



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

RAFAEL MARIMON BOUCINHA

APRENDIZAGEM DO PENSAMENTO COMPUTACIONAL E
DESENVOLVIMENTO DO RACIOCÍNIO

PORTO ALEGRE

2017

APRENDIZAGEM DO PENSAMENTO COMPUTACIONAL E
DESENVOLVIMENTO DO RACIOCÍNIO

RAFAEL MARIMON BOUCINHA

Proposta de 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: Dr. Dante Augusto Couto Barone

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

CIP - Catalogação na Publicação

Boucinha, Rafael Marimon
APRENDIZAGEM DO PENSAMENTO COMPUTACIONAL E
DESENVOLVIMENTO DO RACIOCÍNIO / Rafael Marimon
Boucinha. -- 2017.
151 f.
Orientador: Dante Augusto Couto Barone.

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,
BRRS, 2017.

1. Pensamento Computacional. 2. Inteligência. 3.
Avaliação Cognitiva. I. Barone, Dante Augusto Couto,
orient. II. 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
RAFAEL MARIMON BOUCINHA**

Às nove horas do dia vinte e um 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: Alberto Bastos do Canto Filho, Denise Ruschel Bandeira e Érico Marcelo Hoff do Amaral, para a análise da defesa de Tese de Doutorado intitulada "*Aprendizagem do Pensamento Computacional e Desenvolvimento do Raciocínio*", do doutorando do Programa de Pós-Graduação em Informática na Educação Rafael Marimon Boucinha, sob a orientação do Prof. Dr. Dante Augusto Couto Barone. A Banca, reunida, após a apresentação e arguição, emite o parecer abaixo assinalado.

Considera a Tese aprovada

(X) sem alterações;

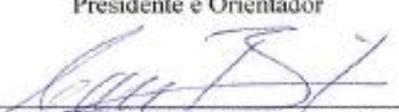
() e recomenda que sejam efetuadas as reformulações e atendidas as sugestões contidas nos pareceres individuais dos membros da Banca;

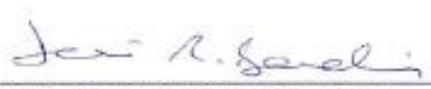
Considera a Tese reprovada.

Considerações adicionais (a critério da Banca):

A banca deliberou que este trabalho é considerado como uma tese de doutorado, tendo como pontos fortes tratar de um tema inovador, abordando três disciplinas diferentes (Educação, Psicologia e Informática), além de propor uma intervenção de pesquisa. A banca aceita o aceite de uma sessão de discussões após os resultados estarem publicados em revista científica.


Prof. Dr. Dante Augusto Couto Barone
Presidente e Orientador


Prof. Dr. Alberto Bastos do Canto Filho
PPGIE/UFRGS


Prof. Dr. Denise Ruschel Bandeira
Psicologia/UFRGS


Prof. Dr. Érico Marcelo Hoff do Amaral
UNIPAMPA

AGRADECIMENTO

O término de um doutorado é o fim de um ciclo, um período de quatro anos, idade da minha pequena Luísa. Durante este período, várias vezes lamentei outras atividades que tomavam meu tempo, mas descobri que isso se chamava vida. Gostaria de agradecer a Deus por ter emprego num período de crise, pela minha esposa que me apoiou em todos os momentos e pela família, por acreditar valer a pena disputar meu tempo.

Preciso agradecer ao professor Valdeni Lima que incentivou meu ingresso no PPGIE e a professora Liane Tarouco pelo esforço em orientar meu primeiro projeto. Por alguma razão divina, cada vez que pensei em parar algo acontecia e as coisas simplesmente se reorganizavam. Agradeço ao colega Christian Brackmann por auxiliar na mudança de rumo deste doutorado e ao professor Dante Barone, que acolheu meu projeto e o orientou com maestria.

É necessário registrar meu agradecimento aos colegas da Companhia Estadual de Geração e Transmissão Energia Elétrica, os quais permitiram que eu frequentasse este doutorado.

No ano de 2007, eu convenci meu irmão Eduardo que estava numa cadeira de rodas a voltar a estudar. Como tinha que levá-lo, frequentei com ele o curso de Análise e Desenvolvimento de Sistema, gostaria de agradecer este meu colega, que agora está no plano espiritual, pelos ensinamentos de vida que deixou.

Rafael Boucinha

RESUMO

Esta tese descreve um estudo quase experimental que teve como objetivo: investigar a relação entre a construção do Pensamento Computacional e o desenvolvimento do raciocínio de estudantes dos últimos anos do Ensino Fundamental. A pesquisa foi realizada utilizando um curso de extensão em Desenvolvimento de Games, ofertado em 2 escolas particulares de Porto Alegre, tendo a participação de 50 alunos. A prática de ensino-aprendizagem proposta foi construída com base em pressupostos teóricos da aprendizagem significativa e aprendizagem experiencial. O Pensamento Computacional e o raciocínio dos alunos foram avaliados antes e após o término do curso, sendo utilizados para este fim o Teste de Pensamento Computacional e as provas que compõe a Bateria de Provas de Raciocínio – BPR-5. A análise estatística dos dados permitiu evidenciar um incremento do Pensamento Computacional, bem como do Raciocínio Verbal, Raciocínio Abstrato e Raciocínio Mecânico dos alunos que participaram do experimento. Comprovou-se também uma correlação positiva entre o Pensamento Computacional e os cinco tipos de raciocínio avaliados. Os resultados deste estudo demonstram como a construção do Pensamento Computacional contribuí no desenvolvimento cognitivo dos alunos e é apresentada uma proposta pedagógica que pode servir de referência para novos estudos na área.

Palavras-chave: Pensamento Computacional. Inteligência. Avaliação Cognitiva.

ABSTRACT

This thesis describes a quasi-experimental study aimed to investigate a relationship between the construction of Computational Thinking and the development of students' reasoning in Middle School. A research was carried out during a course about Games Development, offered in two private schools in Porto Alegre, with 50 students. The proposed teaching-learning practice was built on the theoretical assumptions of meaningful learning and experiential learning. Both, Computational Thinking and reasoning, of the students were measured before and after the course, using a Computational Thinking Test and a set of reasoning evidence tests (BPR-5). The statistical analysis of the data showed an increase in Computational Thinking, as well as Verbal Reasoning, Abstract Reasoning and Mechanical Reasoning of the students participating in the experiment. There was also a positive observation between Computational Thinking and the five types of reasoning. The results of this study demonstrate how the construction of Computational Thinking contributes to the cognitive development of students and presents a pedagogical proposal that can serve as a reference for new studies in the area.

Keywords: Computational Thinking. Intelligence. Cognitive Assessment

LISTA DE ABREVIATURAS E SIGLAS

BNCC – Base Nacional Curricular Comum

BPR5 – Bateria de Provas de Raciocínio

CC – Ciência da Computação

CNCS – Corporação para o Ensino Comunitário Nacional

CT – Computational Thinking

CS – Computer Science

CSTA – Computer Science Teacher Association

ISTE – International Society for Technology in Education

NSF – National Science Foundation

NTIC – Novas Tecnologias da Informação e Comunicação

PC – Pensamento Computacional

PISA – Programme for International Student Assessment

RA – Raciocínio Abstrato

RE – Raciocínio Espacial

RM – Raciocínio Mecânico

RN – Raciocínio Numérico

RV – Raciocínio Verbal

TI – Tecnologia de Informação

TIC – Tecnologias da Informação e Comunicação

LISTA DE FIGURAS

Figura 1: Nível de Integração do ensino de programação	17
Figura 2: Pilares do Pensamento Computacional	22
Figura 3: Conceitos e práticas relacionadas ao Pensamento Computacional	23
Figura 4: Exemplo de Projeto analisado no Dr. Scratch.....	26
Figura 5: Sumário de projeto avaliado no com Scrape.....	27
Figura 6: Exemplo de portfólio	28
Figura 7: Exemplo do teste de realização de tarefas	30
Figura 8: Exemplo de questão utilizada no Teste de Pensamento Computacional	31
Figura 9: Evolução Teoria CHC.....	36
Figura 10: Prova de Raciocínio Abstrato (RA)	39
Figura 11: Prova de Raciocínio Verbal (RV)	40
Figura 12: Prova de Raciocínio Espacial (RE).....	40
Figura 13: Prova de Raciocínio Numérico (RN).....	40
Figura 14: Prova de Raciocínio Mecânico (RM)	41
Figura 15: Desempenho na questão 5: Seja Criativo.....	48
Figura 16: Exemplo de carta para apoio a atividade Scratch	49
Figura 17: Avaliação do desempenho	52
Figura 18: Porcentagem de projetos contendo cada conceito computacional	55
Figura 19: Teoria cognitiva da aprendizagem multimídia.....	60
Figura 20: Subsunçor x Objetivo de aprendizagem	65
Figura 21: Aprendizagem subordinada.....	66
Figura 22: Aprendizagem supraordenada.....	67
Figura 23: Aprendizagem combinatória	67
Figura 24: Ciclo de Aprendizagem Experiencial	69
Figura 25: Etapas da pesquisa	71
Figura 26: Análise da relação entre as variáveis	72
Figura 27: Apresentação do trabalho realizado com os grupos.....	74
Figura 28: Folder para divulgação do curso.	76
Figura 29: Gráficos de normalidade Pré-teste e Pós-Teste	83
Figura 30: Distribuição dos escores do TPC	87
Figura 31: Análise Fatorial.....	93
Figura 32: Carga Fatorial.....	94

LISTA DE TABELAS

Tabela 1: Avaliação das questões de 1 a 4	47
Tabela 2: Recursos utilizados	50
Tabela 3: Conceitos apresentados em Projetos Scratch.....	54
Tabela 4: Inscrições dos alunos e amostra.....	76
Tabela 5: Distribuição dos alunos por gênero	77
Tabela 6: Distribuição por ano escolar	78
Tabela 7: Tempo dedicado a jogar videogame	79
Tabela 8: Média dos alunos	81
Tabela 9: Resultado do Teste Shapiro-Wilk.....	82
Tabela 10: Turma A16 Teste-t: duas amostras em par para médias.....	84
Tabela 11: Turma B16 Teste-t: duas amostras em par para médias	85
Tabela 12: Turma A17 Teste-t: duas amostras em par para médias.....	86
Tabela 13: Turma B17 Teste-t: duas amostras em par para médias	86
Tabela 14: Teste-t pareado	87
Tabela 15: Variação das Médias dos Testes de Raciocínio.....	88
Tabela 16: Teste de Shapiro-Wilk	89
Tabela 17: Análise da Médias do Raciocínio Verbal	89
Tabela 18: Análise da Médias do Raciocínio Abstrato	90
Tabela 19: Análise das Médias do Teste de Raciocínio Mecânico	90
Tabela 20: Análise das Médias do Teste de Raciocínio Espacial	91
Tabela 21: Análise das Médias do Teste de Raciocínio Numérico	91
Tabela 22: Análise das Médias da BPR-5	92
Tabela 23: Correlação de Pearson	92
Tabela 24: Cargas Fatoriais Não Rotacionadas e Itens Comuns	94

LISTA DE QUADROS

Quadro 1: Relação do Pensamento Computacional com outras disciplinas escolares	20
Quadro 2: Exemplo de critérios de avaliação utilizada por Brennan	29
Quadro 3: Definição dos dez fatores amplos da teoria CHC.....	37
Quadro 4: Provas e capacidade específica avaliada	39
Quadro 5: Resumo dos conceitos computacionais presentes nas diferentes tarefas.....	46
Quadro 6: Programação de atividades da Oficina de Produção de Jogos	56
Quadro 7: Pressupostos da Teoria de Carga Cognitiva	59
Quadro 8: Escolha profissional	80

SUMÁRIO

1 INTRODUÇÃO	10
2 PENSAMENTO COMPUTACIONAL	15
2.1 CONCEITO	18
2.2 ESTRATÉGIAS PARA AVALIAÇÃO DO PENSAMENTO COMPUTACIONAL ..	25
2.2.1 Avaliação de artefatos	25
2.2.2 Avaliação com uso de testes	29
3 PARADIGMA DIFERENCIAL DA INTELIGÊNCIA	33
3.1 TEORIA BIFATORIAL	34
3.2 TEORIA DAS APTIDÕES PRIMÁRIAS	34
3.3 TEORIA GF-GC	35
3.4 CARROL TEORIA DOS TRÊS ESTRATOS	35
3.5 TEORIA CATTELL-HORN-CARROLL	36
3.6 AVALIAÇÃO DA INTELIGÊNCIA	37
4 ESTUDOS CORRELATOS	42
5 PRESSUPOSTOS PEDAGÓGICOS PARA CONSTRUÇÃO DA METODOLOGIA DE ENSINO-APRENDIZAGEM	57
5.1 OBJETOS DE APRENDIZAGEM	57
5.2 TEORIA DA CARGA COGNITIVA	58
5.3 MOTIVAÇÃO E APRENDIZAGEM	60
5.3.1 Teoria das necessidades	61
5.3.2 Teoria da fixação de objetivos	62
5.3.3 Teoria do reforço	63
5.3.4 Modelo contingencial	63
5.4 APRENDIZAGEM SIGNIFICATIVA	64
5.5 APRENDIZAGEM EXPERIENCIAL	67
6 METODOLOGIA	70
6.1 METODOLOGIA DE ENSINO PROPOSTA	72
6.2 AMOSTRA	75
7 DESCRIÇÃO E ANÁLISE DOS DADOS	81
7.1 AVALIAÇÃO DO PENSAMENTO COMPUTACIONAL E VALIDAÇÃO DA METODOLOGIA DE ENSINO	81
7.2 PENSAMENTO COMPUTACIONAL E CAPACIDADE DE RACIOCÍNIO DOS ALUNOS	88
7.3 RELAÇÃO ENTRE PENSAMENTO COMPUTACIONAL E OS DIFERENTES TIPOS DE RACIOCÍNIO	92
7.4 DISCUSSÃO DOS RESULTADOS	95
8 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	97
REFERÊNCIAS	99

1 INTRODUÇÃO

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

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

Mitch Resnick (2012) ao discutir o conceito de nativos digitais questiona a fluência digital destes jovens e explica que embora tenham nascidos imersos num ambiente tecnológico e demonstrem muita experiência e facilidade em interagir com tecnologias, não tem esta mesma habilidade para criar e se expressar utilizando novas tecnologias. Afirma que são consumidores de tecnologia e não produtores, comparando este fenômeno com alguém que sabe ler, mas não sabe escrever.

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

Este modo de pensar próprio da Ciência da Computação foi nomeado como Pensamento Computacional, termo que se popularizou a partir do artigo denominado "*Computational Thinking*" publicado em 2006 por Jeannette M. Wing. Cabe destacar que algumas ideias divulgadas por Wing já estavam presentes nos experimentos de Papert sobre a linguagem de programação LOGO, bem como a ideia de que habilidades desenvolvidas ao aprender a programar seriam transpostas para outras esferas da vida (PAPERT, 1980).

O Pensamento Computacional tem sido comparado com a alfabetização do século XXI e possuir habilidades básicas de codificação tornou-se algo necessário para muitos postos de trabalho. Estudos da *European Commission* (2015) estimam que 90% das ocupações profissionais hoje em dia requerem competências digitais, incluindo programação.

No relatório “*A National Talent Strategy*”, organizado pela Microsoft, (2012), se estima que entre 2010 e 2020, a economia americana vai produzir anualmente mais de 120.000 postos de trabalho que exigem um grau de bacharel na área de computação, ao passo que a educação superior dos EUA tem produzindo apenas 40.000 graduados em Ciência da Computação (CC), cenário que pode ser um dos problemas mais graves enfrentados pela indústria de tecnologia.

Em 2015, mais de 600.000 empregos na área de tecnologia com altos salários permaneceram vagos Estados Unidos. Para o ano 2018, é projetado que 51 % de todos os empregos serão em campos relacionados à Ciência da Computação (SMITH, 2016).

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

Outro fator que tem preocupado o mercado tem sido o desinteresse pela área de Ciência da Computação, que gera uma baixa ocupação das vagas disponíveis e alta evasão nos cursos da área de tecnologia da informação (SOFTEX, 2012).

A previsão desta carência de profissionais e o entendimento de que a Ciência da Computação (CC) é uma nova habilidade necessária para as oportunidades econômicas e a mobilidade social, fez com que, o governo estadunidense destinasse 4 bilhões de dólares em financiamento para os Estados e 100 milhões de dólares para os distritos escolares expandirem o ensino de CC nas escolas, através da formação de professores, ampliação do acesso a materiais de instrução de alta qualidade e construção de parcerias regionais eficazes. (SMITH, 2016).

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

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

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

Embora não esteja cientificamente comprovado, supõe-se que o processo de resolução de problemas utilizados nas Ciências da Computação possa ser generalizado e transferido para uma ampla variedade de problemas da vida cotidiana. Desta forma, o Pensamento Computacional não seria uma habilidade relacionada exclusivamente ao curso Superior em Ciência da Computação.

Frente à carência global de profissionais prevista para a área de TI é esperado o benefício da empregabilidade para os jovens que aprenderem habilidades de programação desde a educação básica.

No Brasil, a Ciência da Computação não faz parte do currículo das escolas, havendo apenas iniciativas pontuais de ensino de Pensamento Computacional, muitas vezes propostas como atividade extraclasse.

Neste momento, estão sendo discutidas as alterações necessárias na Base Nacional Curricular Comum (BNCC), documento que deverá nortear o currículo das escolas e dentre as propostas de alteração surgiram sugestões para integrar CC no currículo.

Entende-se como relevante elucidar os efeitos da aprendizagem do Pensamento Computacional sob a cognição, pois caso seja identificado a influência no desenvolvimento do raciocínio, a importância desta disciplina na educação básica superaria uma simples demanda de mercado.

Na literatura científica, são raros os estudos que avaliam o Pensamento Computacional com instrumentos objetivos e praticamente não existem pesquisas sobre os impactos desta aprendizagem na cognição. Diante disso, coloca-se o seguinte problema de pesquisa: **Qual a relação entre a construção do Pensamento Computacional e o desenvolvimento do raciocínio?**

Desta questão de pesquisa, ramificam-se outros questionamentos: Como ensinar Pensamento Computacional? Como avaliar o PC dos alunos? Quais efeitos do PC na capacidade

1 <https://www.code.org>
2 <https://www.codecademy.com/pt>
3 <https://coderdojo.com>
4 <https://girlswhocode.com>
5 <http://www.blackgirlscode.com/>

de raciocínio? Quais tipos de raciocínio estão relacionados ao PC? O Pensamento Computacional é um fator independente ou expressa a capacidade de raciocínio?

Com intuito de investigar o problema proposto e as demais questões, foi formulado o Objetivo Geral: **Investigar a relação entre a construção do Pensamento Computacional e o desenvolvimento do raciocínio.**

Já como objetivos específicos, destacam-se os seguintes:

- Desenvolver metodologia para ensino de Pensamento Computacional;
- Avaliar o Pensamento Computacional dos alunos;
- Verificar o efeito do desenvolvimento do Pensamento Computacional na capacidade de raciocínio dos alunos.
- Analisar a relação entre Pensamento Computacional e os diferentes tipos de raciocínio: raciocínio verbal, raciocínio numérico, raciocínio espacial, raciocínio abstrato e raciocínio mecânico.

Formularam-se duas hipóteses que serão devidamente testadas durante a análise dos dados deste estudo:

- O ensino do Pensamento Computacional melhora a capacidade de raciocínio dos alunos;
- Existe uma correlação entre Pensamento Computacional e capacidade de raciocínio.

O texto deste trabalho foi organizado em capítulos a fim de atender a organização necessária, contendo além do capítulo introdutório, um segundo capítulo que trata sobre o termo pensamento computacional e como pode ser avaliado.

No terceiro capítulo será abordado o paradigma diferencial da inteligência e instrumentos de avaliação que serão utilizados nesta pesquisa.

O levantamento de estudos, correlatos a este projeto de pesquisa, foram apresentados no quarto capítulo, com intuito de investigarmos as pesquisas já realizadas, os métodos e resultados obtidos.

Os pressupostos pedagógicos utilizados para construção da metodologia de ensino-aprendizagem de pensamento computacional estão expostos no quinto capítulo, logo após, a metodologia utilizada nesta pesquisa será detalhada no sexto capítulo.

No sétimo capítulo, será apresentada a descrição e análise dos dados. O último capítulo apresenta as considerações finais com base nos objetivos propostos para esta Tese e algumas sugestões para trabalhos futuros.

2 PENSAMENTO COMPUTACIONAL

O termo Pensamento Computacional se disseminou no meio científico a partir do ano de 2006, quando Jeannette M. Wing publicou o artigo: “Computational Thinking”. Embora Wing tenha sido considerada a cientista que cunhou o termo Pensamento Computacional, Papert (1980) foi o precursor na utilização do conceito Pensamento Computacional com a obra: “Mindstorms: Children, Computers, and Powerful Ideas”.

Em suas pesquisas com ensino de programação para crianças, Papert (1980) descreve um conjunto de habilidades cognitivas que seriam desenvolvidas na aprendizagem de programação e também afirma que ao se tornar um programador mais proficiente em LOGO, possibilitaria transferir automaticamente as habilidades cognitivas desenvolvidas para outros contextos, como planejamento e capacidade de resolução de problemas.

Após uma década do primeiro artigo de Jeannette M. Wing, a capacidade de desenvolver programas de computador tem sido considerada uma parte importante da alfabetização na sociedade de hoje e o "Pensamento Computacional coloca-se como uma habilidade fundamental para todos, não apenas para cientistas da computação" (WING, 2006).

O Pensamento Computacional tem exercido influência em todos os campos de atuação, representando um novo desafio educacional para a nossa sociedade, especialmente para os mais jovens.

Pensamento Computacional para todos abrange: entender que aspectos de um problema são passíveis de computação, avaliar ferramentas e técnicas computacionais para solução de um problema, compreender as limitações e o poder de ferramentas e técnicas computacionais; aplicar ou adaptar uma ferramenta computacional ou técnica para um novo uso, reconhecer a oportunidade de usar a computação numa nova situação; aplicar estratégias computacionais, tais como, examinar, dividir e conquistar em qualquer domínio.

Considerando a aplicação do Pensamento Computacional para cientistas, engenheiros e outros profissionais, abrange também: aplicar novos métodos computacionais para seus problemas, reformular problemas para serem passível de estratégias computacionais; realizar novas descobertas científicas através da análise de grandes volumes de dados; formular novas perguntas não convencionais; explicar problemas e soluções em termos computacionais.

Cabe destacar que embora o uso de computadores possa ser um importante facilitador para o desenvolvimento do Pensamento Computacional, o mesmo pode ser desenvolvido sem a utilização de computadores. Nesta linha, os estudos de Tim Bell, Ian H. Witten e Mike Fellows foram adaptados para uso em sala de aula por Robyn Adams e Jane McKenzie (2011)

originando o livro chamado “Unplugged”, o qual se propõe, através de uma série de atividades previamente estruturadas, desenvolver ativamente habilidades de comunicação, resolução de problemas, criatividade e cognição num contexto significativo. Esta obra representa uma proposta concreta de ensino de conceitos computacionais sem a utilização de computadores

Devido à evolução tecnológica, possuir habilidades básicas de codificação tornou-se algo necessário para muitos postos de trabalho. Conforme estudos da *European Commission* (2015), 90% das ocupações profissionais hoje em dia requerem competências digitais, incluindo programação. Como o sistema de ensino é lento para reagir às novas exigências há uma escassez de talentos, que tende a aumentar à medida que teremos mais de 825 mil ofertas de empregos na área de tecnologia até 2020.

Esta previsão de uma grande escassez de mão de obra na área mobilizou as grandes empresas de tecnologia a iniciarem uma verdadeira jornada para mobilizar os governos e sociedade para ensinarem programação.

Nos últimos anos, o ensino de Ciência da Computação na educação básica, tem sofrido profundas mudanças curriculares. Observa-se a retirada de conteúdos que abordam o uso de aplicativos de escritório e uma ênfase no ensino de programação.

Brown (2014) afirma que no Reino Unido (Inglaterra, Escócia, País de Gales e do Norte Irlanda) as alterações curriculares estão se encaminhando para que CC (Ciência da Computação) se torne disciplina obrigatória para todos os alunos a partir dos cinco anos de idade.

A organização europeia Schoolnet publicou recentemente um panorama atualizado da integração formal do ensino de programação em escolas em diversos países da Europa e de Israel. Atualmente, 16 países integram codificação no currículo em nível nacional, regional ou local: Áustria, Bulgária, República Tcheca, Dinamarca, Estônia, França, Hungria, Irlanda, Israel, Lituânia, Malta, Espanha, Polônia, Portugal, Eslováquia e Inglaterra (SCHOOLNET, 2015).

Figura 1: Nível de Integração do ensino de programação

	NATIONAL	REGIONAL	SCHOOL LEVEL	STARTING YEAR
AUSTRIA	●			
BELGIUM (NL)		●		
BULGARIA	●			
CZECH REPUBLIC			●	
DENMARK	●			2014
ESTONIA	●		●	
FINLAND	●	●	●	2016
FRANCE	●			2016
HUNGARY	●			1995
IRELAND	●		●	2014
ISRAEL	●			1976
LITHUANIA	●		●	1986
MALTA	●			1997
POLAND	●			1985
PORTUGAL	●			2012
SLOVAKIA	●		●	1990
SPAIN	●	●		2015
UK (ENGLAND)	●			2014

Fonte: SCHOOLNET (2015)

Cabe destacar que, no Brasil, o ensino de programação e Pensamento Computacional não estão estabelecidos como parte do currículo do Ensino Básico. As produções científicas brasileiras evidenciam que este tema tem sido trabalhado em práticas pontuais ou oficinas extraclasse.

Embora tenha havido muito foco neste campo nos últimos anos, não há consenso no meio científico que delimite o conceito de Pensamento Computacional, como pode ser desenvolvido e de que forma pode ser mensurado.

2.1 CONCEITO

No livro “*Mindstorms: children, computers, and powerful ideas*”, o pesquisador Papert (1980) discute o impacto do computador na vida das pessoas e como seu uso estava influenciando a forma como as pessoas pensam. Ele observou que as crianças ao aprenderem a programar com LOGO, utilizavam modelos de computador para organizar o pensamento, à medida que “eles programam o computador para tomar decisões mais complexas e encontram-se envolvidos na reflexão sobre os aspectos mais complexos de seu próprio pensamento” (PAPERT, 1980, p. 28).

Em diversos trechos da obra, Papert apresenta temas relacionados ao que hoje denominamos Pensamento Computacional, no entanto não há uma preocupação deste autor em definir o conceito. Cabe destacar que o termo “*Computational Thinking*” é mencionado apenas uma vez, referindo-se à penetração do computador na sociedade e a dificuldade de criar uma experiência engajadora com as tecnologias utilizadas nos clubes de programação.

Ainda, na obra de Papert (1980), entende-se o Pensamento Computacional como uma forma de estruturar o pensamento. Este autor relaciona a raciocínio lógico, resolução de problemas e depuração de erros. “Aprender a ser um mestre da programação é aprender para se tornar altamente qualificados a isolar e corrigir os *erros*” (PAPERT, 1980, p. 23).

O conceito Pensamento Computacional se popularizou através de um artigo de Wing (2006), onde a autora afirma que “Pensamento Computacional envolve a resolução de problemas, desenvolvimento de sistemas e a compreensão do comportamento humano, recorrendo aos conceitos fundamentais a Ciência da Computação” (WING, 2006, p. 33).

O Pensamento Computacional está presente quando analisamos um código para além da sua funcionalidade, verificando a simplicidade e a elegância, desta forma considera o uso de recursividade e paralelismo. Este conceito considera o uso de abstração e decomposição utilizada para resolver uma tarefa complexa ou projetar um sistema complexo. (WING, 2006).

Wing (2006, p. 34) finaliza este artigo pioneiro tentando definir limites do conceito e define o que é e o que não é Pensamento Computacional:

- É um conceito que não se restringe a programação;
- Habilidade fundamental;
- Forma de resolver problemas própria do ser humano e não das máquinas;
- É uma forma de pensar que se complementa e combina com pensamento matemático e o modo de pensar das engenharias;

- Ideias e não artefatos são conceitos utilizados pelo ser humano que ultrapassam o *hardware* e *software*;
- Qualquer pessoa pode se beneficiar do pensamento computacional, pois, é algo para todos, em todas as áreas.

Em novo artigo publicado em 2008, Wing destaca que a essência do pensamento computacional é a abstração, dando especial importância ao raciocínio abstrato envolvido:

Abstrações são as ferramentas “mentais” da computação. O poder de nossas ferramentas *mentais* é amplificado pelo poder de nossas ferramentas *metálicas*. Computar é a automação das nossas abstrações. Operação de mecanizar as nossas abstrações, camadas de abstração e seus relacionamentos. A mecanização é possível devido a nossas notações e modelos precisos e exigentes. Automatizar implica a necessidade de algum tipo de computador para interpretar as abstrações (WING, 2008, p. 3718).

O Pensamento Computacional estaria relacionado ao raciocínio humano e a forma como ser humano processa informação. Neste processo de absorver informações a autora destaca a capacidade do ser humano de executar cálculos, tipo de raciocínio que na psicologia é nomeado de raciocínio numérico.

Os seres humanos processam a informação; os seres humanos calculam. Em outras palavras, pensamento computacional não requer máquinas. Também, pode considerar a combinação de um ser humano e uma máquina como um computador [...] (WING, 2008, p. 3719).

A combinação do ser humano com o computador propicia uma potencialização de habilidades que seriam humanas. "Pensamento Computacional está influenciando pesquisas em quase todas as disciplinas, tanto nas ciências e as humanidades" (BUNDY, 2007).

No artigo *Computational Thinking: What and Why?* Wing (2010, p.1) define “Pensamento Computacional como os processos de pensamento envolvidos na formulação de problemas e as suas soluções de modo a que as soluções são representadas de uma forma que pode ser eficazmente levada a cabo por um agente de processamento de informação”, descrevendo a atividade mental na solução de um problema que admite uma solução computacional.

Novamente são destacados processos mentais inerentes ao Pensamento Computacional quando a autora afirma:

o pensamento computacional sobrepõe-se com o pensamento lógico e pensamento sistêmico. Ele inclui pensamento algoritmo e pensamento paralelo, que por sua vez se envolver outros tipos de processos de pensamento, por exemplo, o raciocínio de composição, correspondência de padrões, o pensamento processual, e pensando recursivo (WING, 2010, p.1).

Esta definição evoluiu no artigo de 2014 publicado por Wing, onde Pensamento Computacional é definido como um processo de resolução de problemas que inclui uma série de características, tais como, lógica, sistematização e análise de dados e criação de soluções que utilizam uma série de passos ordenados (ou algoritmos), e disposições, tais como a capacidade de lidar com segurança com a complexidade e problemas em aberto.

Já *Google for Education* descreve Pensamento Computacional como um processo que inclui quatro técnicas computacionais de pensamento: decomposição, reconhecimento de padrões, generalização e abstração, e projeto de algoritmos (Google, 2015).

Neste sentido, entende-se Pensamento Computacional como:

um conjunto de habilidades e técnicas de solução de problemas usadas por engenheiros de *software* para criar os programas dos aplicativos que você utiliza, como a pesquisa, o *e-mail* e os mapas. O pensamento computacional inclui as habilidades e formas de pensar que são usadas ao escrever programas de computador, mas que ultrapassam o uso dos computadores (GOOGLE, 2015).

Para a Royal Society (2012, p. 29) "O Pensamento Computacional é o processo de reconhecer os aspectos da computação no mundo que nos rodeia, e aplicação de ferramentas e técnicas de Ciência da Computação de entender e razão sobre ambos os sistemas e processos naturais e artificiais".

Pensamento Computacional é considerado essencial para o desenvolvimento de aplicações informáticas, mas também pode ser usado para apoiar a resolução de problemas em todas as disciplinas, incluindo matemática, ciências e humanidades (GOOGLE, 2015). No Quadro 1: Relação do Pensamento Computacional com outras disciplinas são apresentados exemplos da manifestação do Pensamento Computacional em outras áreas.

Quadro 1: Relação do Pensamento Computacional com outras disciplinas escolares

CONCEITO PENSAMENTO COMPUTACIONAL	APLICAÇÃO ÁREA DE ASSUNTO
Quebrar um problema em partes ou etapas	Literatura: Quebrar a análise de um poema em análise de metro, rima, aparência, estrutura, tom, dicção e significado.
Reconhecer e encontrar padrões ou tendências	Economia: encontrar padrões de ciclo de ascensão e queda da economia do país.
Desenvolver instruções para resolver um problema ou passos para uma tarefa	Artes Culinárias: Escrever uma receita para que outros possam usar.
Generalizar padrões e tendências em regras, princípios ou ideias	Matemática: Descobrir as regras de fatoração de polinômios de 2ª ordem
	Química: Determinar as regras para a ligação química e interações.

Fonte: <https://computationalthinkingcourse.withgoogle.com/unit?lesson=8&unit=1>

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

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

Essas habilidades são apoiadas e reforçadas por uma série de disposições ou atitudes que são dimensões essenciais do PC, tais como:

- A confiança em lidar com a complexidade;
- A persistência em trabalhar com problemas difíceis;
- Tolerância à ambiguidade;
- A capacidade de lidar com os problemas abertos;
- A capacidade de comunicar e trabalhar com outros para atingir um objetivo comum ou solução.

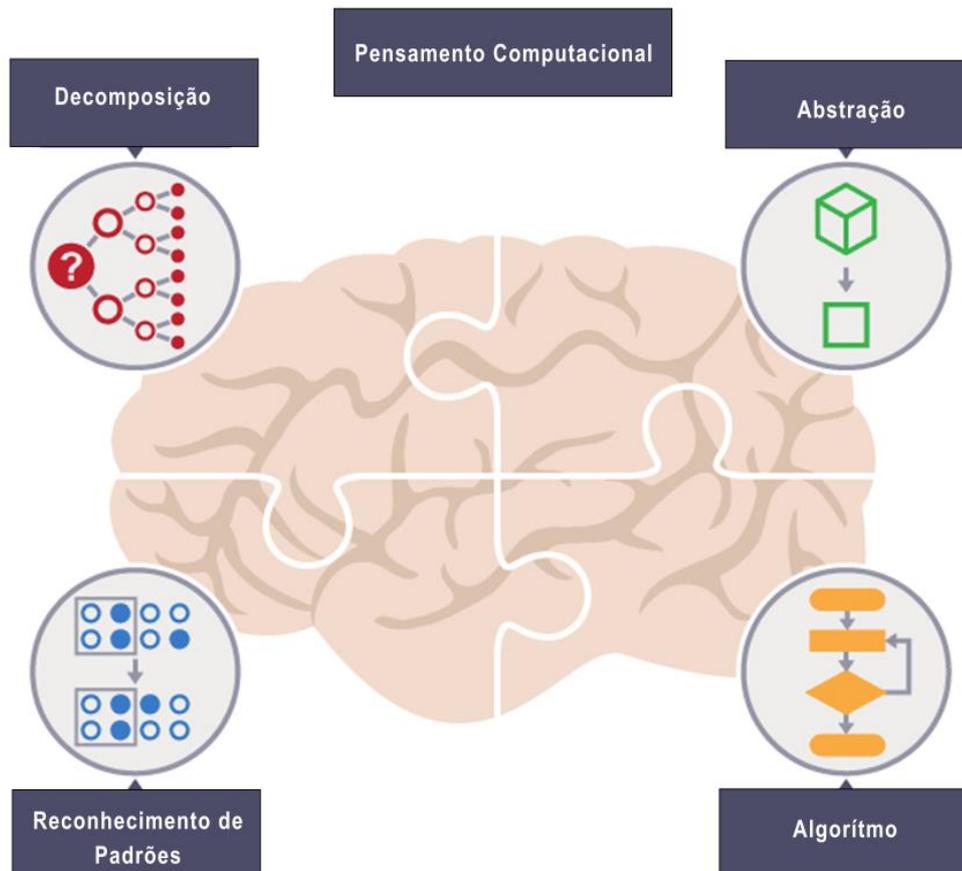
Já o conceito da BBC Learning (2015) considera que existem quatro principais técnicas, que seriam pilares para pensamento computacional:

- Decomposição - quebrar um problema ou um sistema complexo em partes menores, mais manejáveis;
- Reconhecimento de padrões - procurando semelhanças entre problemas;
- Abstração - incidindo sobre a informação importante somente, ignorando detalhes irrelevantes

- Algoritmos - desenvolvimento de uma solução passo a passo para o problema, ou as regras a seguir para resolver o problema

Uma representação destes quatro pilares do pensamento computacional pode ser observada na Figura 2.

Figura 2: Pilares do Pensamento Computacional



Fonte: "Introduction to Computational Thinking". BBC Bitesize. 2015.

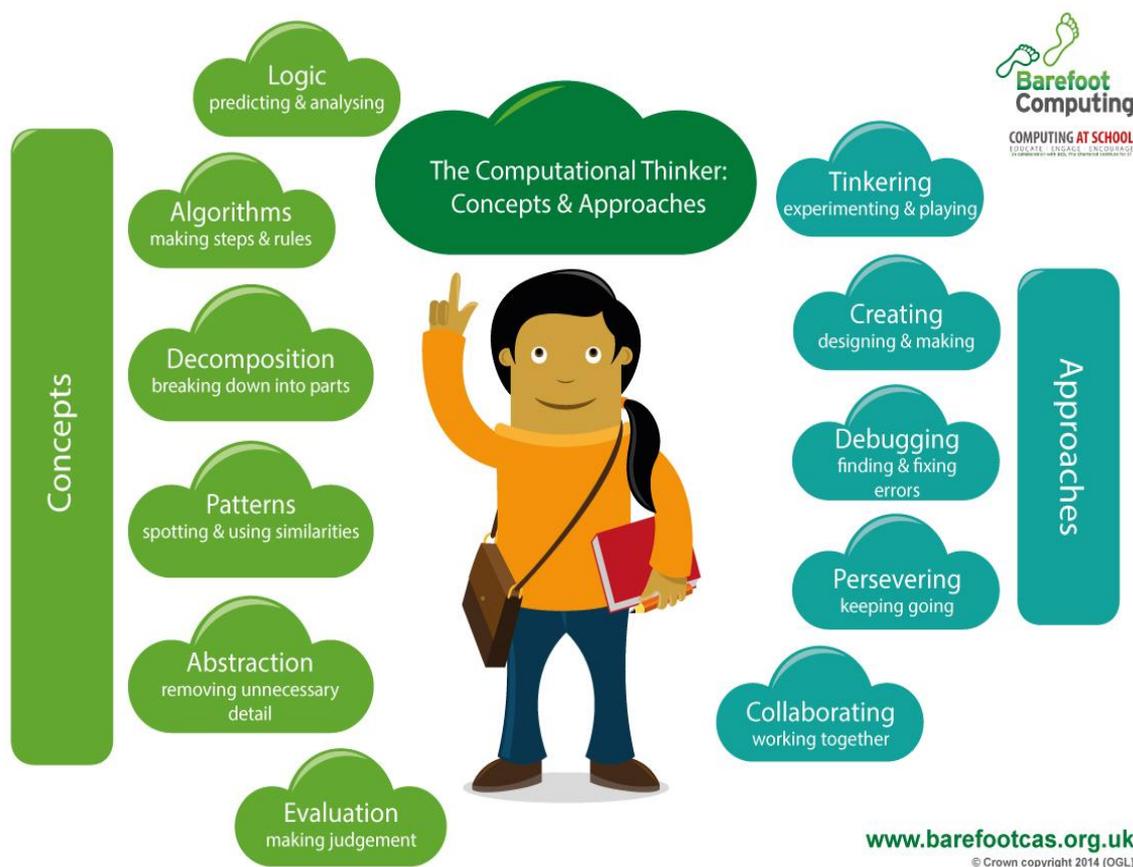
De acordo com Humphreys (2015), Pensamento Computacional é um processo cognitivo envolvendo raciocínio lógico, através do qual problemas são resolvidos, procedimentos e sistemas são melhor compreendidos. Para este autor, estas habilidades de pensamento e resolução de problemas deveria estar presente em todo o currículo e na vida em geral. O PC abrange:

- Capacidade de pensar em termos de algoritmos;
- Capacidade de pensar em termos de decomposição;
- Capacidade de pensar em generalizações, identificando e fazendo uso de padrões;

- Capacidade de pensar em abstrações, a escolha de boas representações;
- Capacidade de pensar em termos de avaliação.

Os pesquisadores do projeto Barefoot (2014), criado com o objetivo de ajudar os professores da escola primária da Inglaterra a se prepararem para o novo currículo de computação, entendem que o Pensamento Computacional envolve 6 diferentes conceitos e 5 abordagens a serem trabalhados. Os conceitos e abordagens encontram-se na Figura 3.

Figura 3: Conceitos e práticas relacionadas ao Pensamento Computacional



Fonte: Barefoot (2015)

Grover & Pea (2013) realizaram uma revisão sobre os trabalhos envolvendo Pensamento Computacional e como resultado desta pesquisa bibliográfica destacaram a falta de consenso internacional sobre uma definição do PC. No entanto, consideram que os seguintes elementos são aceitos como compreendendo PC e formam a base de currículos que visam apoiar a sua aprendizagem, bem como avaliar o seu desenvolvimento:

- Abstrações e generalizações padrão (incluindo modelos e simulações);

- Sistematização e processamento da informação; sistemas de símbolos e representações;
- Noções de algoritmos de fluxo de controle;
- Decomposição de problemas estruturados (modularização);
- Recursividade e pensamento paralelo;
- Lógica condicional;
- Eficiência e restrições de desempenho;
- Depuração e detecção de sistemática de erros.

Barr e Stephenson (2011) no artigo intitulado *Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community* realizaram, com base nas definições da ISTE, CSTA e NSF, uma síntese dos conceitos que envolvem Pensamento Computacional. Neste trabalho são definidos 9 conceitos computacionais:

- Coleta de Dados: O processo de coletar a informação adequada;
- Análise de Dados: Encontrar sentido nos dados, encontrar padrões, e tirar conclusões;
- Representação de Dados: Descrever e organização de dados em gráficos apropriados, texto, palavras ou imagens;
- Decomposição de Problema: Quebrar tarefas em partes menores e gerenciáveis;
- Abstração: Reduzir a complexidade de definir ideia principal
- Algoritmos e Procedimentos: série de passos ordenados tomadas para resolver um problema ou atingir algum fim;
- Automação: ter computadores ou máquinas para fazer tarefas repetitivas ou tediosas;
- Simulação: Representação ou modelo de um processo. Simulação envolve também as experiências em andamento utilizando modelos;
- Paralelismo: Organizar recursos para realizar tarefas de forma simultânea para alcançar um objetivo.

Considerando as diferentes definições para Pensamento Computacional apresentadas neste trabalho, entende-se que esta dificuldade está associada ao fato de agrupar sob o mesmo termo impactos distintos da utilização do computador em nossa sociedade. Pensamento Computacional abarca processos de três categorias distintas: Processos Cognitivos, Processos Comportamentais e Processos Sociais.

- Processos Cognitivos: estão relacionados ao impacto do uso do computador na cognição humana, envolvem abstração, raciocínio lógico, decomposição, algoritmo, depuração de erros e reconhecimento de padrões
- Processos Comportamentais: envolvem as exigências e modificações de comportamentos e atitudes: colaboração, perseverança e experiências de compartilhamento.
- Processos Sociais: referem-se aos impactos do computador sob a sociedade, tais como: automação, simulação, utilização de redes sociais, modificações na organização do trabalho e a influência nos demais ramos do conhecimento.

O Pensamento Computacional seria um conjunto de transformações observadas na forma de pensar, agir e se comportar socialmente, decorrente da utilização dos computadores. Com esta proposta de categorização, torna-se mais fácil para delimitar o campo de estudo de novas pesquisas que versem sobre Pensamento Computacional. O escopo desta pesquisa, por exemplo, se limita a analisar o Pensamento Computacional no que se refere aos processos cognitivos envolvidos.

2.2 ESTRATÉGIAS PARA AVALIAÇÃO DO PENSAMENTO COMPUTACIONAL

Para Werner (2012) um dos fatores que dificulta a abrangência do Pensamento Computacional no Ensino Médio é a falta de avaliações. Wilson (2010, p.16) destaca em seu artigo *“Running on Empty: Failure to Teach K-12 Computer Science in the Digital Age”* que “Avaliações para aprendizagem de Ciência da Computação são praticamente inexistentes, colocando estes cursos numa desvantagem significativa para programas de financiamento que necessitam de dados e para administradores demonstrarem resultados” recomendando que os Estados desenvolvam avaliações científicas para medir o desempenho dos alunos nestes programas.

Com a integração da disciplina de Ciências da Computação nas escolas dos Estados Unidos da América (EUA) e Europa, surgiu a necessidade de estabelecer o currículo mínimo para cada nível de ensino, bem como criar instrumentos adequados para avaliar os alunos.

2.2.1 Avaliação de artefatos

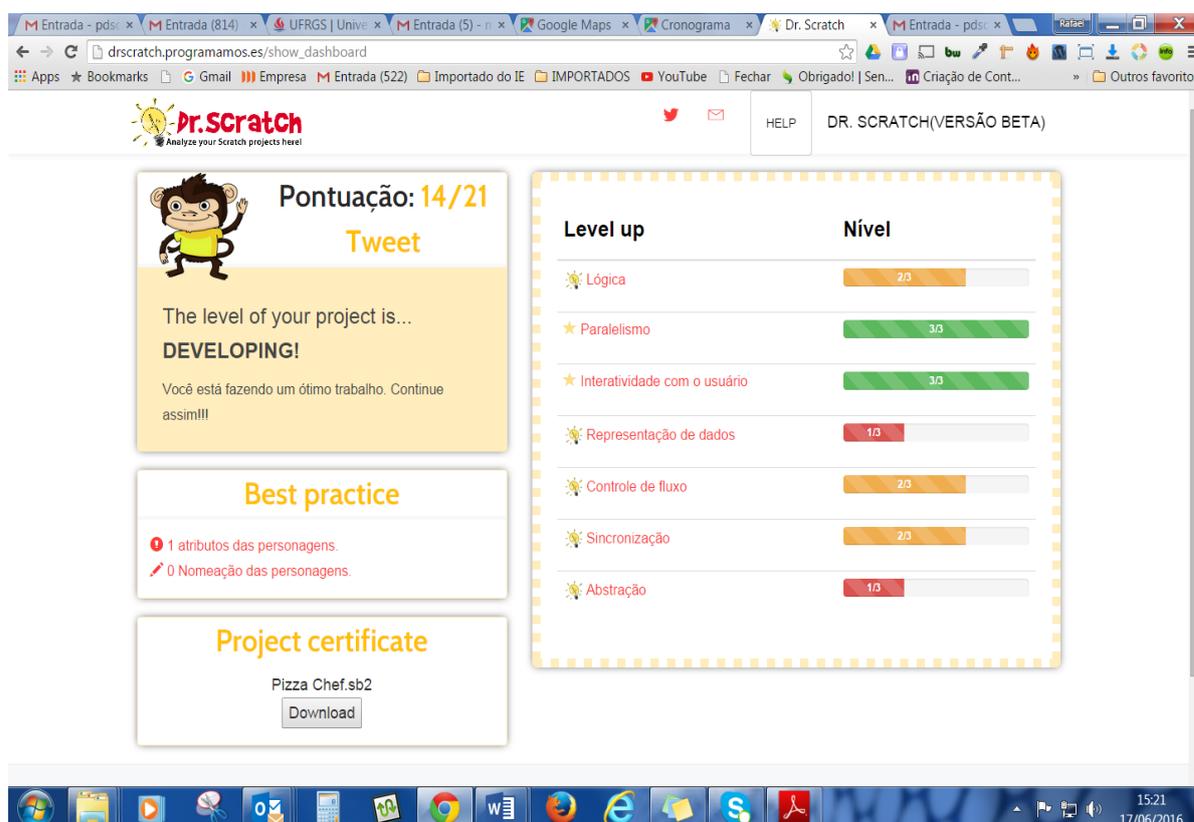
Uma das formas utilizadas para avaliar o desenvolvimento do Pensamento Computacional, difundida pelo grupo de pesquisadores do MIT, responsáveis pelo Scratch, é a avaliação dos artefatos computacionais, ou seja, o professor avalia o código do projeto entregue pelo aluno a fim de identificar os conceitos computacionais utilizados (BRENNAN, 2012).

Ramos (2015) explica que no caso do Scratch, a avaliação do artefato envolve a leitura dos blocos de programação, a interpretação de como o programa foi estruturado e os conceitos computacionais utilizados. Um problema decorrente nesta forma de avaliação é que, quando se trata de avaliar manualmente os blocos utilizados implica em muitas horas de trabalho para o docente.

Esta dificuldade de avaliação dos artefatos motivou o desenvolvimento de ferramentas para análise e avaliação automatizada dos projetos criados com Scratch, dentre as alternativas existentes podemos destacar: Dr. Scratch, Scrap e Hairball.

O Dr. Scratch é um *software open source* destinado a analisar projetos desenvolvidos com Scratch, gerando como resultado um *feedback* sobre a qualidade do programa (MORENO-LEÓN; ROBLES, 2014). Este *software* avalia a lógica, paralelismo, interatividade com o usuário, representação de dados, controle de fluxo, sincronização e abstração. Na figura 5 podemos visualizar um exemplo de avaliação realizada pelo Dr. Scratch.

Figura 4: Exemplo de Projeto analisado no Dr. Scratch

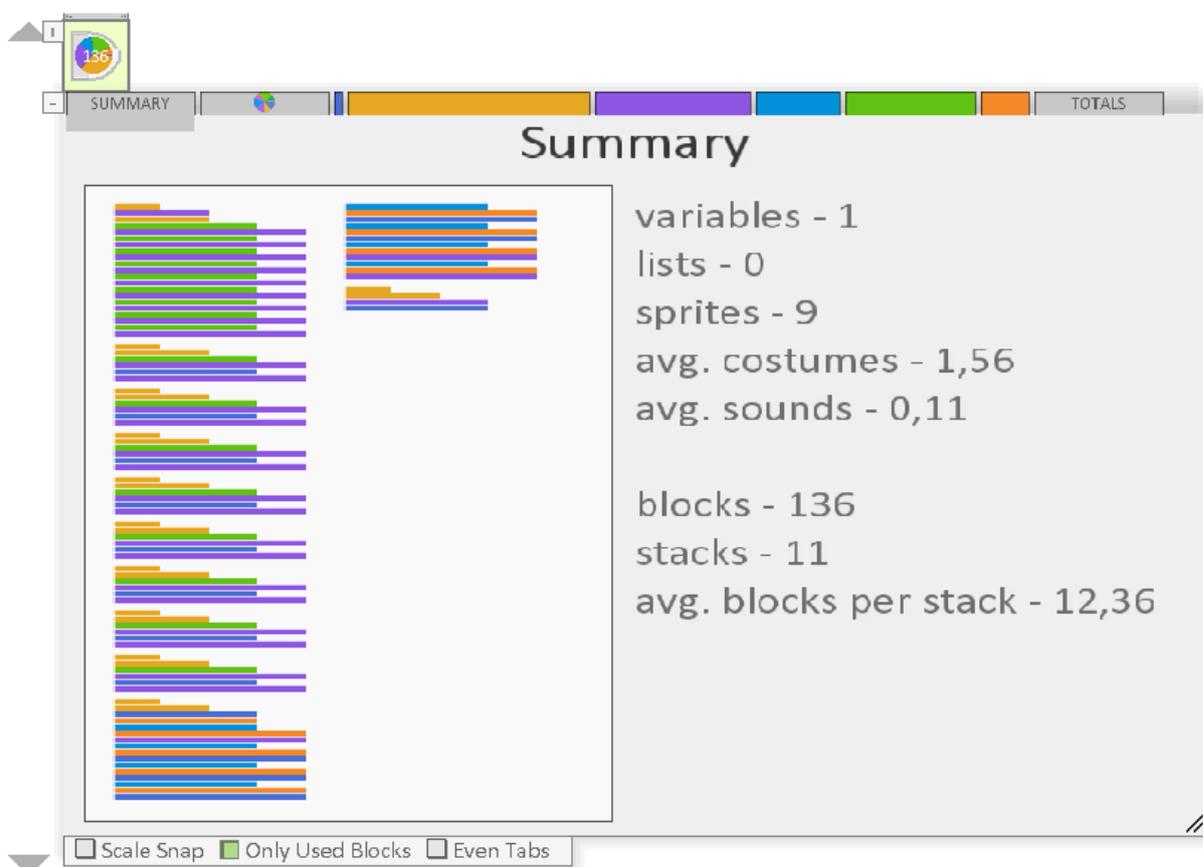


Fonte: desenvolvido pelo autor com aplicativo <http://drscratch.programamos.es/>

Outra alternativa para avaliação automatizada é o Scrape, uma ferramenta desenvolvida para calcular os blocos de programação. Segundo consta no site da ferramenta o Scrape:

é uma ferramenta de análise que permite que você tome projetos de suas unidades locais e visualizar dados concretos de projetos individuais ou de grupos de projetos. Ela também lhe dá a capacidade de comparar qualquer número de projetos, agrupados ou individualmente, a um outro (SCRAPE, 2016).

Figura 5: Sumário de projeto avaliado no com Scrape

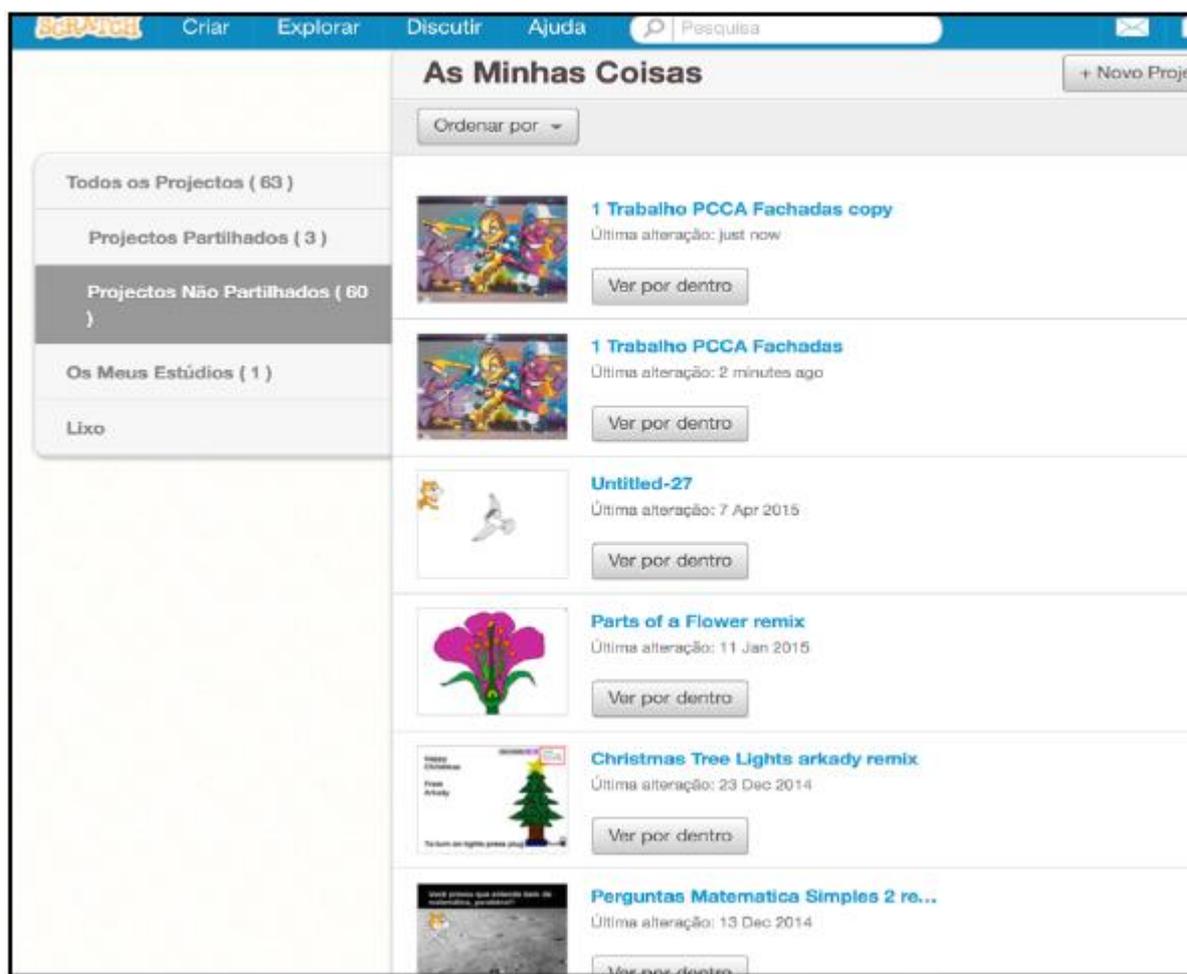


Fonte: <http://happyanalyzing.com/downloads/scrapeLocal/>

O Hairball, disponível em <https://github.com/ucsb-cs-education/hairball>, é uma ferramenta de análise estática, que fornece uma estrutura extensível para analisar automaticamente programas desenvolvidos em Scratch.

Para Ramos (2013) a avaliação de portfólios, o conjunto de projetos que o aluno desenvolveu, permite acompanhar o progresso de um estudante desde patamares iniciais até níveis mais complexos. A figura 6, apresentada a seguir, demonstra um exemplo de portfólio no *software* Scratch.

Figura 6: Exemplo de portfólio



Fonte: Ramos (2013)

Brennan (2012) utiliza entrevistas acerca dos portfólios dos alunos para compreender os conceitos envolvidos em cada projeto. Esta técnica, nomeada de entrevista baseada em artefato, embora agregue dados relevantes, demanda muito tempo.

A abordagem de entrevista baseada em artefato é demorada, à medida que exige pelo menos uma hora com o Scratch. Além de aumentar a carga de tempo, as entrevistas, idealmente, devem ser realizadas várias vezes ao longo do tempo para se ter um retrato do desenvolvimento (BRENNAN, 2012, p.18)

No trabalho de Brennan (2012), também foi utilizada uma tabela com critérios qualitativos para avaliação de projetos, os quais serviram como um roteiro para entrevistas. Nesta proposta avaliativa, após interrogar os alunos acerca do projeto desenvolvido, o pesquisador classifica qualitativamente o nível que cada objetivo foi atingido a exemplo do que está apresentado no quadro 2:

Quadro 2: Exemplo de critérios de avaliação utilizada por Brennan

ABSTRAIR E MODULARIZAR	BAIXO	MÉDIO	ALTO
Como foi decidido que sprites eram necessários para o projeto e onde eram utilizados	Não descreve que sprites foram selecionados	Fornece uma descrição geral da decisão de escolher certos sprites	Dá uma explicação detalhada acerca de como selecionou os sprites em função do objetivo do projeto
Como foi decidido que scripts eram necessários para o projeto e onde eram utilizados	Não descreve que scripts foram criados	Fornece uma descrição geral da decisão de criar scripts	Dá uma explicação detalhada acerca de como criou os scripts em função do objetivo do projeto
Como foram organizados os scripts de forma a terem significado para o estudante e para os outros	Não descreve como os scripts foram organizados	Fornece uma descrição geral da forma como foram organizados scripts	Dá uma explicação detalhada acerca de como organizou os scripts e porquê.

Fonte: Adaptado a partir de Brennan (2012).

Os artefatos construídos pelos alunos têm sido largamente utilizados para mensurar os conceitos computacionais aprendidos, embora sejam uma fonte rica para coleta de dados, sua análise demanda muito tempo do pesquisador e prejudica seu uso em sala de aula.

Cabe destacar que o aluno pode dominar um conceito computacional e não empregar em nenhum projeto, deste modo, mesmo que tenha o domínio deste conceito não será possível observá-lo pela avaliação dos artefatos construídos. Outra limitação na avaliação de artefatos é que não permite avaliar conhecimentos anteriores a prática de programação, à medida que depende de projetos como material de análise.

2.2.2 Avaliação com uso de testes

A utilização de testes padronizados para avaliar Pensamento Computacional, contribuem para uma análise objetiva de práticas pedagógicas, pois permitem comparar resultados de testagens realizadas antes e após uma intervenção.

Um dos trabalhos pioneiros envolvendo testes foi desenvolvido Gouws (2013), o qual realizou uma pesquisa para investigar o PC de estudantes do primeiro ano da universidade. A partir de um banco de itens liberados por “*La Computer Olympiad Talent Search*”¹, uma olimpíada de informática, foi elaborado para esta pesquisa o “*Test for Computational Thinking*”.

¹ Disponível em <http://www.olympiad.org.za/talent-search/>

Na Universidade de Kentucky, Walden (2013) investigou a relação entre Pensamento Computacional e Pensamento Crítico. Para esta pesquisa utilizaram um teste validado para Pensamento Crítico e desenvolveram um instrumento, com objetivo de avaliar o domínio de conceitos computacionais.

Uma avaliação de Pensamento Computacional em grande escala foi realizada em Taiwan, entre alunos de cursos técnicos e de graduação, utilizando para isso 17 tarefas extraídas do “*International Bebras Contest*”¹, um concurso de conhecimento de informática para estudantes de todas idades.

Na Colômbia, Bernarte (2016) desenvolveu um instrumento para investigar o desenvolvimento do Pensamento Computacional em estudantes de ensino fundamental, com o uso do Scratch, parte do projeto de pesquisa que integra: “*El uso de Scratch como promotor del Desarrollo del Pensamiento Computacional en Niños*” realizado pela Universidad Icesi no Instituto Nuestra Señora de la Asunción (INSA).

As avaliações consistiam em um conjunto de 5 tarefas, sendo muitas delas compostas de questões dissertativas e, por este motivo, dificultando a aplicação e correções em grandes grupos. Um exemplo destas tarefas pode ser observado na figura 8.

Figura 7: Exemplo do teste de realização de tarefas

Tarea 3. Organiza los animales

En este cuadro hay un grupo llamado nombres de animales:

Nombres de animales

caballo, oso, abeja, tigre, piraña, perro, zancudo, serpiente, mosca, lombriz, ballena, águila, gato, pingüino, rana, lobo, delfin, vaca, león, oveja, búho, cocodrilo.

Divide los animales en tres grupos y escribe cada grupo de animales dentro de los cuadros de abajo

1.	2.	3.

Debes poner un nombre diferente a cada grupo, que identifique los animales que pusiste ahí.

Fonte: Bernarte (2016)

1 O material do Bebras está disponível em <http://www.bebas.org>

Diante da ausência de instrumentos destinados a avaliar estudantes de ensino médio, Roman (2015) desenvolveu o Teste de Pensamento Computacional, contendo 28 itens com 4 alternativas de resposta (apenas 1 correta), com tempo máximo de 45 minutos para conclusão da tarefa. Este teste é destinado a crianças em idade escolar entre 12 e 13 anos (1ª e 2ª ESO) e tem como objetivo medir o nível de desenvolvimento do Pensamento Computacional do sujeito.

Considera-se neste teste, o PC como a capacidade de formular e resolver problemas, baseando-se em conceitos fundamentais da Computação e usando sintaxe lógica de linguagens de programação de computador. A figura 8 apresenta um exemplo de questão utilizada no Teste de Pensamento Computacional.

Figura 8: Exemplo de questão utilizada no Teste de Pensamento Computacional

<p><i>¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?</i></p>	<p>Opción A</p> <pre> avanzar girar a la izquierda ⌵ avanzar avanzar </pre>	<p>Opción B</p> <pre> avanzar girar a la derecha ⌵ avanzar avanzar </pre>
	<p>Opción C</p> <pre> avanzar avanzar girar a la izquierda ⌵ avanzar </pre>	<p>Opción D</p> <pre> avanzar avanzar girar a la derecha ⌵ avanzar </pre>

Fonte: <http://goo.gl/IYEKMB>

Todos os itens deste instrumento estão alinhados com normas estabelecidas pela CSTA (ROMAN, 2015) para a educação em Ciência da Computação nestas idades, sendo dispostos em dificuldade crescente, cada item aborda um ou mais dos setes conceitos computacionais:

- Direções básicas (4 itens);
- Loop – Repetir Vezes (4 itens);
- Loop 'repetir até ' (4 itens);
- Condicional simples – if (4 itens);
- Condicional composto – if /else (4 itens);
- Enquanto - 'while" (4 itens);

- Funções simples (4 itens).

Considerando as qualidades apresentadas por este teste e a inexistência de instrumentos similares, nesta pesquisa foi o instrumento utilizado para mensurar o PC.

3 PARADIGMA DIFERENCIAL DA INTELIGÊNCIA

Primi (2002) afirma que a psicologia busca há décadas responder à pergunta sobre a natureza da inteligência, sendo este o tema central de inúmeras pesquisas. Consultando a base de dados da Associação Americana de Psicologia PsycINFO¹ da Associação Americana de Psicologia estão disponíveis 131.176 publicações com a palavra inteligência em seu título, revelando uma multiplicidade de teorias.

Para facilitar a compreensão das teorias acerca da inteligência humana, Afonso (2007) propõe classificar as diferentes teorias em 4 paradigmas:

- Paradigma Biológico: remete para a compreensão da inteligência enquanto fenômeno decorrente de fatores biológicos, desde o nível neuronal – anatomia, fisiologia e funcionamento do sistema nervoso, até os níveis mais elementares, genético e bioquímico, ou mais macroscópicos, desenvolvimentista e evolucionista.
- Paradigma Construtivista ou Psicogenético: considera a inteligência como uma forma de adaptação ao meio, em que o conhecimento é construído pelo indivíduo, através dos dois processos complementares de assimilação e acomodação;
- Paradigma Informacional: o qual toma como metáfora o processamento computadorizado de dados, procura entender a inteligência em termos de processos mentais de tratamento da informação
- Paradigma Diferencial: emerge das evidências de diferenças individuais no funcionamento cognitivo, constatado através de provas e uso instrumentos psicológicos.

O presente estudo, a medida que se propõe a mensurar efeitos do ensino do Pensamento Computacional em aspectos da inteligência, estará embasado no Paradigma Diferencial de inteligência.

De acordo com Primi (2003) a abordagem psicométrica utiliza um conceito de inteligência baseado na análise fatorial, a qual se sustenta nas diferenças individuais reveladas através de centenas de testes criados para avaliar as capacidades cognitivas.

Para compreensão da abordagem diferencial ou psicométrica, torna-se importante apresentar as principais teorias desta abordagem nos subcapítulos a seguir.

¹ Consulta realizada em 21/07/2017 disponível em: <<http://psycnet.apa.org/search/results?id=a164ee68-ba21-551e-7b7f-875535d982ac>>

3.1 TEORIA BIFATORIAL

Spearman (1904, 1927) descobriu em seus experimentos que as pontuações de testes de diferentes atividades intelectuais tinham uma correlação entre as demais que permaneciam constantes. Para este autor, todos ramos da atividade intelectual teriam uma função fundamental comum, a qual denominou de Fator Geral ou Fator G.

Desta forma, na resolução de um teste, dois tipos de fatores estariam presentes: um fator de inteligência geral (Fator G) e outros fatores específicos (Fator S). O desempenho em qualquer medida de inteligência estaria associado a inteligência geral e habilidades específicas, por esta razão a teoria de Spearman ficou conhecida como teoria dos 2 fatores ou Teoria G.

3.2 TEORIA DAS APTIDÕES PRIMÁRIAS

Thurstone (1938) utilizando novos métodos de análise fatorial, formulou uma nova teoria denominada *Teoria das Aptidões Primárias*, na qual identificou a existência de 7 fatores primários independentes:

1. Compreensão verbal (V): medida por testes de vocabulário;
2. Fluência verbal (W): medida por testes (cronometrados) para mensurar a produção e compreensão da linguagem escrita;
3. Raciocínio (R): capacidade para tirar conclusões a partir de informações gerais e conclusões a partir de exemplos particulares;
4. Visualização espacial (S): medida por testes que exigem a rotação mental de figuras;
5. Aptidão numérica (N): capacidade de fazer cálculos e resolver operações aritméticas simples;
6. Memória (M): capacidade de reter e recordar informações;
7. Rapidez perspectiva (P): medida por teste que exige o reconhecimento de diferenças e semelhanças entre figuras.

Thurstone (1938) acreditava que destes fatores primários, seriam derivados mais de 200 fatores da inteligência e não existiria um Fator Geral, no entanto, os resultados de suas pesquisas demonstraram uma grande dificuldade de identificar certos fatores, bem como uma certa correlação entre os fatores já identificados.

Em 1941, Thurstone reconheceu a correlação entre as aptidões primárias e a possível identificação, em análise fatorial de segunda ordem, de um fator geral da inteligência.

3.3 TEORIA GF-GC

Pesquisando a análise das correlações entre as capacidades mentais primárias de Thurstone e o fator g da Teoria bifatorial de Spearman, Cattell constatou a existência de dois fatores gerais: inteligência fluida e inteligência cristalizada criando um modelo hierárquico de inteligência chamado de *Teoria Gf-Gc*, referindo-se à inteligência fluída e inteligência cristalizada. (SCHELINI, 2006).

Neste modelo teórico, a Inteligência Fluida (Gf) está associada a componentes não verbais, sendo pouco dependente de conhecimento prévio e da influência da cultura. Caracteriza-se por operações mentais que são utilizadas frente a uma tarefa nova, sendo determinada por aspectos biológicos e por este motivo mais suscetível a lesões e alterações orgânicas que inteligência Cristalizada.

A Inteligência Cristalizada (Gc) representa tipos de capacidades exigidas na solução da maioria dos problemas cotidianos complexos, sendo conhecida como “inteligência social” ou “senso comum”. Este fator é desenvolvido a partir de experiências culturais e educacionais, estando presente na maioria das atividades escolares, motivo pelo qual tende a evoluir com aumento da idade.

Este modelo recebeu importantes contribuições de Horn (1965), aluno de Cattell, o qual expandiu o modelo Gf-Gc adicionando novas capacidades cognitivas: processamento visual, memória a curto prazo, armazenamento e recuperação a longo prazo; velocidade de processamento; rapidez para a decisão correta; processamento auditivo; conhecimento quantitativo e leitura-escrita.

3.4 CARROL TEORIA DOS TRÊS ESTRATOS

Carrol propôs uma teoria que explica a inteligência através de um modelo hierárquico composto por uma hierarquia com três níveis ou estratos (SCHELINI, 2006; ALMEIDA, 2002).

- Camada I: formada por capacidades específicas, as quais representam especializações das capacidades, refletindo os efeitos da experiência e da aprendizagem.
- Camada II: capacidades amplas ou gerais, que exercem influência numa grande variedade de comportamentos, sendo composta por 8 fatores: Inteligência fluida; Inteligência cristalizada; Memória e aprendizagem; Percepção visual; Percepção auditiva; Recuperação da informação; Rapidez cognitiva; Rapidez de decisão.

- Camada III: relativa a uma única capacidade geral ou fator geral, chamado de Fator G.

3.5 TEORIA CATTELL-HORN-CARROLL

Em 1998, McGrew e Farnham propuseram a integração das teorias GF-GC e a Teoria dos 3 estratos, resultando numa visão multidimensional da inteligência com 10 fatores.

Figura 9: Evolução Teoria CHC

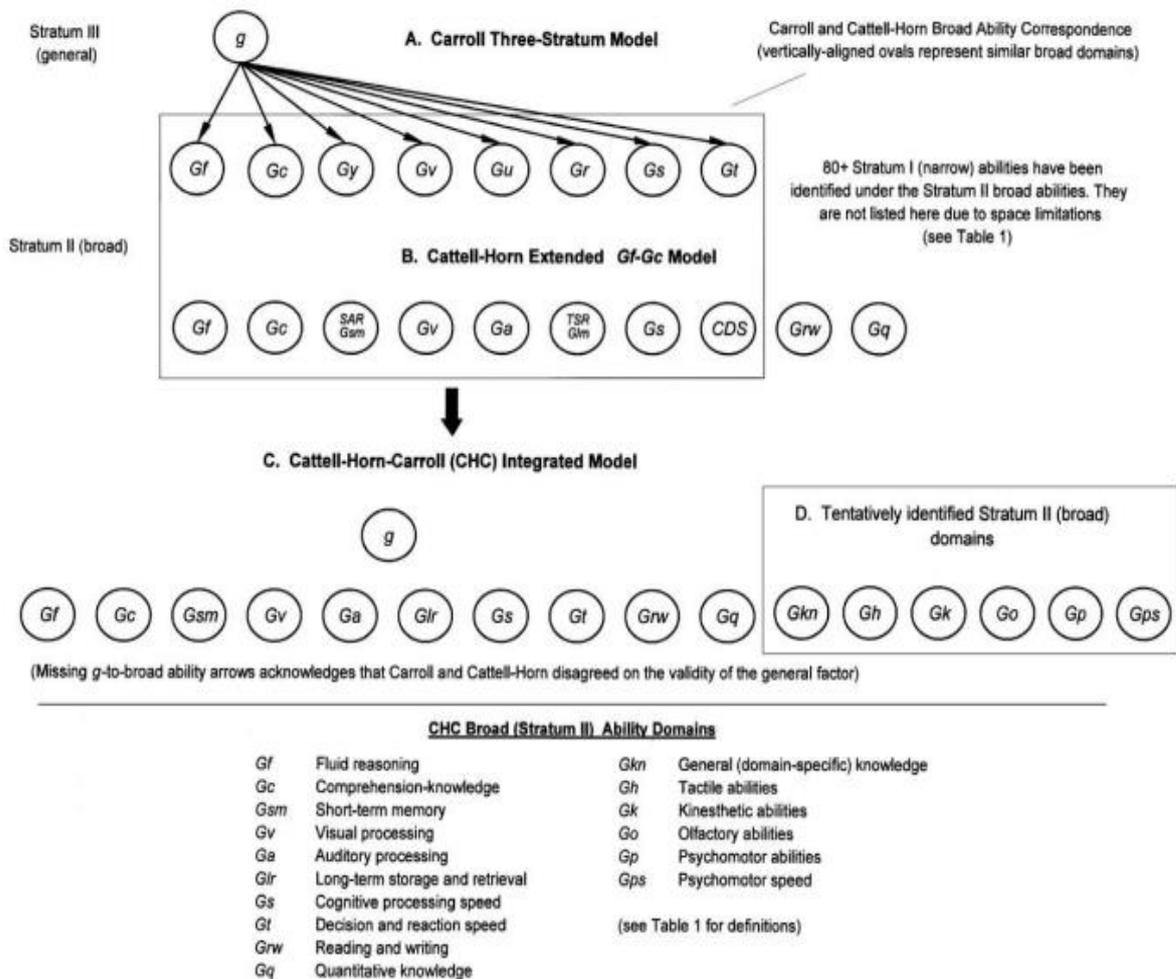


Fig. 1. Schematic representation and comparisons of Carroll's Three-Stratum, Cattell-Horn's Extended Gf-Gc, and the integrated Cattell-Horn-Carroll models of human cognitive abilities.

Fonte: MCGREW (2009) <http://iapsych.com/articles/mcgreww2009.pdf>

De acordo com Schelini (2006, p.328),

A teoria das capacidades cognitivas de Cattell-Horn-Carroll (teoria CHC) representa uma evolução do modelo dicotômico Gf-Gc, podendo ser descrita como uma integração, empiricamente avaliada, das concepções desenvolvidas por Raymond Cattell, John Horn e John Carroll.

A partir da contribuição do modelo teórico CHC a inteligência deixa de ser compreendida como uma capacidade única, inata e estática, passando a ser entendida como sendo composta por capacidades múltiplas e passíveis de estimulação (SCHELINI, 2006, p.328).

Conforme McGrew & Flanagan (1998), os modelos mais recentes sobre a estrutura da inteligência confirmam a ideia de que as capacidades humanas são organizadas de uma forma hierárquica, existindo, além do fator g, pelo menos dez áreas amplas de raciocínio.

3.6 AVALIAÇÃO DA INTELIGÊNCIA

De acordo com Primi (2003), nenhuma das baterias de testes psicológicos disponíveis no mercado brasileiro apresenta uma avaliação completa das dez capacidades amplas propostas na Teoria GHC.

Quadro 3: Definição dos dez fatores amplos da teoria CHC

Fator do Estrato II	Descrição
Inteligência Fluida (Gf)	Capacidade ligada às operações mentais de raciocínio em situações novas minimamente dependente de conhecimentos adquiridos. Capacidade de resolver problemas novos, relacionar idéias, induzir conceitos abstratos, compreender implicações, extrapolação e reorganização de informações (os testes psicométricos do fator g geralmente avaliam a inteligência fluida tais como Raven, a maioria dos testes da BPR-5, Colúmbia, R1, INV, etc...).
Inteligência Cristalizada (Gc)	Extensão e profundidade dos conhecimentos adquiridos de uma determinada cultura e a aplicação efetiva deste conhecimento. Capacidade de raciocínio adquirida pelo investimento da capacidade geral em experiências de aprendizagem. Primariamente baseada na linguagem. Está associado ao conhecimento declarativo (conhecimento de fatos, idéias, conceitos) e ao conhecimento de procedimentos (raciocinar com procedimentos aprendidos previamente para transformar o conhecimento).
Conhecimento Quantitativo (Gq)	Estoque de conhecimentos declarativos e de procedimentos quantitativos. Capacidade de usar informação quantitativa e manipular símbolos numéricos.
Leitura e Escrita (Grw)	Conhecimento adquirido em competências básicas da compreensão de textos e expressão escrita. Inclui desde habilidades elementares como decodificação em leitura e ortografia até habilidades mais complexas como a compreensão de textos e a composição de histórias.
Memória de Curto Prazo (Gsm)	Capacidade associada à manutenção de informações na consciência por um curto espaço de tempo para poder recuperá-las logo em seguida.
Processamento Visual (Gv)	Capacidade de gerar, perceber, armazenar, analisar, e transformar imagens visuais, isto é, os processos cognitivos específicos de processamento mental de imagens (geração, transformação, armazenamento e recuperação).
Processamento Auditivo (Ga)	Capacidade associada à percepção, análise e síntese de padrões sonoros. Capacidade discriminativa de padrões sonoros (incluindo a linguagem oral) particularmente quando apresentados em contextos mais complexos como, por exemplo, a percepção de nuances em estruturas musicais complexas.
Capacidade e Armazenamento e Recuperação da Memória de Longo Prazo (Glr)	Extensão e fluência que itens de informação ou conceitos são recuperados da memória de longo prazo por associação. Está ligada ao processo de armazenamento e recuperação posterior por associação. Capacidade de recuperar os itens de informação da base de conhecimentos por meio de associações. Este fator agrupa os testes psicométricos criados de avaliação da criatividade sendo muitas vezes chamado de domínio da produção de idéias.
Velocidade de Processamento (Gs)	Capacidade de manter a atenção e realizar rapidamente tarefas simples automatizadas em situações que pressionam o foco da atenção. Está geralmente ligado a situações em que há um intervalo fixo definido para que a pessoa execute o maior número possível de tarefas simples e repetitivas (sustentabilidade).
Rapidez de Decisão (Gt)	Rapidez em reagir ou tomar decisões envolvendo processamentos mais complexos. Refere-se à reação rápida a um problema envolvendo processamento e decisão (imediatez).

Para Almeida (2000), o BPR-5 – Bateria de Provas de Raciocínio, é o teste mais completo disponível no Brasil. Este teste fundamenta-se nas concepções fatoriais mais recentes da inteligência, possibilitando a avaliação da inteligência geral: fator g de Spearman, bem como de fatores mais específicos da inteligência.¹

Os 5 subtestes que compõe o BPR-5: RA – Raciocínio Abstrato, RV – Raciocínio Verbal, RN – Raciocínio Numérico, RE – Raciocínio Espacial e RM – Raciocínio Mecânico, relacionam-se com fatores específicos da inteligência:

- O subteste RA associa-se principalmente à inteligência fluida (Gf) definida como capacidade de raciocinar em situações novas, criar conceitos e compreender implicações.
- O subteste RV associa-se à inteligência fluida e à inteligência cristalizada (Gc), definida como extensão e profundidade do conhecimento verbal vocabular, e à capacidade de raciocinar utilizando conceitos previamente aprendidos.
- O subteste RN associa-se à inteligência fluida e em parte à habilidade quantitativa (Gq) definida como a compreensão de conceitos quantitativos básicos como soma, subtração, multiplicação, divisão e manipulação de símbolos numéricos.
- O subteste RE associa-se em parte à inteligência fluida, mas, principalmente, à capacidade de processamento visual (Gv) definida como a habilidade de representar e manipular imagens mentais.
- O subteste RM associa-se em parte à inteligência fluida e aos conhecimentos práticos mecânicos.

Almeida (2001) destaca que todos os subtestes se associam, em maior ou menor grau, à inteligência fluida, habilidade que é mais semelhante ao fator g de Spearman. O quadro 4 apresenta uma síntese dos atributos medidos em cada prova do BPR-5.

¹ Devido a estas características este foi o instrumento utilizado para avaliar o raciocínio na presente tese.

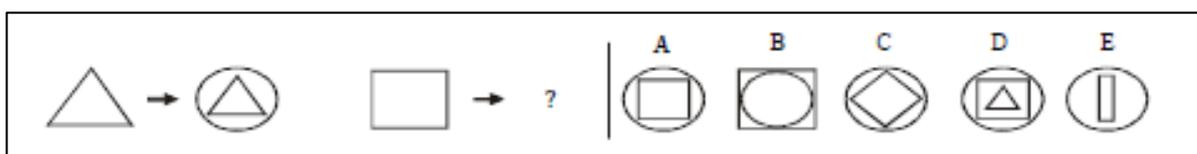
Quadro 4: Provas e capacidade específica avaliada

Prova	Capacidade Específica
RV <u>Raciocínio Verbal</u>	<ul style="list-style-type: none"> • Extensão do vocabulário • Capacidade de estabelecer relações abstratas entre conceitos verbais
RA <u>Raciocínio Abstrato</u>	<ul style="list-style-type: none"> • Capacidade de estabelecer relações abstratas em situações novas para as quais se possui pouco conhecimento previamente aprendido
RE <u>Raciocínio Espacial</u>	<ul style="list-style-type: none"> • Capacidade de visualização, isto é, de formar representações mentais visuais e manipulá-las transformando-as em novas representações
RM <u>Raciocínio Mecânico</u>	<ul style="list-style-type: none"> • Conhecimento prático de mecânica e física (adquirido principalmente em experiências cotidianas e práticas) • Capacidade de integrar as informações em textos com a figura descritiva da situação-problema
RN <u>Raciocínio Numérico</u>	<ul style="list-style-type: none"> • Capacidade de raciocinar indutiva e dedutivamente com símbolos numéricos em problemas quantitativos • Conhecimento de operações aritméticas básicas

Fonte: Almeida e Primi (2000)

Para compreensão de cada teste que compõe este instrumento, é interessante a apresentação de alguns exemplos elaborados por Almeida e Primi (2000) em seu Manual Técnico:

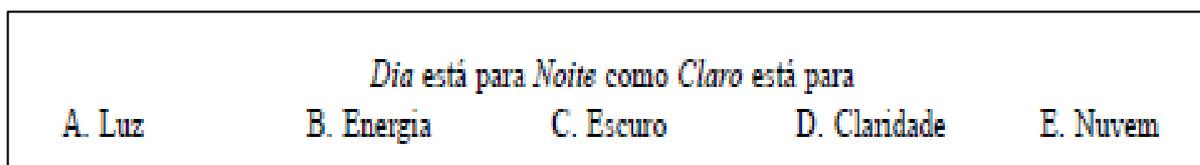
Raciocínio Abstrato (Prova RA) – composta por 25 itens de conteúdo abstrato, envolvendo analogias com figuras geométricas. Solicita-se descobrir a relação existente entre os dois primeiros termos da analogia e aplicá-la ao terceiro, para se identificar a quarta figura entre as 5 alternativas de resposta.

Figura 10: Prova de Raciocínio Abstrato (RA)

Fonte: PRIMI, R., & Almeida, L. S. (2000, p.10)

Raciocínio Verbal (Prova RV) – esta prova contém 25 itens envolvendo analogias, como na prova de raciocínio abstrato, no entanto solicita as relações semânticas entre palavras. A relação analógica existente entre um primeiro par de palavras deverá ser descoberta e aplicada de forma a identificar a quarta palavra entre as cinco alternativas de resposta, de modo a manter a relação com a palavra apresentada.

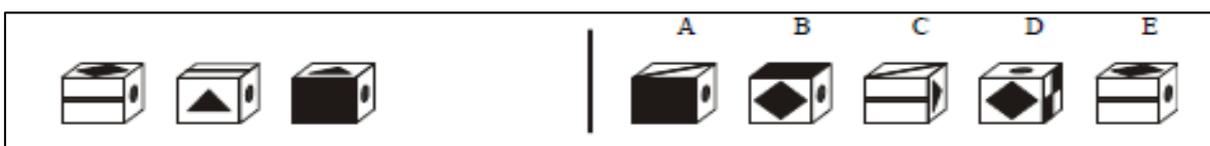
Figura 11: Prova de Raciocínio Verbal (RV)



Fonte: PRIMI, R., & Almeida, L. S. (2000, p.10).

Raciocínio Espacial (Prova RE) - composta por 20 itens nos quais existem séries de cubos tridimensionais em movimento. Os movimentos podem ser constantes, por exemplo sempre para a direita, ou alternados, por exemplo para esquerda e para cima. Analisando as posições das diferentes faces dos cubos na série apresentada, deve-se descobrir o movimento e escolher entre as alternativas de resposta a representação do cubo que se seguiria aplicando o movimento ao último cubo da sequência.

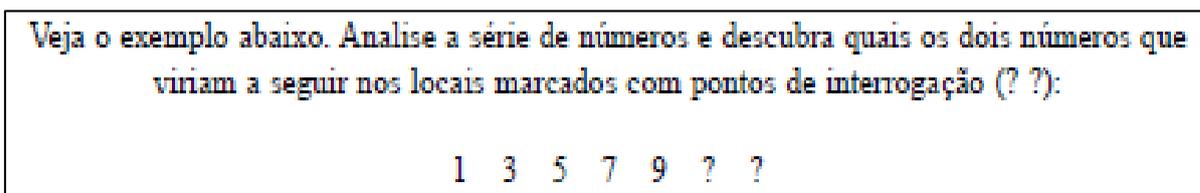
Figura 12: Prova de Raciocínio Espacial (RE)



Fonte: PRIMI, R., & Almeida, L. S. (2000, p.10).

Prova de Raciocínio Numérico (Prova RN) - Prova composta por 20 itens na forma de séries de números lineares ou alternadas, onde o sujeito deve descobrir qual a relação aritmética que rege as progressões nas séries e aplicá-la, escrevendo os dois últimos números que completariam a série.

Figura 13: Prova de Raciocínio Numérico (RN)



Fonte: PRIMI, R., & Almeida, L. S. (2000, p.10).

Prova de Raciocínio Mecânico (Prova RM) - Prova composta por 25 itens constituídos por gravuras que retratam um problema e opções de resposta. Os itens apresentam problemas

práticos que envolvem conteúdos físico-mecânicos e espaciais (roldanas, direção de movimentos, resistência de materiais, etc.). A resposta é dada escolhendo-se entre as alternativas de resposta aquela que melhor responde à questão formulada no problema.

Figura 14: Prova de Raciocínio Mecânico (RM)



Fonte: PRIMI, R., & Almeida, L. S. (2000, p.10).

4 ESTUDOS CORRELATOS

Ambrósio (2014) realizou uma pesquisa visando identificar competências cognitivas básicas relacionadas ao pensamento computacional. Este estudo contou com a participação de 12 estudantes calouros da Universidade do Minho em Portugal: cinco alunos da Licenciatura em Engenharia Informática, e sete estudantes da Licenciatura em Ciência da Computação.

Os estudantes que participaram deste experimento foram submetidos no início do semestre a uma bateria de testes psicológicos: Teste D.48, um teste não-verbal que mede o fator g; Teste de Raciocínio Espacial, um subteste da Bateria de Provas de Raciocínio (BPR-5) e 2 subtestes da versão em Português GATB - General Aptitude Test Battery ¹, o subteste raciocínio aritmético e o subteste percepção de Formas.

Ao término do segundo semestre letivo os resultados dos testes foram comparados com a avaliação final do aluno e com a avaliação de desempenho acadêmico realizada pelo professor. Os resultados da pesquisa sugerem que o raciocínio espacial e inteligência geral são dimensões cruciais para a aprendizagem de programação, havendo uma correlação com o sucesso acadêmico dos alunos nesta área. No entanto, não foram encontradas correlações com raciocínio aritmético e atenção ao detalhe testes (AMBRÓSIO, 2014).

Park (2015) elaborou um estudo para investigar a carga cognitiva durante o processo de resolução de problemas. Este experimento contou com dois grupos de alunos que foram submetidos a cursos distintos:

- Curso Básico de Perícia do Computador (grupo controle): um curso de introdução a informática para professores.
- Curso de Pensamento Computacional (grupo experimental): este curso foi desenhado para ensinar conceitos de CT para professores. Estes estudantes tiveram 10 semanas de aula, nas primeiras 5 semanas aprenderam conceitos teóricos sobre Pensamento Computacional e as últimas 5 semanas foram destinadas para prática de programação com Scratch.

Após as 10 sessões de aprendizagem (30 horas no total), foi aplicado em todos os alunos um teste de resolução de problemas elaborado a partir de questões do OCDE / PISA Resolução de Problemas, as quais foram devidamente adaptadas para estudantes de graduação.

Este mesmo teste foi utilizado durante as medições realizadas com dispositivos de EEG, as quais não revelaram correlação entre as ondas cerebrais e estados cognitivos. Foi verificado

¹ General Aptitude Test Battery (GATB) consiste em 12 subtestes separadamente cronometrados que compõem nove dezenas de aptidões (capacidade de aprendizagem em geral, verbal de aptidão, a aptidão numérica, aptidão espacial, percepção da forma, a percepção clerical, motor coordenação, destreza dos dedos e destreza manual).

um reforço das capacidades de pensamento computacional para os alunos submetidos 10 sessões de programação Scratch integrados com curso de CS introdutório. O grupo experimental apresentou melhorias em relação aos componentes: solução estratégica, que consiste em pensamento estratégico e simultâneo, bem como no domínio do pensamento estratégico (PARK, 2015).

Neves (2010) conduziu um experimento com alunos do 2º ano do ensino fundamental a fim de avaliar se a aprendizagem de conceitos de programação de computadores apresenta algum impacto no tempo para a resolução de desafios que exigem utilização do raciocínio lógico.

Participaram deste experimento 17 alunos, sendo que os alunos do grupo experimental (9 alunos) participaram de um treinamento voltado ao ensino de conceitos de programação de computadores por meio da criação de jogos digitais com Scratch e o grupo controle (8 alunos) não recebeu nenhum treinamento.

Após o treinamento, os alunos dos dois grupos foram submetidos a um desafio de lógica (“O lobo, a ovelha e a couve”), extraído do *site* Racha Cuca. Os alunos que participaram do treinamento resolveram o desafio com um tempo 52% inferior (média) em comparação com o grupo que não recebeu, sendo comprovada uma diferença estatisticamente significativa entre os grupos, demonstrando um melhor desempenho no raciocínio lógico das crianças que participaram do treinamento na ferramenta *Scratch*. (NEVES, 2010).

Voskoglou (2012) escreveu um artigo no qual relata duas experiências de sala de aulas realizadas no Instituto Tecnológico da Educação (TEI) de Patras, na Grécia. Os sujeitos do primeiro experimento foram 90 alunos da disciplina de Matemática I do curso de engenharia. Estes estudantes, não tinham nenhuma experiência em programação e foram divididas em dois grupos de 45 alunos.

No grupo (controle) as palestras foram realizadas utilizando o quadro negro, seguido por uma série de exercícios e exemplos relacionados a aplicação da matemática com problemas cotidianos. Enquanto no grupo experimental 1/3 das aulas foi ministrada no laboratório de informática, com apoio de ferramentas tecnológicas adequadas, fazendo uso de *softwares* para resolver os problemas matemáticos com a ajuda dos computadores (VOSKOGLU, 2012).

Os dois grupos foram submetidos a um exame abrangendo todos os tópicos ensinados e três problemas cotidianos que exigem técnicas de modelagem matemática para a sua solução. Os resultados da experiência de Voskoglou (2012) indicam que o uso de computadores reforça habilidades na resolução de problemas matemáticos do mundo real.

O segundo experimento foi realizado nas mesmas condições utilizando dois grupos de alunos da Escola de Gestão e Economia do TEI de Patras (100 alunos em cada grupo) e apresentou resultados semelhantes.

Wang (2014) realizou um experimento com 91 alunos do ensino secundário, incluindo 43 estudantes medianos e 48 estudantes talentosos de matemática, baseado em 12 sessões de aprendizagem (50 minutos cada). Depois de receberem aulas sobre vírus de computador e internet os alunos foram incentivados a criarem com sua equipe uma história digital com uso do Scratch, tendo como base a compreensão do conteúdo que aprenderam.

Os alunos foram submetidos a um teste de 12 itens destinado a investigar a capacidade de resolução de problemas (pré-teste e pós-teste). O teste t de amostras pareadas foi utilizado para comparar avaliações dos alunos antes e depois da atividade e os resultados estatísticos demonstraram uma melhoria significativa no desempenho de resolução de problemas.

Pardamean (2014) realizou um estudo para investigar os efeitos da aprendizagem de programação LOGO na criatividade das crianças. Participaram deste experimento 85 estudantes da 5ª série, divididos em 2 grupos: grupo experimental que recebeu aulas de programação (43 estudantes) e grupo controle (42 estudantes).

Para analisar os resultados da intervenção foi utilizado o Teste de Pensamento Criativo Figural (CTFT) para medir a criatividade e a Avaliação de Programação LOGO para estudantes jordanianos para medir a habilidade de programação LOGO. Os estudantes foram testados antes e depois do experimento, sendo observado aumento dos escores nos aspectos de flexibilidade e originalidade das habilidades de criatividade do grupo experimental. Desta forma ficou demonstrado que aprender a programar em Logo incrementa a criatividade dos alunos (PARDAMEAN, 2014)

Akcaoglu (2014) realizou um estudo pré-experimental com um grupo de estudantes (n = 21) que participaram de um curso de verão oferecido em Istambul, Turquia. Todos os alunos estudavam numa escola privada de ensino fundamental e eram provenientes de famílias de classe média ou média-alta. Nenhum dos estudantes tinha conhecimento prévio do *Microsoft Kodu* ou de *game design* utilizados no experimento.

Para medir as mudanças nas habilidades de resolução de problemas dos alunos, foi realizada uma avaliação de resolução de problemas utilizada pelo Programa Internacional de Avaliação de Alunos (PISA). O instrumento tinha um total de 19 itens: análise de sistemas e *design* (sete itens), solução de problemas (cinco itens) e de tomada de decisão (sete itens). As perguntas eram em forma de múltipla escolha e itens de resposta curta.

Os resultados indicaram que houve melhorias significativas nas habilidades de resolução de problemas dos alunos depois de assistir ao programa de verão que tinha como objetivo ensinar noções básicas de *game design* e programação, bem como ensinar a usar o *software do game design: Microsoft Kodu* (AKCAOGLU, 2014).

Taborda (2012) observou o desempenho de um estudante de 8 anos de idade que cursava a 3ª série do primário no Colégio INSA durante as aulas de Scratch. Para esta observação foi utilizada a Análise de Tarefas, desenvolvida em psicologia para examinar em profundidade a estrutura de uma situação-problema, sua demanda cognitiva e os níveis de aprendizagem dos sujeitos em desempenhos reais. Os resultados da análise de tarefas demonstraram em detalhe a forma como o uso do Scratch, promoveu o desenvolvimento do pensamento computacional, a aquisição de conhecimento conceitual e planejamento cognitivo (TABORDA, 2012).

Gülbahar (2014) realizou uma pesquisa que tinha como objetivo analisar a influência do Scratch na habilidade de resolver problemas. A amostra deste estudo foram 49 alunos do 5º ano do ensino primário, que estavam participando de um curso de informática em uma escola particular na Turquia.

A habilidade de resolver problemas foi mensurada pelo Quantitative – Problem Solving Inventory (PSI). As médias do pré-teste e pós-teste do PSI foram comparadas com o uso do teste t, não comprovando diferença na habilidade de resolução de problemas. Também foi realizado o teste t para pré e pós-teste entre as subescalas do PSI, encontrando-se apenas aumentos não significativos na média do fator de autoconfiança na capacidade de resolução de problemas.

Ambrósio (2011) realizou uma pesquisa na qual entrevistou 5 professores e 9 alunos do da disciplina de Introdução a Programação para identificar dificuldades na aprendizagem de programação.

Como resultado desta pesquisa, a autora relata que:

dois processos cognitivos foram identificados como sendo responsáveis pelas dificuldades de reveladas pelos alunos, a capacidade de organizar a solução em passos sequenciais, de elaborar um algoritmo e a capacidade de abstração, entendida como a capacidade de encontrar soluções genéricas para problemas aparentemente diferentes (AMBRÓSIO, 2011, p. 192)

Calder (2010) descreve uma prática onde empregou o Scratch com um grupo de 26 estudantes do 6º ano do Ensino Fundamental. Estes estudantes trabalharam em dupla e durante as duas semanas da pesquisa foram incentivados a criar um diário com seus progressos e suas reflexões, ao mesmo tempo que eram observados e entrevistados pelos professores.

Como resultado deste estudo Calder (2010) concluiu que o Scratch é autêntico facilitador para resolução de problemas e que o pensamento matemático é incentivado neste processo de resolução de problemas com Scratch.

Sáez-López (2016) realizou um estudo para avaliar o uso do Scratch na prática de sala de aula, analisando os resultados e atitudes de 107 alunos do ensino fundamental da 5ª a 6ª série, em cinco escolas diferentes da Espanha. A intervenção ocorreu em dois anos letivos sendo as aulas de Scratch integradas as disciplinas de ciências e de artes. O principal objetivo do estudo foi analisar os benefícios e as possibilidades de codificação com uma linguagem de programação visual através de projetos e atividades de ensino primário.

Os resultados obtidos através da comparação das médias do pré e pós-teste demonstraram melhorias estatisticamente significativas na compreensão de conceitos e práticas computacionais nesta etapa educacional. Desta forma o estudo recomenda implementar a programação em ambientes educacionais na 5ª e 6ª série do ensino fundamental (SÁEZ-LÓPEZ, 2016).

Souza (2014) apresentou uma pesquisa realizada com três turmas do 8º ano do ensino fundamental (69 alunos) de uma escola na cidade do Porto. A intervenção proposta foi baseada em resolução de problemas com o uso do Scratch, compreendendo 4 tarefas predefinidas.

Ao término do curso os resultados das tarefas foram analisados com base nos conceitos computacionais utilizados, conforme quadro a seguir:

Quadro 5: Resumo dos conceitos computacionais presentes nas diferentes tarefas

Conceitos computacionais	Tarefa 1	Tarefa 2	Tarefa 3	Tarefa 4
1. Sequências	✓	✓	✓	✓
2. Ciclos			✓	✓
3. Execução em paralelo	✓	✓		✓
4. Eventos			✓	✓
5. Condições			✓	✓
6. Operadores			✓	✓
7. Dados				✓

Fonte: Sousa (2014) p.262

Segundo Sousa (2014) a estratégia adotada permitiu o desenvolvimento do pensamento computacional e da competência de resolução de problemas, uma vez que as três turmas conseguiram atingir resultados bastante satisfatórios, com valores médios entre 67,9% e 88,0%.

No Brasil, Scaico (2013) realizou uma pesquisa utilizando como instrumento de intervenção um curso de programação com um grupo de 32 alunos com idade entre 15 a 19 anos. Estes alunos não possuíam conhecimentos prévios sobre a área de Computação, sendo apenas usuários de informática e da internet.

Este curso foi elaborado com carga horária de 20 horas (distribuídas em 10 semanas). Ao todo foram ministradas quatro turmas com professores diferentes, porém todos seguiram os mesmos planos de aula, utilizaram os mesmos exercícios e a mesma avaliação.

Nesta intervenção Scaico (2013) utilizou uma Olimpíadas de Scratch composta de 5 questões como fator motivacional do curso e instrumento de avaliação. As questões de 1 a 4 foram avaliadas com base em uma escala likert de 3 pontos, conforme tabela 1.

Tabela 1: Avaliação das questões de 1 a 4

Questão	Concluído	Parcialmente	Não concluído
1	88%	12%	0%
2	100%	0%	0%
3	75%	25%	0%
4	88%	12%	0%

Fonte: Scaico (2013, p.9)

A questão 5 solicitava a elaboração de um projeto criativo necessitando, desta forma, de uma tabela mais complexa para sua análise, conforme segue:

Figura 15: Desempenho na questão 5: Seja Criativo

Critérios de uso	X	Y	Z	K
Elementos de Repetição	0%	25%	50%	25%
Domínio Condicional	24%	38%	0%	38%
Mudança de Trajes	12%	25%	38%	25%
Destrezas com os movimentos	0%	25%	63%	12%
Controle no Diálogo	0%	50%	38%	12%
Cores	12%	25%	63%	0%
Associação da história com o plano de fundo utilizado	12%	38%	50%	0%
Originalidade ao criar e utilizar objetos	12%	50%	38%	0%
Utilização de Sprites	0%	63%	25%	12%
Criatividade	12%	50%	13%	25%
MÉDIA	8,4%	38,9%	37,8%	14,9%

Tabela 2. Desempenho dos participantes na questão Seja criativo, onde: X representa um recurso que não foi utilizado; Y representa um recurso foi utilizado de forma básica; Z representa um recurso que foi utilizado de forma intermediária e K um recurso utilizado de forma avançada.

Fonte: Scaico (2013, p.9)

De acordo com Scaico (2013) o Scratch contribuiu para o desenvolvimento de boas práticas e técnicas de programação, tais como: leitura e interpretação de código alheio, de depuração e correção de erros, além da implementação de algoritmos simples.

A pesquisa de NETO (2013) realizada na cidade de São Luiz com 36 alunos entre 16 a 17 anos de idade, do curso Técnico em Informática, propôs como intervenção a utilização da

ferramenta *Scratch* nas primeiras aulas de Lógica de Programação como apoio no ensino da disciplina.

Para a presente intervenção, a disciplina foi estruturada distribuindo as atividades em 5 aulas para ensino do Scratch, dos conceitos básico (operadores, variáveis, comandos de entrada e saída, etc.) e comandos de seleção e repetição e 10 aulas para o ensino de Visualg.

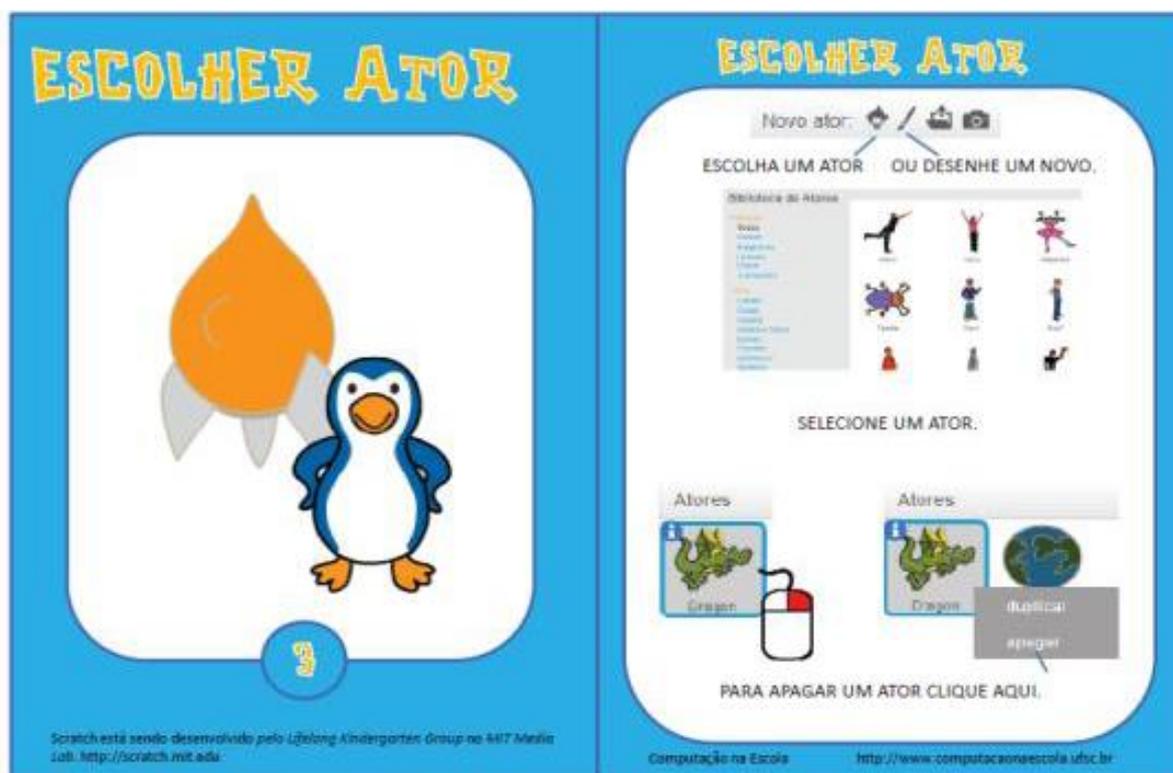
Os efeitos do Scratch foram analisados com base num questionário aplicado com os alunos, permitindo com que Neto (2013) constatasse que esta ferramenta facilitou o entendimento de conceitos de lógica de programação pelos alunos.

A pesquisa de Gresse Von Wangenheim (2013) utilizando o Scratch, foi realizada a partir da aplicação de uma unidade instrucional projetada para o primeiro ano do Ensino Fundamental, tendo a participação de 24 alunos com idades entre 6 a 7 anos.

Gresse Von Wangenheim (2013) explica que a unidade instrucional foi desenvolvida, aplicada e avaliada seguindo o modelo *Instructional System Design* (ISD).

Para apoio as atividades foram desenvolvidas cartas, conforme exemplo constante na Figura 17: Exemplo de carta para apoio a atividade Scratch.

Figura 16: Exemplo de carta para apoio a atividade Scratch



Fonte: Gresse Von Wangenheim (2013. p.120)

Todos os projetos desenvolvidos no Scratch foram analisados de acordo com os recursos utilizados pelos alunos, conforme tabela 2.

Tabela 2: Recursos utilizados

	Quantidade de alunos que utilizaram o comando/recurso
Desenhar um personagem	24
Desenhar um novo fundo	24
Incluir/excluir personagem do catálogo	23
Controlar o movimento via teclas	23
Incluir/excluir um fundo do catálogo	17
Incluir falas ou pensamentos as personagens	15
Mudar o fundo durante a história	13
Esconder alguma personagem quando a história se inicia	13
Fazer uma personagem ir para determinada coordenada	11
Mudar pano de fundo quando a personagem tocar na borda	9
Mover personagem com o comando mover	8
Mostrar alguma personagem quando um fundo é mostrado	5
Fazer uma personagem desaparecer quando toca em outro personagem	4
Sincronizar a fala de uma personagem com outro (conversar)	2
Fazer uma personagem andar sozinha	2

Fonte: Gresse Von Wangenheim (2013)

Com base nos dados coletados, Gresse Von Wangenheim (2013) afirma que o Scratch além de despertar o interesse dos alunos pela área de conhecimento a programação possibilitou a aprendizagem de conceitos básicos de computação.

Rocha (2013) relata uma intervenção realizada utilizando uma oficina de Scratch com apenas 4 horas de duração, a qual contou com 26 participantes sem experiência em programação e foi facilitada por 6 estudantes.

A oficina foi iniciada com a apresentação da ferramenta Scratch e também do seu ambiente colaborativo na internet, após a apresentação foi proposto aos estudantes o desenvolvimento do jogo *Arrow and Bow*.

De acordo com Rocha (2013) foi percebido uma dificuldade dos estudantes em visualizar a ordem correta para a introdução de comandos, bem como a tradução da lógica para código, barreira que teria sido superada com o auxílio dos monitores.

Armoni (2015) realizou uma pesquisa tendo como objetivo analisar os efeitos de se estudar Scratch antes da aprendizagem de uma linguagem de programação textual profissional como C # ou Java na escola secundária.

A pesquisa foi realizada durante o ano 2011-12 incluindo 120 alunos, estudando em cinco classes, em quatro escolas, e ensinados por quatro professores. Todos os alunos que

participaram no estudo eram estudantes de 10º grau (15 a 16 anos) que participam do curso Fundamentos da Ciência da Computação.

De acordo com Armani (2015) o conhecimento de programação e experiência obtida com Scratch facilitou a aprendizagem do material mais avançado: demandando menos tempo para aprendizagem de novos tópicos, reduzindo as dificuldades de aprendizagem e permitindo que os alunos alcançassem níveis cognitivos superiores de compreensão dos conceitos. No entanto, no final do processo de ensino, não houve diferenças significativas nos resultados em comparação com estudantes de outras turmas que não tinham estudado Scratch.

Aureliano (2012) relata um estudo realizado com a participação de 29 alunos matriculados na disciplina de Lógica de programação, disciplina introdutória de programação de um curso técnico profissionalizante em Informática, no Estado de Pernambuco.

Neste experimento, os alunos foram divididos em dois grupos através da realização de um sorteio, o grupo experimental teve dois dias de aulas com *Scratch*, no primeiro dia uma ambientação com o *software* e no segundo foi abordado o conteúdo *arrays*. Enquanto o grupo controle, teve aula sobre array utilizando a abordagem tradicional.

Para medir o aprendizado dos alunos em ambas as abordagens, foi aplicada uma avaliação pós-aula envolvendo questões conceituais e de implementação sobre o assunto *arrays* para mensurar o conhecimento dos alunos em relação ao conteúdo ministrado. Embora tenham sido encontradas diferenças de desempenho entre os grupos, a diferença entre as notas, ao ser submetida a testes estatísticos, não foi significativamente relevante (AURELIANO, 2012).

Com intuito de verificar o impacto do ensino de Scratch na resolução de problemas e aquisição de conhecimentos matemáticos Conole (2015) realizou um experimento numa Escola da Colômbia, no qual participaram 42 alunos da 6º série do ensino fundamental. Os alunos foram divididos em grupo experimental (24 alunos) e grupo controle (18 alunos).

De acordo com Conole (2015) os objetivos desta pesquisa visavam responder os seguintes questionamentos:

1. Modelagem: O uso de programação nas aulas de matemática melhora a modelagem de processos e fenômenos da realidade?
2. Raciocínio: O uso de programação nas aulas de matemática melhora o raciocínio?
3. A resolução de problemas: O uso de programação nas aulas de matemática melhora a formulação e resolução de problemas?
4. Exercício: O uso de programação nas aulas de matemática melhora a comparação e execução de procedimentos e algoritmos?

Neste experimento, foi realizado um pré e pós-teste baseado em questionários com questões relacionadas ao uso da matemática em situações da vida real e aplicações da matemática a situações habituais.

Os resultados demonstraram um ganho estatisticamente significativo no entendimento do conhecimento matemático do grupo experimental, que recebeu formação em Scratch. O desenvolvimento do pensamento computacional usando o ambiente de programação Scratch permitiu que os alunos do ensino fundamental melhorassem o seu desempenho em termos de processos matemáticos de modelagem, raciocínio, resolução de problemas e exercícios, além de facilitar a geração de um ambiente de aprendizagem motivador (CONOLE, 2015)

Oliveira (2014) apresentou o relato de um projeto de extensão que teve como objetivo a realização de um curso envolvendo os conteúdos de lógica de programação. O curso teve 10 encontros semanais, totalizando 20 horas de conteúdo, tendo iniciado com 20 alunos de escola pública do 9º ano do Ensino Fundamental e finalizado com apenas 11 alunos.

Embora os alunos tenham apresentado alguma evolução de desempenho no pensamento computacional, os resultados dos alunos foram baixos se compararmos com outras pesquisas, conforme observado na figura 18. Este resultado talvez esteja relacionado a falhas na prática pedagógica.

Figura 17: Avaliação do desempenho

Conteúdos Computacionais	Comandos no Scratch	Média de Aproveitamento (%)
Números Binários	Não se aplica	75%
Seguindo Instruções	Não se aplica	81,2%
Laços de repetição e estruturas condicionais	Controle	54,4%
Passagem de parâmetro e movimentação.	Movimento	80%
Interatividade com usuário e design (personalização) de personagens	Aparência	60%
Eventos ativados através dos dispositivos de entrada (mouse e teclado)	Sensores	30%

Fonte: Oliveira (2014)

Considerando a evasão, esta pesquisadora realizou uma série de entrevistas com o grupo, para adequar as aulas de forma a motivar a permanência dos alunos. Como solução Oliveira (2014) relata que:

Para que o curso fosse mais atrativo e despertasse o interesse do participante em continuar, foi despertada a possibilidade de criar jogos simples e animações, mesmo considerando que não possuíssem conhecimento algum sobre algoritmos.

Marques (2013) publicou os resultados de um estudo implementado com uma turma do 12º ano do ensino secundário, no qual participaram 26 alunos da disciplina de Aplicações Informáticas B e foram utilizadas vinte e cinco sessões de noventa minutos. Como tarefa desta intervenção foi solicitado, num primeiro momento, para cada aluno desenvolver um jogo individual no Scratch. Na segunda fase, os alunos trabalharam em grupos de cinco e seis elementos, desenvolvendo um jogo global para toda a turma intitulado Zschool.

A atividade foi avaliada utilizando uma grelha com os critérios de avaliação dos projetos individuais e de grupo. Utilizou-se também, inquérito por entrevista coletiva tipo Focus Group para aferir as percepções e opiniões dos alunos sobre o processo de intervenção.

Os resultados do experimento demonstraram que através do Scratch os alunos foram capazes de programar e encontrar soluções para os problemas com que se depararam, através da exploração criativa do ambiente de trabalho. A análise dos trabalhos permitiu constatar que os alunos desenvolveram o raciocínio lógico/abstrato exigido para atividade de programação, sendo capazes de implementar um sistema de pontuação ou vidas através do uso de variáveis, considerados conceitos mais abstratos da programação (MARQUES, 2013).

Schoeffel (2015) realizou um experimento utilizando um curso com carga horária de 48 horas, aplicado em dois dias por semana, totalizando oito semanas. Participaram desta pesquisa 34 alunos dos 7º, 8º e 9º anos do Ensino Fundamental do município de Ibirama – Santa Catarina.

O objetivo do curso era o ensino lúdico de tecnologia para jovens, sendo dividido em:

- Lógica computacional: ensino de raciocínio lógico e pensamento computacional de forma lúdica e prática;
- Programação de computadores: introdução ao ensino de programação, com introdução de linguagens de programação;
- Robótica: ensino e prática de montagem e programação de computadores, utilizando os *kits LEGO Mindstorms®*. (SCHOEFFEL, 2015, p.1477).

Todos os alunos da escola foram incentivados a participar das Olimpíadas Brasileira de Informática - OBI. O desempenho dos alunos nas OBI foi comparado utilizando os alunos que

não participaram do curso como grupo controle. Com base em testes estatísticos, constatou-se que os alunos que realizaram o curso proposto obtiveram notas médias significativamente maiores do que as obtidas por aqueles que não o realizaram, indicando que o curso melhorou o desempenho dos alunos na resolução de exercícios envolvendo problemas de lógica (SCHOEFFEL, 2015).

Maloney (2008) relatou as experiências de programação com Scratch realizadas em computadores de um *Club House* frequentado por jovens entre 8 e 18 anos durante um período de 18 meses. De acordo com Maloney (2008) as intervenções pedagógicas eram raras e o clube não contava com mentores experientes. Apenas a cada dois ou três meses eram organizados encontros durante o qual todos membros do clube trabalhavam com Scratch por 3 a 4 horas e compartilhavam publicamente os projetos desenvolvidos.

Foram recolhidos 536 projetos, no entanto 111 foram descartados por não conterem qualquer script. “Dos 425 projetos restantes, todos eles faziam uso de execução sequencial (isto é, o uso de mais de um bloco) e a maioria (374 projetos, 88%) mostram o uso de tópicos (ou seja, vários scripts rodando em paralelo)” (MALONEY, 2008, p. 369).

Também foram identificados o uso de outros conceitos de programação: interação com usuário (uso de teclado ou entrada mouse), loops, declarações condicionais, comunicação e sincronização (broadcast), lógica booleana (e, ou, e não), variáveis e números aleatórios. Os resultados estão expressos na tabela 3.

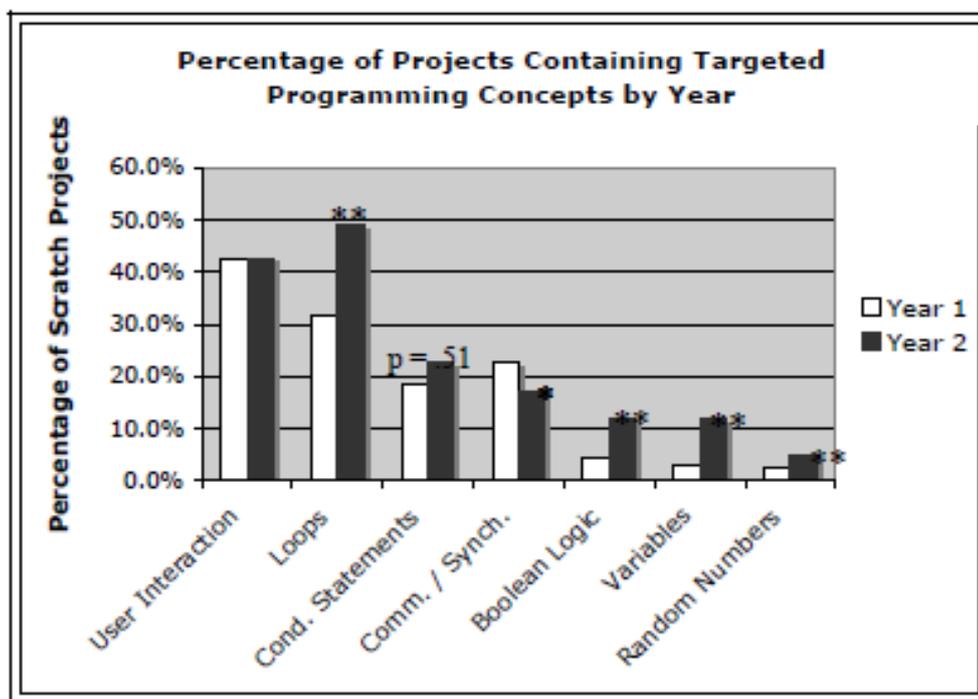
Tabela 3: Conceitos apresentados em Projetos Scratch

Conceito Computacional	Número de projetos	Percentual (N=425)
Interação com usuário	228	53,6%
Loop	220	51,8%
Instruções condicionais	111	26,1%
Comunicação e sincronização	105	24,7%
Lógica booleana	46	10,8%
Variáveis	41	9,6%
Números aleatórios	20	4,7%

Fonte: MALONEY (2008, p. 370).

Comparando a utilização dos conceitos computacionais, no período de dois anos, observa-se que no segundo ano os alunos apresentam um maior domínio de conceitos computacionais, o que demonstra um incremento do pensamento computacional, conforme figura 19.

Figura 18: Porcentagem de projetos contendo cada conceito computacional



Fonte: MALONEY (2008, p. 370).

Em sua dissertação de mestrado, Marques (2009) relata um experimento com 35 alunos do 5º ano do ensino fundamental utilizando o Scratch. A turma foi dividida em grupo experimental que realizou atividades com Scratch durante o período letivo (19 alunos) e grupo controle (16 alunos).

As notas da disciplina de matemática dos dois grupos de alunos foram comparadas, demonstrando-se que o grupo que utilizou o Scratch apresentou melhor desempenho. No entanto, não foi possível concluir que o uso do Scratch tenha relação direta com estes resultados.

Marques (2009) argumenta que o Scratch parece ter sido um meio potencializador e capaz de gerar motivação para além da presença do professor e dos pares, incentivando um maior investimento em casa em tarefas relacionadas com a escola (suportadas na utilização do Scratch).

Barcelos (2014) realizou em sua tese de doutorado um estudo envolvendo a relação entre Pensamento Computacional e pensamento matemático. Esta pesquisa contou com a participação de 24 alunos brasileiros do Instituto Federal de São Paulo e 25 alunos chilenos da Universidad de Valparaiso, os quais frequentaram uma oficina de 12 aulas de Produção de jogos com Scratch abordando os conteúdos apresentados no Quadro 6.

Quadro 6: Programação de atividades da Oficina de Produção de Jogos

Semana	Atividade / conteúdo
1	Conceito de algoritmo. Familiarização com o ambiente do Scratch. Conceitos de <i>sprite</i> e colisão entre <i>sprites</i> .
2	Variáveis e laços de repetição
3	Laços de repetição e estruturas condicionais
4	Criar o jogo Pedra-Papel-Tesoura
5-6	Criar o jogo Simulação de Guerra
7-8	Criar o jogo Breakout
9	Pacman – Criar a mecânica básica de movimentação do personagem
10-11	Pacman – Implementar as demais características do jogo (Projeto final)
12	Apresentação dos projetos finais

Fonte: Barcelos (2014)

Para analisar o pensamento matemático Barcelos (2014) utilizou um pré-teste e pós-teste elaborado a partir de questões extraídas da Prova Brasil e ENEM. Foi constatado com o teste t Student uma melhora no pensamento matemático após os alunos frequentarem a oficina de Produção de Jogos proposta pelo pesquisador.

5 PRESSUPOSTOS PEDAGÓGICOS PARA CONSTRUÇÃO DA METODOLOGIA DE ENSINO-APRENDIZAGEM

Para atingir os objetivos propostos no presente estudo foi necessário a realização de atividades didáticas com objetivo de ensinar conceitos básicos envolvendo o Pensamento Computacional.

Considerando que este tema não faz parte do currículo das escolas brasileiras, foi necessário adaptar atividades para nossa realidade e estruturar a prática pedagógica de forma que pudesse ser ofertado como um curso extraclasse.

Neste capítulo serão abordadas teorias de aprendizagem e conceitos fundamentais que foram utilizados para construção da Oficina de Pensamento Computacional com Scratch, cujo planejamento consta no Apêndice A deste trabalho.

5.1 OBJETOS DE APRENDIZAGEM

Uma das características básicas do método científico é o fato de ser sistematizado permitindo, desta forma, que a experiência possa ser replicada por outro pesquisador. Para facilitar a reprodução do experimento em mais de um grupo, ou mesmo por outro pesquisador, todo o plano de aula utilizado nas atividades didáticas foi documentado na própria plataforma Scratch e foram construídos códigos no Scratch que serviram como objetos de aprendizagem para apoiar a intervenção do professor.

Segundo a IEEE LOM (2010), um objeto de aprendizagem é qualquer entidade, computacional ou não, que pode ser usada para a aprendizagem. Hodgins (2002) explica que os objetos de aprendizagem são pequenos pedaços de instruções que podem ser montados em conjunto em estruturas instrucionais maiores e reutilizados em outras estruturas instrucionais. Este autor utiliza a metáfora do LEGO, destacando as semelhanças dos objetos de aprendizagem com as peças de montar são: duráveis, compartilháveis e intercambiáveis.

A capacidade de capturar o conhecimento de tal forma que ele pode ser analisado, reutilizados e compartilhados com os outros, desenvolvendo assim uma espiral de criação para novos conhecimentos, é talvez a mais poderosa promessa que a tecnologia da informação pode proporcionar (HODGINS, 2002, p.7).

De acordo com Wiley (2000) os objetos de aprendizagem permitem aos projetistas a construção de pequenos componentes instrucionais, os quais podem ser reutilizados em diferentes contextos de aprendizagem.

Para Tarouco (2006) os objetos de aprendizagem são elementos de um novo tipo de instrução construído sobre o paradigma da orientação a objetos da Ciência da Computação.

Singh (2000) propõe critérios para avaliar os Objetos de Aprendizagem considerando três aspectos:

- Objetivos – o que vai apresentar ao aluno;
- Conteúdo instrucional – material necessário para aprendizagem;
- Prática e realimentação – tarefas que permitam ao aluno verificar se atingiu os objetivos propostos.

Para Handa e Silva (2006) as principais características que devem ser apresentadas por objetos de aprendizagem são:

- Acessibilidade: habilidade de localizar e acessar componentes instrucionais de uma localização remota;
- Adaptabilidade: habilidade para adequar a instrução para necessidades individuais e organizacionais;
- Sustentabilidade: habilidade para incrementar eficiência e produtividade, por meio da redução de tempo e custo envolvidos na distribuição de instrução;
- Durabilidade: habilidade para suportar a evolução tecnológica e mudanças sem custo de replanejamento, reconfiguração e recodificação;
- Interoperabilidade: habilidade para pegar componentes instrucionais em uma locação e num conjunto de ferramentas ou plataforma e usá-los em outra locação com outro conjunto de ferramentas ou plataformas;
- Reusabilidade: flexibilidade para incorporar componentes em múltiplas aplicações e contextos.

5.2 TEORIA DA CARGA COGNITIVA

Segundo a teoria da carga cognitiva, sobrecarga cognitiva ocorre quando se excede os recursos cognitivos disponíveis, sendo a sobrecarga cognitiva considerada uma das principais causas de falha de aprendizagem (SWELLER, 1988)

Por esta razão, para construção de objetos de aprendizagem eficazes, é importante que sejam adequados ao processo de cognição humano, pois deste modo demandam menor esforço para atividade de aprendizagem.

Ao estudar limitações inerentes a cognição Miller (1956) apresentou um número mágico de informações que o sistema cognitivo pode manipular 7 ± 2 , apresentando uma preocupação

em relação à carga cognitiva envolvida no processo de aprendizagem. De acordo com pesquisas deste autor, o sistema cognitivo humano consegue processar um número limitado de informações, o qual varia entre 5 a 9 elementos por vez.

Para Sweller *et al.* (1998) a teoria da carga cognitiva tem como objetivo a descoberta de princípios que resultam em um ambiente de aprendizagem eficiente, ou seja, onde as informações estejam adequadas ao processo cognitivo humano. O número de elementos que podemos manipular estaria limitado a memória de trabalho, a qual é utilizada para processar todo material. No entanto, para este autor, a capacidade cognitiva pode ser ampliada se a informação é processada usando tanto o canal visual, como o verbal.

A aprendizagem multimídia de Mayer (2001) assenta em três pressupostos, os quais estão apresentados no Quadro 7: Pressupostos da Teoria de Carga Cognitiva.

Quadro 7: Pressupostos da Teoria de Carga Cognitiva

Pressuposto	Definição
Dois canais	Os seres humanos possuem o processamento de informações em separado em canal verbal e canal visual.
Limite de capacidade	Existe um limite na capacidade de processamento disponível no canal verbal e canal visual
Processo ativo	Aprendizagem requer processamento cognitivo substancial no canal verbal e canal visual

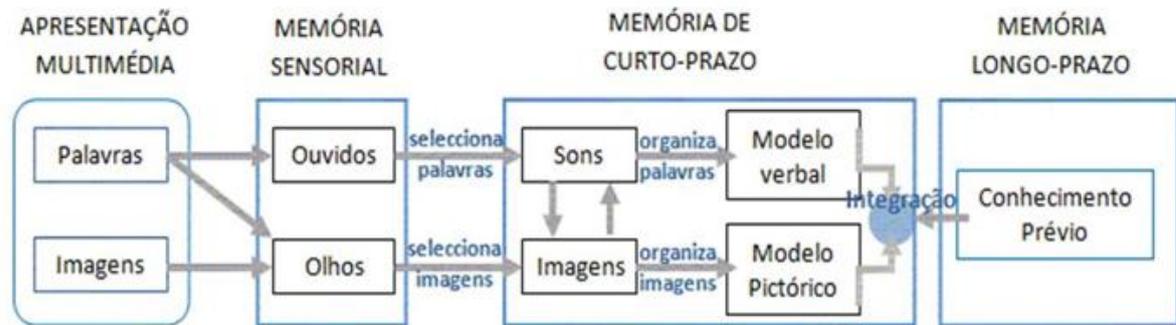
Fonte: adaptado de Mayer (2001).

Mayer (2001) aplicou os conhecimentos de carga cognitiva à produção de material multimídia. Para este autor, a carga cognitiva pode ser dividida em 3 tipos:

- Carga intrínseca (material)
- Carga natural (atividades)
- Carga externa (irrelevante)

Na figura 20, Richard Mayer (2001) revela como imagens e palavras são assimiladas através de uma apresentação multimídia entrando na memória sensorial através da audição e visão.

Figura 19: Teoria cognitiva da aprendizagem multimídia



Fonte: adaptado de Mayer (2001).

Para reduzir a carga psíquica devem ser eliminadas as cargas externas, as quais apenas desperdiçam recursos. Mayer (2001) apresenta princípios para redução da carga cognitiva:

- Representação múltipla – a aprendizagem é facilitada com uso de palavras e imagens;
- Proximidade espacial – apresentar palavras e imagens próximas, em vez de distanciadas;
- Proximidade temporal – palavras e imagens são apresentadas simultaneamente em vez de sucessivamente;
- Princípios da coerência – excluir informações não relevantes;
- Diferenças individuais – aprendizagem é diferente para cada pessoa,

As diferenças individuais podem ser explicadas pela existência de subsunçores, conhecimentos prévios que servem de base para aprendizagem significativa (AUSUBEL, 1983).

5.3 MOTIVAÇÃO E APRENDIZAGEM

Segundo Davidoff (2001, p.386) "Motivo, ou motivação refere-se a um estado interno que resulta de uma necessidade e que ativa ou desperta comportamento usualmente dirigido ao cumprimento da necessidade ativante". Esta autora classifica quatro tipos de motivos:

- Impulsos básico: motivos relacionados à sobrevivência, com raízes na fisiologia
- Motivos sociais: depende do contato com o outro, necessidade de sentir-se amado, aprovado, estimado, dentre outras;
- Motivos para estimulação sensorial: atividade autoestimulante;
- Motivos de crescimento: pessoas lutam por maestria.

Na visão de Arnold *et al.* (1994, p.428) “Motivação é um dos processos hipotéticos envolvidos na determinação do comportamento, além do efeito de um estímulo ou de uma situação percebida, e além dos processos da aprendizagem e outros fatores, tais como as capacidades. Mas enquanto as capacidades influenciam o resultado ou nível de adaptação de um determinado padrão de comportamento, motivação determina seu nível de ativação, intensidade e consistência, assim como sua direção geral”.

De acordo com Arnold *et al.* (1994, p.428), podemos dividir os termos que se referem a motivação em dois grupos, os que se referem ao polo interno (impulsos e esforço do indivíduo), ou ao polo externo (valência do objeto) do processo de motivação.

- Polo interno: necessidade, tensão, instinto, *drive* (impulso), instinto, desejos, vontade, plano, intenção;
- Polo externo: valência (desafio), valor, valor afetivo, incentivo, interesse.
- Nos estudos atuais sobre motivação estes polos têm sido nomeados de motivação intrínseca e motivação extrínseca (MATLIN, 2004).
- Motivação intrínseca: motivação para trabalhar com aquilo que é interessante, empolgante ou pessoalmente desafiador.
- Motivação extrínseca: motivação para trabalhar em um determinado assunto a fim de obter uma recompensa ou vencer um concurso.

Embora existam inúmeras teorias para explicar o que é motivação e a forma de funcionamento, para a presente pesquisa é de especial importância os estudos sobre motivação extrínseca.

5.3.1 Teoria das necessidades

Os estudos de McClelland (1987) sobre motivação apresentam 3 motivos sociais:

- Necessidade de realização: o impulso para superar-se, o impulso de atingir determinados padrões, o impulso de lutar pelo sucesso;
- Necessidade de afiliação: o desejo de relacionamentos interpessoais próximos e amigáveis.
- Necessidade de poder: a necessidade de induzir outras pessoas a se comportarem de uma maneira que não é própria do comportamento delas.

5.3.2 Teoria da fixação de objetivos

Uma das teorias da motivação que exerceu grande influência sobre a gestão de pessoas e tem sido amplamente empregada é a teoria da fixação de objetivos. Através de algumas descobertas de pesquisas de Locke (1996, p.118) é possível compreendermos a forma adequada de fixar objetivos:

1. Quanto mais difícil a meta, maior a realização;
2. Quanto mais específica ou explícito o objetivo, mais precisamente é regulada a performance;
3. Os objetivos que são específicos e difíceis promovem o alto desempenho;
4. Objetivos difíceis quando aceitos melhoram mais o desempenho do que aqueles mais fáceis;
5. O alto compromisso com os objetivos é alcançado quando
 - (a) o indivíduo está convencido de que a meta é importante, e
 - (b) o indivíduo está convencido de que a meta é possível (ou que, pelo menos, o progresso pode ser feito em direção a ela)
6. Adicionalmente tem um efeito direto sobre o desempenho, as influências de autoeficácia:
 - (a) O nível de dificuldade do objetivo escolhido ou aceito;
 - (b) O compromisso com as metas;
 - (c) A resposta ao feedback negativo ou fracasso e
 - (d) A escolha de estratégias de tarefas;
7. O estabelecimento de metas é mais eficaz quando há *feedback* mostrando os progressos em relação à meta;
8. O estabelecimento de metas (junto com a autoeficácia) media o efeito de conhecimento do desempenho passado sobre o desempenho subsequente;
9. Metas alteram o desempenho, afetando a direção da ação, o grau de esforço exercido, e a persistência de ação ao longo do tempo;
 - (a) Objetivos estimulam o planejamento em geral. Muitas vezes, a qualidade de planejamento é maior do que aquela que ocorre sem metas
 - (b) Quando as pessoas possuem planos de tarefa ou meta relevantes como resultado da experiência ou treinamento, podem ativá-los de forma quase automática, quando confrontados com uma meta de desempenho;
 - (c) Planos ou estratégias recém-adquiridas são mais susceptíveis de serem utilizados, sob o estímulo de uma meta difícil específico;

10. Quando as pessoas se esforçam para objetivos em tarefas complexas, eles são menos eficazes na tarefa de descobrir estratégias adequadas se:
 - (a) eles não têm experiência ou formação na tarefa anterior;
 - (b) há uma alta pressão para um bom desempenho;
 - (c) não é de alta pressão do tempo (para um bom desempenho imediatamente);
11. Metas (incluindo o compromisso de meta), em combinação com a auto eficácia, media ou media parcialmente os efeitos de diversos traços de personalidade e os incentivos sobre o desempenho;
12. Mecanismos de definição de objetivo e meta relacionada podem ser treinados ou adotados na ausência de treinamento com a finalidade de autorregulação;
13. As metas servem como padrões de autossatisfação, com metas mais duras que exigem maior realização

5.3.3 Teoria do reforço

Para esta teoria o reforço condiciona comportamento, visto que o comportamento é um produto do ambiente. Os behavioristas ignoram as condições internas do indivíduo, o objeto de estudo desta corrente psicológica é o comportamento observável.

Embora se acredite na possibilidade de influenciar o comportamento através de reforço e o adestramento animal demonstre que isso é possível, no ser humano é mais complexo.

Os behavioristas apresentam que um mesmo estímulo poder ter resultados diferentes em cada indivíduo, devido a história de reforços e punições vivenciadas.

5.3.4 Modelo contingencial

A teoria da expectativa ou modelo contingencial explica que a intensidade da tendência para agir de determinada maneira depende da intensidade da expectativa de que essa ação trará um dado resultado e da atração que esse resultado exerce sobre o indivíduo.

De acordo com Robbins (2002, p. 60) para esta teoria é importante a compreensão de três variáveis:

- **Atração:** a importância que o indivíduo dá ao resultado ou recompensa potencial a ser alcançado. Essa variável se baseia nas necessidades não satisfeitas do indivíduo;
- **Relação desempenho recompensa:** o grau em que o indivíduo acredita que um determinado nível de desempenho vai levar a obtenção de um resultado desejado;
- **Relação esforço-desempenho:** a probabilidade percebida pelo indivíduo de uma certa quantidade de esforço vai levar ao desempenho;

Considerando a importância de manter a motivação dos alunos, Marques (2011) cita sua experiência em uma oficina de programação em Python para alunos do ensino médio, onde utilizou codificação de jogos como fator motivacional.

Oliveira (2014), após ter evasão de vários alunos que participavam de um curso para despertar o interesse dos alunos e manter o interesse dos participantes em continuar, utilizou como estratégia motivacional o desenvolvimento de jogos simples e animações.

Barcelos (2014) ao estudar a relação do pensamento matemático com Pensamento Computacional, utilizou uma oficina de desenvolvimento de jogos no Scratch, tendo conseguido despertar o interesse dos alunos.

Em estudo de Scaico (2011) foi utilizada uma Olimpíadas de Scratch para motivar os alunos, sendo relatado resultados positivos.

Considerando os resultados dos trabalhos mencionados, o material pedagógico utilizado nesta Tese foi baseado na construção de pequenos jogos, onde todas as aulas tinham pequenos desafios e o certificado final foi condicionado a frequência em todas as aulas.

5.4 APRENDIZAGEM SIGNIFICATIVA

Considerando as teorias acerca da motivação expostas, fica constatada a importância de organizar os conteúdos de forma a adequar o esforço do aluno aos objetivos e desafios propostos em cada lição, tendo especial importância para isso os conhecimentos que ele já possui.

Para Ausubel, Novak & Hanesia (1980), "o fator isolado mais importante que influencia a aprendizagem é aquilo que o aprendiz já conhece. Descubra o que ele sabe e baseie nisso os seus ensinamentos". A ênfase desta teoria está na organização do conhecimento em estruturas e nas reestruturações que se produzem no sujeito com a aquisição de novas informações.

Aprendizagem significativa ocorre quando uma nova informação se relaciona com algum aspecto relevante da estrutura de conhecimento do indivíduo, as novas ideias podem ser aprendidas na medida em que conceitos relevantes e inclusivos estejam claros e disponíveis na estrutura cognitiva do indivíduo, também chamado de ancoragem de conceitos

O outro tipo de aprendizagem existente é a aprendizagem mecânica, processo de aprendizagem, onde as novas informações têm pouca, ou nenhuma relação com conceitos relevantes existentes na estrutura cognitiva. Neste caso, o conhecimento é armazenado de maneira arbitrária e como não há interação entre a nova informação e aquela já armazenada a retenção é prejudicada.

Segundo a teoria de Ausubel (1980), a aprendizagem pode acontecer através da recepção, processo pelo qual o conhecimento é apresentado em sua forma final para o aprendiz

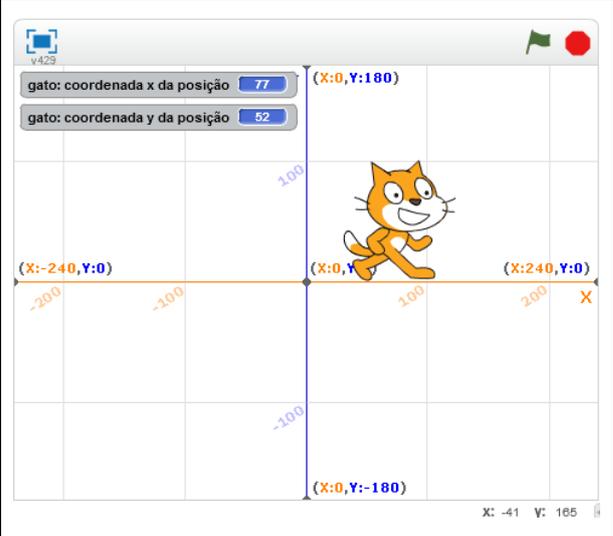
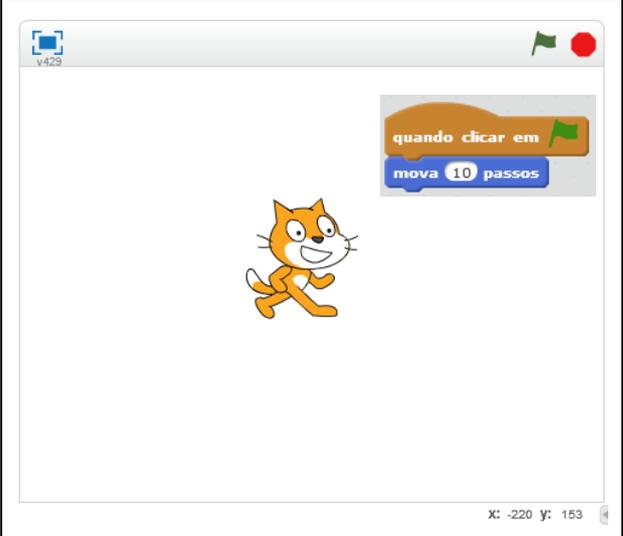
ou através da descoberta. É importante salientar que "tanto a aprendizagem receptiva como a por descoberta podem ser automáticas ou significativas, dependendo das condições sob as quais a aprendizagem ocorre" (AUSUBEL, 1980, p. 23).

A aprendizagem é significativa se o conteúdo se ligar a conceitos, subsunçores relevantes existentes na estrutura cognitiva, por este motivo o conceito de organizadores prévios tem especial importância. Os organizadores prévios são conhecimentos que tem a função de facilitar a aprendizagem sobre um domínio que pode ser completamente desconhecido, servindo de ponte entre o que o aprendiz já sabe e o que ele deve saber, estes organizadores funcionam como uma ponte cognitiva

Para Ausubel (1980), os organizadores prévios verdadeiros são aqueles destinados a facilitar a aprendizagem significativa de tópicos específicos, ou ideias estreitamente relacionadas. Enquanto, os materiais introdutórios utilizados para facilitar a aprendizagem de vários tópicos seriam denominados de pseudo-organizadores prévios.

Como exemplo prático podemos citar o ensino de código para movimentação de *sprites* no Scratch. Para aprender a movimentar os personagens é necessário conhecer o plano cartesiano, bem como a resolução de tela utilizada no Scratch (subsunçores), este exemplo está demonstrado na figura 21.

Figura 20: Subsunçor x Objetivo de aprendizagem

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

Fonte: desenvolvido pelo autor

A aprendizagem significativa é mais eficaz que a aprendizagem mecânica devido a três vantagens essenciais:

- 1) Produz retenção mais duradoura da informação;
- 2) Facilita novas aprendizagens relacionadas;
- 3) Produz mudanças profundas na estrutura cognitiva

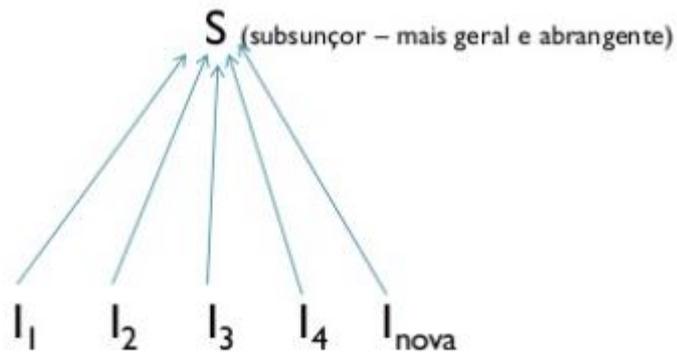
Para ocorrer a aprendizagem significativa é necessário haver uma predisposição do aluno para aprender.

O aluno deve manifestar uma predisposição para relacionar o substancial e não arbitrariamente o novo material com sua estrutura cognitiva, como o material deve ser potencialmente significativo para ele, ou seja, relacionar com sua estrutura de conhecimento sobre uma base não arbitrária. (Ausubel 1983, p. 48).

Dependendo como o material instrucional for organizado pode facilitar diferentes relações entre conceitos, processos de aprendizagem que Ausubel (1983) definiu como: aprendizagem subordinada, aprendizagem supraordenada e aprendizagem combinatória.

Aprendizagem subordinada: a nova ideia produzida está associada a preexistente: É produzida uma diferenciação progressiva de conceitos já existentes, utilizando conceitos de nível inferior

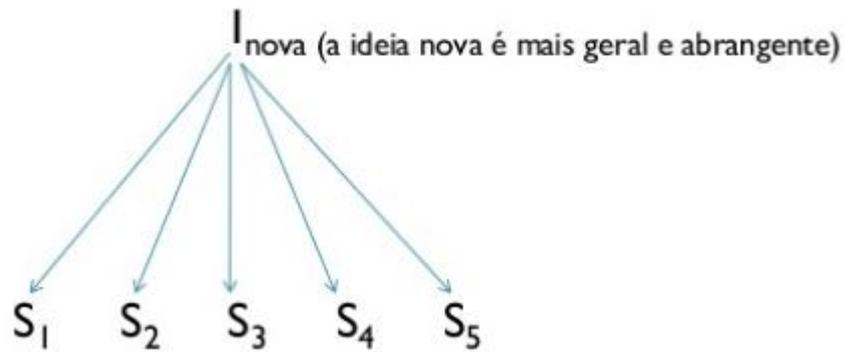
Figura 21: Aprendizagem subordinada



Fonte: desenvolvido pelo autor

Aprendizagem supraordenada: as ideias preexistentes são mais específicas que a ideia que se pretende adquirir. É produzida através da integração de uma série de conceitos, de forma a produzir um conceito mais geral.

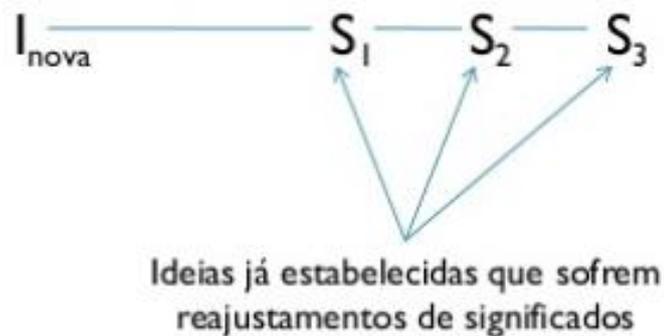
Figura 22: Aprendizagem supraordenada



Fonte: desenvolvido pelo autor

Aprendizagem combinatória: o novo conceito se encontra no mesmo nível hierárquico, possuindo atributos comuns com conceito preexistente.

Figura 23: Aprendizagem combinatória



Fonte: desenvolvido pelo autor

5.5 APRENDIZAGEM EXPERIENCIAL

Esta teoria é chamada de Aprendizagem Experiencial para destacar o papel central que experiência desempenha no processo de aprendizagem, o processo pelo qual o conhecimento é criado através da transformação da experiência (KOLB, 1984). Outra razão da teoria ser chamada de “experimental” refere-se a suas origens intelectuais nos trabalhos experimentais de Lewin, Piaget, Dewey, Freire e James, formando uma perspectiva única sobre a aprendizagem e desenvolvimento

Para Kolb (1984) o conhecimento é criado pela transformação da experiência, sendo resultado da combinação da compreensão e transformação da experiência.

simples percepção da experiência por si só não é suficiente para a aprendizagem; algo deve ser feito com ele. Da mesma forma, a transformação por si só não pode representar a aprendizagem, pois deve haver algo a ser transformado, algum estado ou experiência que está sendo posta em prática (KOLB, 1984, p.42).

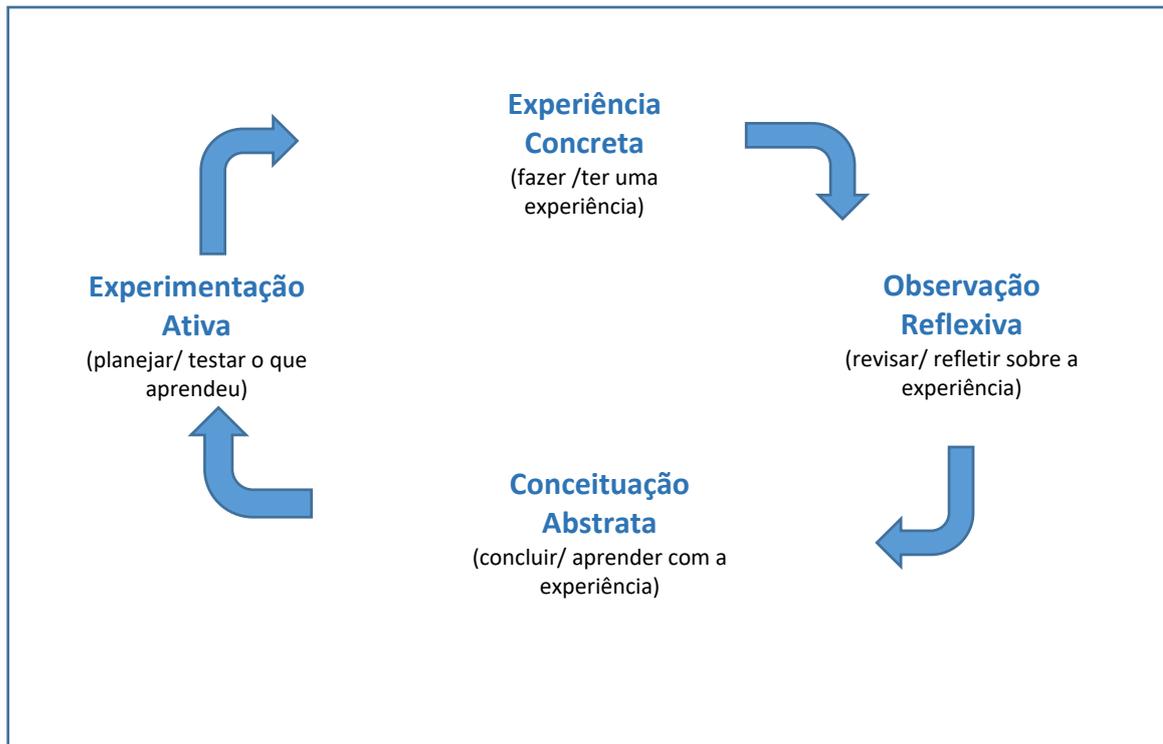
Seguindo Dewey, a Teoria da Aprendizagem Experiencial de Kolb (1984) descreve como a experiência é transformada em aprendizagem através de um ciclo de aprendizagem envolvendo experimentar, refletir, pensar e agir.

O modelo proposto pela Teoria de Aprendizagem Experiencial retrata dois modos opostos de incorporar a experiência: Experiência Concreta (CE) e Conceituação Abstrata (AC) experiência concreta, assim como, dois modos opostos de transformá-la: Observação Reflexiva (RO) e Experimentação Ativa (AE).

Aprender com a experiência é um processo de construção conhecimento que envolve uma tensão criativa entre esses quatro modos de aprendizagem. Este processo é retratado como um ciclo de aprendizagem idealizado ou em espiral em que o aluno percorre as quatro fases:

- Experiência Concreta: relacionada às experiências pessoais e os sentimentos envolvidos na situação de aprendizagem;
- Observação Reflexiva: implica na solução de problemas revisando e refletindo sobre a experiência;
- Conceituação Abstrata: o entendimento é baseado na compreensão intelectual de uma situação, ou seja, nas conclusões construídas com base na experiência, sendo o nível de abstração bastante elevado;
- Experimentação Ativa: envolve a aprendizagem de forma ativa, onde os alunos planejam novas experiências, modificam variáveis e influenciam situações, para experimentar o que aprendeu e formular hipóteses.

Para este autor a aprendizagem será eficaz quando o aluno progride através de um ambiente composto por etapas de experiência concreta, observação reflexiva, conceituação e atividade prática, conforme demonstrado na figura 24.

Figura 24: Ciclo de Aprendizagem Experiencial

Fonte: Adaptado de Kolb (1984)

Kolb e Fry (1975) argumentam que o ciclo de aprendizagem pode começar em qualquer um dos quatro pontos, e que estas etapas devem ser abordadas como uma espiral contínua. No entanto, sugerem que o material didático deve ser planejado respeitando todo o processo de aprendizagem atendendo a sequência do ciclo de aprendizagem experiencial, de forma a oferecer para cada aprendiz a oportunidade de desenvolver habilidades de cada fase do ciclo de aprendizagem experiencial.

Conforme será apresentado no capítulo seguinte, o curso de Desenvolvimento de Games utilizado nesta pesquisa, ao buscar subsunçores para apoiar os novos conhecimentos, iniciava a aprendizagem a partir da Experiência Concreta.

6 METODOLOGIA

O presente estudo pode ser classificado como uma pesquisa quantitativa, porque recorre à linguagem matemática para descrever as causas de um fenômeno e as relações entre variáveis. Fonseca (2002, p. 20) explica que este tipo de pesquisa está centrado na objetividade e “influenciada pelo positivismo, considera que a realidade só pode ser compreendida com base na análise de dados brutos, recolhidos com o auxílio de instrumentos padronizados e neutros”.

Em relação à natureza do estudo, esta pesquisa pode ser considerada uma pesquisa aplicada à medida que objetiva gerar conhecimentos para aplicação prática, dirigidos à solução de um problema específico.

Considerando os objetivos propostos, este estudo trata de uma pesquisa explicativa, a qual segundo Gil (2010) se determina um objeto de estudo, seleciona-se as variáveis que seriam capazes de influenciá-lo, define-se as formas de controle e de observação dos efeitos que a variável produz no objeto. Este é o tipo de pesquisa que mais aprofunda o conhecimento da realidade, porque explica a razão, o porquê das coisas.

Conforme será explicado a seguir, a natureza deste estudo não permitia a seleção aleatória da amostra e não foi utilizado grupo controle, por esta razão, esta pesquisa caracteriza-se como um estudo quase-experimental. Segundo Shadish et al. (2001), os desenhos quase-experimentais são experimentos que não realizam a distribuição aleatória das unidades, porém possuem os mesmos propósitos e atributos estruturais dos experimentos.

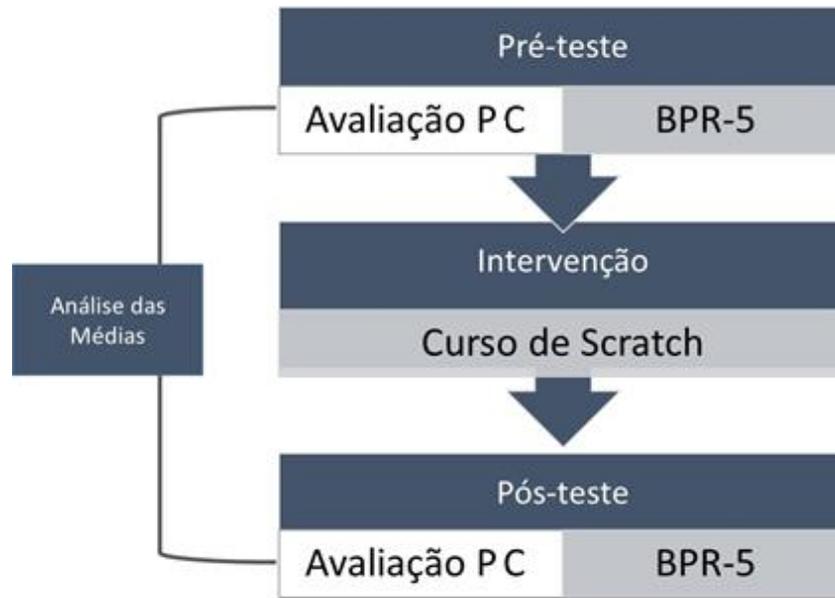
Shadish et al. (2001) diferencia os experimentos e quase experimentos:

- No experimento as explicações alternativas tornam-se pouco plausíveis porque o delineamento assegura que essas explicações alternativas estejam distribuídas aleatoriamente nas condições.
- No quase experimento, como não é feita distribuição aleatória de unidades nas condições, são usados outros princípios para mostrar que explicações alternativas não são plausíveis.

Para verificar o efeito da intervenção pedagógica proposta, foram realizadas duas avaliações dos 50 alunos que participaram do experimento. A primeira antes do Curso de Desenvolvimento de Games e a segunda após a intervenção pedagógica, posteriormente a média dos grupos foi comparada com base em testes estatísticos.

Com intuito de assegurarmos que os resultados obtidos eram relacionados a pesquisa e não a variáveis externas, o Curso de Desenvolvimento de Games foi ofertado em quatro edições, conforme será abordado no capítulo seguinte. A figura 25 apresenta de forma gráfica o delineamento desta pesquisa.

Figura 25: Etapas da pesquisa



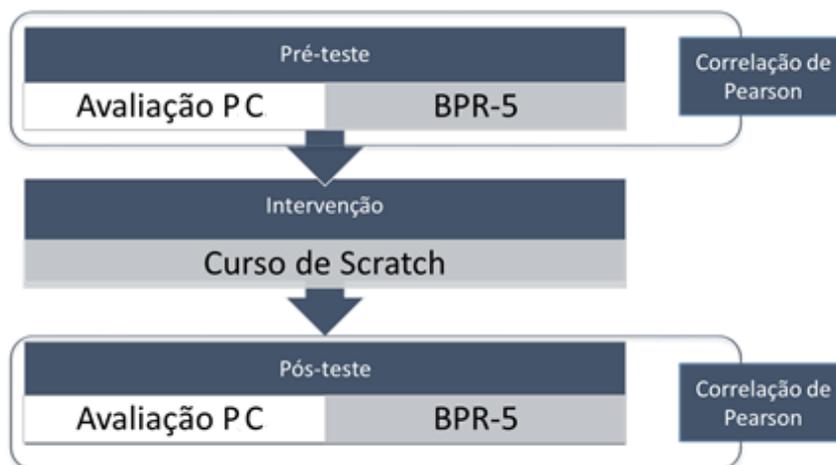
Fonte: desenvolvido pelo autor

Rúdio (2007) explica que uma hipótese: “é uma suposição que se faz na tentativa de explicar o que se desconhece. Esta suposição tem por característica o fato de ser provisória, devendo, portanto, ser testada para a verificação de sua validade”. Os dados coletados neste estudo tinham como objetivo testar duas hipóteses formuladas no início:

- O ensino do Pensamento Computacional melhora a capacidade de raciocínio dos alunos;
- Existe uma correlação entre Pensamento Computacional e capacidade de raciocínio.

Para isso foi previsto no desenho de pesquisa analisar a relação do score obtido no Teste de Pensamento Computacional e as provas BPR-5, conforme ilustrado na Figura 26.

Figura 26: Análise da relação entre as variáveis



Fonte: desenvolvido pelo autor

As duas primeiras turmas deste experimento ocorreram entre os meses de outubro e novembro de 2016 e as demais entre abril e maio de 2017. O experimento teve a duração de 7 semanas, com encontros semanais com os alunos. A primeira e última semana foram dedicadas a avaliação com BPR-5. Sendo cinco semanas utilizadas para realização do curso de Desenvolvimento de Games, conforme detalhado no próximo tópico.

6.1 METODOLOGIA DE ENSINO PROPOSTA

Todas as lições do curso proposto por este projeto foram planejadas para iniciar buscando resgatar experiências passadas dos alunos, para encontrar subsunçores e gerar discussões de experiências fora do contexto computacional. Posteriormente, os alunos interagem com fragmentos de código. No momento posterior eram indagados sobre o funcionamento do código e testavam modificações na programação. Por último, cada lição finalizava com desafios sobre o tópico abordado no Scratch.

O Quadro 1 apresenta uma síntese de cada aula do Curso de Desenvolvimento de Games exemplificando a articulação dos subsunçores, experiências e objetivos de aprendizagem.

Quadro 8: Síntese do plano de Aula do Curso de Desenvolvimento de Games

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

Fonte: desenvolvido pelo autor

Seguindo o exemplo do trabalho de Kologeski et al. (2016) para motivar os alunos na realização de atividades, bem como a permanência no curso, foi explicado que eles receberiam

um certificado de conclusão, o qual foi entregue no encerramento do curso de acordo com modelo (Apêndice E).

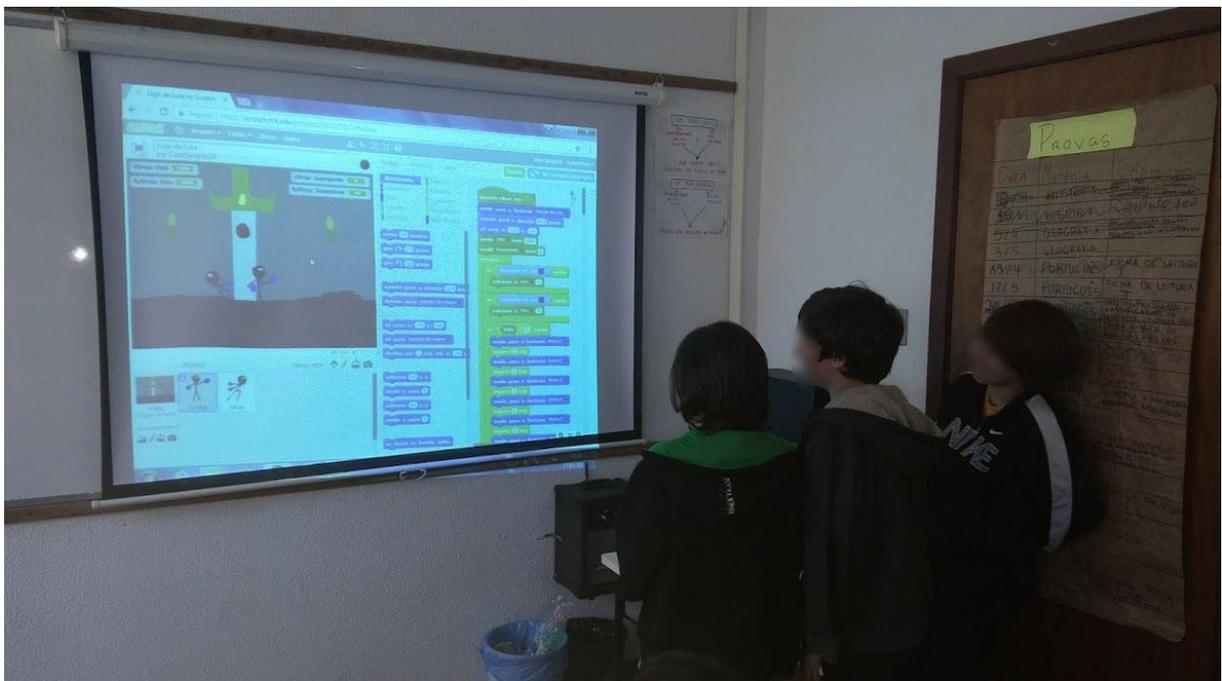
Todos os testes de raciocínio que compõe a BPR 5- Bateria de Provas de Raciocínio foram aplicados pelo autor desta pesquisa, o qual possui graduação em psicológica e trabalha há muitos anos com avaliação psicológica.

A BPR-5 é composta de 5 testes e exige quase 2 horas para ser finalizada. Devido a exigência cognitiva da avaliação, é previsto um intervalo após os primeiros 3 testes e com base na experiência profissional do autor, a expectativa era que os alunos saíam cansados da prova e poderia gerar uma expectativa negativa do curso.

Com intuito de minimizar esta impressão negativa, foi previsto um *coffe break* para os alunos. Este mesmo procedimento se repetiu no dia de entrega de certificados, após a última avaliação.

Com intuito de valorizar o conhecimento construído pelos alunos, foi realizada uma apresentação do trabalho dos grupos após a entrega dos certificados. Embora não faça parte do escopo deste estudo, o pesquisador foi surpreendido com a qualidade de alguns jogos apresentados e o interesse dos alunos em explicar seu jogo. A figura 27 ilustra uma apresentação de trabalho.

Figura 27: Apresentação do trabalho realizado com os grupos



Fonte: foto do autor.

6.2 AMOSTRA

Conforme Rúdio (2007, p.31) “uma pesquisa não é feita com todos os elementos que compõe uma população. Costuma-se neste caso, selecionar uma parte representativa dela, denominada amostra”.

Neste estudo, necessitávamos realizar uma intervenção pedagógica formatada como um curso de Desenvolvimento de Games. Como o curso foi modelado para utilizar o *software* Scratch, era necessário que as escolas tivessem um laboratório de informática. Para padronizar às aulas foi exigido a disponibilidade de um computador por aluno. Para obter uma amostra adequada, foi exigido que o laboratório de informática possuísse no mínimo 10 computadores.

Considerando que a amostra não seria aleatória o pesquisador inicialmente entrou em contato por telefone com várias escolas, utilizando o critério de conveniência onde considerou a proximidade da UFRGS e de sua residência. Deste contato inicial foram agendadas reuniões com diretoras de três escolas, sendo que apenas 2 tinham laboratórios em condições de uso. As duas escolas que participaram desta pesquisa, são escolas particulares, situadas em Porto Alegre, e serão designadas neste estudo como Escola A e Escola B.

A primeira reunião ocorreu com a Escola A, na qual foi acordado que o curso deveria ser oferecido como atividade extraclasse e no turno inverso as aulas, sendo imobilizado o laboratório de informática nas quintas-feiras, dos meses de outubro e novembro de 2016, para esta atividade.

Na reunião com a Escola B, foi ajustado o cronograma do curso para ocorrer simultaneamente com o do Escola A, sendo fixado as quartas-feiras para sua realização. Com objetivo de ampliar a amostra, foi disponibilizada mais uma turma para cada escola, as quais ocorreram entre abril e maio de 2017.

As inscrições no curso foram ofertadas para alunos matriculados a partir do sexto ano do Ensino Fundamental, uma vez que a BPR-5 está padronizada para sujeitos com esta escolaridade. Para divulgar o curso, o pesquisador entrou em todas salas de aula, explicando a pesquisa e os objetivos do curso para os alunos, sendo necessário a criação de um folder para esta finalidade, conforme apresentado na figura 28.

Figura 28: Folder para divulgação do curso.

Curso de Desenvolvimento de
GAMES

APRENDA A PROGRAMAR DESENVOLVENDO GAMES

Este curso é parte de um projeto de pesquisa que tem como objetivo: **Investigar a relação entre aprendizagem do Pensamento Computacional e desenvolvimento do raciocínio.** Com a metodologia proposta espera-se obter um **Incremento em habilidades relacionadas ao raciocínio numérico, raciocínio espacial e raciocínio abstrato.** Os voluntários da pesquisa serão avaliados, com testes psicométricos, antes e depois da realização do curso.

Público
Estudantes do 6 ao 9 ano do Ensino Fundamental

Carga Horária
10 horas aula

Custo
Fornecido gratuitamente

Informações:
51 91494040 ou rafael@ieduca.com.br

UFRGS ieduc

Fonte: elaborado pelo autor

As inscrições no curso foram realizadas através de uma lista que ficou na Secretaria de cada escola. Para efetivar a inscrição foi exigido que os alunos interessados tivessem a autorização dos pais para participar da pesquisa, desta forma foi solicitada a assinatura de um Termo de Consentimento Livre e Esclarecido, conforme Apêndice F.

O laboratório de informática da Escola A contava com 20 computadores e o da Escola B a 16 computadores, como maior parte das desistências ocorreram após o pré-teste foi possível trabalhar com um aluno por computador. Para facilitar o tratamento dos dados, as turmas foram nomeadas com uma letra para designar cada escola e dois dígitos referenciando o ano de sua ocorrência.

Tabela 4: Inscrições dos alunos e amostra

Turma	Escola	Ano	Inscritos	Amostra
TurmaA16	Escola A	2016	26	16
TurmaA17	Escola A	2017	19	16
TurmaB16	Escola B	2016	17	11
TurmaB17	Escola B	2017	10	7
Total			72	50

Fonte: elaborado pelo autor

Considerando que para atender os objetivos da presente pesquisa existia a necessidade de comparar os resultados do pré-teste com o teste realizado após a finalização do curso de Desenvolvimento de Games, foi utilizado como critério de exclusão da amostra a ausência numa das etapas de avaliação. Desta forma, a amostra ficou limitada ao número de 50 alunos.

Para obter informações complementares sobre a amostra, foi solicitado que todos alunos preenchessem um questionário (Anexo G), onde forneceram informações que serão apresentadas a seguir.

A distribuição da amostra por gênero evidenciou que maior parte dos interessados no curso foram do sexo masculino. Conforme demonstrado na tabela 5, os sujeitos do sexo masculino totalizavam 76% da amostra:

Tabela 5: Distribuição dos alunos por gênero

Turma	Masculino	Feminino
TurmaA16	14	2
TurmaA17	9	7
TurmaB16	8	3
TurmaB17	7	0
Total	38	12

Fonte: elaborado pelo autor

Moreira (2014) afirma que no Brasil 79,9% dos alunos em cursos relacionados a tecnologia são homens e que em várias partes do mundo como: Estados Unidos, Europa e Ásia, as mulheres são uma minoria nesta área. Portanto, o maior interesse do sexo masculino por cursos na área de Ciência da Computação é amplamente conhecido e à medida que o curso proposto não era uma disciplina obrigatória, era esperado maior interesse dos meninos.

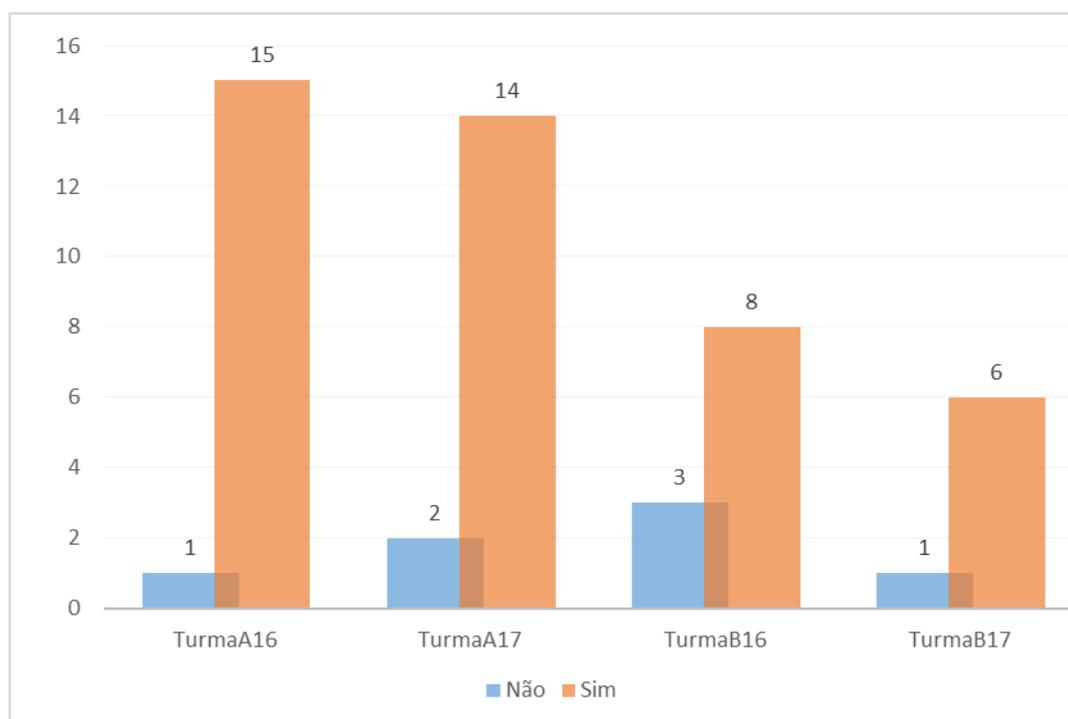
A idade dos participantes da pesquisa estava situada entre 11 e 16 anos, sendo a idade média de 12 anos. Este fato está associado a delimitação da amostra, a qual tinha como critério de inclusão estar cursando entre o sexto e nono ano do Ensino Fundamental. Observou-se que em todas edições realizadas deste curso, maior parte dos interessados (62%), eram alunos do 7 ano. No entanto, os dados desta pesquisa não permitem compreender porque que havia maior interesse neste ano letivo.

Tabela 6: Distribuição por ano escolar

Turma	6 ano	7 ano	8 ano	9 ano
TurmaA16	-	12	-	4
TurmaA17	-	10	-	6
TurmaB16	-	7	3	1
TurmaB17	5	2	-	-
Total	5	31	3	11

Fonte: elaborado pelo autor

Uma vez que a intervenção pedagógica proposta seria um Curso de Desenvolvimento de Games, os alunos foram questionados se costumavam jogar videogame (ou no computador). Dentre os 50 participantes, apenas 7 alunos não tinham o hábito de jogar videogame, o que pode ser observado no Gráfico 1. Este dado demonstra que jogar videogames estava associado ao interesse no Curso.

Gráfico 1: Hábito de jogar videogame

Fonte: elaborado pelo autor

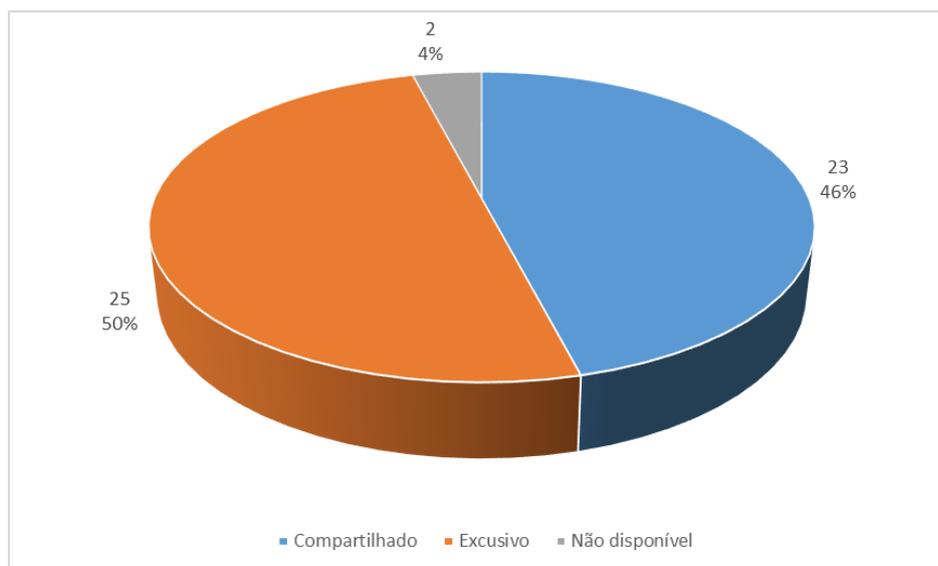
Ao serem questionados sobre o tempo semanal que destinavam para jogar videogame, verificou-se que os alunos dedicavam em média 19 horas semanais para esta atividade. Na tabela 7 é apresentada esta distribuição de frequência:

Tabela 7: Tempo dedicado a jogar videogame

Quantidade de horas/semana	Número de Alunos
Até 5	14
6 - 10	5
11 - 15	7
16 - 20	5
21 - 25	10
Mais de 25	7
Não sabe	2

Fonte: elaborado pelo autor

Constatou-se que 93 % dos alunos tinham computadores em sua residência, sendo que 50% tem um computador para seu uso exclusivo, o que pode ser observado no Gráfico 2:

Gráfico 2: Disponibilidade de computador na residência

Fonte: elaborado pelo autor

Para verificar se existia uma relação entre a escolha profissional do aluno e a adesão ao curso, os alunos foram questionados se já tinham uma escolha profissional e, em caso positivo, qual era a profissão. Apenas 18 alunos manifestaram saber a profissão que pretendiam seguir, sendo que 4 participantes informaram mais de uma profissão nesta escolha.

Quadro 9: Escolha profissional

1. Arquitetura
2. Cirurgia plástica ou Engenharia de Petróleo
3. Criador de games
4. Edição de vídeo
5. Empresário
6. Engenharia aeroespacial ou Astronomia
7. Engenheiro
8. Gamer
9. Jogador de futebol
10. Medicina
11. Medicina
12. Piloto de avião
13. Programador
14. Programador
15. Veterinária
16. Veterinária e Bombeira
17. Veterinária ou Programador de Games
18. Youtuber

Fonte: elaborado pelo autor

Analisando o quadro 8, observa-se que vários alunos tinham interesse por cursos na área de saúde, demonstrando que o curso de Desenvolvimento de Games nem sempre estava associado a um projeto profissional futuro. Cabe destacar que nenhum participante do curso manifestou interesse profissional por cursos na área de humanas, o que indica que a abordagem proposta não despertou interesse por alunos identificados com estas disciplinas.

Com intuito de identificar o conhecimento prévio dos alunos, eles foram indagados se tinham alguma experiência prévia com programação e se já tinham utilizado o Scratch. Apenas 8 alunos relataram conhecer alguma linguagem de programação, 7 estavam se referindo ao Scratch e 1 aluno respondeu que sabia programar servidores de game. Quanto ao uso do Scratch, 11 alunos relataram já ter alguma experiência prévia com o *software*.

Estas informações sobre a amostra permitem situar as particularidades destes alunos, bem como contextualizar os resultados apresentados no próximo capítulo.

7 DESCRIÇÃO E ANÁLISE DOS DADOS

Neste capítulo serão apresentados os dados coletados na pesquisa e o tratamento estatístico ao qual foram submetidos. Visando uma melhor exposição do assunto, o texto será dividido em tópicos de acordo com os objetivos específicos.

7.1 AVALIAÇÃO DO PENSAMENTO COMPUTACIONAL E VALIDAÇÃO DA METODOLOGIA DE ENSINO

Conforme mencionado nos procedimentos metodológicos do presente trabalho, os alunos deste experimento foram submetidos ao Teste de Pensamento Computacional antes de iniciarem o curso de Desenvolvimento de Games (pré-teste) e depois de finalizado o curso (pós-teste). Esta avaliação atende o objetivo específico: **2. Avaliar o Pensamento Computacional dos alunos.**

Ao comparar os resultados deste teste, obtivemos dados para validar se a metodologia de ensino, prevista no objetivo: **1. Desenvolver metodologia para ensino de Pensamento Computacional**, foi eficaz.

A Tabela 8 apresenta uma análise preliminar das médias obtidas em cada momento de testagem com o Teste de Pensamento Computacional, o qual conta com o total de 28 questões:

Tabela 8: Média dos alunos

Escola / Turma	Total de alunos	Pré-teste	Pós-teste	Diferença
Escola A – 2016	16	17,43	19,69	2,26
Escola B – 2016	11	14,82	16,09	2,03
Escola A – 2017	16	14,06	16,69	2,63
Escola B – 2017	7	15,14	17,29	2,15
Total	50	15,46	17,60	2,14

Fonte: o autor

Considerando a diferença entre as médias, pode-se afirmar que em todas as turmas do Curso de Desenvolvimento de Games, houve um incremento do Pensamento Computacional dos alunos. No entanto, são necessários testes complementares para determinar a relevância estatística deste achado.

Com apoio do *software Past*¹, buscou-se verificar se os dados coletados no Teste de Pensamento Computacional atendiam uma distribuição normal, para determinar os testes estatísticos possíveis de serem utilizados.

¹ Disponível em <https://folk.uio.no/ohammer/past/>

Utilizou-se o teste Shapiro-Wilk (1965) para testar a normalidade da amostra. Este teste foi escolhido por apresentar melhor desempenho para discriminar a distribuição de amostra normal (RAZALI, 2011). Cabe destacar, que em sua versão original o teste Shapiro-Wilk estava limitado a amostras de menos de 50 elementos, mas posteriormente teve seu algoritmo alterado por Rahman e Govidarajulu (1997) de forma a atender amostras maiores. A implementação do Shapiro-Wilk no Past incorpora as correções no algoritmo, permitindo a utilização para amostra com até 5000 observações (Hammer, 2017).

Para realizar o teste de Shapiro-Wilk, definiu-se duas hipóteses:

H0: A amostra provém de uma população normal;

H1: A amostra não provém de uma população normal.

Tabela 9: Resultado do Teste Shapiro-Wilk

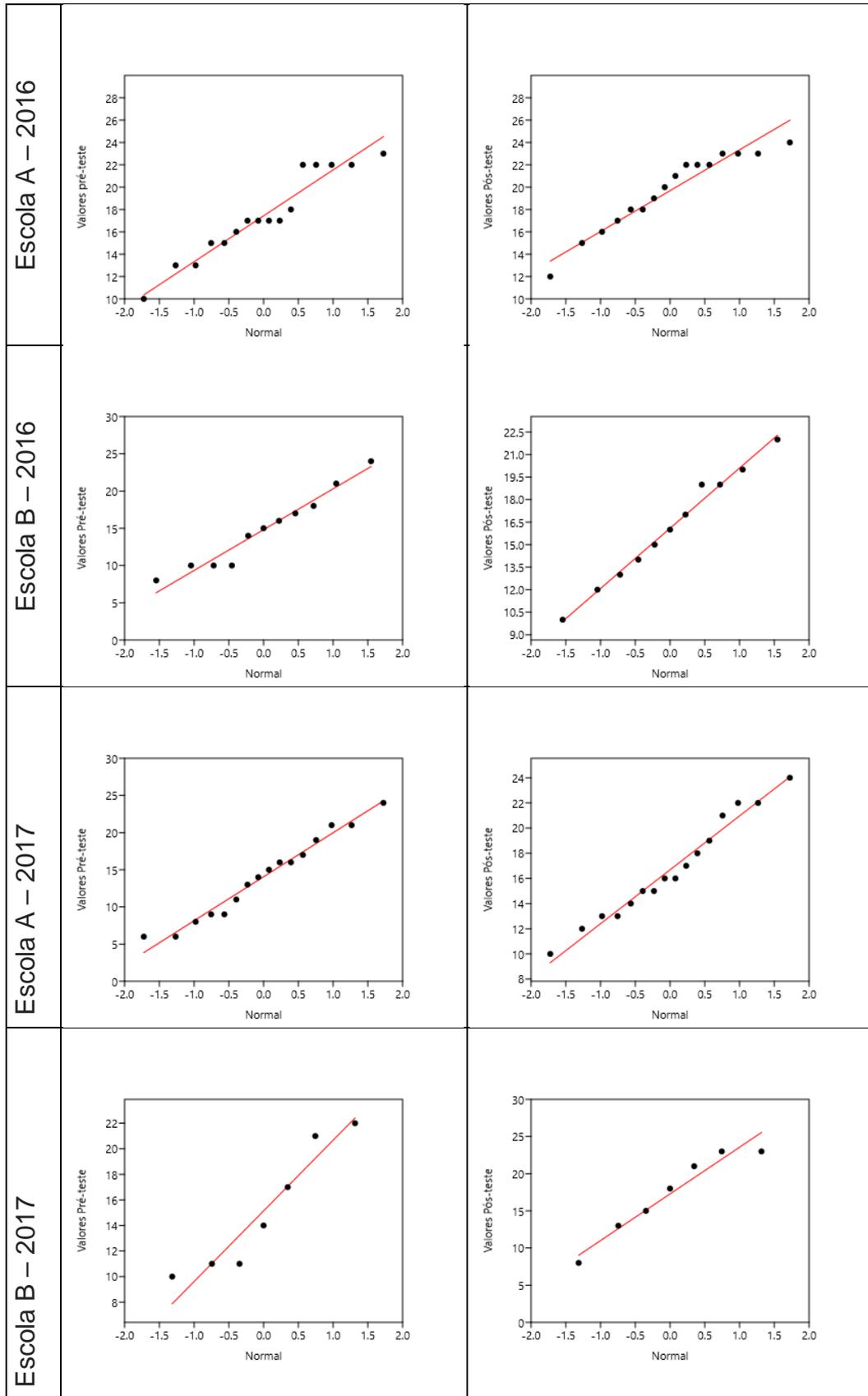
Escola	TPC	N	Shapiro-Wilk W	p(normal)
Escola A 2016	Pré-teste	16	0,9225	0,1851
	Pós-teste	16	0,9242	0,1973
Escola B 2016	Pré-teste	11	0,9468	0,6034
	Pós-teste	11	0,9802	0,9672
Escola A 2017	Pré-teste	16	0,9585	0,6341
	Pós-teste	16	0,9656	0,7633
Escola B 2017	Pré-teste	7	0,8779	0,2171
	Pós-teste	7	0,9231	0,4939

Fonte: desenvolvido pelo autor

O teste de Shapiro-Wilk permitiu validar o pressuposto da normalidade das variáveis em todas as amostras, pois a medida que $P[W > W_{\text{calculado}}]$ é maior que $\alpha = 0,05$ não é possível rejeitar H0. Desta forma, podemos afirmar com nível de significância de 5% que a amostra provém de uma população normal.

Esse fato pode ser visualizado pela aleatoriedade dos pontos em torno da reta, conforme exposto na figura 29.

Figura 29: Gráficos de normalidade Pré-teste e Pós-Teste



Fonte: o autor

A normalidade da amostra possibilitou a utilização de testes paramétricos. Dentre as opções disponíveis de teste foi utilizado o *teste t pareado* para analisar a diferença entre as médias do pré-teste e pós-teste. Conforme Galtz (2013) o teste *t pareado* é adequado em experimentos onde os sujeitos são observados antes e depois de uma intervenção, servindo para comparar a mudança média em um par de valores.

Para a execução deste teste, definiu-se como hipótese nula, a inexistência de diferença entre as médias dos grupos e, como hipótese 1, médias distintas, como segue:

$$H0: \mu_{Pré-teste\ TPC} = \mu_{Pós-teste\ TPC}$$

$$H1: \mu_{Pré-teste\ TPC} \neq \mu_{Pós-teste\ TPC}$$

A partir dos dados apresentados na Tabela 10, pode-se concluir que na Turma A16 existem evidências ao nível de 5% de um aumento na média do score do TPC após os alunos frequentaram as aulas de Desenvolvimento de Games, pois a medida que o $P(T \leq t)$ uni-caudal é 0,00117527 rejeita-se a hipótese nula. Considerando o Intervalo de Confiança, as médias aumentaram entre 0,94 e 3,56 pontos.

Tabela 10: Turma A16 Teste-t: duas amostras em par para médias

	Pré-teste	Pós-teste
Média	17,4375	19,6875
Variância	14,92916667	11,82916667
Observações	16	16
Diferença das médias	2,25	
Correlação de Pearson	0,778521515	
Hipótese da diferença de média	0	
Gl	15	
Stat t	-3,653990743	
$P(T \leq t)$ uni-caudal	0,00117527	
t crítico uni-caudal	1,753050356	
$P(T \leq t)$ bi-caudal	0,002350541	
t crítico bi-caudal	2,131449546	
95% conf.:	(0,93753	3,5625)

Fonte: elaborado pelo autor

Os resultados demonstrados na Tabela 11 demonstram que a variação da média do pré-teste e pós-teste na Turma B16 foi de apenas 1,27 ponto. O Teste T pareado evidenciou que esta diferença não tem relevância estatística, sendo que o $P(T \leq t)$ uni-caudal 0,209221 é maior que 0,05 comprovando a validade de H0.

Tabela 11: Turma B16 Teste-t: duas amostras em par para médias

<i>Resultado</i>	<i>Pré-teste</i>	<i>Pós-teste</i>
Média	14,81818	16,09091
Variância	25,56364	13,69091
Observações	11	11
Diferença das médias	1,2727	
Correlação de Pearson	0,380489	
Hipótese da diferença de média	0	
gl	10	
Stat t	-0,84392	
P(T<=t) uni-caudal	0,209221	
t crítico uni-caudal	1,812461	
P(T<=t) bi-caudal	0,418442	
t crítico bi-caudal	2,228139	
95% conf.:	(-2,0875	4,633)

Fonte: elaborado pelo autor

Cabe destacar que o pesquisador, ao ministrar as aulas para a Turma B16, observou que os alunos eram mais dispersos que os alunos da Turma A16. Estes resultados confirmam a impressão subjetiva do professor de que a aprendizagem destes alunos seria muito inferior à da Turma A16.

Com intuito de ampliar a amostra e verificar se os resultados se confirmavam em outras turmas, foram ofertadas mais duas turmas do curso de Desenvolvimento de Games. Desta forma, foram ministradas as turmas: A17 e B17, no ano de 2017.

Na Tabela 12, estão demonstrados os resultados do teste t pareado da Turma A17. Cabe destacar que a hipótese nula afirma que não existe diferença média dos testes. Como o valor de p é 0,003 é menor do que o nível de significância de 0,05, este resultado permite rejeitar a hipótese nula e concluir que há uma diferença entre as médias obtidas antes e depois do Curso de Desenvolvimento de Games. Considerando um Intervalo de Confiança de 95% pode-se afirmar que o curso proporciona um aumento no score do TPC entre 0,83 e 4,42 pontos.

Tabela 12: Turma A17 Teste-t: duas amostras em par para médias

	<i>Pré-teste</i>	<i>Pós-teste</i>
Média	14,0625	16,6875
Variância	30,99583333	16,22916667
Observações	16	16
Diferença das médias	2,625	
Correlação de Pearson	0,800507645	
Hipótese da diferença de média	0	
Gl	15	
Stat t	-3,121260649	
P(T<=t) uni-caudal	0,003503085	
t crítico uni-caudal	1,753050356	
P(T<=t) bi-caudal	0,00700617	
t crítico bi-caudal	2,131449546	
95% conf.:	(0,83244 4,4176)	

Fonte: elaborado pelo autor

Na Turma B17 verifica-se um aumento na média do score do TPC de 2,1429 após os alunos frequentarem as aulas de Desenvolvimento de Games. Considerando que o P(T<=t) uni-caudal é 0,042, rejeita-se a hipótese nula. Considerando o Intervalo de Confiança, as médias sofreram variação entre -0,39 e 4,67 pontos.

Tabela 13: Turma B17 Teste-t: duas amostras em par para médias

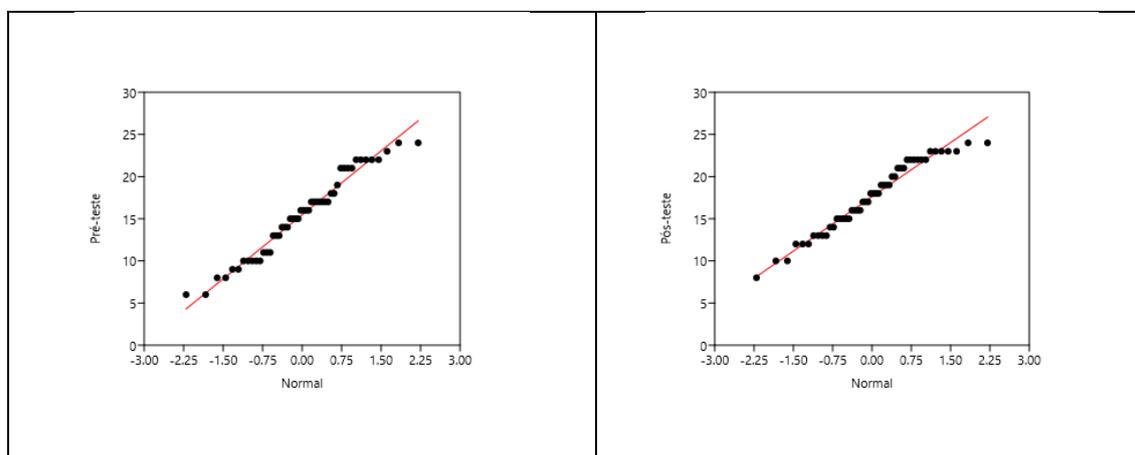
	<i>Pré-teste</i>	<i>Pós-teste</i>
Média	15,14285714	17,28571429
Variância	24,47619048	31,57142857
Observações	7	7
Diferença das médias	2,1429	
Correlação de Pearson	0,873638677	
Hipótese da diferença de média	0	
gl	6	
Stat t	-2,073490549	
P(T<=t) uni-caudal	0,041741629	
t crítico uni-caudal	1,943180281	
P(T<=t) bi-caudal	0,083483258	
t crítico bi-caudal	2,446911851	
95% conf.:	(-0,38591 4,6716)	

Fonte: elaborado pelo autor

Visando integrar os resultados obtidos nesta pesquisa, comparou-se as médias do TPC do pré-teste e pós-teste integrando o resultado das 4 turmas. Através do teste de Shapiro-Wilk obteve-se um p(normal) de 0,102 para o Pré-teste e 0,065 no pós-teste, confirmando a normalidade da amostra.

A figura 30 apresenta a distribuição dos escores do teste TPC, evidenciando que a população estudada atende uma distribuição normal.

Figura 30: Distribuição dos escores do TPC



Fonte: elaborado pelo autor

Foi realizado o teste T pareado, tendo como resultado $P(T \leq t)$ uni-caudal de 0,000025. Desta forma, ficou comprovado que existem evidências ao nível de 5% de aumento na média do escore do TPC após os alunos frequentarem as aulas de Desenvolvimento de Games. Considerando o Intervalo de Confiança, as médias aumentaram entre 1,17 e 3,11 pontos.

Tabela 14: Teste-t pareado

	<i>Pré-teste</i>	<i>Pós-teste</i>
Média	15,46	17,6
Variância	24,29429	17,46939
Observações	50	50
Diferença das Médias	2,14	
Correlação de Pearson	0,732276	
Hipótese da diferença de média	0	
gl	49	
Stat t	-4,44442	
$P(T \leq t)$ uni-caudal	0,000025	
t crítico uni-caudal	1,676551	
$P(T \leq t)$ bi-caudal	0,000050	
t crítico bi-caudal	2,009575	
95% conf.:	(1,1724 3,1076)	

Fonte: elaborado pelo autor

A variação da média através dos testes estatísticos apresentados corrobora a validação da metodologia de ensino-aprendizagem utilizada neste estudo, demonstrando sua eficiência para promover o Pensamento Computacional.

7.2 PENSAMENTO COMPUTACIONAL E CAPACIDADE DE RACIOCÍNIO DOS ALUNOS.

Um dos objetivos propostos neste estudo era: **3. Verificar o efeito da aprendizagem do Pensamento Computacional na capacidade de raciocínio dos alunos.** A análise das médias dos resultados brutos dos testes RV - Raciocínio Verbal, RA – Raciocínio Abstrato, RM- Raciocínio Mecânico, RE – Raciocínio Espacial e RN – Raciocínio Numérico apresentada na tabela 15, demonstra que houve um aumento na média em todos os testes.

Tabela 15: Variação das Médias dos Testes de Raciocínio

	N	Min	Max	Sum	Mean	Std. error	Variance
Pré-RV	50	3	22	768	15.36	0.5998367	179.902
RV	50	9	24	834	16.68	0.5146348	1.324.245
Pré-PRA	50	4	24	854	17.08	0.5382056	1.448.327
RA	50	7	23	918	18.36	0.5560502	1.545.959
Pré-RM	50	2	15	481	9.62	0.3968138	7.873.061
RM	50	3	19	543	10.86	0.5229508	1.367.388
Pré-RE	50	3	20	645	12.9	0.6089469	1.854.082
RE	50	4	20	658	13.16	0.6641613	2.205.551
Pré-RN	50	1	19	484	9.68	0.7041799	2.479.347
RN	50	0	20	529	10.58	0.7714762	2.975.878
Pré-BPR5	50	13	91	3232	64.64	2.286.564	2.614.188
BPR5	50	29	98	3482	69.64	2.379.635	2.831.331
Pré-TPC	50	6	24	773	15.46	0.697055	2.429.429
TPC	50	8	24	880	17.6	0.5910903	1.746.939

Fonte: elaborado pelo autor

Para determinar se esta diferença das médias tinha relevância estatística, ou ocorreu por acaso, foi necessário testar a distribuição de frequência utilizando o teste de Shapiro-Wilk, no intuito de verificar a possibilidade de utilizar testes paramétricos.

Considerando a regra de decisão do teste Shapiro-Wilk, podemos afirmar com nível de significância de 5%, que a amostra obtida nos pré-testes RV, RA e EPN5, não segue uma distribuição normal. Analisando os resultados do pós-teste, verificamos que nos testes RA e RV, também não foi possível rejeitar a hipótese nula. Estes dados podem ser observados na tabela 16, apresentada a seguir:

Tabela 16: Teste de Shapiro-Wilk

		N	Shapiro-Wilk W	(normal)
Pré-teste	RV	50	0.9438	0.01913
	RA	50	0.9535	0.04756
	RM	50	0.9672	0.1772
	RE	50	0.9545	0.05227
	RN	50	0.9621	0.109
	BPR5	50	0.9533	0.04675
Pós-teste	RV	50	0.98	0.5531
	RA	50	0.8602	0,0296
	RM	50	0.9866	0.8393
	RE	50	0.9334	0.007453
	RN	50	0.954	0.05001
	BPR5	50	0.9626	0.1138

Fonte: elaborado pelo autor

A consequência de não seguir uma distribuição normal é o impedimento de utilizar testes paramétricos, os quais precisam atender esta premissa. Neste caso específico, não podemos fazer uso do teste t pareado para comparar as médias do pré-teste e pós-teste. Por este motivo, para padronizar a análise estatística das médias obtidas nos testes de raciocínio, utilizou-se o teste de Wilcoxon para comparar a diferença das médias.

Comparando as médias obtidas no teste de Raciocínio Verbal, constata-se com o teste de Wilcoxon que houve um aumento da média, comprovado em termos estatísticos com um nível de significância de 5%. Para este teste é esperada uma diferença de pontuação de 1,5 ponto com margem de erro de 1 ponto, ou seja, os alunos que frequentaram o curso obtiveram um incremento no escore deste teste entre 0,5 e 2,5 pontos superior ao resultado da primeira avaliação. Estes resultados estão apresentados na tabela 17:

Tabela 17: Análise da Médias do Raciocínio Verbal

<i>Tabela da Estatística do Teste (Wilcoxon)</i>	
<i>Informações</i>	<i>Valores</i>
Estatística	806
P-valor	0,0036
Hipótese Nula	0
Limite Inferior	0,499974285
(Pseudo) Mediana	1,499974734
Limite Superior	2,499952547
Nível de Confiança	0,95

Fonte: elaborado pelo autor

Em relação à diferença de médias do teste de Raciocínio Abstrato, o teste de Wilcoxon comprovou que o aumento da média é relevante em termos estatístico, com um nível de significância de 5%. Neste teste os alunos obtiveram um incremento em torno de 1,5 ponto em relação à primeira testagem realizada, conforme demonstrado na tabela 18:

Tabela 18: Análise da Médias do Raciocínio Abstrato

<i>Tabela da Estatística do Teste (Wilcoxon)</i>	
<i>Informações</i>	<i>Valores</i>
Estatística	814
P-valor	0,0027
Hipótese Nula	0
Limite Inferior	0,500039073
(Pseudo) Mediana	1,999999486
Limite Superior	2,500061646
Nível de Confiança	0,95

Fonte: elaborado pelo autor

O incremento da média no teste Raciocínio Mecânico, também foi comprovado através do teste de Wilcoxon com um nível de significância de 5%. Para este teste é esperada uma diferença de pontuação de 1,5 ponto com margem de erro de 1 ponto, ou seja, os alunos que frequentaram o curso obtiveram um incremento no escore deste teste entre 0,5 e 2,5 pontos superior ao resultado da primeira avaliação. Estes resultados estão apresentados na tabela 19:

Tabela 19: Análise das Médias do Teste de Raciocínio Mecânico

<i>Tabela da Estatística do Teste (Wilcoxon)</i>	
<i>Informações</i>	<i>Valores</i>
Estatística	726,5
P-valor	0,0068
Hipótese Nula	0
Limite Inferior	0,499976114
(Pseudo) Mediana	1,500079589
Limite Superior	2,500008729
Nível de Confiança	0,95

Fonte: elaborado pelo autor

O teste de Wilcoxon não comprovou a existência de diferença de pontuação em relação a primeira e segunda avaliação do teste de Raciocínio Espacial, pois a medida que o p-valor é superior a 0,05, confirma-se H0. Os resultados deste teste são apresentados na tabela 20:

Tabela 20: Análise das Médias do Teste de Raciocínio Espacial

<i>Tabela da Estatística do Teste (Wilcoxon)</i>	
<i>Informações</i>	<i>Valores</i>
Estatística	607
P-valor	0,4683
Hipótese Nula	0
Limite Inferior	-0,500038266
(Pseudo) Mediana	0,499933548
Limite Superior	1,499944641
Nível de Confiança	0,95

Fonte: elaborado pelo autor

A diferença de média do teste de Raciocínio Numérico não teve relevância em termos estatísticos, pois com o teste Wilcoxon p-valor foi maior que 0,05, não permitindo rejeitar a hipótese nula. A tabela 21 demonstra os resultados deste teste:

Tabela 21: Análise das Médias do Teste de Raciocínio Numérico

<i>Tabela da Estatística do Teste (Wilcoxon)</i>	
<i>Informações</i>	<i>Valores</i>
Estatística	718,5
P-valor	0,0508
Hipótese Nula	0
Limite Inferior	-7,64E-05
(Pseudo) Mediana	1,000038909
Limite Superior	1,99997262
Nível de Confiança	0,95

Fonte: elaborado pelo autor

Ao serem comparadas as médias do conjunto de testes de raciocínio constata-se com o teste de Wilcoxon que houve um aumento da média, comprovado em termos estatístico com um nível de significância de 5%. Para este teste é esperada uma diferença de pontuação de 5,5 considerando a margem de erro, os alunos que frequentaram o Curso de Desenvolvimento de Games, obtiveram um incremento na BPR-5 entre 3,0 e 7,5 pontos superiores ao resultado da primeira avaliação. Estes resultados estão apresentados na tabela 22:

Tabela 22: Análise das Médias da BPR-5

<i>Tabela da Estatística do Teste (Wilcoxon)</i>	
<i>Informações</i>	<i>Valores</i>
Estatística	1015,5
P-valor	1,00E-04
Hipótese Nula	0
Limite Inferior	3,000005013
(Pseudo) Mediana	5,49951932
Limite Superior	7,500006083
Nível de Confiança	0,95

Fonte: elaborado pelo autor

Com base nos resultados expostos é possível constatar que o curso proposto favoreceu um incremento do Pensamento Computacional e do raciocínio dos alunos, sendo a relação entre estas avaliações tratada no próximo capítulo.

7.3 RELAÇÃO ENTRE PENSAMENTO COMPUTACIONAL E OS DIFERENTES TIPOS DE RACIOCÍNIO

Um dos objetivos específicos propostos para esta pesquisa era: **Analisar a relação entre Pensamento Computacional e os diferentes tipos de raciocínio: Verbal, Raciocínio Numérico, Raciocínio Espacial, Raciocínio Abstrato e Raciocínio Mecânico.** Para atender este objetivo específico foi utilizado a correlação de Pearson.

O coeficiente de correlação do produto de momentos de Pearson, ou coeficiente de correlação linear (r), é uma medida numérica da força da relação entre duas variáveis que representam dados quantitativos, ou seja, mede intensidade da relação linear entre os valores quantitativos emparelhados x e y em uma amostra.

Conforme a tabela 23, o Teste de Pensamento Computacional apresentou uma correlação com todos testes de raciocínio, atingindo uma correlação muito elevada com a soma dos escores da BPR-5. Este fato demonstra que existe uma relação muito estreita com o Pensamento Computacional e um fator geral de inteligência.

Tabela 23: Correlação de Pearson

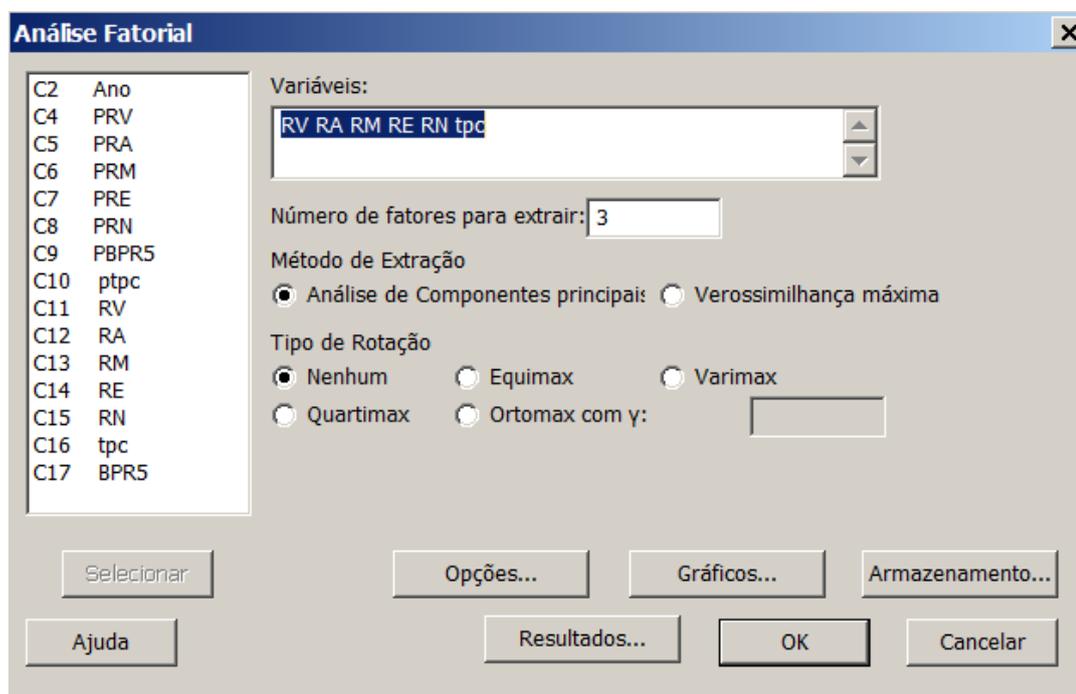
TPC	PRV	PRA	PRM	PRE	PRN	PBPR5
Pré-teste	0,505	0,649	0,405	0,630	0,562	0,696
Pós-teste	0,462	0,670	0,457	0,629	0,646	0,742

Fonte: elaborado pelo autor

Cabe destacar que a relação entre o TCP e os diferentes tipos de raciocínio, mantiveram-se constantes antes e após o curso de Desenvolvimento de Games, fato que demonstra que a correlação não se altera com o desenvolvimento do Pensamento Computacional.

Como existia uma correlação muito grande entre o TCP e os demais testes de raciocínio foi realizada análise fatorial, com apoio do *software* estatístico Minitab. As opções do *software* foram selecionadas para reduzir as variáveis em 3 fatores, utilizando o Método de extração: Análise de componentes principais, conforme figura 31:

Figura 31: Análise Fatorial



Fonte: elaborado pelo autor a partir da tela do Minitab.

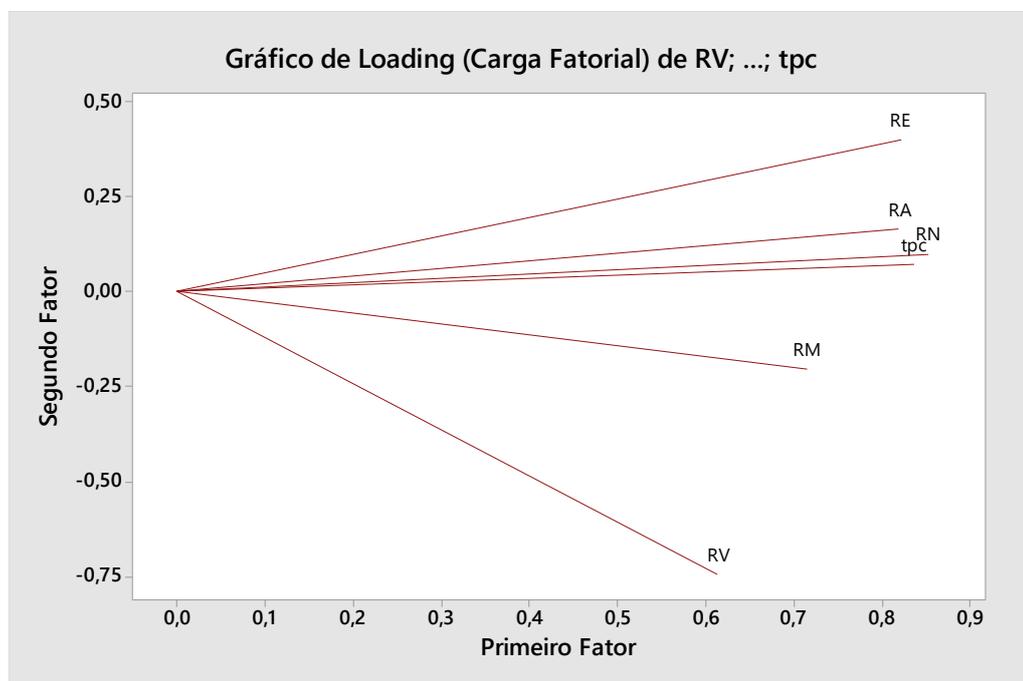
Com a análise fatorial do componente principal da matriz de correlação de RV, RA, RE, RN e TPC, foi possível constatar que apenas 3 fatores explicam 84% das variações nos escores. Este resultado está demonstrado na tabela 24:

Tabela 24: Cargas Fatoriais Não Rotacionadas e Itens Comuns

Variável	Fator1	Fator2	Fator3	Comum
RV	0,614	-0,743	0,189	0,965
RA	0,819	0,163	0,111	0,710
RM	0,715	-0,206	-0,652	0,978
RE	0,822	0,398	-0,114	0,846
RN	0,853	0,099	0,144	0,758
TPC	0,837	0,071	0,274	0,780
Variância	3,6628	0,7942	0,5819	5,0388
% Var	0,610	0,132	0,097	0,840

Fonte: elaborado pelo autor

Na figura 32 é apresentado o Gráfico de Loading que representa a carga fatorial. Com esta representação é possível constatar que os testes que apresentaram maior correlação com o TCP são Raciocínio Abstrato, Raciocínio Espacial e Raciocínio Numérico, os quais formam um fator em conjunto com o Teste de Pensamento Computacional.

Figura 32: Gráfico de Loading

Fonte: elaborado pelo autor

7.4 DISCUSSÃO DOS RESULTADOS

O principal desafio deste trabalho foi desenvolver uma metodologia para apoiar o desenvolvimento do Pensamento Computacional de alunos do Ensino Fundamental e mensurar objetivamente sua eficácia.

Conforme já abordado, todo o plano de Ensino foi baseado na Teoria de Aprendizagem Significativa de Ausubel (1980), havendo uma preocupação em cada lição de buscar os subsunçores para apoiar os novos conhecimentos que seriam apresentados em cada atividade. A prática de programação com Scratch possibilitou ricas experiências práticas, as quais foram exploradas com base na Teoria da Aprendizagem Experiencial de Kolb (1984).

A utilização do Teste de Pensamento Computacional concebido por Roman (2015) como instrumento de pré-teste e pós-teste permitiu demonstrar que a metodologia proposta possibilitou um incremento do Pensamento Computacional dos alunos.

Neste estudo não foi possível a utilização de um grupo controle, procedimento que permitiria isolar fatores externos que pudessem influenciar o desenvolvimento do Pensamento Computacional, no entanto, como o experimento ocorreu em 7 semanas, este intervalo reduziu a possibilidade da aprendizagem escolar ter sido responsável pelo incremento observado.

Cabe destacar que a metodologia foi utilizada em 4 turmas distintas, apresentando incremento do Pensamento Computacional e de diferentes tipos de raciocínio em todas as turmas. Considerando que as edições do curso ocorreram em escolas diferentes e em 2 anos consecutivos, este fato reforça o papel do Curso de Desenvolvimento de Games nos resultados mensurados, afastando a hipótese de ser decorrente de fatores externos como a aprendizagem escolar.

No presente estudo todas as turmas do Curso de Desenvolvimento de Games foram ministradas pelo próprio pesquisador, o que facilitou a padronização das aulas. Ao replicar este experimento para sala de aula torna-se relevante atentar para o plano de aula utilizado e talvez seja necessário um maior detalhamento para o mesmo ser replicado por terceiros.

Os alunos não receberam os testes realizados e não tiveram acesso aos resultados obtidos até o término da pesquisa. Como o intervalo temporal utilizado entre o teste e reteste foi de 7 semanas, este tempo foi suficiente para impedir que os alunos lembrassem as questões utilizadas nos testes. Cabe destacar que em estudos de precisão da BPR5 - Bateria de Provas de Raciocínio Lemos (2006) utilizou o intervalo de apenas 1 mês entre teste e reteste encontrando um coeficiente de correlação superior a 0.75.

Diversos estudos ao conceituar o Pensamento Computacional utilizam conceitos relacionados a habilidades de raciocínio abordando aspectos como: decomposição,

reconhecimento de padrões, abstração e pensamento algorítmico (BBC Learning; 2015; Google;2015; ISTE e CSTA, 2011). Nesta pesquisa, encontramos uma correlação do Teste de Pensamento, com todos os testes de raciocínio, sendo que a correlação com a soma dos escores do BPR-5 foi de 0,74, o que demonstra sua relação estreita com um fator geral de inteligência.

Na pesquisa bibliográfica não foram encontrados estudos demonstrando esta correlação entre inteligência e Pensamento Computacional. A pesquisa de Ambósio (2014) demonstrou a existência de uma correlação entre inteligência e o desempenho acadêmico na disciplina de programação, sendo o que estudo que mais se aproxima dos resultados apresentados neste trabalho. Destaca-se como importante contribuição desta Tese demonstrar que com o desenvolvimento do Pensamento Computacional é possível gerar um incremento na capacidade de raciocínio.

Os resultados apresentados neste tópico serão devidamente integrados e discutidos no próximo capítulo.

8 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Esta tese teve como objetivo geral investigar a relação entre a construção do Pensamento Computacional e o desenvolvimento do raciocínio, tendo testado e comprovado as duas hipóteses formuladas no início do estudo: 1. O ensino do Pensamento Computacional melhora a capacidade de raciocínio dos alunos e 2. Existe uma correlação entre Pensamento Computacional e capacidade de raciocínio.

A revisão da literatura acerca das teorias de aprendizagem, em especial sobre a Teoria de Aprendizagem Significativa de Ausubel (1980) e Teoria da Aprendizagem Experiencial de Kolb (1984), serviram de base para elaboração de uma proposta pedagógica para ensino do Pensamento Computacional, a qual resultou no Curso de Desenvolvimento de Games utilizado nesta pesquisa.

Considera-se um diferencial deste estudo a utilização do Teste de Pensamento Computacional de Roman (2015) e do BPR-5 - Bateria de Provas de Raciocínio, pois estes instrumentos permitiram mensurar o Pensamento Computacional e o raciocínio dos alunos de forma objetiva. Deste modo, foi possível ultrapassar a descrição subjetiva do pesquisador e realizar elaborado tratamento estatístico dos dados.

Uma vez que o escore no Teste de Pensamento Computacional foi significativamente superior após a realização do curso, os resultados obtidos permitiram validar a presente metodologia de ensino e inferir acerca dos possíveis benefícios que serão obtidos com sua adoção nas salas de aula do Ensino Fundamental. A medida que esta pesquisa permite sua adoção por professores e pesquisadores, ultrapassa os limites meramente acadêmicos.

Com este trabalho foi constatado que até mesmo intervenções breves, como a utilizada nesta pesquisa, quando elaboradas com uma metodologia adequada, podem produzir efeitos relevantes para seus participantes. O fato de um curso de 10 horas de duração gerar alterações na capacidade cognitiva, possíveis de serem mensuradas e estatisticamente comprovadas, confirmam a importância de inserir conteúdos para o desenvolvimento do Pensamento Computacional nas escolas brasileiras.

Outro dado relevante desta pesquisa foi demonstrar a correlação existente entre o Pensamento Computacional e os demais tipos de raciocínio, reforçando sua importância no desenvolvimento cognitivo. A medida que o incremento do Pensamento Computacional favoreceu o desenvolvimento do Raciocínio Verbal, Raciocínio Abstrato e Raciocínio Mecânico, é possível concluir que a melhora na cognição do sujeito relacionada com a construção do Pensamento Computacional possa favorecer a aprendizagem de outros componentes curriculares.

A análise fatorial, realizada com os dados deste estudo, permitiu reduzir as variações dos escores a somente três fatores, os quais seriam responsáveis por mais de 84% das variações de escore dos sujeitos. O fato do Pensamento Computacional compor um fator em conjunto com Raciocínio Abstrato, Raciocínio Espacial e Raciocínio Numérico, sugere que o Teste de Pensamento Computacional talvez seja apenas uma forma de medir aspectos da inteligência relacionados com estes tipos de raciocínio.

Considerando que o número de fatores identificados nesta análise fatorial foi inferior a divisão dos tipos de raciocínio propostos pela BPR-5, a partir deste dado, seria possível retomar a discussão acerca da natureza da inteligência humana, ou seja: existem vários fatores que compõem a inteligência, ou apenas uma inteligência geral?

Seria importante que novos estudos ampliem a amostra e utilizem também alunos com características sociodemográficas distintas da população aqui apresentada. Desta forma, será possível confirmar os resultados desta pesquisa em outras amostras.

Um tema que merece ser devidamente estudado é o desenvolvimento do Pensamento Computacional em adultos, ou mesmo da população idosa, pois seria interessante verificar se ocorrem os mesmos benefícios na capacidade de raciocínio numa população adulta.

A fim de reduzir os custos envolvidos em novos estudos, bem como abreviar o tempo necessário para realização das avaliações, o Teste de Pensamento Computacional pode ser relacionado apenas a um teste de Inteligência Geral, ou com teste de Raciocínio Abstrato.

A partir dos resultados deste estudo, o Curso de Desenvolvimento de Games está sendo modelado para um Ambiente Virtual de Aprendizagem. Pretende-se verificar a possibilidade de obter os mesmos benefícios utilizando aulas EAD. Desta forma, os benefícios desta metodologia podem ser estendidos para um maior número de participantes.

Como resultado principal desta pesquisa, espera-se que sirva de incentivo para novas práticas de ensino do Pensamento Computacional no Brasil. Com intuito de facilitar a reprodução da metodologia proposta neste trabalho e sua adoção em salas de aula, está sendo elaborado um livro didático com a descrição detalhada do plano de aula utilizado, bem como de propostas desplugadas para complementar as atividades no Scratch.

REFERÊNCIAS

AFONSO, Maria João **Paradigmas diferencial e sistêmico de investigação da inteligência humana: perspectivas sobre o lugar e o sentido do construto**. 2007. Tese (doutorado em Psicologia – Psicologia Diferencial), Faculdade de Psicologia e de Ciências da Educação, Universidade de Lisboa, 2007

AKCAOGLU, M. Learning problem-solving through making games at the game design and learning summer program. **Educational technology research and development**, out. 2014. v. 62, n. 5, p. 583–600.

AMBROSIO, A. P. et al. **Exploring core cognitive skills of computational thinking**. [S.l.]: [s.n.], 2014. p. 25–34. Disponível em: <<http://repositorium.sdum.uminho.pt/handle/1822/30076>>. Acesso em: 2 jul. 2016.

AMBRÓSIO, A., Almeida, L., Macedo, J., Santos, A., Franco, A. (2011). **Programação de computadores: compreender as dificuldades de aprendizagem dos alunos**. Revista Galego Portuguesa de Psicoloxía e Educación, 19 (1). 185-197.

ARMONI, M.; MEERBAUM-SALANT, O.; BEN-ARI, M. **From scratch to “real” programming**. ACM transactions on computing education, 24 fev. 2015. v. 14, n. 4, p. 1–15.

ARNOLD, W.; EYSENCK, H.J.; MEILI, R. **Dicionário de psicologia**. São Paulo: Loyola, 1994.

AURELIANO, V. C. O.; TEDESCO, P. **Avaliando o uso do scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação**. [S.l.]: [s.n.], 2012. V. 32, p. 1–10. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2012/006.pdf>>. Acesso em: 3 jul. 2016.

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

BARCELOS, Thiago Schumacher. **Relações entre o pensamento computacional e a matemática em atividades didáticas de construção de jogos digitais** / Thiago Schumacher Barcelos. -- São Paulo; SP: [s.n.], 2014. 276 p. Orientador: Ismar Frango Silveira. Tese (doutorado) - Programa de Pós-Graduação em Ensino de Ciências e Matemática, Universidade Cruzeiro do Sul

BAREFOOT. **Computational Thinking** (2015) Disponível em: <http://barefootcas.org.uk/wp-content/uploads/2014/10/Computational-thinking-Barefoot-Computing.pdf> Acesso em: 2 jul. 2016.

BARR V. and Stephenson C., **Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community?** ACT Inroads, vol. 2., no. 1, (2011), pp. 48-54.

BBC LEARNING, Bitesize. **What is computational thinking?** s.d. Disponível em: <<http://www.bbc.co.uk/education/guides/zp92mp3/revision>>. Acesso em: 23 nov. 2015.

BERNATE, Sandra Patricia Peña **Análisis de tareas para instrumento e medición de pensamiento computacional** (2016) Disponível em:
<<http://eduteka.icesi.edu.co/articulos/investigacion-scratch-analisis-tareas>> Acesso em: 08/06/2016.

BOE, B. et al. **Hairball: lint-inspired static analysis of scratch projects**. [S.l.]: ACM, 2013. p. 215–220. Disponível em: <<http://dl.acm.org/citation.cfm?id=2445265>>. Acesso em: 10 jul. 2016.

BRENNAN, K., & Resnick, M. (2012). **New frameworks for studying and assessing the development of computational thinking**. Paper presented at annual American Educational Research Association meeting, Vancouver, BC, Canada.

BUNDY, A. Computational thinking is pervasive. **Journal of Scientific and Practical Computing**, 2007. v. 1, n. 2, p. 67–69.

CALDER, N. **Using Scratch: An Integrated Problem-solving Approach to Mathematical Thinking**. APMC, 2010, 15 (4), 9-14.

CATTELL, R. B. (1998). **Where is intelligence?** Some answers from the triadic theory. In J. J McArdle & R. W. Woodcock (Orgs.), *Human cognitive abilities in theory and practice* (pp. 29-38). New Jersey: Erlbaum.

Computational Thinking in K–12 Education teacher resources second edition
http://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf

CONOLE, G. et al. (Org.). **Design for teaching and learning in a networked world**. Cham: Springer International Publishing, 2015. V. 9307.

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

DAVIDOFF, Linda L. **Introdução a Psicologia**. 3ª ed. São Paulo: Pearsen Educacions, 2001

EUROPEAN SCHOOLNET. **Computing our future**. Disponível em:
<http://www.eun.org/c/document_library/get_file?uuid=3596b121-941c-4296-a760-0f4e4795d6fa&groupId=43887>. Acesso em: 15/03/2015.

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

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

GLANTZ, S. A. **Princípios de bioestatística**. 7. ed. Porto Alegre: AMGH, 2013. 320p.

GOUWS, L., Bradshaw, K., & Wentworth, P. (2013b). **First year student performance in a test for computational thinking**. Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference, 271-277.

GRESSE VON WANGENHEIM, C.; RODRIGUES NUNES, V.; DOS SANTOS, G. D. **Ensino de computação com scratch no ensino fundamental** – um estudo de caso. Revista brasileira de informática na educação, 23 nov. 2014. v. 22, n. 3, p. 115.

GROVER, Shuchi; PEA, Roy. **Computational Thinking in K-12: A Review of the State of Field.** Educational Researcher. Vol 42. No. 1. P. 38-43. 2013.

GÜLBAHAR, Y.; KALELIOĞLU, F.; OTHERS. **The effects of teaching programming via scratch on problem solving skills:** a discussion from learners' perspective. Informatics in education-an international journal, 2014. n. Vol13_1, p. 33–50.

HANDA, Jaime Kenji; SILVA, Jaime Balbino G. **Objetos de Aprendizagem** (Learning Objects). Boletim EAD – UNICAMP. 31 de janeiro de 2003. Disponível em: Acesso em: 04 jan 2006.

HODGINS, H. W. **The future of learning objects.** In: WILEY, D. A. (Ed.). The instructional use of learning objects: online version. 2000. Disponível em: <http://reusability.org/read/chapters/hodgins.doc> . Acesso em: 15/03/2015.

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.

HU, Chenglie 2011. **Computational thinking: what it might mean and what we might do about it.** In Proceedings of the 16th annual joint conference on Innovation and technology in computer science education (ITiCSE '11). ACM, New York, NY, USA, 223-227. DOI=<http://dx.doi.org/10.1145/1999747.1999811>

HUMPHREYS, Simon et alli. **Computational thinking: A guide for teachers 2015.** Disponível em: <<http://computingschool.org.uk/computationalthinking>>

IEEE 1484.12.1-2002. Standard for Information Technology --**Education and Training Systems -- Learning Objects and Metadata.** Disponível em: <http://ltsc.ieee.org/wg12/>. Acesso em 21/12/2015.

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

KOLB. D. A. and Fry, R. (1975) **‘Toward an applied theory of experiential learning;**, in C. Cooper (ed.) Theories of Group Process, London: John Wiley.

LEE, G., Lin, Y., & Lin, J. (2014). **Assessment of computational thinking skill among high school and vocational school students in Taiwan.** World Conference on Educational Multimedia, Hypermedia and Telecommunications, 2014(1), 173-180

LEMOS, Gina C.; ALMEIDA, Leandro S.; GUISANDE, M.Adelina. Bateria de Provas de Raciocínio: suas versões, validação e normalização. (2006). Disponível em: <<https://dspace.uevora.pt/rdpc/handle/10174/1824>> Acesso em: 12 de outubro de 2017.

LOCKE, Edwin A. **Motivation through conscious goal setting.** (1996) Disponível em: <<http://expand.nu/wpcontent/uploads/M%C3%A5ls%C3%A6tning-review.pdf>> Acesso em: 20 de abril de 2013.

MALONEY, J. H. et al. **Programming by choice:** urban youth learning programming with scratch. [S.l.]: ACM, 2008. V. 40.

MARQUES, M. C. P. de O. **O ensino da programação no desenvolvimento de jogos através do ambiente scratch.** 2013. Disponível em: <<http://repositorium.sdum.uminho.pt/handle/1822/29142>>. Acesso em: 3 jul. 2016.

MATLIN, M. W. (2004). **Psicologia Cognitiva.** Journal of Personality. 5 ed. Rio de Janeiro: LTC, 403 p.

MAYER, R. E. (2001). **Multimedia learning.** New York: Cambridge University Press.

MCCLELLAND, David C., **Human Motivation,** 1987, Cambridge University Press, Cambridge

MCGREW, Kevin S. **CHC theory and the human cognitive abilities project:** Standing on the shoulders of the giants of psychometric intelligence research. *Intelligence*, Vol 37(1), Jan 2009, 1-10. Disponível em: <http://iapsych.com/articles/mcgrew2009.pdf> Acesso em 08 jan. 2016.

MICROSOFT. **A National Talent Strategy Ideas For Securing U.S. Competitiveness and Economic Growth Microsoft Report.** September, 2012. Disponível em: <<https://news.microsoft.com/download/presskits/citizenship/MSNTS.pdf>> Acesso em: 2 jul. 2016.

MILLER, G. A. (1956). "**The magical number seven, plus or minus two:** Some limits on our capacity for processing information". *Psychological Review* 63 (2): 81–97. doi:10.1037/h0043158. PMID 13310704. Disponível em: <<http://psychclassics.yorku.ca/Miller/>> Acesso em: 2 jun: 2015.

MOREIRA, J. A.; OLIVEIRA MATTOS, G. DE; REIS, L. **Um Panorama da Presença Feminina na Ciência da Computação.** [s.d.]. Disponível em: <<http://paradoxzero.com/zero/redor/wp-content/uploads/2015/07/1935-4736-1-PB.pdf>>. Acesso em: 24 jul. 2017.

MORENO, J., Robles, G. e Chusig, C. (2014) **Dr Scratch,** Automatic analysis of Scratch projects to assess the development of CT. MIT, Scratch Conference, Boston.

MORENO, R., & Mayer, R. (2003). "**Nine Ways to Reduce Cognitive Load in Multimedia Learning** " *Educational Psychologist* 38 (1), 43–52.

NETO, V. D. S. M. **A utilização da ferramenta Scratch como auxílio na aprendizagem de lógica de programação.** [S.l.]: [s.n.], 2013. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/2675>>. Acesso em: 3 jul. 2016.

NEVES, L. et al. **Um relato de experiência do uso de programação de jogos para a melhoria do raciocínio lógico em crianças.** 2010. Disponível em: <<http://www.tise.cl/volumen11/TISE2015/578-583.pdf>>. Acesso em: 2 jul. 2016.

OLIVEIRA, M. L. S. DE et al. **Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência.** [S.l.]: [s.n.], 2014. Disponível em: <http://www.academia.edu/download/34381042/Ensino_de_logica_de_programacao_no_ensino_fundamental.pdf>. Acesso em: 3 jul. 2016.

PARDAMEAN. **Enhancement of creativity through logo programming.** American journal of applied sciences, 1 abr. 2014. v. 11, n. 4, p. 528–533.

PARK, S.-Y.; SONG, K.-S.; KIM, S.-H. **Cognitive load changes in pre-service teachers with computational thinking education.** International journal of software engineering and its applications, 31 out. 2015. v. 9, n. 10, p. 169–178.

PRIMI, R., & Almeida, L. S. (2000). **BPR-5 Bateria de Provas de Raciocínio – Manual Técnico.** São Paulo: Casa do Psicólogo.

PRIMI, Ricardo. **Inteligência: avanços nos modelos teóricos e nos instrumentos de medida.** Aval. psicol., Porto Alegre , v. 2, n. 1, p. 67-77, jun. 2003. Disponível em <http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1677-04712003000100008&lng=pt&nrm=iso>. Acesso em 14 mar. 2016.

RAHMAN und Govidarajulu (1997). **A modification of the test of Shapiro and Wilk for normality.** Journal of Applied Statistics. 24 (2): 219–236. doi:10.1080/02664769723828

RAZALI, Nornadiah; Wah, Yap Bee (2011). **Power comparisons of Shapiro–Wilk, Kolmogorov–Smirnov, Lilliefors and Anderson–Darling tests** (PDF). Journal of Statistical Modeling and Analytics. 2 (1): 21–33. Retrieved 30 March 2017 Disponível em: <https://www.researchgate.net/profile/Bee_Yap/publication/267205556_Power_Comparisons_of_Shapiro-Wilk_Kolmogorov-Smirnov_Lilliefors_and_Anderson-Darling_Tests/links/5477245b0cf29afed61446e1/Power-Comparisons-of-Shapiro-Wilk-Kolmogorov-Smirnov-Lilliefors-and-Anderson-Darling-Tests.pdf> Acesso em: 13 abr. 2017.

RESNICK, M. **Rethinking learning in the digital age.** [S.l.]: The Global Information Technology Report: Readiness for the Networked World. Oxford University Press, 2002.

ROBBINS, Stephen Paul. **Comportamento organizacional.** São Paulo, São Paulo: Prentice Hall, 2002

ROCHA, A. S. et al. (2013) **Utilização do Scratch como Ferramenta de Auxílio à Aprendizagem de Programação.** Anais do COBENGE 2013 - XLI Congresso Brasileiro de Educação em Engenharia, Gramado, RS, Brasil.

ROMÁN, M. (2015b). **Test de Pensamiento Computacional** [Test en línea]. Recuperado de <http://goo.gl/IYEKMB>

ROYAL SOCIETY. (2012). **Shut down or restart: The way forward for computing in UK schools.** Disponível em: <<http://royalsociety.org/education/policy/computing-in-schools/report/>> Acesso em: 03 jun. 2015.

RUDIO, F. V. **Introdução ao projeto de pesquisa científica.** Petropolis: Vozes, 2007.

SÁEZ-LÓPEZ, J.-M.; ROMÁN-GONZÁLEZ, M.; VÁZQUEZ-CANO, E. **Visual programming languages integrated across the curriculum in elementary school: a two year case study using “scratch” in five schools.** Computers & education, jun. 2016. v. 97, p. 129–141.

SCAICO, P. D. et al. **Ensino de programação no ensino médio: uma abordagem orientada ao design com a linguagem scratch.** Revista brasileira de informática na educação, 30 set. 2013. v. 21, n. 2. Disponível em: <<http://www.br-ie.org/pub/index.php/rbie/article/view/2364>>. Acesso em: 3 jul. 2016.

SHADISH, William R., Cook, Thomas D., Campbell, Donald T. **Experimental and Quasi-Experimental Designs for Generalized Causal Inference**, 2nd Edition, Boston: Cengage Learning, 2001.

SCHELINI, Patrícia Waltz. **Teoria das inteligências fluida e cristalizada: início e evolução.** Estud. psicol. (Natal), Natal, v. 11, n. 3, p. 323-332, Dec. 2006 Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-294X2006000300010&lng=en&nrm=iso>. Acesso: 14 Mar. 2016. <http://dx.doi.org/10.1590/S1413-294X2006000300010>.

SCHOEFFEL, P. et al. **Uma experiência no ensino de pensamento computacional para alunos do ensino fundamental.** [S.l.]: [s.n.], 2015. p. 1474. Disponível em: <<http://br-ie.org/pub/index.php/wcbie/article/view/6333>>. Acesso em: 3 jul. 2016.

SOUSA, R. M.; LENCASTRE, J. A. **Scratch: uma opção válida para desenvolver o pensamento computacional e a competência de resolução de problemas.** 2o encontro sobre jogos e mobile learning, 2014. p. 256–267.

SPEARMAN, C. (1927). **The abilities of man: Their nature and measurement.** New York: Macmillan.

SPEARMAN, C. E. (1904). **‘General intelligence’ objectively determined and measured.** American Journal of Psychology, 15, 201-293. Disponível em: <<http://psychclassics.yorku.ca/Spearman/>> Acesso em 05 jan. 2016.

SHAPIRO, S. S. & Wilk, M. B. 1965. **An analysis of variance test for normality** (complete samples). Biometrika 52:591–611.

SMITH, Megan. **Computer Science For All.** 2016. Disponível em: <<https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all>>. Acesso em: 11 fev. 2016.

SWELLER, John and Merrienboer, Jeroen J.G. van and Paas, Fred G.W.C. (1998) **Cognitive Architecture and Instructional Design.** Educational Psychology Review, 10 (3). pp. 251-296. ISSN 1040-726X SWELLER, John “Cognitive load during problem solving: Effects on learning”, Cognitive Science, vol. 12, (1988), pp. 257-285.

TABORDA, H.; MEDINA, D. **Programación de computadores y desarrollo de habilidades de pensamiento en niños escolares: fase exploratòria.** Julio de, 2012. Disponível em: <http://www.academia.edu/download/33887265/WP-CIES_3-FINAL.pdf>. Acesso em: 2 jul. 2016.

THURSTONE, L. L. (1938). **Primary mental abilities.** Chicago: University of Chicago Press.

VILLELA, Paulo R. C. **Escassez de mão de obra em TI: uma perspectiva quantitativa.** In: Cadernos temáticos do observatório: mercado de trabalho e formação de mão de obra em TI.

SOFTEX, 2013. Disponível em: <http://www.softex.br/wp-content/uploads/2015/10/Cadernos_Tematicos_Mercado_de_trabalho.pdf> Acesso em: 05 de junho de 2016.

VOSKOGLU, M. G.; BUCKLEY, S. **Problem solving and computational thinking in a learning environment**. Arxiv preprint arxiv:1212.0750, 2012. Disponível em: <<http://arxiv.org/abs/1212.0750>>. Acesso em: 2 jul. 2016.

WALDEN, J., Doyle, M., Garns, R., & Hart, Z. (2013). **An informatics perspective on computational thinking**. Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education, 4-9.

WANG, H. Y.; HUANG, I.; HWANG, G. J. **Effects of an integrated scratch and project-based learning approach on the learning achievements of gifted students in computer courses**. [S.l.]: IEEE, 2014. p. 382–387. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6913329>>. Acesso em: 2 jul. 2016.

WERNER, L., Denner, J., Campe, S. (2012). **The Fairy Performance Assessment: Measuring computational thinking in middle school**, Proceedings of Special Interest Group in Computer Science Education, Feb. 29 – Mar. 3, Raleigh, N. Carolina, USA Wing, J. (2006).

WILEY, D. A. **Connecting learning objects to instructional design theory : A definition, a metaphor, and a taxonomy**. 2000. Disponível em: http://wesrac.usc.edu/wired/bldg-7_file/wiley.pdf. Acesso em: 26 de abril de 2014.

WILSON, C., Sudol, L., Stephenson, C., and Stehlik, M., 2010. **Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age**. ACM Available. Disponível em: <<http://www.acm.org/runningonempty/fullreport.pdf>> Acesso em. 20 abr. 2015.

Wing, J. (January de 2008). **Five Deep Questions in Computing**. Communications Of The ACM, pp. Vol. 51, No. 1.

WING, J. M. (2008b). **Computational thinking and thinking about computing**. Philosophical transactions. Series A, Mathematical, physical, and engineering sciences, pp. 366(1881), 3717–25.

WING, J. M. (2011). **Computational Thinking—What and Why?** The Link - Carnegie Mellon School of Computer Science. Disponível em: <<http://link.cs.cmu.edu/article.php?a=600>> Acesso em: 05 abr. 2015.

WING, J., (2014) **Computational Thinking Benefits Society**. Social Issues in Computing. New York: Academic Press. Artigo disponível e consultado em: socialissues.cs.toronto.edu., 2014.

APÊNDICE A - Plano de aula

01 Interface do Scratch e Movimentação

Subsunçor: Conhecimento do Plano Cartesiano

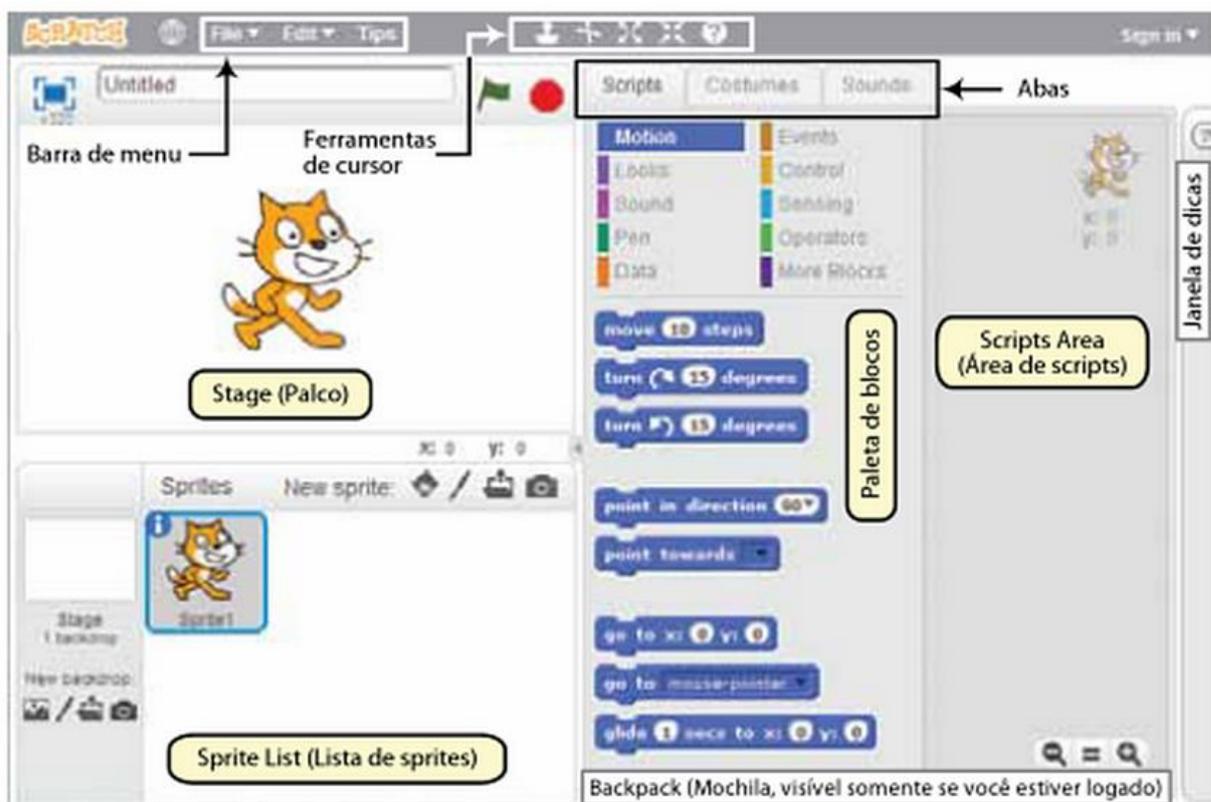
Experiência com jogo Batalha Naval (papel)

Conceito computacional: Sequência

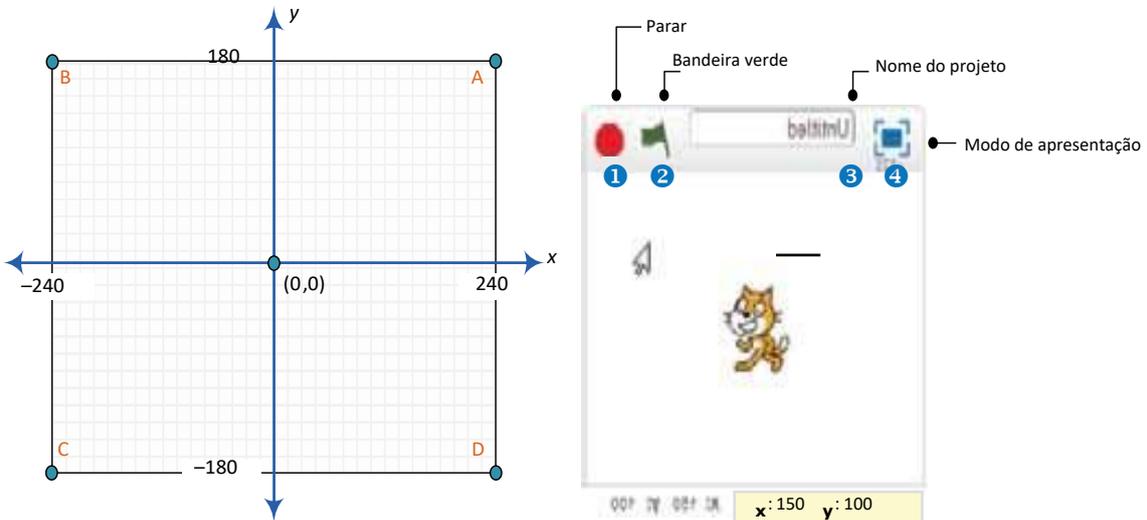
Objetivos:

- Conhecer a interface do Scratch de forma a conseguir alterar palco, substituir sprites e adicionar scripts;
- Movimentar sprites
- Movimentos com teclado e mouse.

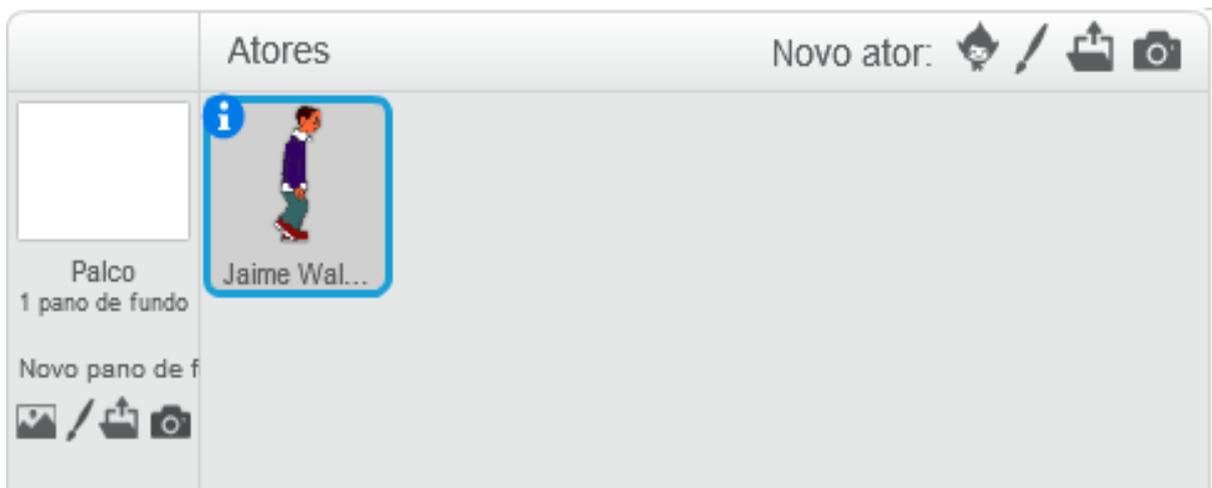
Para iniciar o Scratch, acesse o seu site (<http://scratch.mit.edu/>) e clique no link *Create*.



Palco: O Palco é o local em que os seus *sprites* se movem, são desenhados e interagem. O *Stage* tem 480 passos de largura e 360 passos de altura. O centro do Palco tem coordenada x igual a 0 e coordenada y igual a 0.



Atores: A Lista de *Sprites* (*Sprite List*) exibe os nomes e as miniaturas (*thumbnails*) de todos os sprites de seu projeto. Novos projetos começam com o Palco em branco e um único *Sprite* representado por um gato



Blocos: Os blocos no Scratch estão divididos em dez categorias: Movimento, Aparência, Som, Caneta, Variáveis, Eventos, Controle, Sensores, Operadores e Mais Blocos.

Área de Scripts: Para fazer com que um *sprite* faça algo interessante, é necessário programá-lo arrastando blocos da aba Blocos para a *Scripts Area* (Área de Scripts), unindo-os.

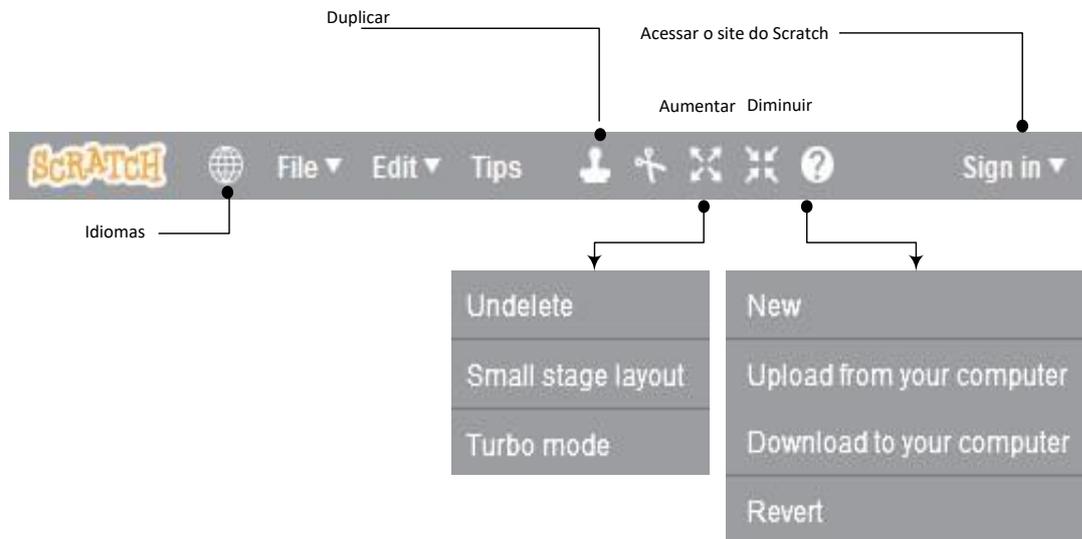


Fantasia: A aparência de um *sprite* pode ser alterada se mudarmos a sua fantasia (costume)

Sons: A aba Sons permite definir os sons que serão utilizados no projeto.

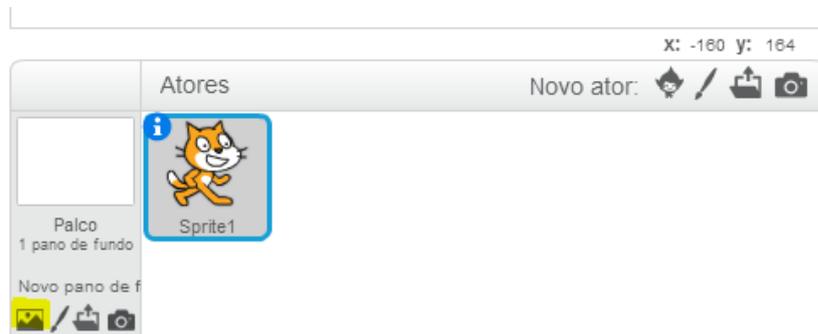
Panos de Fundo: A aba Panos de fundo o selecionar a miniatura do Stage na Lista de Sprites, o nome da aba do meio mudará para *Backdrops* (Panos de fundo).

Barra de ferramentas



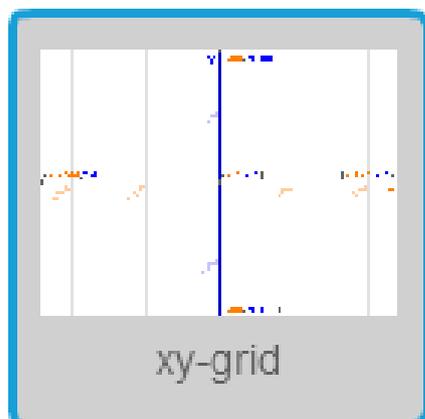
Atividade prática

Modificar Palco



Escolha o palco com o nome xy-grid e pressione

OK



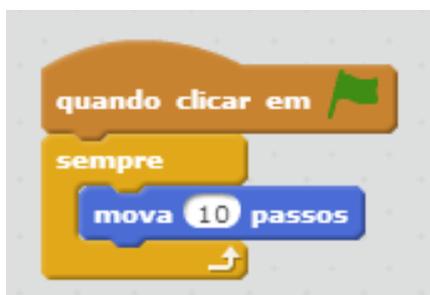
Primeiro código:



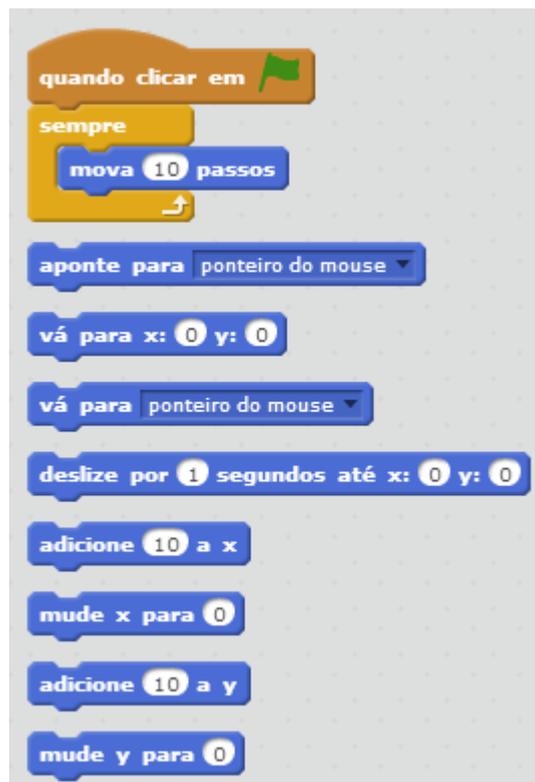
O que aconteceu?



Agora vamos testar desta forma:

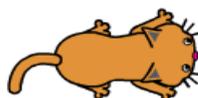


Modifique o bloco por estes:

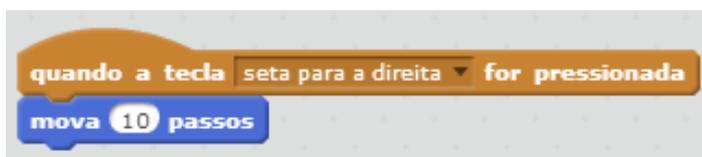


Desafio:

1. Troque o *Sprite* por



2. Teste estes blocos:



3. O que acontece?
4. Será que você consegue descobrir como movimentar um Sprite utilizando as 4 setas?

02 Repetir, Animar, Rotar

Subsunçor: repetir, rotação de figuras (direções), animação

Experiência: carrinho bate e volta

Rotação

Utilizar no Scratch se bater volta

Animação

Mostrar desenho <https://www.youtube.com/watch?v=vndinDhSUSE>

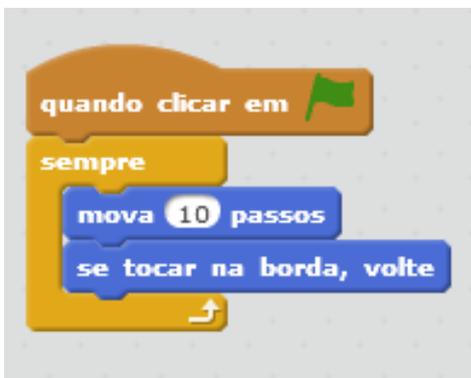
<https://www.youtube.com/watch?v=Njl-uqnmBGA>

Objetivo:

1. Repetir
2. Movimentos complexos
3. Animação de sprites

Código bate-volta

Teste o *script*:

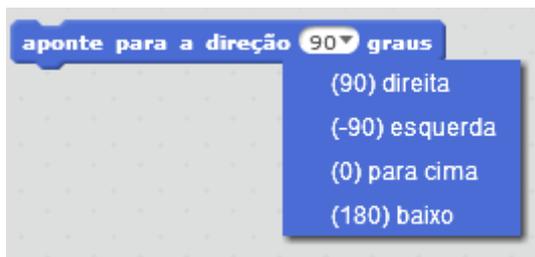


Observe que o personagem volta de cabeça para baixo. Para corrigir, precisamos definir o sentido de rotação do personagem.



Você já sabe mover o Sprite com o teclado, tente fazer isso.

Para mover de forma adequada precisamos incluir um bloco para modificar o sentido de rotação.

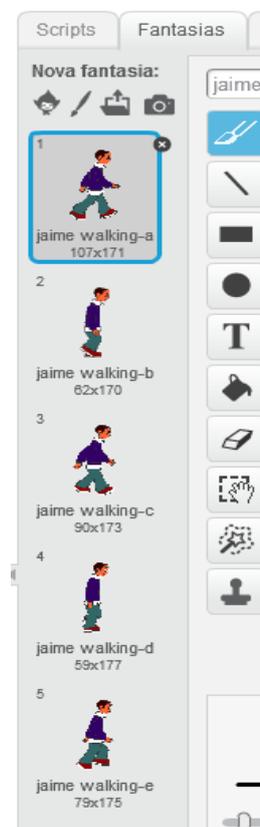
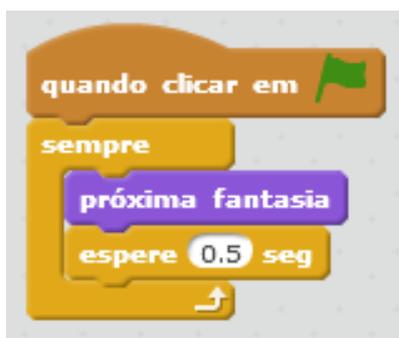


Vamos explorar as fantasias do personagem?

Animar um Sprite é semelhante ao que vimos no vídeo.

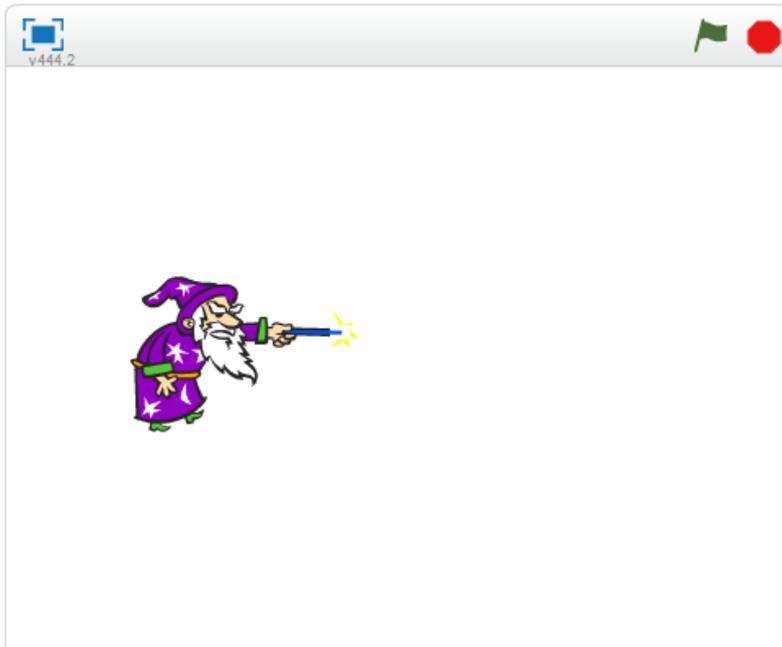
Como vocês acham que seria para animar um Sprite?

Este é o código que faz ele andar no mesmo lugar:



Desafio:

1. Tente movimentar o Sprite e utilizar animação.
2. Como fazemos para o Mago atirar uma estrela?
3. Como fazemos para o personagem dar um pulo?



03 Condicionais

Subsunçor: Colisão

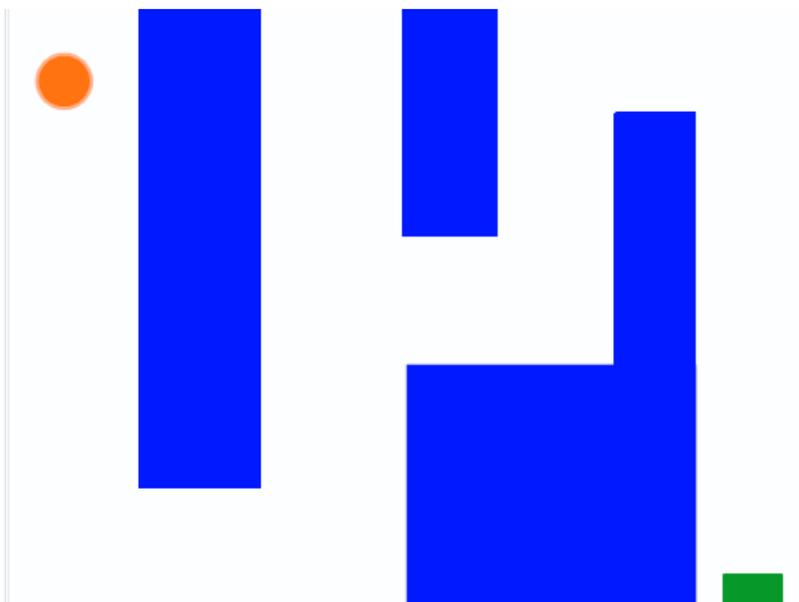
Conceito computacional
Condicional simples–'if'

Objetivo:

- Condicional simples
- Se tocar na cor azul trocar o palco

Jogo de Labirinto:

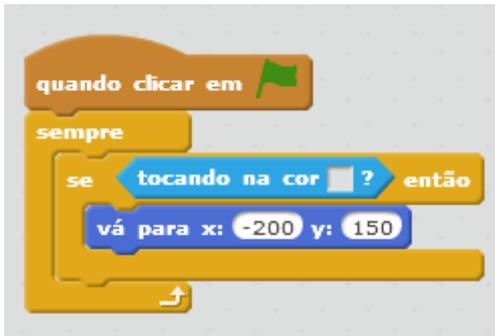
1. Vamos desenhar um palco para nosso labirinto e colocar 2 sprites como na figura abaixo.



2. Adicione os comandos para movimentar a bolinha com o teclado
3. Utilize o script para posicionar a bolinha no topo:

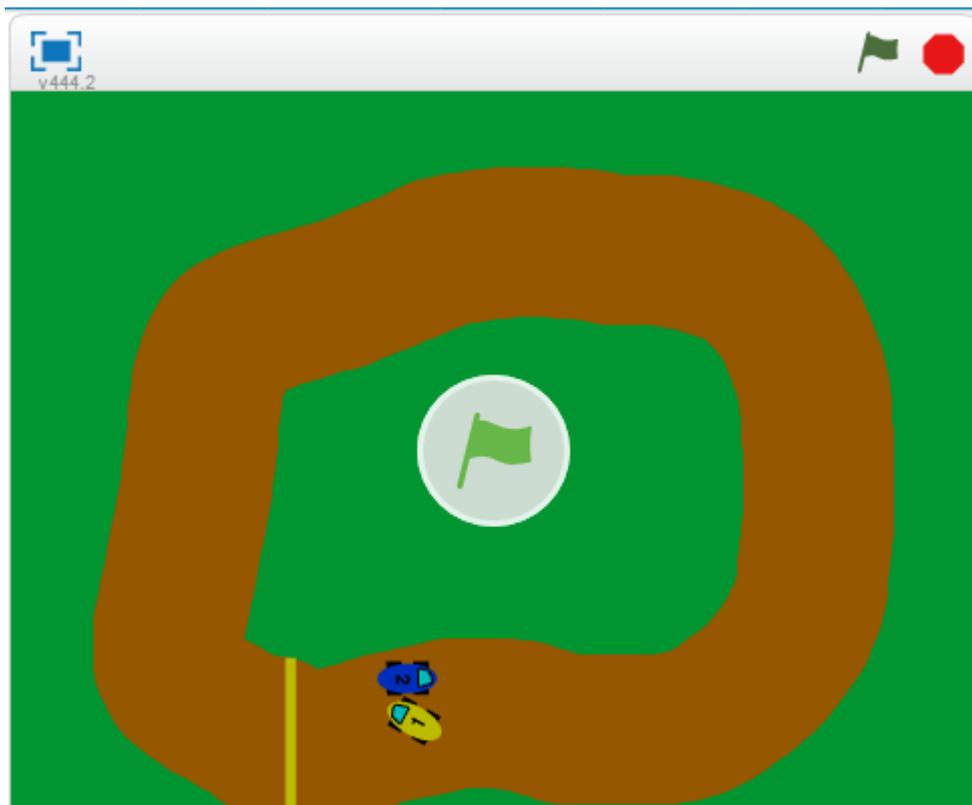


1. Agora vamos utilizar um script para detectar colisão:



DESAFIO:

1. Crie um script para detectar se a bolinha tocou na cor verde. Caso toque na cor verde troque o palco apresentando uma mensagem de vitória.
2. Tente fazer uma pista de corrida e movimentar um carrinho.



04 Variáveis e condicionais

Subsunçor: Conceito de variável

Experiência: Placar eletrônico de um jogo de futebol

Conceito computacional

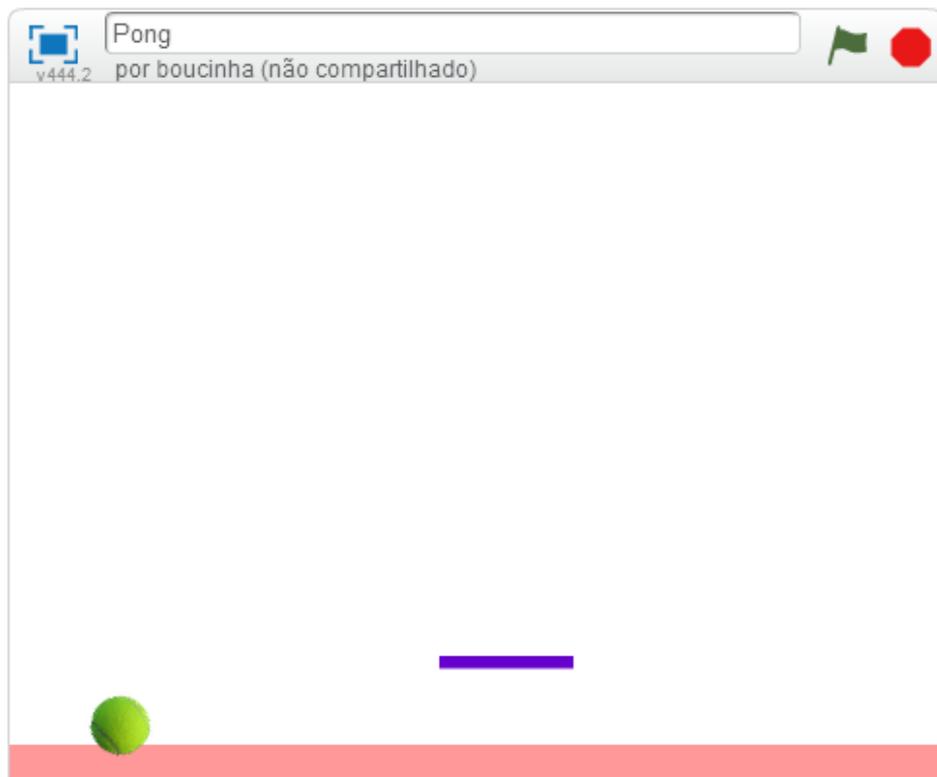
Condicional simples repetir até:

Objetivo:

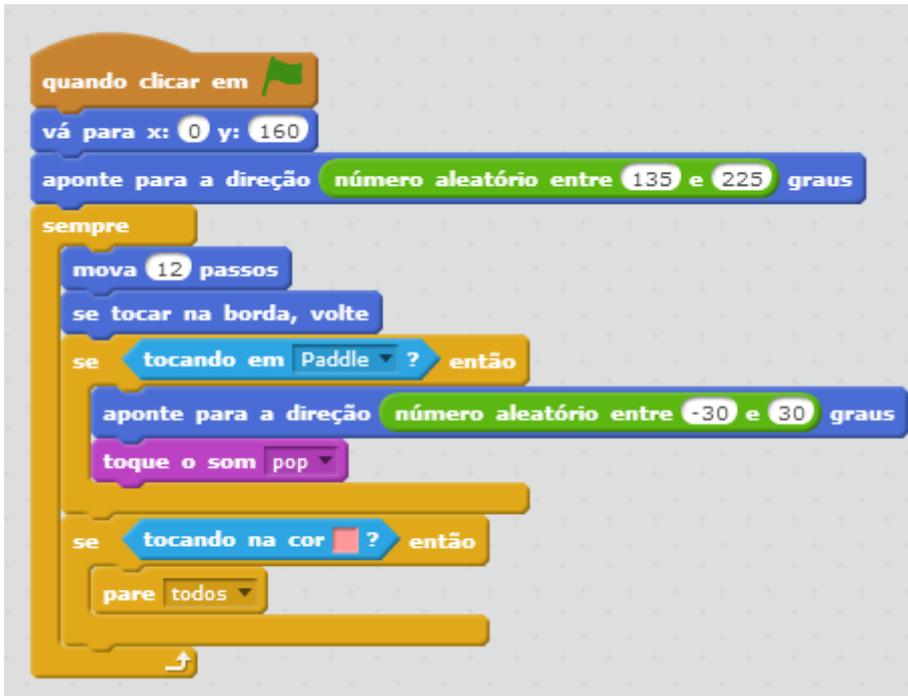
- Utilizar variável para controlar placar
- Condicional repetir até

JOGO PONG

Vamos criar a estrutura deste jogo:



Este é um exemplo de código para movimentar a bolinha:



Tente substituir o bloco sempre por:



O que acontece? O que precisa modificar?

VARIÁVEIS

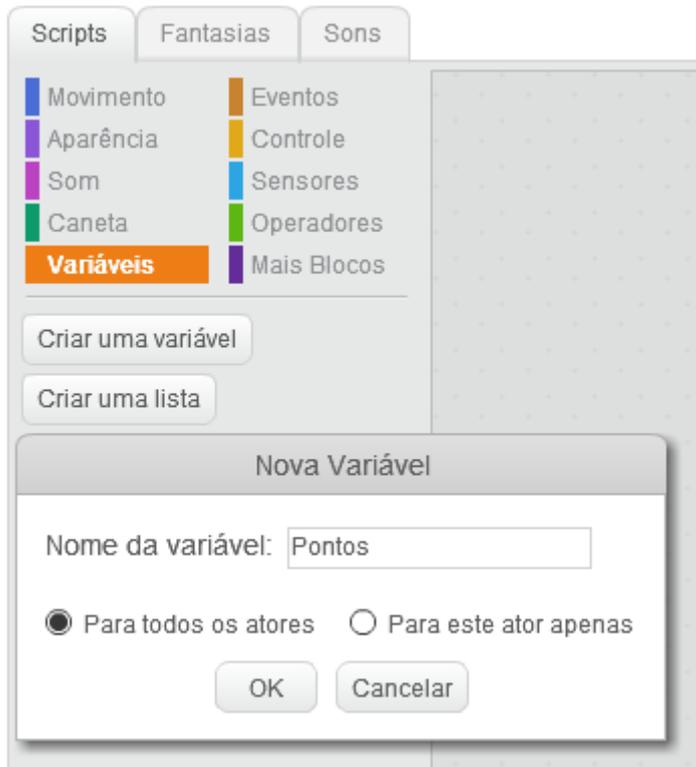


Variável é um número que não conhecemos. Pense no resultado de um jogo de futebol, antes de iniciar a partida não sabemos o número de gols de cada time, no entanto o placar já tem um lugar reservado para armazenar o resultado.

Neste caso temos quantas variáveis?

Como é alterado o valor de cada variável?

Vamos criar uma variável para armazenar os pontos do jogo Pong



Nesta primeira versão ganhará um ponto cada vez que tocar na raquete,
Desta forma, cada vez que bater na raquete, temos que adicionar 1 ponto no placar.



DESAFIO

1. Modifique o palco para exibir uma mensagem de vitória quando atingir 10 pontos.
2. Crie um novo palco para apresentar mensagem de Fim de jogo quando a bolinha cair.

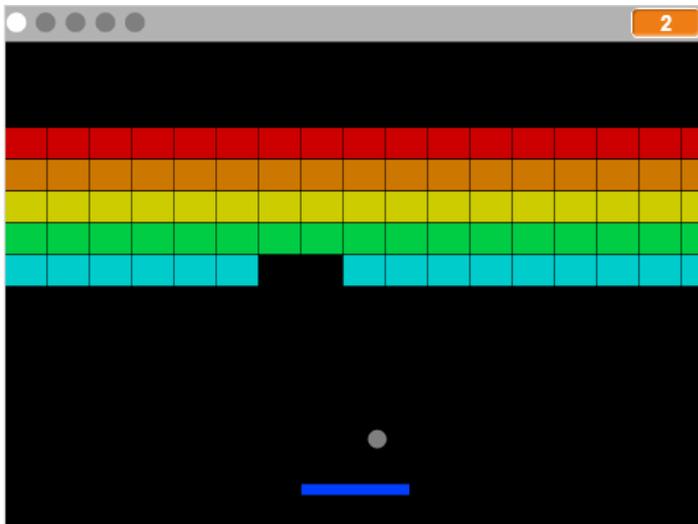
05 Condicional composto

Subsunçor: Conceito de variável

Objetivo:

- Utilizar variável para controlar placar
- Utilizar variável para controlar número de bolas

JOGO BREAKOUT

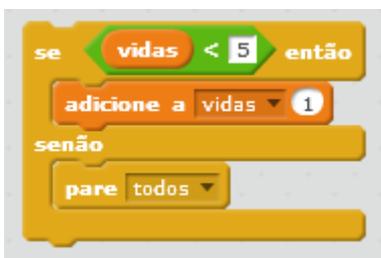


Vamos criar o jogo *Breakout* a partir da estrutura que aprendemos do *Pong*.

Você precisa criar um Sprite para cada quadrado, lembre quando a bolinha bater no quadrado ele tem que sumir.

Utilize a opção duplicar para facilitar o trabalho.

Implemente uma variável para contar quantas bolinhas caíram, quando perder 5 vidas, acaba o jogo.



DESAFIO:

1. Adicione pontos para cada quadrado conforme a tabela

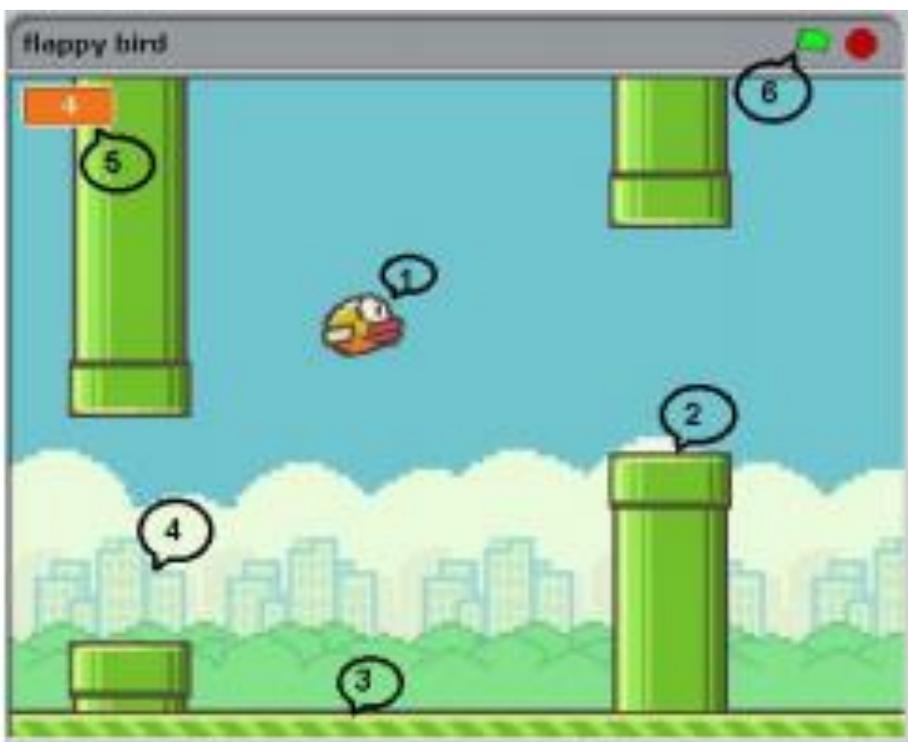
Cor	Azul	Verde	Amarelo	Laranja	Vermelho
Pontos	1	2	3	4	5

06 Depuração

Subsunçor: Colisão

Objetivo:
– Analisar o código

Abra o arquivo: <https://scratch.mit.edu/projects/101551499/>



Observe como funciona.

Como é realizado o movimento de cada objeto?

DESAFIO:

1. Faça o jogo parar caso o pássaro bata no cano.

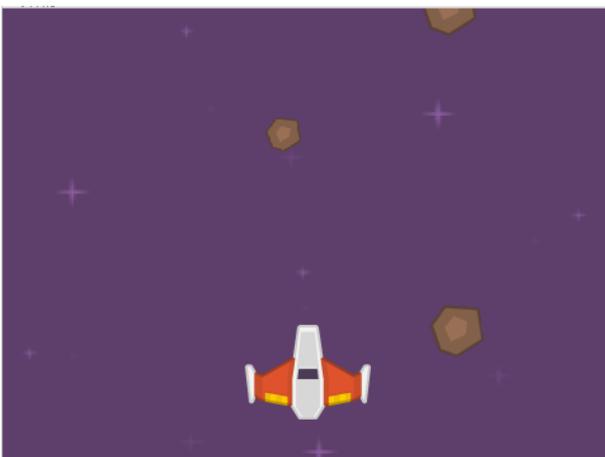
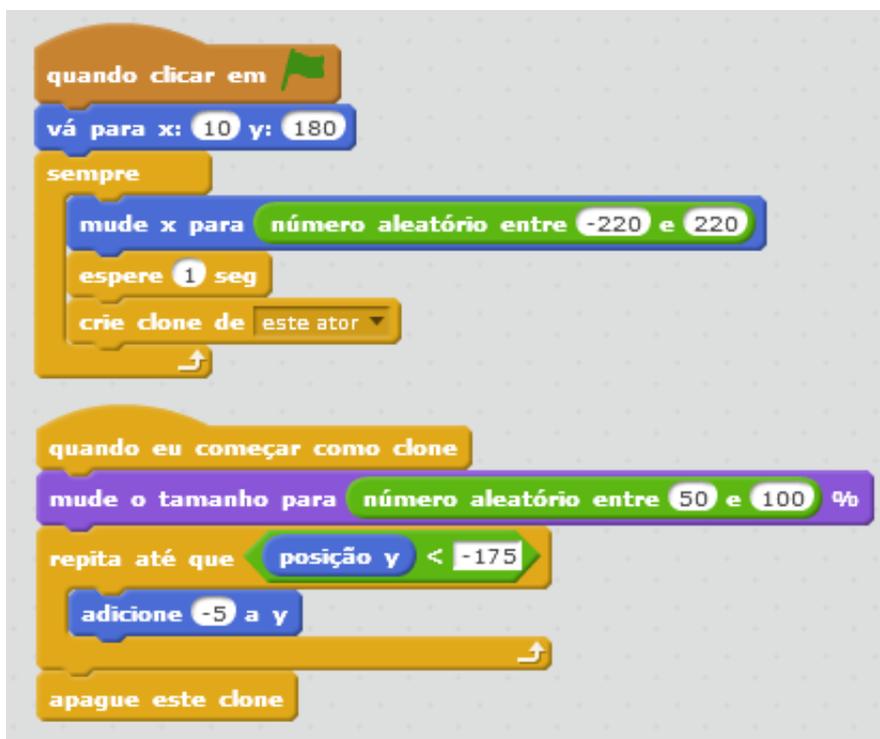
07 Clones

Subsunçor: Colisão

Objetivo:

- Uso de clones
- Alterar tamanho

Abra o projeto: <https://scratch.mit.edu/projects/101551950/>



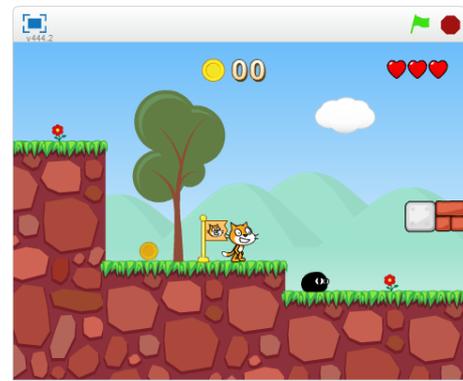
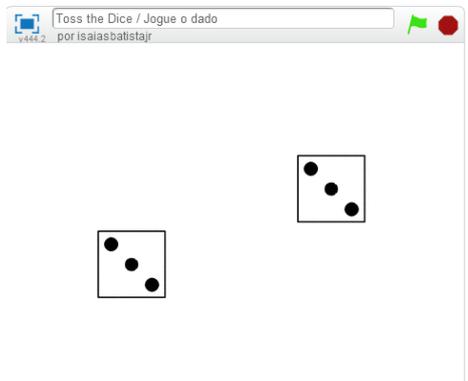
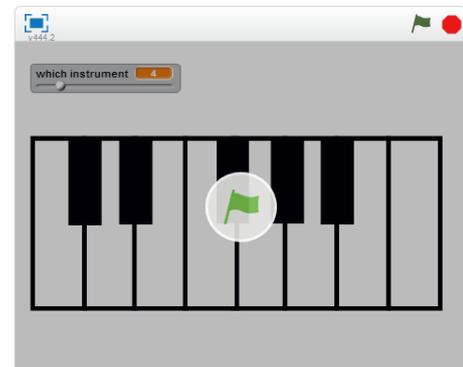
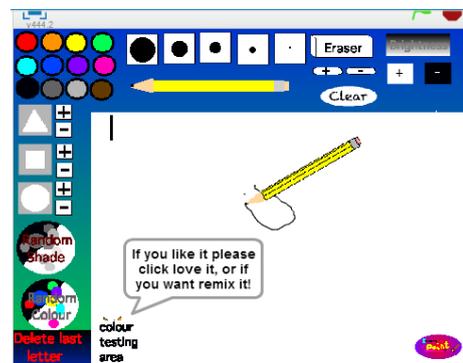
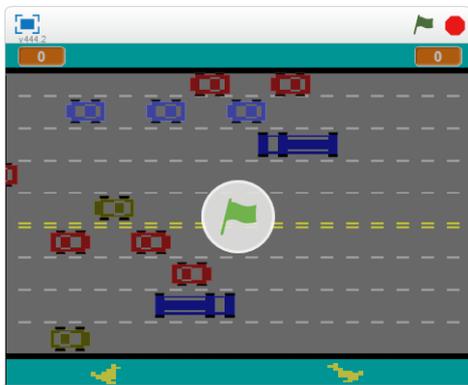
DESAFIO:

1. Implemente um painel para marcar o número de vidas, utilizando 3 corações. Cada vez que uma pedra colidir, apague um coração.
2. Tente implementar uma arma para nave.

09 Exemplos de Games

Objetivo:

- Explorar projetos no Scratch
- Utilizar figuras



APÊNDICE B – Teste de Pensamento Computacional

TESTE DE PENSAMENTO COMPUTACIONAL

INSTRUÇÕES

O teste é composto por 28 perguntas, distribuídas em 12 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.

Qual seqüência leva 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>

Exemplo I A B C 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.

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

	<p>Alternativa A</p> <pre> avance vire à esquerda avance avance </pre>	<p>Alternativa B</p> <pre> avance vire à direita avance avance </pre>
	<p>Alternativa C</p> <pre> avance avance vire à esquerda avance </pre>	<p>Alternativa D</p> <pre> avance avance vire à direita avance </pre>

Exemplo II A B C 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 alto ao artista sem desenhar. 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.

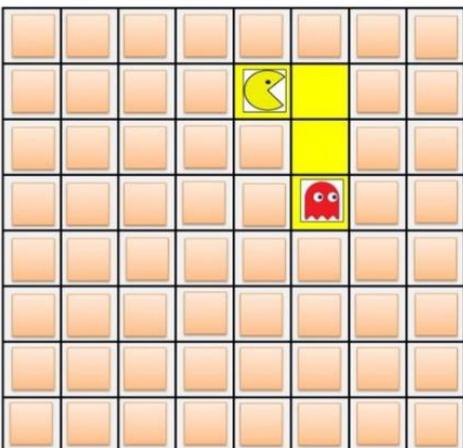
Qual sequência o artista deve seguir para desenhar a figura abaixo? O lado menor mede 50 pixels e o maior mede 100 pixels.

	<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>

Exemplo III A B C D

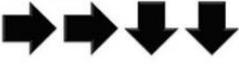
QUESTÃO 1

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



Alternativa A

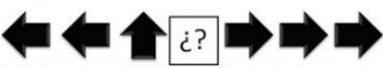
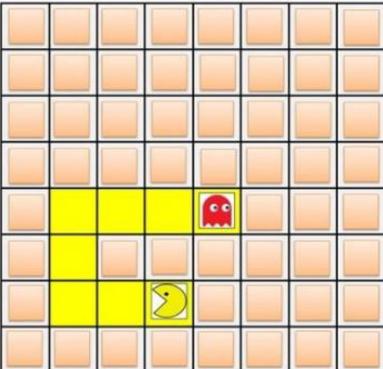

Alternativa B


Alternativa C


Alternativa D


QUESTÃO 2

Qual comando está faltando na sequência para levar o "Pac-Man" até o fantasma pelo caminho indicado?

Alternativa A

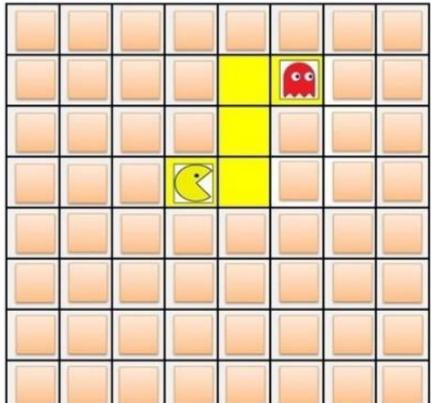

Alternativa B


Alternativa C


Alternativa D

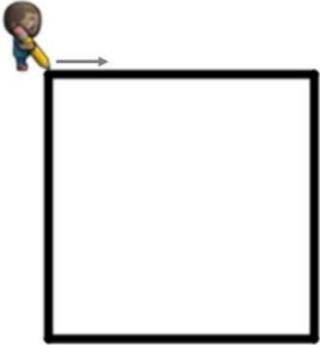

QUESTÃO 3

Para levar o "Pac-Man" até o fantasma pelo caminho indicado, qual passo da sequência está incorreto?

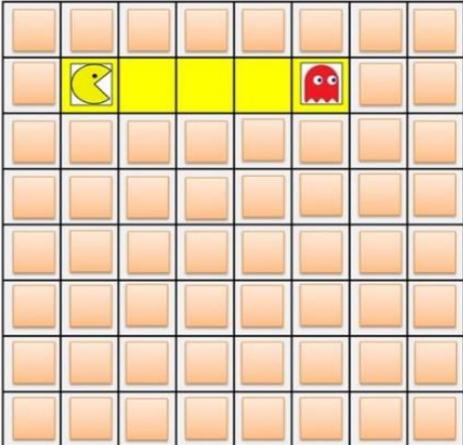




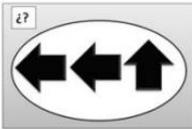
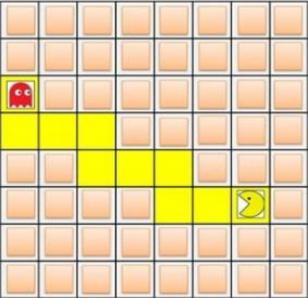
QUESTÃO 4

<p>Qual sequência o artista deve seguir para desenhar o quadrado abaixo? Cada um dos lados mede 100 pixels.</p> 	<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 25 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 	<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

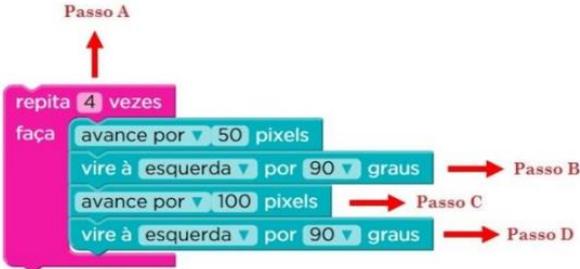
<p>Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?</p> 	<p>Alternativa A</p> 	<p>Alternativa B</p> 
	<p>Alternativa C</p> 	<p>Alternativa D</p> 

QUESTÃO 6

<p>Quantas vezes a sequência abaixo deve ser repetida para levar o "Pac-Man" até o fantasma pelo caminho indicado?</p>  	<p>Alternativa A</p> <p>× 2</p>
	<p>Alternativa B</p> <p>× 1</p>
	<p>Alternativa C</p> <p>× 4</p>
	<p>Alternativa D</p> <p>× 3</p>

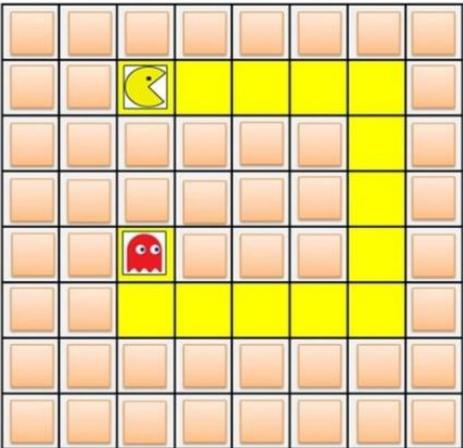
QUESTÃO 7

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*?

QUESTÃO 8

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



Alternativa A



Alternativa B



Alternativa C

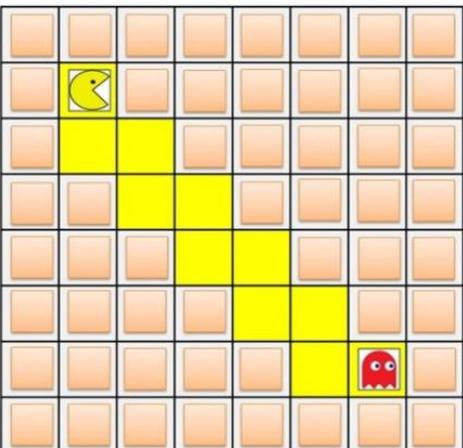


Alternativa D

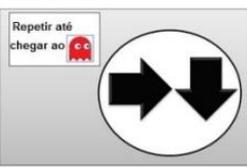


QUESTÃO 9

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



Alternativa A



Alternativa B



Alternativa C

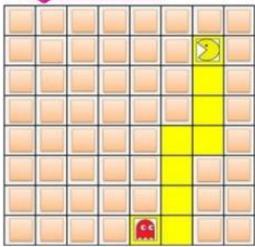


Alternativa D



QUESTÃO 10

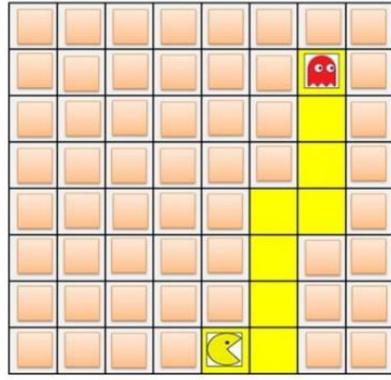
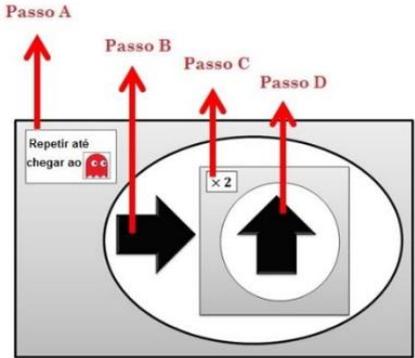
Qual bloco está faltando na sequência abaixo para levar o "Pac-Man" até o fantasma pelo caminho indicado?

<p>Alternativa A</p> 	<p>Alternativa B</p> 
<p>Alternativa C</p> 	<p>Alternativa D</p> <p>Não falta nenhum bloco</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> 	<p>Alternativa B</p> 
<p>Alternativa C</p> 	<p>Alternativa D</p> 

QUESTÃO 13

Qual sequência leva 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>

QUESTÃO 14

Qual sequência leva 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>

QUESTÃO 15

O que falta na seguinte sequência 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>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*?

QUESTÃO 17

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

Alternativa A

Alternativa B

Alternativa C

Alternativa D

QUESTÃO 18

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

Alternativa A

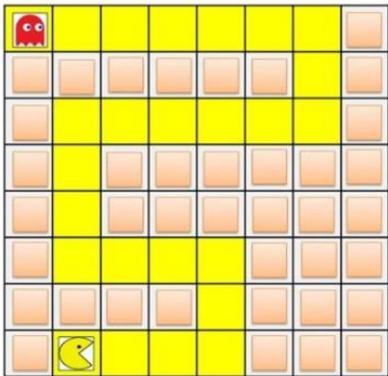
Alternativa B

Alternativa C

Alternativa D

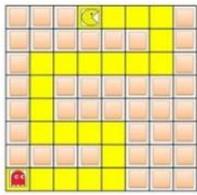
QUESTÃO 19

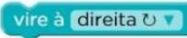
Para que o "Pac-Man" chegue até o fantasma pelo caminho indicado, qual passo da sequência está incorreto?




QUESTÃO 20

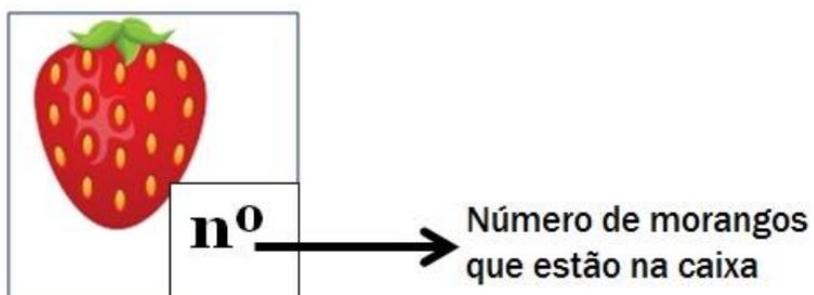
Qual bloco está faltando na sequência abaixo para levar o "Pac-Man" até o fantasma pelo caminho indicado?

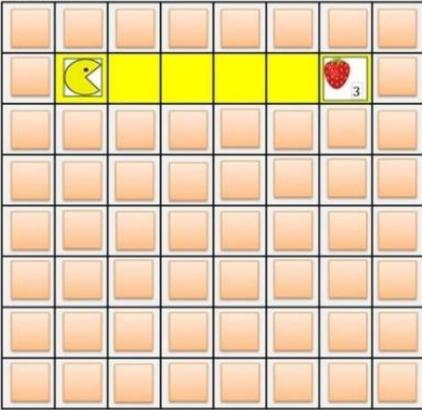
Alternativa A 	Alternativa B 
Alternativa C 	Alternativa D Não falta nenhum bloco

IMPORTANTE: LEIA COM ATENÇÃO

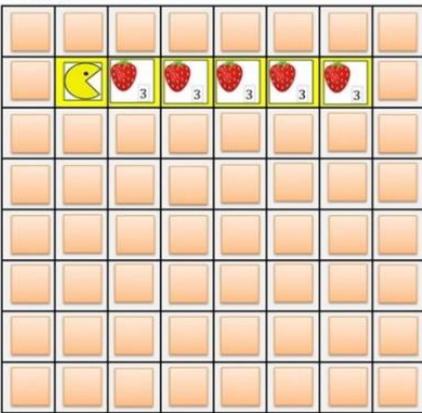
A imagem abaixo irá aparecer nas próximas questões. O número que está no canto inferior direito indica quantos morangos existem na caixa.



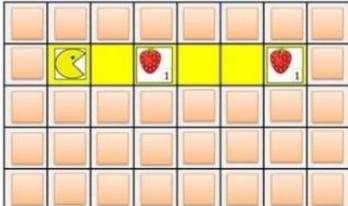
QUESTÃO 21

<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 faça 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 faça 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

<p>O que falta na seguinte sequência para que "Pac-Man" avance pelo caminho assinalado comendo o número de morangos indicados?</p> <pre> enquanto houver caminho em frente faça repita ??? vezes faça avance se existe algum morango faça coma 1 morango </pre> 	<p>Alternativa A</p> <p>1 vez</p> <hr/> <p>Alternativa B</p> <p>2 vezes</p> <hr/> <p>Alternativa C</p> <p>3 vezes</p> <hr/> <p>Alternativa D</p> <p>5 vezes</p>
---	---

QUESTÃO 27

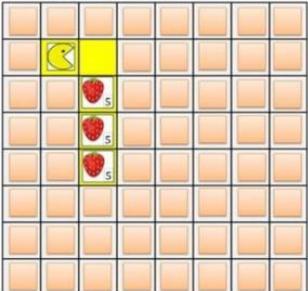
Se temos o seguinte conjunto de instruções, que chamamos de "get 5":

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

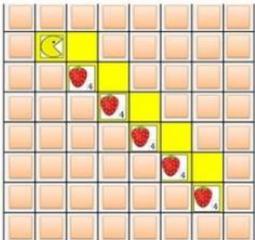
```

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?

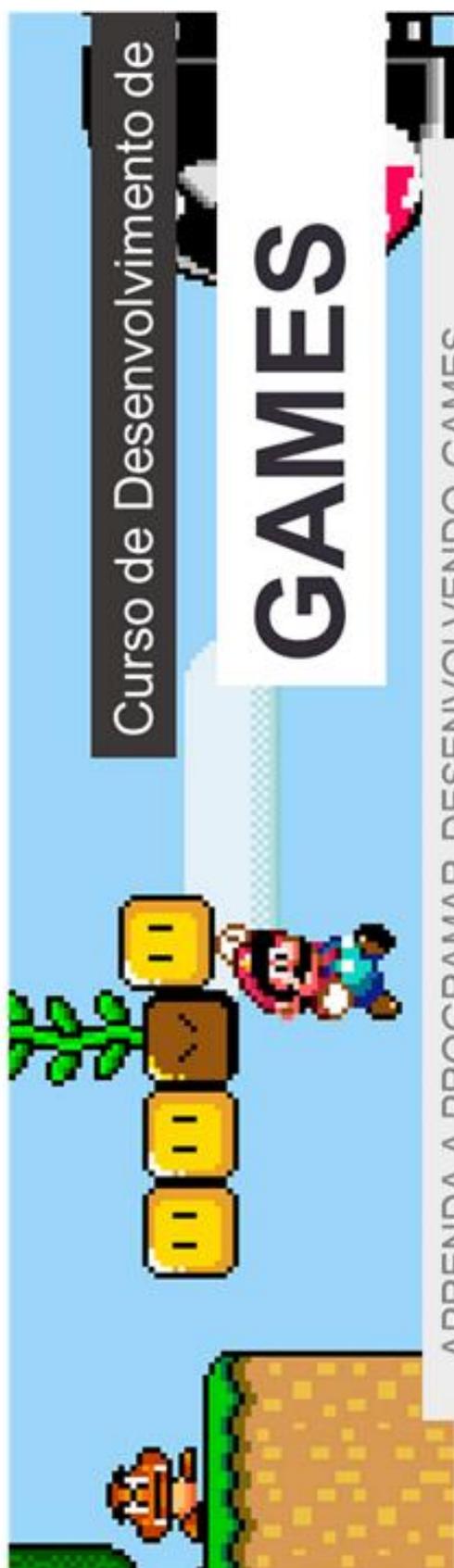
```

repetir ??? vezes
  faça
    move and get 4
        
```



Alternativa A	Alternativa B
3	4
Alternativa C	Alternativa D
5	6

Teste baseado na pesquisa de Marcos Román-Gonzalez, Juan Carlos Pérez-González, Carmen Jiménez-Fernández intitulada "Test de Pensamiento Computacional: diseño y psicométrica general".
Tradução: Rafael M. Boucinha



APRENDA A PROGRAMAR DESENVOLVENDO GAMES

Este curso é parte de um projeto de pesquisa que tem como objetivo: **Investigar a relação entre aprendizagem do Pensamento Computacional e desenvolvimento do raciocínio.** Com a metodologia proposta espera-se obter um **Incremento em habilidades relacionadas ao raciocínio numérico, raciocínio espacial e raciocínio abstrato.** Os voluntários da pesquisa serão avaliados, com testes psicométricos, antes e depois da realização do curso.



Público

Estudantes do 6 ao 9 ano
do Ensino Fundamental



Carga Horária

10 horas aula



Custo

Fornecido gratuitamente

Informações:
51 91494040 ou rafael@ieduca.com.br



ieduca

APÊNDICE D – Parece Câmara de Pesquisa

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
CONSELHO DE ENSINO, PESQUISA E EXTENSÃO
CÂMARA DE PESQUISA

Processo eletrônico

Assunto: Projeto de pesquisa 31947

Requerente: Rafael Marimon Boucinha

Origem: Centro de Estudos Interdisciplinares em Novas Tecnologias da Educação.

PARECER N° 66/2016

PROJETO N° 31947

TÍTULO DO PROJETO: Avaliação do processo ensino-aprendizagem do pensamento computacional e desenvolvimento do raciocínio Professor Coordenador: Prof. Dr. Dante Augusto Couto Barone

Doutorando CINTED: Rafael Marimon Boucinha

Instituição de origem: Universidade Federal do Rio Grande do Sul, Centro de Estudos Interdisciplinares em Novas Tecnologias da Educação (CINTED), Programa de Pós-Graduação em Informática na Educação

ASPECTOS CIENTÍFICOS:

Título: O título (“Avaliação do processo ensino-aprendizagem do pensamento computacional e desenvolvimento do raciocínio”) é coerente com a proposição do estudo. Trata-se de projeto que dá base à tese de doutorado em informática na educação, aprovada em banca de qualificação.

Introdução, revisão teórica e justificativa: O projeto inclui a revisão de referências sobre o pensamento computacional na formação escolar e na vida cotidiana. Pretende “contribuir para uma maior reflexão sobre a inclusão do Pensamento Computacional (PC) como uma habilidade básica, assim como ler e escrever, desde a Educação Básica”. O projeto contextualiza a temática em estudo, revisando o conceito de Pensamento Computacional, teorias sobre inteligência e um levantamento de estudos correlatos, e justifica abordagem da pesquisa.

Objetivos: Objetivo geral (“Investigar a relação entre aprendizagem do Pensamento Computacional e desenvolvimento do raciocínio”) é coerente e compatível com os objetivos específicos (“Desenvolver metodologia para ensino de Pensamento Computacional; Avaliar aprendizagem de Pensamento Computacional; Verificar o efeito da aprendizagem do Pensamento Computacional na capacidade de raciocínio dos alunos; Analisar a relação entre pensamento computacional e os diferentes tipos de raciocínio: raciocínio verbal, raciocínio numérico, raciocínio espacial, raciocínio abstrato e raciocínio mecânico; Verificar se pensamento computacional é uma variável independente da capacidade de raciocínio”) e com o delineamento do projeto.

Aspectos Metodológicos: Apresenta-se como um estudo de natureza quantitativa, no qual será utilizada uma amostra de 100 alunos, que serão avaliados com o Teste de Pensamento Computacional e a Bateria de Provas de Raciocínio-BPR-5, descritos na literatura. Os resultados destes testes serão analisados com a utilização de procedimentos estatísticos, com os quais pretende-se evidenciar a relação entre Pensamento Computacional e raciocínio. Os sujeitos da pesquisa serão aproximadamente 100 estudantes de três escolas de educação básica que aderirem ao projeto, avaliados antes e depois de oficinas sobre a aprendizagem do pensamento computacional. Como impactos esperados do projeto, estão previstas: Conscientização da importância e maior divulgação do Pensamento Computacional no meio acadêmico e escolas de âmbito municipal, estadual, federal e particular; Maior interesse de



APÊNDICE E – Modelo de Certificado

ieduc@

CERTIFICADO

Certificamos que

participou do curso de DESENVOLVIMENTO DE GAMES COM SCRATCH, com carga horária de 10 horas aula.

Porto Alegre, 11 de maio de 2017.

Aluno



Prof. Rafael Boucinha

APÊNDICE F – Termo de Consentimento

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 “APRENDIZAGEM DO PENSAMENTO COMPUTACIONAL E DESENVOLVIMENTO DO RACIOCÍNIO”, que tem como objetivo analisar a relação entre Pensamento Computacional e os diferentes tipos de raciocínio em alunos do 6º ao 9º ano do Ensino Fundamental

O Pensamento Computacional tem sido considerado como uma habilidade essencial para qualquer pessoa, independentemente da área de atuação, por este motivo foi incluído como disciplina obrigatória em vários países desenvolvidos.

Para abordarmos este tema, propomos um curso de Desenvolvimento de Games abordando os principais conceitos utilizados na programação de computadores, oferecido como atividade extracurricular para os alunos das escolas participantes.

Para medir os efeitos deste curso, os voluntários desta pesquisa serão submetidos a um teste de Pensamento Computacional, avaliação do raciocínio e capacidade de atenção. Estas capacidades serão medidas antes e depois da realização do curso.

Os dados desta pesquisa serão utilizados como subsídios para tese de Doutorado de Rafael Marimon Boucinha e produção de artigos científicos, com objetivo de preservar o sigilo dos dados o nome dos participantes não será divulgado em qualquer divulgação científica.

Como a atividade proposta é um curso de programação, a pesquisa não envolve esforço físico que possa levar a lesões, nem quaisquer conteúdos inapropriados para a idade do público-alvo. Para evitar comparação do desempenho entre alunos, os dados individuais de cada aluno não serão divulgados em sala de aula.

Ao término da pesquisa os participantes receberão um resumo dos resultados obtidos com a pesquisa. A partir deste momento, os pais poderão agendar entrevista para solicitar resultados individuais de cada aluno.

A participação desta pesquisa é voluntária e o participante tem total liberdade de recusar tomar parte na Pesquisa em qualquer uma de suas fases, sem que isto lhe acarrete qualquer prejuízo ou constrangimento.

No que diz respeito aos benefícios da pesquisa para o aluno são, porém, não limitados a esses:

- Entender um modelo estruturado de pensamento, auxiliando no processo de aprender a aprender;

- Melhorar o processo de formulação do problema;
- Aperfeiçoar a capacidade de abstração e criatividade;
- Dar ênfase na criação do conhecimento e não apenas no seu uso.

Com o benefício em longo prazo, destacam-se:

- Aprender a lidar com problemas variados e complexos, de forma crítica;
- Seguir tendências mundiais de ensino (competências do século XXI);
- Atender uma forte demanda de mercado;
- Aumentar a compreensão da fusão da computação com outras áreas, não apenas dentro da sua área de atuação.

Os procedimentos da Pesquisa obedecem aos Critérios de Ética em Pesquisa com Seres Humanos conforme Resolução nº 466/12 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 durante a Pesquisa serão utilizados apenas para fins de investigação e de produção de conhecimento. Todo o material obtido será arquivado por um período de 05 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 Rafael Marimon Boucinha, pelo telefone (51) 9149-4040.

Data: ____/____/____

EU _____, concordo com a participação do meu filho/a _____ na Pesquisa APRENDIZAGEM DO PENSAMENTO COMPUTACIONAL E DESENVOLVIMENTO DO RACIOCÍNIO.

Observação: 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 a participação no Curso de Desenvolvimento de Games. O nome de seu filho/a não será veiculado em qualquer material gerado a partir desta pesquisa.

Nome do responsável pela criança: _____

Assinatura do responsável pela criança: _____

Assentimento

EU, _____, concordo em participar desta pesquisa.

Assinatura da criança: _____

APÊNDICE G – Questionário

QUESTIONÁRIO

1. Nome:
2. Sexo:
3. Idade
4. Série/ano:
5. Costuma jogar videogame (ou no computador)?
 Sim. Quantas horas por semana? ____
 Não.
6. Possui computador em casa?
 Não.
 Sim, compartilhado com pais ou irmãos
 Sim, de uso exclusivo;.
7. Já definiu a profissão que deseja seguir?
 Não.
 Sim. Qual?
8. Tem experiência com alguma linguagem de programação?
 Não.
 Sim. Qual linguagem?
9. Já utilizou o software Scratch destinado a programação visual?
 Não.
 Sim.