÍVEL	ANO	COMPUTAÇÃO
ENSINO FUNDAMENTAL	1	<ul style="list-style-type: none"> Entender o que é um algoritmo e ser capaz de expressar linearmente simples (sem ramificações) algoritmos simbólicos; Entender que os computadores precisam de instruções precisas e demonstrar cuidado e precisão para evitar erros; Saber que as pessoas podem desenvolver seus próprios programas; Executar, verificações e mudar algoritmos simples; Entender que os algoritmos são executados, seguindo instruções precisas; Entender que os computadores não possuem inteligência e que não podem fazer nada a menos que um programa seja executado; Reconhecer que todos os softwares executados em dispositivos digitais são algoritmos.
	2	<ul style="list-style-type: none"> Entender que os aplicativos são implementados em dispositivos digitais através de algoritmos; Projetar algoritmos simples usando repetições e condições; Usar o raciocínio lógico para prever resultados; Detectar e corrigir erros, ou seja, depuração de algoritmos e semânticas; Usar o raciocínio lógico para prever o comportamento dos algoritmos; Reconhecer que uma variedade de dispositivos digitais pode ser considerado um computador; Reconhecer e poder usar uma variedade de dispositivos de entrada e saída.
	3	<ul style="list-style-type: none"> Projetar soluções (algoritmos) que usem repetição pré-testada, repetição pós-testada e seleção simples; Usar diagramas para expressar soluções; Usar o raciocínio lógico para prever saídas, informadas as entradas; Criar algoritmos simples para alcançar determinadas metas; Saber que os computadores coletam dados de diversos dispositivos de entrada, incluindo sensores e software de aplicação; Compreender a diferença entre hardware e software, e sua função em computadores; Entender o que é a internet, a rede mundial de computadores.
	4	<ul style="list-style-type: none"> Demonstrar a capacidade de identificar quais tarefas são melhor concluídas por seres humanos ou computadores; Projetar soluções através da decomposição de um problema maior e criar uma sub-solução para cada uma destas partes; Reconhecer que existem soluções diferentes para o mesmo problema; Projetar e depurar algoritmos modulares (procedimentos) e saber que um procedimento pode ser usado para ocultar detalhes de uma sub-solução; Entender por que e em que situações os computadores são utilizados; Compreender as principais funções do sistema operacional; Entender como usar eficazmente os motores de busca, e saber como resultados de pesquisa são selecionados.
	5	<ul style="list-style-type: none"> Entender que programas preenchem a lacuna entre soluções algorítmicas e computacionais; Reconhecer que existem diferentes algoritmos para a resolver o mesmo problema; Representar soluções usando uma notação estruturada (pseudocódigo); Poder identificar semelhanças e diferenças em situações e poder usá-las para resolver problemas (reconhecimento de padrões); Diferenciar e selecionar os tipos de dados apropriados (texto e números); Definir novos tipos de dados: números reais e booleano; Usar o operador booleano e aplicá-lo no contexto de controle do programa; Saber que os computadores são digitais e utiliza binários para representar todos os dados, inclusive números, imagens e durante a transferência entre outros computadores; Reconhecer e compreender a função das principais peças internas do computador; Compreender como motores de busca classificam os resultados da pesquisa;

	6	<ul style="list-style-type: none"> • Entender que em uma solução recursiva para um problema, pode-se aplicar repetidamente a mesma solução para instâncias menores do problema; • Reconhecer alguns problemas que compartilham as mesmas características e utilizar o mesmo algoritmo para resolver ambos (generalização); • Compreender a noção de desempenho para os algoritmos e identificar que alguns algoritmos têm diferentes características de desempenho para a mesma tarefa; • Utilizar seleção composta e negação com os operadores; • Perceber a necessidade de, e escrever, funções personalizadas, incluindo o uso de parâmetros; • Usar e manipular estruturas de dados unidimensional; • Compreender a relação entre a resolução e profundidade de cor, incluindo os efeitos sobre o tamanho do arquivo; • Distinguir variáveis e estruturas de armazenamento de dados.
	7	<ul style="list-style-type: none"> • Reconhecer que a concepção de um algoritmo e suas expressões são diferentes nas variadas linguagens de programação; • Avaliar a eficácia de algoritmos e modelos para problemas semelhantes; • Reconhecer onde e quais informações podem ser filtradas ao generalizar soluções de problemas (abstração); • Usar o raciocínio lógico para explicar como funciona um algoritmo; • Representar algoritmos usando linguagem estruturada (pseudocódigo); • Entender como ocorre a transmissão de dados entre computadores; • Compreender a relação entre os circuitos e lógica booleana.
	8	<ul style="list-style-type: none"> • Entender que alguns problemas não podem ser resolvidos computacionalmente; • Projetar e escrever programas modulares que impõem a reutilização de código; • Compreender a diferença entre repetições 'enquanto' e 'para'; • Entender e utilizar estruturas de dados bidimensionais; • Realizar operações utilizando padrões de bits (ex: conversão entre binário e hexadecimal, subtração binária, etc.); • Compreender o modelo cliente-servidor (páginas web dinâmicas do lado do servidor), processam e armazenam os dados inseridos pelos usuários.
	9	<ul style="list-style-type: none"> • Ter experiência prática com uma linguagem de programação simples; • Compreender como os dados são armazenados em memória; • Projetar uma solução para um problema que dependa de sub-soluções para instâncias menores do mesmo problema (recursão); • Saber que processadores têm conjuntos de instruções; • Entender e poder explicar o que são computadores multitarefa; • Entender e poder explicar métodos de compressão simples.

Anexo C: PROPOSTA DAS COMPETÊNCIAS POR ANO ESCOLAR (TECNOLOGIA DA INFORMAÇÃO)

NÍVEL	ANO	TECNOLOGIA DA INFORMAÇÃO
ENSINO FUNDAMENTAL	1	<ul style="list-style-type: none"> • Reconhecer que o conteúdo digital pode ser representado de maneiras diferentes e poder explicar de que forma eles trocam informações; • Obter conteúdo da Internet usando um navegador; • Utilizar programas para criar, editar e guardar conteúdo digital em arquivos e pastas apropriadas; • Reconhecer que computadores interagem com pessoas; • Compartilhar descobertas tecnológicas na escola, inclusive a respeito de assuntos extraclasse.
	2	<ul style="list-style-type: none"> • Reconhecer diferentes tipos de dados: textos e números; • Reconhecer que uma variedade de dispositivos digitais pode ser considerado um computador; • Reconhecer dispositivos de entrada e saída; • Navegar na web e realizar pesquisas na web simples; • Usar a tecnologia com aumento da independência para organizar o conteúdo digital com um propósito específico (Abstração);
	3	<ul style="list-style-type: none"> • Entender a diferença entre dados e informação; • Saber por que a classificação de dados pode melhorar a busca de informações; • Usar critérios individuais para obter informações através de pesquisas; • Coletar, organizar e apresentar dados e informações em conteúdo digital; • Fazer melhorias para soluções com base no feedback recebido;
	4	<ul style="list-style-type: none"> • Reconhecer que os algoritmos podem trabalhar com diferentes tipos de dados; • Reconhecer que os dados podem ser estruturados em uma tabela para torná-los mais úteis. • Analisar e avaliar os dados e informações, e reconhecer que os dados de má qualidade levam a resultados não confiáveis e conclusões imprecisas; • Realizar pesquisas mais complexas de informação usando operadores; • Saber a diferença entre as diferentes redes de computadores;
	5	<ul style="list-style-type: none"> • Utilizar critérios para avaliar a qualidade das soluções, podendo identificar melhorias através do refinamento da solução, e para as soluções futuras. • Consultar dados em uma tabela; • Saber que há uma gama de sistemas operacionais e software aplicativo para o mesmo hardware;
	6	<ul style="list-style-type: none"> • Reconhecer o público-alvo na concepção e criação de conteúdo digital; • Projetar critérios para avaliar criticamente a qualidade das soluções, utiliza os critérios para identificar melhorias e pode fazer aperfeiçoamentos adequados à solução; • Avaliar a confiabilidade do conteúdo digital e considerar a possibilidade de utilização de recursos de design visual na criação de artefatos digitais para um público-alvo;
	7	<ul style="list-style-type: none"> • Avaliar a adequação dos dispositivos digitais, serviços de internet e software de aplicação para alcançar determinadas metas; • Critérios de design para os usuários avaliarem a qualidade das soluções; • Usar o feedback dos usuários para identificar melhorias e poder fazer aperfeiçoamentos adequados à solução;
	8	<ul style="list-style-type: none"> • Realizar projetos criativos que recolha, analisa e avalia dados para atender às necessidades de um grupo de pessoas específico; • Considerar as propriedades da mídia quando importados para artefatos digitais; • Projetar e criar artefatos digitais para um público mais amplo ou remoto.
	9	<ul style="list-style-type: none"> • Saber o que é um banco de dados relacional e compreender os benefícios de armazenar dados em mais de uma tabela; • Documentar feedback do usuário, identificando as melhorias e os aperfeiçoamentos feitos para a solução; • Conhecer a finalidade de equipamentos associados a sistemas de computadores em rede;

Adaptado de CAS, 2014

Anexo D: LICENÇA MAURÍCIO DE SOUZA PRODUÇÕES



TERMO DE AUTORIZAÇÃO

MAURICIO DE SOUSA PRODUÇÕES S.A., com sede nesta capital, à Rua Werner Von Siemens, n.º 111 – Prédio 19 – Espaço 1 - Lapa de Baixo, CEP 05069-010, inscrita no CNPJ sob o n.º 47.257.902/0001-71, neste ato representada por seu Presidente Sr. Mauricio Araujo de Souza, que também assina como “Mauricio de Sousa”; vem pelo presente termo, **AUTORIZAR**, sem custos de Direitos Autorais o Sr. **Christian Puhlmann Brackmann**, brasileiro, professor, casado, portador da Cédula de Identidade R.G. sob o n.º 3081050175 SSP/RS, inscrito no CPF/MF sob o n.º 990.653.040-53, residente e domiciliado à Rua Benjamin Constant, n.º 881/203 – Centro – Santa Maria – Rio Grande do Sul – CEP 97050021; a veicular os seus artigos/projeto intitulados: a.) - **AVALIAÇÃO DO PROCESSO ENSINO-APRENDIZAGEM DO PENSAMENTO COMPUTACIONAL DESPLUGADO NA EDUCAÇÃO BÁSICA**; b.) - **PENSAMENTO COMPUTACIONAL DESPLUGADO PARA CRIANÇAS: ENSINO DA COMPUTAÇÃO SEM O PROTAGONISTA** e c.) - **PENSAMENTO COMPUTACIONAL PARA TODOS**; com os personagens de criação e autoria de “Mauricio Araujo de Sousa”, conforme Anexo (Doc. 01 a 04), nos “sites” da rede internet: <http://www.wipsce.org/2017/index.php>, do evento/workshop denominado: 12th Workshop in Primary and Secondary Computing Education (WiPSCE 2017) e <http://seer.ufrgs.br/RENOTE>, da Revista Acadêmica “Novas Tecnologias na Educação da Universidade Federal do Rio Grande do Sul (RENOTE)”, para servir de **apoio, consulta e inspirações** para alunos e professores de escolas públicas e particulares de ensino do Brasil e exterior e sem quaisquer fins partidários e políticos.

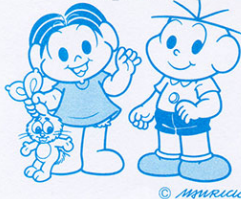
Na referida veiculação, deverá ser incluso o crédito para: **MAURICIO DE SOUSA PRODUÇÕES S.A. – 2017**, pelo uso dos personagens.

Qualquer outra destinação contrária ao exposto neste Termo de Autorização deverá ser solicitada expressamente.

Esta autorização é válida até a data de **11 de abril de 2019**, sendo vetado o “download” das páginas acima citadas e objeto deste Termo.

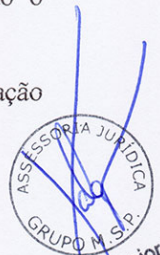
Findo o prazo especificado acima, se faz necessário nova solicitação de autorização expressa, para continuidade da mesma.

São Paulo, 11 de abril de 2017.



MAURICIO

 - MAURICIO DE SOUSA PRODUÇÕES S.A. -
 MAURICIO ARAUJO DE SOUZA



Valquensir Tatyano Junior
 OAB/SP n. 91.492

Anexo E: Licença Thinkfun Inc.



May 23, 2017

Dear Mr. Brackmann:

We hereby give permission to you, Christian Brackmann, to use game concepts, puzzles, and challenges developed by Thinkfun to serve as support, reference, and inspiration for students, teachers, and researchers of schools in Brazil. The uses are exclusive for education, research, and non-commercial purposes.

Specifically this permission includes Brackmann's efforts to adapt, distribute, display, make derivative works and remixes, and to publish academic papers based on ThinkFun game concepts. Mr. Brackmann agrees to attribute credits from ThinkFun games and derivatives to ThinkFun, and to give permission to ThinkFun to use these derivative ideas in their own game development work attributing credits.

Sincerely,

A handwritten signature in blue ink, appearing to read "Bill Ritchie", with a large, sweeping flourish extending to the right.

Bill Ritchie

President, ThinkFun Inc.

Anexo F: Relação de questões e conceitos computacionais envolvidos no Teste de PC

	Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado										Existencia de anidamiento	Tarea requerida	Opción correcta
			Direcciones		Bucles (loops)		Condicionales (conditionals)			Funciones (functions)					
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicionales simple (if)	Condicionales compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros						
Item 1	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	No	No	Secuenciación	B
Item 2	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	No	No	Complejamiento	C
Item 3	Laberinto	Visual por bloques	Sí	No	No	No	No	No	No	No	No	No	No	Depuración	D
Item 4	Lienzo	Visual por bloques	Sí	No	No	No	No	No	No	No	No	No	No	Secuenciación	D
Item 5	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	No	No	Secuenciación	C
Item 6	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	No	No	Complejamiento	D
Item 7	Lienzo	Visual por bloques	Sí	No	No	No	No	No	No	No	No	No	No	Depuración	A
Item 8	Laberinto	Visual por bloques	Sí	No	No	No	No	No	No	No	No	No	Sí	Secuenciación	B
Item 9	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	No	No	Secuenciación	D
Item 10	Laberinto	Visual por bloques	Sí	No	No	No	No	No	No	No	No	No	No	Complejamiento	C
Item 11	Laberinto	Visual por flechas	Sí	Sí	Sí	No	No	No	No	No	No	No	Sí	Depuración	C
Item 12	Lienzo	Visual por bloques	Sí	Sí	Sí	No	No	No	No	No	No	No	No	Secuenciación	A
Item 13	Laberinto	Visual por flechas	No	Sí	Sí	No	No	No	No	No	No	No	Sí	Secuenciación	B
Item 14	Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	No	No	Sí	Secuenciación	A
Item 15	Laberinto	Visual por flechas	Sí	Sí	Sí	Sí	No	No	No	No	No	No	Sí	Complejamiento	D
Item 16	Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	No	No	Sí	Depuración	D
Item 17	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	No	No	Sí	Secuenciación	B
Item 18	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	No	No	Sí	Secuenciación	A
Item 19	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	No	No	Sí	Depuración	B
Item 20	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	No	No	Sí	Complejamiento	C
Item 21	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	No	Sí	No	No	Sí	Secuenciación	A
Item 22	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	No	Sí	No	No	Sí	Secuenciación	B
Item 23	Laberinto	Visual por bloques	Sí	No	No	Sí	No	No	No	Sí	No	No	Sí	Complejamiento	A
Item 24	Laberinto	Visual por bloques	Sí	No	No	Sí	No	No	No	Sí	No	No	Sí	Complejamiento	C
Item 25	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	No	No	Sí	No	Sí	Secuenciación	B
Item 26	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	No	No	Sí	No	Sí	Complejamiento	B
Item 27	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	No	No	Sí	No	Sí	Secuenciación	A
Item 28	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	No	No	Sí	No	Sí	Complejamiento	C

APÊNDICES

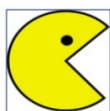
Apêndice A: TRADUÇÃO DO TESTE DE PENSAMENTO COMPUTACIONAL

Devido a formatação exigida pela ABNT, foi necessária a ampliação do número de páginas

TESTE DE PENSAMENTO COMPUTACIONAL

INSTRUÇÕES

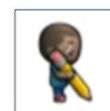
O teste é composto por 28 perguntas, distribuídas em 226 páginas com aproximadamente 3 perguntas em cada uma. Todas as perguntas têm 4 alternativas de resposta (A, B, C e D) das quais só uma é correta. A partir do início do teste, você dispõe de até 45 minutos para fazer o melhor que puder. Não é imprescindível que você responda a todas as perguntas. Antes de começar o teste, vamos ver 3 exemplos para que lhe familiarize com o tipo de perguntas que vai encontrar, nas quais aparecerão os personagens que lhe apresentamos.



'Pac-Man'



Fantasma



Artista

EXEMPLO I

Neste primeiro exemplo se pergunta quais são os comandos que levam o 'Pac-Man' até o fantasma pelo caminho indicado. Ou seja, levar 'Pac-Man' exatamente à caixa em que o fantasma está (sem passar, nem parar), e seguindo estritamente o caminho marcado em amarelo (sem sair e sem tocar nas paredes, representadas pelos quadrados laranja).

A alternativa correta neste exemplo é a B. Marque a alternativa correspondente, na folha de respostas.

<p>Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?</p>	<p>Alternativa A </p> <p>Alternativa B ✓</p> <p>Alternativa C </p> <p>Alternativa D </p>
--	---

Exemplo I	<input type="checkbox"/> A	<input checked="" type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
-----------	----------------------------	---------------------------------------	----------------------------	----------------------------

EXEMPLO II

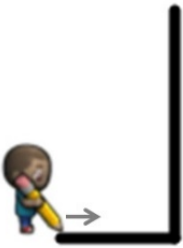

Neste segundo exemplo, se pergunta de novo quais são os comandos que levam o 'Pac-Man' até o fantasma pelo caminho assinalado. Mas neste caso, as opções de resposta, em vez de ser flechas, são blocos que encaixam uns nos outros. Lembramos que a pergunta pede para levar o 'Pac-Man' EXATAMENTE a casa em que se encontra o fantasma (sem passar nem parar), e seguindo estritamente o caminho marcado em amarelo (sem sair e sem tocar nas paredes, representadas pelos quadrados laranja). A alternativa correta neste exemplo é a C. Marque a alternativa correspondente, na folha de respostas.

<p>Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?</p>	<p>Alternativa A</p>	<p>Alternativa B</p>
	<p>Alternativa C </p>	<p>Alternativa D</p>

Exemplo II	<input type="checkbox"/> A	<input type="checkbox"/> B	<input checked="" type="checkbox"/> C	<input type="checkbox"/> D
------------	----------------------------	----------------------------	---------------------------------------	----------------------------

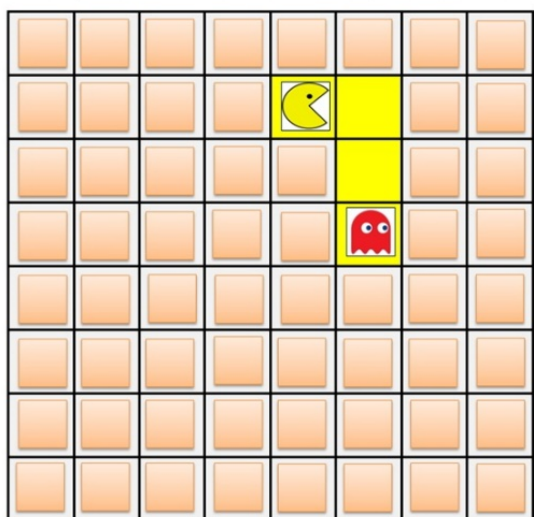




EXEMPLO III

Neste terceiro exemplo se pergunta que comandos deve seguir o artista para desenhar a figura que aparece na tela. Ou seja, como deve MOVER o lápis para que se desenhe a figura. O comando MOVER empurra o lápis desenhando, enquanto que o comando SALTAR faz um salto ao artista sem desenhando. A seta cinza indica a direção do primeiro movimento da caneta. A alternativa correta neste exemplo é A. Marque a alternativa correspondente, na folha de respostas.

<p>Qual sequência o artista deve seguir para desenhar a figura abaixo? O lado menor mede 50 pixels e o maior mede 100 pixels.</p> 	<p>Alternativa A </p> <p>avance por 50 pixels vire à esquerda por 90 graus avance por 100 pixels</p>	<p>Alternativa B</p> <p>avance por 50 pixels vire à direita por 90 graus avance por 100 pixels</p>
	<p>Alternativa C</p> <p>avance por 100 pixels vire à esquerda por 90 graus avance por 50 pixels</p>	<p>Alternativa D</p> <p>avance por 100 pixels vire à direita por 90 graus avance por 50 pixels</p>

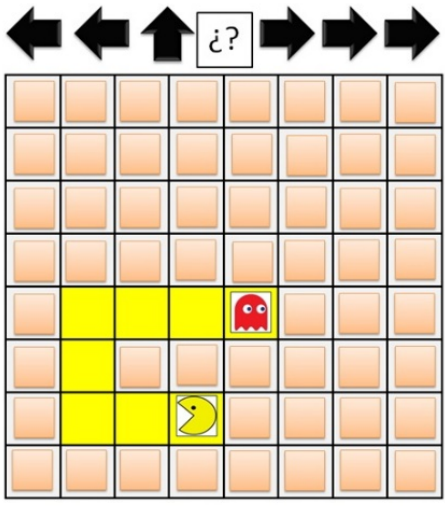
Exemplo III A B C D


QUESTÃO 1


<p>Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?</p> 	<p>Alternativa A</p> <p></p>
	<p>Alternativa B</p> <p></p>
	<p>Alternativa C</p> <p></p>
	<p>Alternativa D</p> <p></p>


QUESTÃO 2


Qual comando está faltando na sequência para levar o "Pac-Man" até o fantasma pelo caminho indicado?



Alternativa A


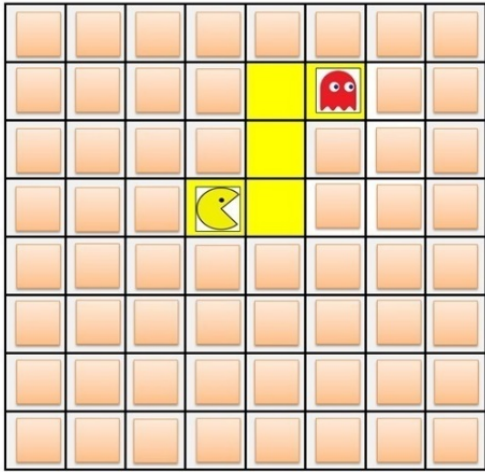
Alternativa B


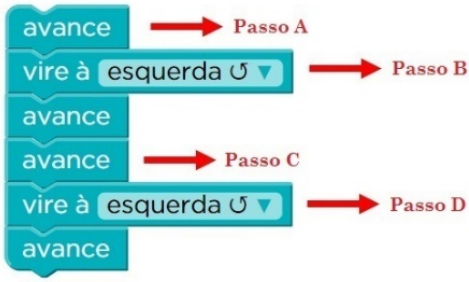
Alternativa C


Alternativa D


QUESTÃO 3

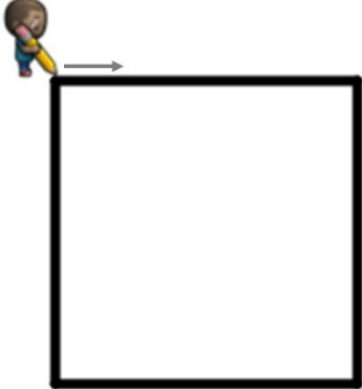
Para levar o "Pac-Man" até o fantasma pelo caminho indicado, qual passo da sequência está *incorreto*?





QUESTÃO 4

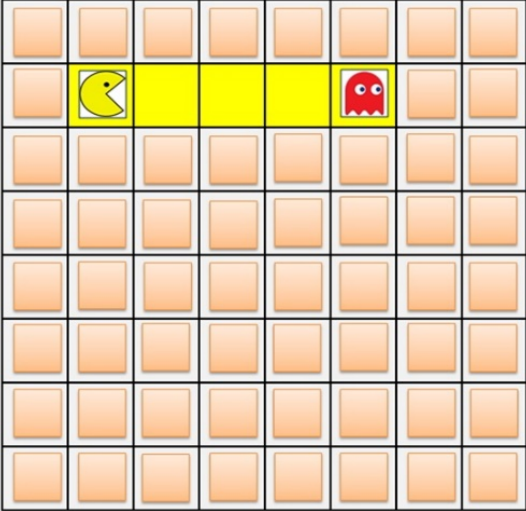
Qual sequência o artista deve seguir para desenhar o quadrado abaixo?
Cada um dos lados mede 100 pixels.







<p>Alternativa A</p> <ul style="list-style-type: none"> avance por 100 pixels vire à direita por 90 graus avance por 100 pixels vire à esquerda por 90 graus avance por 100 pixels vire à direita por 90 graus avance por 100 pixels 	<p>Alternativa B</p> <ul style="list-style-type: none"> avance por 25 pixels vire à direita por 90 graus avance por 25 pixels vire à esquerda por 90 graus avance por 100 pixels vire à direita por 90 graus avance por 25 pixels
<p>Alternativa C</p> <ul style="list-style-type: none"> avance por 50 pixels vire à direita por 90 graus avance por 50 pixels vire à direita por 90 graus avance por 50 pixels vire à direita por 90 graus avance por 50 pixels vire à direita por 90 graus avance por 50 pixels 	<p>Alternativa D</p> <ul style="list-style-type: none"> avance por 100 pixels vire à direita por 90 graus avance por 100 pixels vire à direita por 90 graus avance por 100 pixels vire à direita por 90 graus avance por 100 pixels

QUESTÃO 5

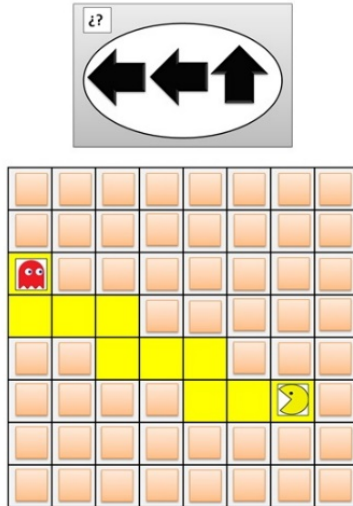
Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?



<p>Alternativa A</p> <p>× 5</p> 	<p>Alternativa B</p> <p>× 3</p> 
<p>Alternativa C</p> <p>× 4</p> 	<p>Alternativa D</p> <p>× 2</p> 

QUESTÃO 6

Quantas vezes a sequência abaixo deve ser repetida para levar o "Pac-Man" até o fantasma pelo caminho indicado?



Alternativa A

× 2

Alternativa B

× 1

Alternativa C

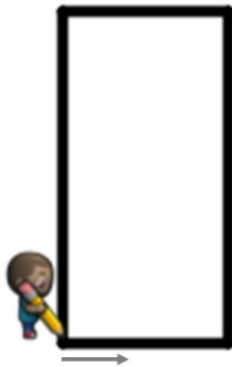
× 4

Alternativa D

× 3

QUESTÃO 7

Para que o artista desenha uma vez o seguinte retângulo (50 pixels de largura e 100 pixels de altura), qual passo da sequência está *incorreto*?



Passo A

repita 4 vezes

faça

avance por 50 pixels

vire à esquerda por 90 graus

→ Passo B

avance por 100 pixels



→ Passo C

vire à esquerda por 90 graus

→ Passo D

QUESTÃO 8

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?

Alternativa A

```

repita 4 vezes
  faça
    repita 3 vezes
      faça
        avance
      vire à direita 90°
    avance
          
```

Alternativa B

```

repita 3 vezes
  faça
    repita 4 vezes
      faça
        avance
      vire à direita 90°
    avance
          
```

Alternativa C

```

repita 3 vezes
  faça
    repita 4 vezes
      faça
        avance
      vire à direita 90°
    avance
          
```



Alternativa D

```


repita 4 vezes
  faça
    avance
  repita 4 vezes
    faça
      vire à direita 90°
    avance
          
```

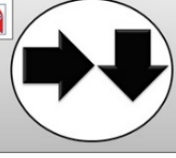
QUESTÃO 9

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?


							
							

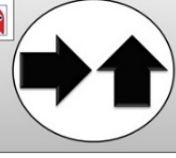
Alternativa A

Repetir até chegar ao 





Alternativa B

Repetir até chegar ao 





Alternativa C

Repetir até chegar ao 



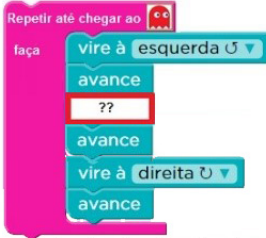
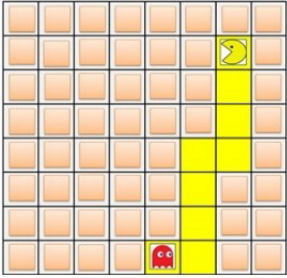
Alternativa D

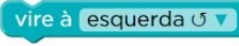
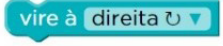

Repetir até chegar ao 



QUESTÃO 10

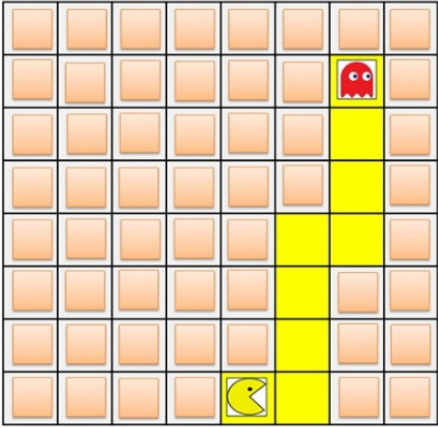
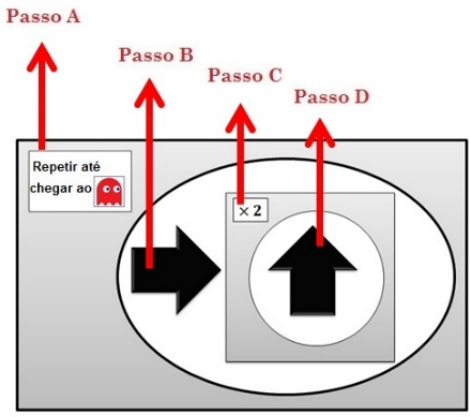
Qual bloco está faltando na sequência abaixo para levar o "Pac-Man" até o fantasma pelo caminho indicado?

<p>Alternativa A</p> 	<p>Alternativa B</p> 
<p>Alternativa C</p> 	<p>Alternativa D</p> <p><i>Não falta nenhum bloco</i></p>


QUESTÃO 11

Para que o "Pac-Man" chegue até o fantasma pelo caminho indicado, qual passo da sequência está *incorreto*?

QUESTÃO 12

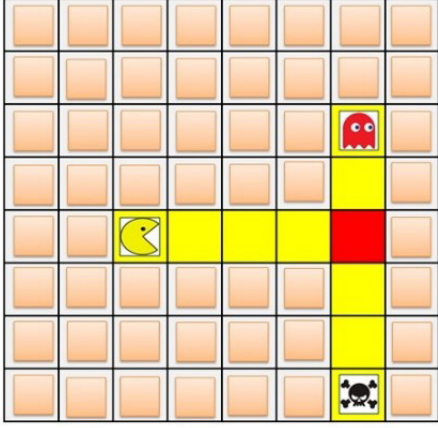
Qual sequência o artista deve seguir para desenhar a escada que leva até a flor? Cada degrau sobe 30 pixels.



<p>Alternativa A</p> <pre> Repetir até a flor faça repita 4 vezes faça avance por 30 pixels vire à direita por 90 graus pula para a frente por 30 pixels </pre>	<p>Alternativa B</p> <pre> Repetir até a flor faça repita 4 vezes faça avance por 120 pixels vire à direita por 90 graus pula para a frente por 30 pixels </pre>
<p>Alternativa C</p> <pre> Repetir até a flor faça repita 4 vezes faça avance por 30 pixels vire à direita por 90 graus pula para a frente por 210 pixels </pre>	<p>Alternativa D</p> <pre> Repetir até a flor faça repita 7 vezes faça avance por 30 pixels vire à direita por 90 graus pula para a frente por 30 pixels </pre>

QUESTÃO 13

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?



<p>Alternativa A</p> <pre> repetir até chegar no se passar por [seta para cima] [seta para direita] </pre>	<p>Alternativa B</p> <pre> repetir até chegar no se passar por [seta para cima] [seta para cima] [seta para direita] </pre>
<p>Alternativa C</p> <pre> repetir até chegar no se passar por [seta para cima] [seta para direita] </pre>	<p>Alternativa D</p> <pre> repetir até chegar no se passar por [seta para cima] [seta para cima] [seta para direita] </pre>

QUESTÃO 14

<p>Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?</p>	<p>Alternativa A</p>	<p>Alternativa B</p>
	<p>Alternativa C</p>	<p>Alternativa D</p>

QUESTÃO 15

<p>O que falta na seguinte sequência para levar o "Pac-Man" até o fantasma pelo caminho indicado?</p>	<p>Alternativa A</p>
	<p>Alternativa B</p>
	<p>Alternativa C</p>
	<p>Alternativa D</p> <p>Tanto a alternativa A como a alternativa C estão corretas</p>

QUESTÃO 16

Para que o "Pac-Man" chegue até o fantasma pelo caminho indicado, qual passo da sequência está *incorreto*?

```

    repita até [ghost]
    faça
      avance
      Se houver caminho à esquerda
      faça vire à esquerda
      Se houver caminho à direita
      faça avance
  
```

Passo A → Passo B → Passo C → Passo D

QUESTÃO 17

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?

Alternativa A

```

        repita até [ghost]
        faça
          se houver caminho à frente
          faça avance
          se não vire à esquerda
      
```

Alternativa B

```

        repita até [ghost]
        faça
          se houver caminho à frente
          faça avance
          se não vire à direita
      
```

Alternativa C

```

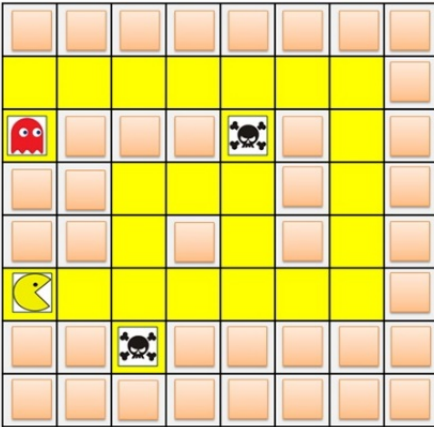



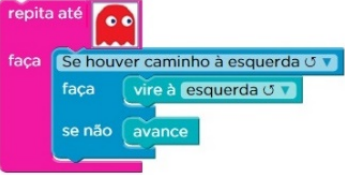
        repita até [ghost]
        faça
          Se houver caminho à direita
          faça vire à direita
          se não avance
      
```

Alternativa D

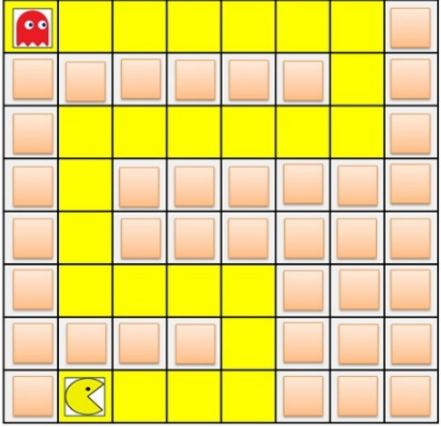
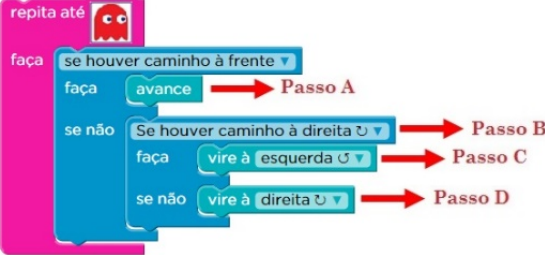
```

        repita até [ghost]
        faça
          Se houver caminho à esquerda
          faça vire à esquerda
          se não avance
      
```

QUESTÃO 18

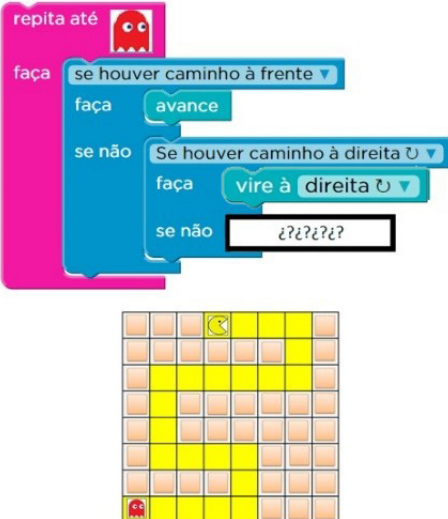
<p>Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?</p> 	<p>Alternativa A</p> 	<p>Alternativa B</p> 
	<p>Alternativa C</p> 	<p>Alternativa D</p> 

QUESTÃO 19

<p>Para que o "Pac-Man" chegue até o fantasma pelo caminho indicado, qual passo da sequência está <i>incorreto</i>?</p> 	
---	--

QUESTÃO 20

Qual bloco está faltando na sequência abaixo para levar o "Pac-Man" até o fantasma pelo caminho indicado?



The Scratch script is as follows:

```

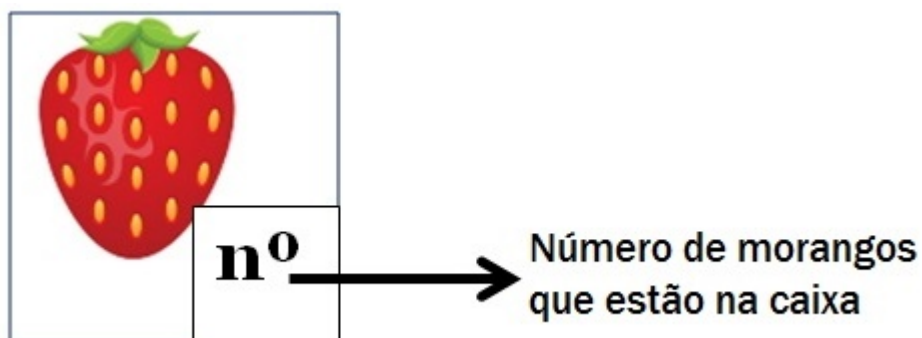
    repita até [1] vezes
      faça
        se houver caminho à frente
          faça
            avance
        se não
          Se houver caminho à direita
            faça
              vire à direita
          se não
            [??????]
  
```

The maze grid is 10x10. Pac-Man starts at (10,1) and the ghost is at (1,10). The path is highlighted in yellow: (10,1) → (9,1) → (8,1) → (8,2) → (8,3) → (8,4) → (8,5) → (8,6) → (8,7) → (8,8) → (8,9) → (8,10) → (9,10) → (10,10).

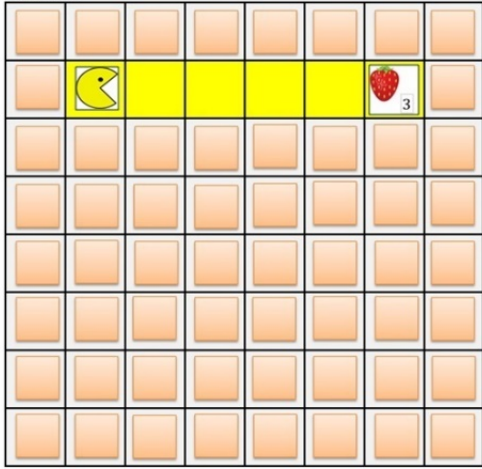
Alternativa A	Alternativa B
avance	vire à direita
Alternativa C	Alternativa D
vire à esquerda	Não falta nenhum bloco

IMPORTANTE: LEIA COM ATENÇÃO

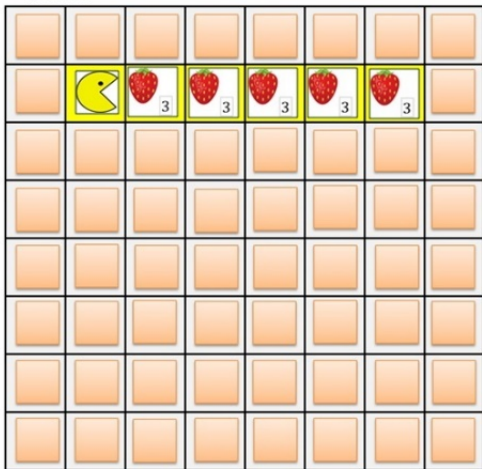
A IMAGEM ABAIXO IRÁ APARECER NAS PRÓXIMAS QUESTÕES. O NÚMERO QUE ESTÁ NO CANTO INFERIOR DIREITO INDICA QUANTOS MORANGOS EXISTEM NA CAIXA.



QUESTÃO 21

<p>Qual sequência leva o "Pac-Man" pelo caminho indicado até os morangos e faz o "Pac-Man" comer o número de morangos indicado?</p> 	<p>Alternativa A</p> <pre> enquanto houver caminho em frente faça avance repita 3 vezes faça coma 1 morango </pre>	<p>Alternativa B</p> <pre> enquanto houver caminho em frente faça avance repita 4 vezes faça coma 1 morango </pre>
	<p>Alternativa C</p> <pre> enquanto houver caminho em frente faça avance repita 5 vezes faça coma 1 morango </pre>	<p>Alternativa D</p> <pre> enquanto houver caminho em frente faça avance repita 3 vezes faça coma 1 morango </pre>

QUESTÃO 22

<p>Qual sequência leva o "Pac-Man" pelo caminho indicado até os morangos e faz o "Pac-Man" comer o número de morangos indicado?</p> 	<p>Alternativa A</p> <pre> enquanto houver caminho em frente repita 5 vezes faça avance repita 3 vezes faça coma 1 morango </pre>	<p>Alternativa B</p> <pre> enquanto houver caminho em frente faça avance repita 3 vezes faça coma 1 morango </pre>
	<p>Alternativa C</p> <pre> enquanto houver caminho em frente repita 3 vezes faça avance repita 5 vezes faça coma 1 morango </pre>	<p>Alternativa D</p> <pre> enquanto houver caminho em frente faça avance repita 3 vezes faça coma 1 morango </pre>

QUESTÃO 23

O que falta na seguinte sequência para que "Pac-Man" avance pelo caminho assinalado comendo o número de morangos indicados?

Alternativa A

1 vez

Alternativa B

2 vezes

Alternativa C

3 vezes

Alternativa D

5 vezes

QUESTÃO 24

O que falta na seguinte sequência para que "Pac-Man" avance pelo caminho assinalado comendo o número de morangos indicados?

Alternativa A

enquanto houver caminho em frente

Alternativa B

enquanto não houver caminho em frente

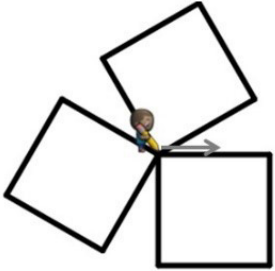
Alternativa C

enquanto houver morangos


Alternativa D

enquanto não houver morangos

QUESTÃO 25

<p>Se temos o seguinte conjunto de instruções, que chamamos de "my function", e que desenha um quadrado de 100 pixels de lado.</p> <div data-bbox="300 434 624 600"> <p>Função</p> <pre> my function repetir 4 vezes faça avance por 100 pixels vire à direita por 90 graus </pre> </div> <p>Qual sequência o artista deve seguir para desenhar a seguinte figura? Cada um dos lados mede 100 pixels.</p> 	<p>Alternativa A</p> <pre> repetir 3 vezes faça my function vire à direita por 120 graus </pre>	<p>Alternativa B</p> <pre> repetir 3 vezes faça my function vire à direita por 120 graus </pre>
	<p>Alternativa C</p> <pre> repetir 4 vezes faça my function vire à direita por 90 graus </pre>	<p>Alternativa D</p> <pre> repetir 4 vezes faça my function vire à direita por 90 graus </pre>

QUESTÃO 26

<p>Se temos o seguinte conjunto de instruções, que chamamos de "my function", e que desenha um triângulo de 50 pixels de lado:</p> <div data-bbox="252 1182 660 1384"> <p>Função</p> <pre> my function repetir 3 vezes faça avance por 50 pixels vire à esquerda por 120 graus </pre> </div> <p>O que falta na seguinte sequência para que o artista desenhe a seguinte figura? Cada um dos lados de cada triângulo mede 50 pixels.</p> <pre> repetir ??? vezes faça my function pule para a frente por 50 pixels </pre> 	<p>Alternativa A</p> <p>15</p>	<p>Alternativa B</p> <p>5</p>
	<p>Alternativa C</p> <p>4</p>	<p>Alternativa D</p> <p>3</p>

QUESTÃO 27

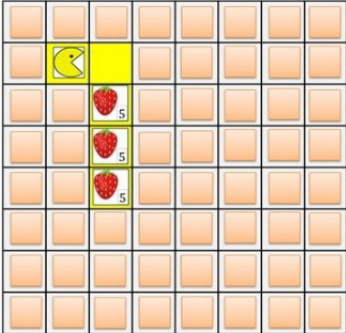
Se temos o seguinte conjunto de instruções, que chamamos de "get 5":

Função

```

get 5
  repita 5 vezes
    faça
      coma 1 morango
  
```

Qual sequência leva o "Pac-Man" pelo caminho indicado e faz com ele coma o número de morangos correspondentes?



<p>Alternativa A</p> <pre> avance vire à direita 90 repita 3 vezes faça avance get 5 </pre>	<p>Alternativa B</p> <pre> avance vire à direita 90 repita 3 vezes faça get 5 avance </pre>
<p>Alternativa C</p> <pre> avance vire à direita 90 repita 5 vezes faça avance get 5 </pre>	<p>Alternativa D</p> <pre> avance vire à direita 90 repita 5 vezes faça get 5 avance </pre>

QUESTÃO 28

Se temos a seguinte sequência de instruções que chamamos de "move and get 4":

Função

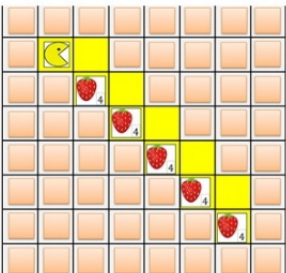
```

move and get 4
  avance
  vire à direita 90
  avance
  repita 4 vezes
    faça
      obtenha néctar
      vire à esquerda 90
  
```

O que falta na seguinte sequência para levar o "Pac-Man" pelo caminho indicado comendo todos os morangos?

```

repetir ??? vezes
  faça
    move and get 4
  
```



<p>Alternativa A</p> <p style="font-size: 2em; text-align: center;">3</p>	<p>Alternativa B</p> <p style="font-size: 2em; text-align: center;">4</p>
<p>Alternativa C</p> <p style="font-size: 2em; text-align: center;">5</p>	<p>Alternativa D</p> <p style="font-size: 2em; text-align: center;">6</p>

Este teste foi desenvolvido pelo Prof. Dr. Marcos Román-Gonzales da Universidad Nacional de Educación a Distancia (UNED) intitulada "Test de Pensamiento Computacional: diseño y psicometría general" e traduzido/adaptado pelos pesquisadores Rafael Marimon Boucinha e Christian Puhmann Brackmann da Universidade Federal do Rio Grande do Sul (UFRGS). É necessário solicitar autorização do uso do teste para o autor do mesmo.

Apêndice B: FOLHA DE RESPOSTA DO TESTE DE PC

TESTE DE PENSAMENTO COMPUTACIONAL

FOLHA DE RESPOSTAS

DADOS PESSOAIS


Nome:	
Sexo:	<input type="checkbox"/> Masculino <input type="checkbox"/> Feminino
Ano escolar:	<input type="checkbox"/> Primeiro <input type="checkbox"/> Segundo <input type="checkbox"/> Terceiro <input type="checkbox"/> Quarto <input type="checkbox"/> Quinto <input type="checkbox"/> Sexto <input type="checkbox"/> Sétimo <input type="checkbox"/> Oitavo <input type="checkbox"/> Nono
Turno:	<input type="checkbox"/> Manhã <input type="checkbox"/> Tarde

RESPOSTAS

Exemplo I	<input type="checkbox"/> A	<input checked="" type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 14	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Exemplo II	<input type="checkbox"/> A	<input type="checkbox"/> B	<input checked="" type="checkbox"/> C	<input type="checkbox"/> D	Questão 15	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Exemplo III	<input checked="" type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 16	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Questão 1	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 17	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Questão 2	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 18	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Questão 3	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 19	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Questão 4	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 20	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Questão 5	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 21	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Questão 6	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 22	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Questão 7	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 23	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Questão 8	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 24	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Questão 9	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 25	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Questão 10	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 26	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Questão 11	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 27	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Questão 12	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	Questão 28	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D
Questão 13	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D					


Apêndice C: ATIVIDADE “DECOMPOSIÇÃO”

PESCAR UM PEIXE



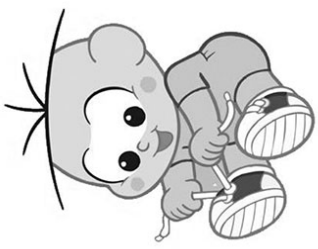
1. _____
 2. _____
 3. _____
 4. _____
 5. _____
 6. _____
 7. _____
 8. _____

CHAMAR E ANDAR DE ELEVADOR




1. _____
 2. _____
 3. _____
 4. _____
 5. _____
 6. _____
 7. _____
 8. _____

ATAR O TÊNIS




1. _____
 2. _____
 3. _____
 4. _____
 5. _____
 6. _____
 7. _____
 8. _____

PLANTAR UMA ÁRVORE




1. _____
 2. _____
 3. _____
 4. _____
 5. _____
 6. _____
 7. _____
 8. _____

LAVAR AS MÃOS



1. _____
 2. _____
 3. _____
 4. _____
 5. _____
 6. _____
 7. _____
 8. _____


PREPARAR CAFÉ DA MANHÃ



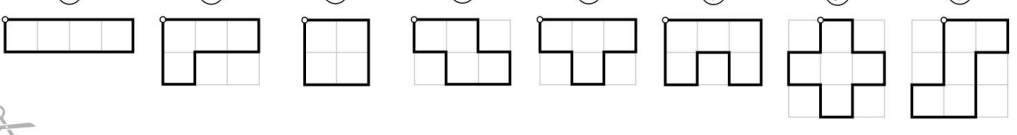
1. _____
 2. _____
 3. _____
 4. _____
 5. _____
 6. _____
 7. _____
 8. _____

Apêndice E: ATIVIDADE “TETRIS (INSTRUÇÕES)”


RODADA 1:



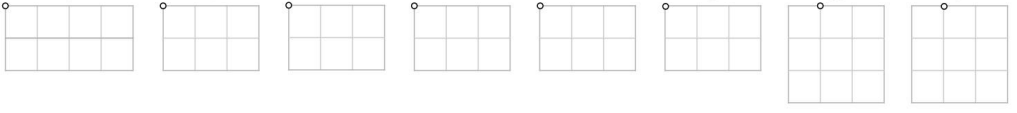
① ② ③ ④ ⑤ ⑥ ⑦ ⑧



A




① ② ③ ④ ⑤ ⑥ ⑦ ⑧

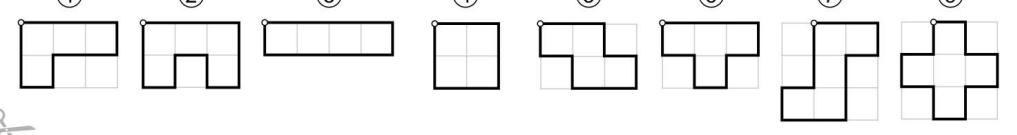


A


RODADA 2:



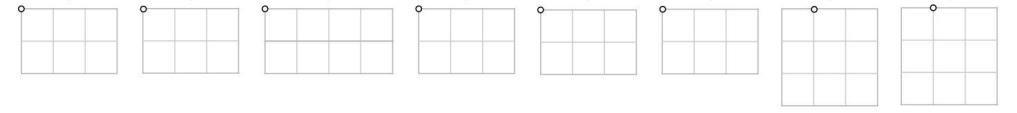
① ② ③ ④ ⑤ ⑥ ⑦ ⑧



B




① ② ③ ④ ⑤ ⑥ ⑦ ⑧

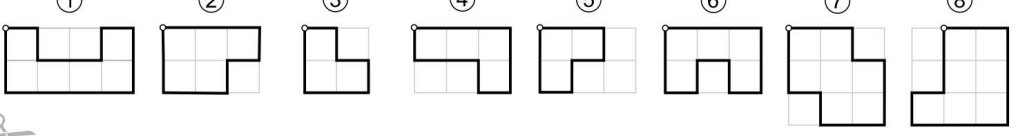


B


RODADA 3:



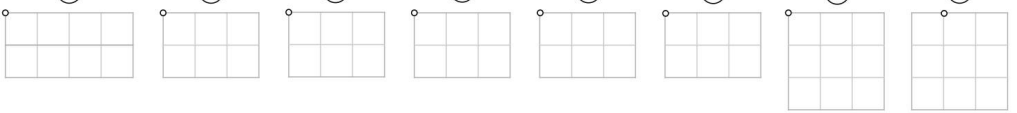
① ② ③ ④ ⑤ ⑥ ⑦ ⑧



C

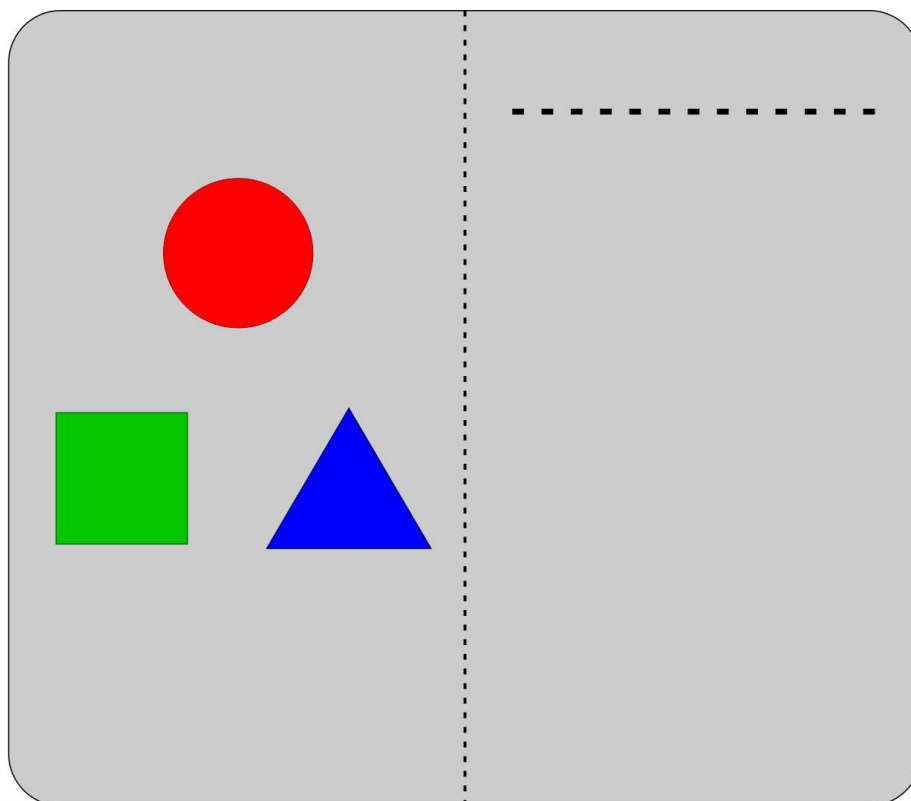











① ② ③ ④ ⑤ ⑥ ⑦ ⑧



C

Apêndice F: ATIVIDADE “CONTROLE REMOTO”



 Quando o botão é pressionado, faça isso: _____	 Quando o botão é pressionado, faça isso: _____	 Quando o botão é pressionado, faça isso: _____
 Quando o botão é pressionado, faça isso: _____	 Quando o botão é pressionado, faça isso: _____	 Quando o botão é pressionado, faça isso: _____
 Quando o botão é pressionado, faça isso: _____	 Quando o botão é pressionado, faça isso: _____	 Quando o botão é pressionado, faça isso: _____

Apêndice G: ATIVIDADE “OS ELEFANTES”



OS ELEFANTES

Um elefante se equilibrava
Em cima da teia de uma aranha
E como via que não caía
Foi chamar outro elefante



Dois elefantes se equilibravam
Em cima da teia de uma aranha
E como viam que não caíam
Foram chamar outro elefante



Três elefantes se equilibravam
Em cima da teia de uma aranha
E como viam que não caíam
Foram chamar outro elefante



Pausa



Quatro elefantes se equilibravam
Em cima da teia de uma aranha
E como viam que não caíam
Foram chamar outro elefante



Cinco elefantes se equilibravam
Em cima da teia de uma aranha
E como viam que não caíam
Foram chamar outro elefante



Seis elefantes se equilibravam
Em cima da teia de uma aranha
E como viam que não caíam
Foram chamar outro elefante



Pausa



Sete elefantes se equilibravam
Em cima da teia de uma aranha
E como viam que não caíam
Foram chamar outro elefante



Oito elefantes se equilibravam
Em cima da teia de uma aranha
E como viam que não caíam
Foram chamar outro elefante



Nove elefantes se equilibravam
Em cima da teia de uma aranha
E como viam que não caíam
Foram chamar outro elefante

OS ELEFANTES

X =



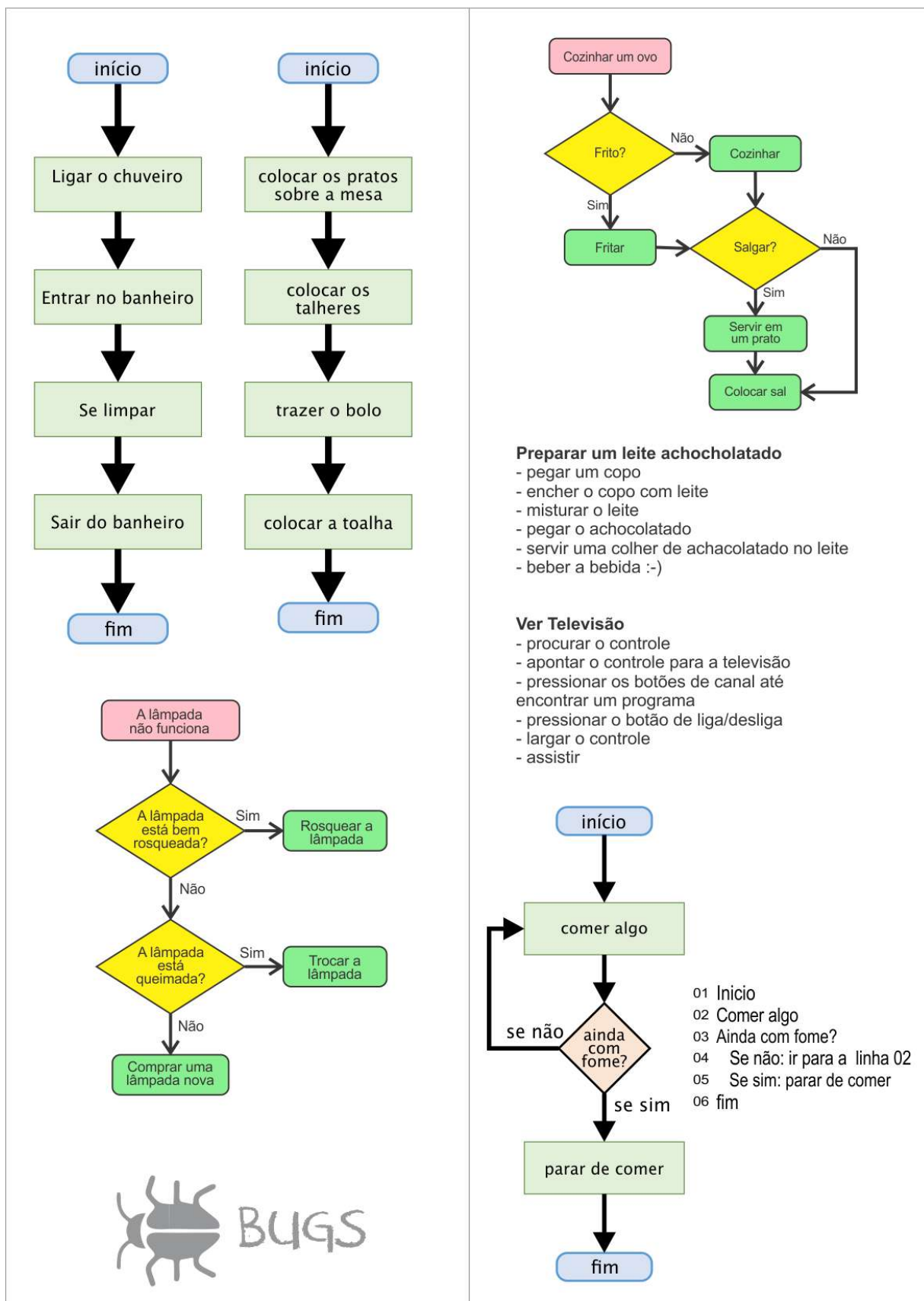
X elefante(s) se equilibrava(m)
Em cima da teia de uma aranha
E como via(m) que não caía(m)
Foram chamar outro elefante

SE X=3 OU X=6: PAUSA

X ← X + 1

SE X<10, REPETIR ESTROFE







Apêndice H: ATIVIDADE “BUGS”



Apêndice I: ATIVIDADE “BONECA DE PAPEL” / TABULEIRO E ROUPAS







<p>se está chovendo <input type="text"/> , então</p> <p>se fico em casa <input type="text"/> , então</p> <p>senão</p>							
<p>senão,</p> <p>se fico em casa <input type="text"/> , então</p> <p>senão</p>							
<p>se vou a uma festa <input type="text"/> , então</p>							
<p>se vou brincar <input type="text"/> , então</p>							
<p>se tenho aula <input type="text"/> , então</p> <p>senão,</p>							

Apêndice J: ATIVIDADE “CUPCAKES”

forma massa cobertura cupcake

Se está pronto, utilize simplesmente um traço (-).







A

Cobertura

Cobertura

Massa + Cobertura

Massa + Cobertura

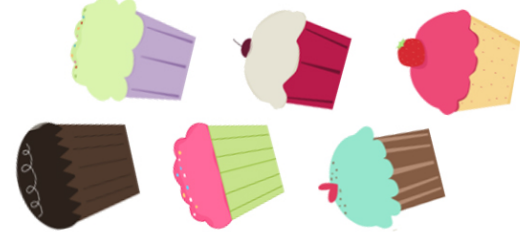
B











Massa + Cobertura

Cobertura

Massa + Cobertura

Cobertura
























D

P1

P1:

Massa + Cobertura





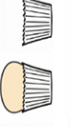
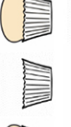




E

2X P2

P2:

Massa + Cobertura

Cobertura

F

G

H

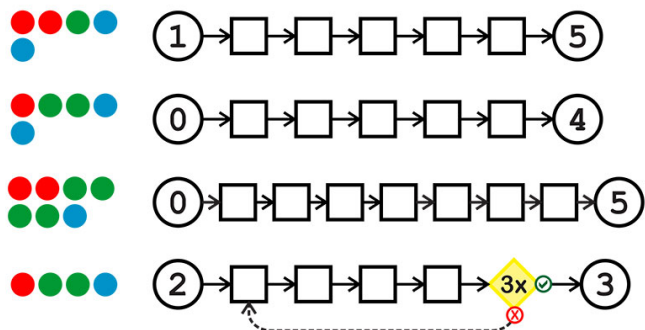
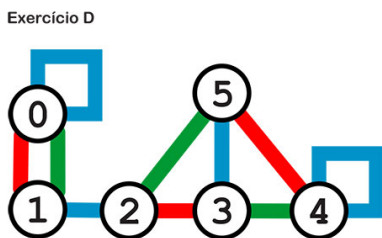
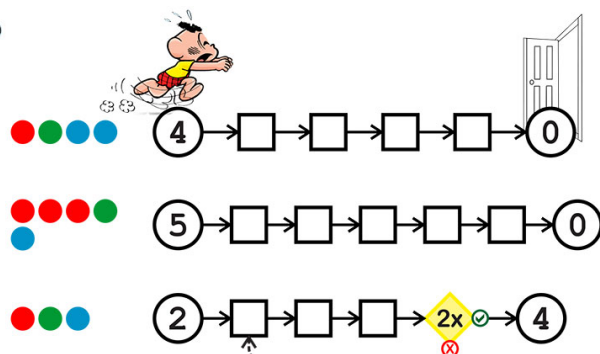
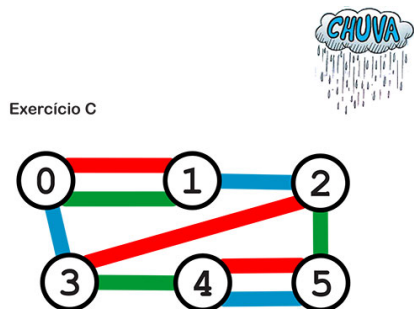
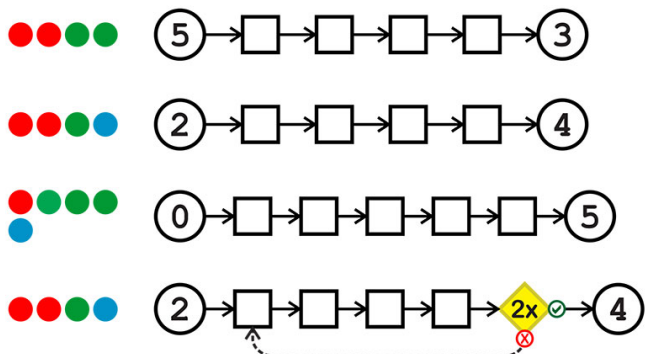
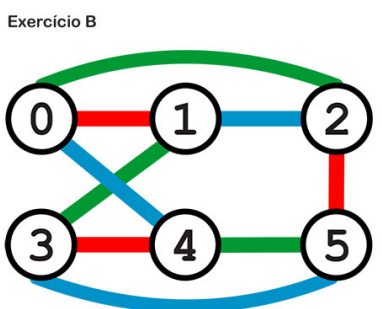
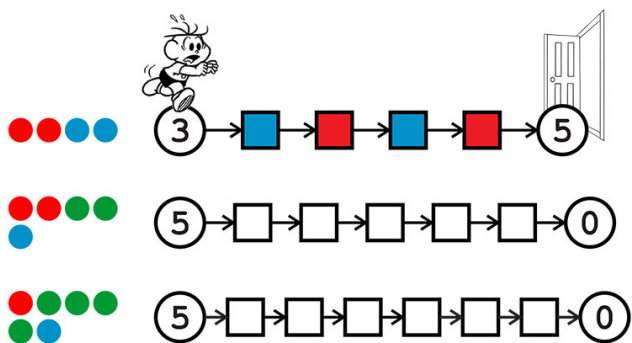
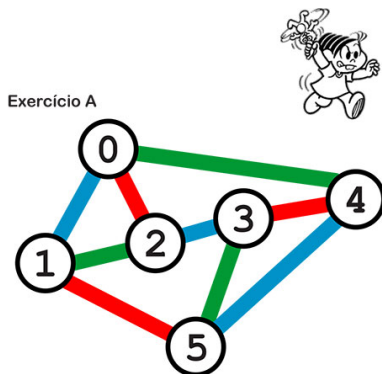
I

J

K

L

Apêndice L: ATIVIDADE “AUTÔMATAS DA MÔNICA”



Apêndice M: CARTA DE PRESENTACIÓN



UNIVERSIDADE POLITÉCNICA DE MADRID
EN ASOCIACIÓN CON LA
UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
Y
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA ERASMUS MUNDUS SMART²



A la atención de la Dirección del Centro.

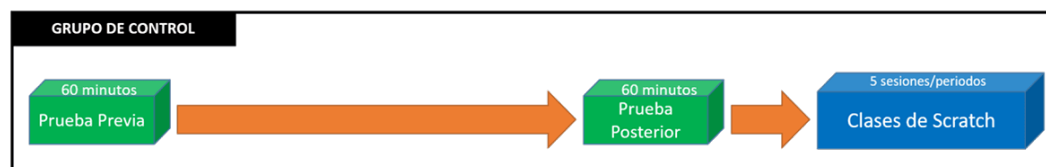
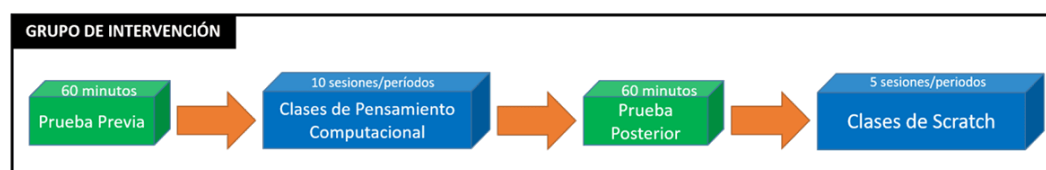
Asunto: Solicitud de realización de actividades de investigación en su institución

Estimado(a) Señor(a),

Yo, Christian Puhlmann Brackmann, alumno de doctorado de la Universidad Federal del Rio Grande do Sul (Brasil), en colaboración con la Universidad Politécnica de Madrid (UPM) y la Universidad Nacional de Educación a Distancia (UNED), estoy llevando a cabo una investigación de doctorado titulada “Evaluación del Proceso Enseñanza-Aprendizaje del Pensamiento Computacional Sin Ordenador en la Educación Primaria” que como su nombre indica estudia el desarrollo de pensamiento computacional en niños de educación primaria.

Dentro de nuestra investigación tenemos la intención de impartir unos seminarios breves (una clase puede variar entre 45 y 60 minutos, dependiendo de la escuela) en dos grupos de quinto o sexto curso (tercer ciclo) de Educación Primaria durante los próximos meses de 2016. La planificación de la experiencia sería la siguiente:

GRUPO/CLASE 1 (INTERVENCIÓN)	GRUPO/CLASE 2 (CONTROL)
<ul style="list-style-type: none"> • 1 clase en Sala de Ordenadores con internet para la prueba previa; • 10 clases en aula convencional para actividades de Pensamiento Computacional sin ordenador; • 1 clase en la Sala de Ordenadores con internet para la prueba posterior; • 5 clases en la Sala de Ordenadores para clases de Pensamiento Computacional con el uso de equipos 	<ul style="list-style-type: none"> • 1 clase en Sala de Ordenadores con internet para la prueba previa; • 1 clase en la Sala de Ordenadores con internet para la prueba posterior; • 5 clases en la Sala de Ordenadores para clases de Pensamiento Computacional con el uso de equipos



Contactamos con su colegio porque lo hemos seleccionado por sus características como candidato para ofrecerles la posibilidad de participar en esta investigación. Ustedes podrán solicitar cualquier aclaración que necesiten y también optar por no aceptar participar en esta investigación. Confiamos en que este ofrecimiento sea de su interés puesto que está totalmente financiado, no implicaría ningún gasto para su institución, ofrecería a los alumnos interesados la posibilidad de participar en las clases y recibir los cursos totalmente gratis y conllevaría un plus de innovación para su institución por participar de primera mano con investigadores de tres instituciones punteras en educación en el área de las nuevas tecnologías. Los datos recabados serán protegidos y se garantizará el anonimato del estudiante. Tras la finalización del estudio, los resultados se analizarán y formalizarán y se enviarán al colegio.

Necesitamos de su respuesta tan pronto como sea posible. Estamos disponibles para hacer una reunión virtual (por Skype) o presencial donde aclarar cualquiera duda. Yo personalmente puedo ir a visitarles a su centro o bien podemos recibirles en la Escuela Técnica Superior de Ingenieros Informáticos de la Universidad Politécnica de Madrid, donde además podrían conocer también a la Profesora Susana Muñoz Hernández que supervisa mi investigación.

Para contactarnos, utilice los siguientes datos:

Christian Puhlmann Brackmann

✉ christian.brackmann@ufrgs.br

☎ 692 37 29 84

Prof. Dra. Susana Muñoz Hernández

✉ susana@fi.upm.es

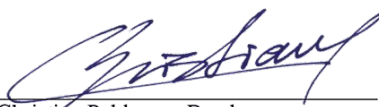
☎ 913 36 36 69

📍 Facultad de Informática de la UPM. Campus de Montegancedo s/n,
Bloque 2, Despacho 2310, Boadilla del Monte 28660, España

A la espera de contar con su inestimable colaboración, aprovecho la ocasión para agradecerles por la atención prestada y quedo a la espera de su contestación a nuestro ofrecimiento para incluir su centro en el proyecto.

Atentamente,

Madrid, 17 de septiembre de 2016.



Christian Puhlmann Brackmann
Universidade Federal do Rio Grande do Sul

✉ christian.brackmann@ufrgs.br

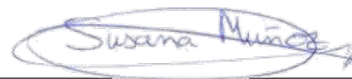
☎ 692 37 29 84



Prof. Dr. Marcos Román González
Investigador Colaborador de la Universidad Nacional de Educación a Distancia

✉ mroman@edu.uned.es

☎ 913 98 90 37



Prof. Dra. Susana Muñoz Hernández
Facultad de Informática de la UPM

✉ susana@fi.upm.es

☎ 913 36 36 69

Más informaciones acerca del proyecto

“Evaluación del Proceso Enseñanza-Aprendizaje del Pensamiento Computacional Sin Ordenador en la Educación Primaria”

¿QUE ES PENSAMIENTO COMPUTACIONAL?

El Pensamiento Computacional es pensar en los problemas de forma que un ordenador consiga solucionarlos. El Pensamiento Computacional es ejecutado por personas y no por ordenadores incluye el pensamiento lógico, la habilidad de reconocimiento de patrones, razonar través de algoritmos, descomponer y abstraer un problema. Actualmente es una de las tendencias mundiales en enseñanza.

¿LOS ALUMNOS VAN A APRENDER A UTILIZAR EL ORDENADOR?

El Pensamiento Computacional jamás puede ser confundido con el manejo de aplicaciones o partes físicas de dispositivos electrónicos (Alfabetismo Digital). Este manejo implica una manera de pensar de forma mecánica como usuario de tecnología, mientras que el pensamiento computacional contempla la capacidad de resolver problemas de forma creativa.

¿CUÁLES SON LOS PRINCIPIOS FUNDAMENTALES DEL PENSAMIENTO COMPUTACIONAL?

El Pensamiento Computacional implica identificar un problema complejo y dividirlo en partes menores y más sencillo de gestionar (DECOMPOSICIÓN). Cada uno de estos problemas menores puede ser analizado individualmente con mayor profundidad, identificando problemas parecidos que ya se han solucionado anteriormente (RECONOCIMIENTO DE PATRONES), centrándose en los detalles que sean importantes, mientras que las informaciones irrelevantes son ignoradas (ABSTRACCION). Finalmente, se pueden crear pasos o reglas simples para resolver cada uno de los subproblemas encontrados (ALGORITMOS).

¿QUÉ BENEFICIOS VAN A OBTENER EL COLEGIO Y LOS ALUMNOS QUE PARTICIPEN EN LA EXPERIENCIA?

Este proyecto fue criado con el propósito didáctico de enseñarlos distintas formas de buscar soluciones para problemas más complejos, facilitando su comprensión en manosear datos digitalizados en un mundo moderno.

Incluyen los siguientes beneficios a los alumnos en corto plazo:

- Comprender un modelo estructurado de pensamiento, auxiliando en el proceso de aprender a aprender;
- Mejorar el proceso de formulación de problemas;
- Perfeccionar la capacidad de abstracción y creatividad;
- Dar énfasis en la creación del conocimiento y no solo en su uso.
- Aprender a tratar con problemas variados y complejos de forma crítica;
- Seguir tendencias mundiales de enseñanza (competencias del siglo XXI);
- Ayudar a atender una fuerte demanda del mercado;
- Aumentar la comprensión de la fusión de la informática con otras áreas.

¿HAY CUSTOS PARA MÍ COLEGIO O ALUMNO?

No. Los materiales son gratuitos y ofrecidos por las universidades.

Apêndice N: Atividades usadas após a aplicação do teste

Exercício 1: Desenhar!

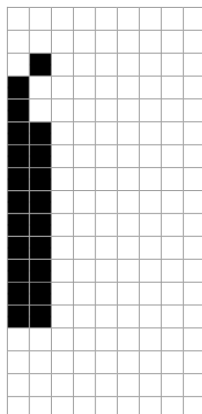
Desenhe como você imagina o funcionamento de um computador por dentro. Como você acredita que a máquina é por dentro? Como funciona? Há magia?

Q QUE SERÁ?

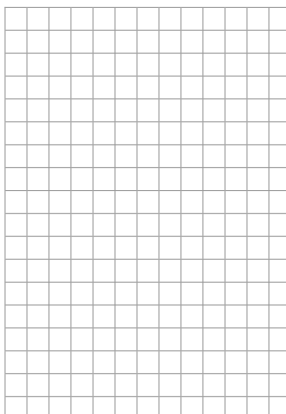


10 Quantos quadrados brancos?

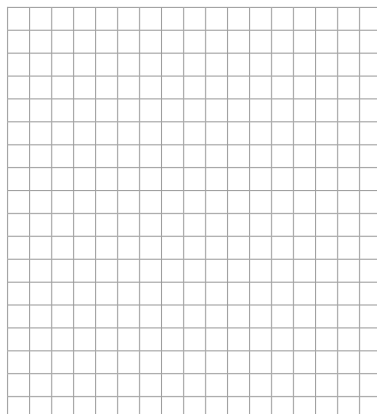
20 Quantos quadrados pretos?



- 4, 11 (4 brancos e 11 pretos)
- 4, 9, 2, 1 (4 brancos, 9 pretos, 2 brancos, 1 preto)
- 4, 9, 2, 1
- 4, 11
- 4, 9
- 5, 7
- 0, 17
- 1, 15



- 6, 5, 2, 3
- 4, 2, 5, 2, 3, 1
- 3, 1, 9, 1, 2, 1
- 3, 1, 9, 1, 1, 1
- 2, 1, 11, 1
- 2, 1, 10, 2
- 2, 1, 9, 1, 1, 1
- 2, 1, 8, 1, 2, 1
- 2, 1, 7, 1, 3, 1
- 1, 1, 1, 1, 4, 2, 3, 1
- 0, 1, 2, 1, 2, 2, 5, 1
- 0, 1, 3, 2, 5, 2
- 1, 3, 2, 5



- 6, 6
- 5, 1, 2, 2, 2, 1
- 6, 6
- 4, 2, 6, 2
- 3, 1, 10, 1
- 2, 1, 12, 1
- 2, 1, 3, 1, 4, 1, 3, 1
- 1, 2, 12, 2
- 0, 1, 16, 1
- 0, 1, 6, 1, 2, 1, 6, 1
- 0, 1, 7, 2, 7, 1
- 1, 1, 14, 1
- 2, 1, 12, 1
- 2, 1, 5, 2, 5, 1
- 3, 1, 10, 1
- 4, 2, 6, 2
- 6, 6

Exercício 2: Explique

O que é a tecnologia? Para que se usa? Quem a utiliza?

Como você se sente com estes exercícios? Faça um círculo ao redor do rosto



Apêndice O: Termo de Consentimento Livre e Esclarecido

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA NA EDUCAÇÃO TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Convidamos seu/sua filho/a a participar da pesquisa “AVALIAÇÃO DO PROCESSO ENSINO-APRENDIZAGEM DO PENSAMENTO COMPUTACIONAL NA EDUCAÇÃO BÁSICA”, que tem como objetivo contribuir para um aprofundamento no tema e auxiliar na fundamentação e argumentação para conscientização dos dirigentes de escolas e até mesmo pais na inserção do Pensamento Computacional na sala de aula.

Neste sentido, a pesquisa pretende levar para a sala de aula diversas atividades, inclusive lúdicas, envolvendo o pensamento crítico juntamente com conceitos da computação como conteúdo complementar ao que é apresentado aos alunos em sala de aula. A pesquisa será composta de 5 sessões de duas horas cada.

Considerada atualmente como uma habilidade essencial para qualquer pessoa, independentemente da área e não somente as pessoas que estão de uma forma ou de outra, relacionadas com a informática, está presente em currículos de outros países de primeiro mundo como disciplina obrigatória.

A experimentação em que os alunos tomarão parte é voluntária. Uma vez de acordo em participar, os voluntários farão parte das intervenções em sala de aula, resolvendo problemas de Pensamento Computacional (PC) e demais atividades relacionadas a abordagem do PC em diversas áreas do conhecimento. Este projeto foi desenvolvido com o propósito didático de ensiná-los diferentes formas de encontrar soluções para problemas mais complexos, facilitando sua compreensão ao manusear dados digitalizados. Eles também deverão responder questionários com perguntas referentes ao PC. Todas as sessões serão gravadas em vídeo apenas para fins de monitoramento dos alunos.

Desta maneira, entende-se que a natureza da pesquisa apresenta riscos pequeno para os alunos, dentre os quais, há o risco mínimo das imagens e vídeos por eventual quebra de sigilo e confidencialidade. Todavia, salienta-se que os dados da pesquisa estarão sempre sob sigilo ético. Não serão mencionados nomes de participantes em qualquer apresentação oral ou trabalho acadêmico que venha a ser publicado com base na pesquisa. O participante também tem total liberdade de recusar tomar parte da pesquisa em qualquer uma de suas fases, sem que isto lhe acarrete qualquer prejuízo ou constrangimento, porém sua participação é essencial para o bom andamento da pesquisa. Além disso, a pesquisa também não envolve conteúdos inapropriados para a idade do público-alvo ou esforço físico que possa levar a lesões, porém não se deve descartar completamente essa possibilidade.

É possível ainda que alguns se sintam alvo da crítica dos colegas por não obter uma pontuação semelhante à deles. Todavia, deve-se ter em mente que devido ao fato da pontuação ser mantida em sigilo até o final da pesquisa e entregues ao professor, minimizando-se potenciais discriminações dos colegas em razão de erros cometidos, dado que, a natureza experimental de tal ambiente torna o erro parte do processo de aprendizado.

No que diz respeito aos possíveis benefícios da pesquisa para o estudante são, porém, não limitados a esses:

- Desenvolvimento do raciocínio lógico;
- Aumento de notas dos estudantes em outras disciplinas como efeito colateral;
- Melhoria no processo de formulação e solução de problemas variados;
- Auxiliar na aproximação da educação brasileira aos novos padrões de ensino de países desenvolvidos;
- Aperfeiçoar a capacidade de abstração e criatividade;
- Dar ênfase na criação do conhecimento e não apenas no seu uso.

Os procedimentos da Pesquisa obedecem aos Critérios de Ética em Pesquisa com Seres Humanos conforme Resolução nº 466/12 e 510/2016 do Conselho Nacional de Saúde, de forma que nenhum dos procedimentos usados oferece risco à dignidade do participante. As informações e os materiais de registro coletados (documentos, vídeos, etc.) durante a pesquisa serão utilizados apenas para fins de monitoramento, investigação e de produção de conhecimento. Todo o material obtido será arquivado por um período de até 5 anos a contar do término desta pesquisa junto ao programa de Programa de Pós-Graduação Em Informática na Educação da UFRGS, situado a Avenida Paulo Gama, 110 – prédio 12105 – 3º andar sala 332, CEP: 90040-060 Porto Alegre – RS.

Caso você tenha dúvidas ou deseje maiores esclarecimentos, por favor, entre em contato com o professor responsável pela pesquisa, Prof. Dr. Dante Barone, pelo telefone (51) 3308-9478, com o pesquisador titular, o doutorando Christian Puhlmann Brackmann, pelo telefone (55) 98100-1819, ou com o Comitê de Ética em Pesquisa da UFRGS, pelo telefone (51) 3308-3738.

Data : ____ / ____ / ____

EU _____,
 concordo com a participação do meu filho/a
 _____ nesta pesquisa.

Obs.: A concordância em participar da pesquisa não implica em dolo caso o mesmo opte por se retirar da mesma em momento futuro. A participação na pesquisa garante a seu filho o uso dos materiais desenvolvidos e distribuídos pelos pesquisadores na escola. O nome de seu filho/a não será veiculado em qualquer material gerado a partir desta pesquisa.

Apêndice P: Planejamento Pedagógico / Plano de Aula



PLANEJAMENTO SIMPLIFICADO

- Teste de Pensamento Computacional (Pré-teste)

- Primeiro Encontro: Aula de Pensamento Computacional Desplugado
 - Atividade 01: Decomposição
 - Atividade 02: Mapa de Rotas da Turma da Mônica

- Segundo Encontro: Aula de Pensamento Computacional Desplugado
 - Atividade 03: Tetris - Instruções
 - Atividade 04: Controle Remoto

- Terceiro Encontro: Aula de Pensamento Computacional Desplugado
 - Atividade 05: Os Elefantes
 - Atividade 06: Bugs
 - Atividade 07: Condicional/Trajes

- Quarto Encontro: Aula de Pensamento Computacional Desplugado
 - Atividade 08: Cupcakes
 - Atividade 09: Tetris 2

- Quinto Encontro: Aula de Pensamento Computacional Desplugado
 - Atividade 10: Autômatos da Mônica

- Teste de Pensamento Computacional (Pós-teste)

- Sexto Encontro: Aulas de Scratch

- Sétimo Encontro: Aulas de Scratch