

UNIVERSIDADE FEDERAL DO PARANÁ

SIDNÉIA VALERO EGIDO

EDUCAÇÃO MATEMÁTICA E DESENVOLVIMENTO DO PENSAMENTO
COMPUTACIONAL NO 3º ANO DO ENSINO FUNDAMENTAL: CRIANÇAS
PROGRAMANDO JOGOS COM SCRATCH

CURITIBA

2018

SIDNÉIA VALERO EGIDO

EDUCAÇÃO MATEMÁTICA E DESENVOLVIMENTO DO PENSAMENTO
COMPUTACIONAL NO 3º ANO DO ENSINO FUNDAMENTAL: CRIANÇAS
PROGRAMANDO JOGOS COM SCRATCH

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Educação em Ciências e em Matemática, no Curso de Pós-Graduação em Educação em Ciências e em Matemática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Orientadora: Profa. Dra. Luciane Mulazani dos Santos

CURITIBA

2018

FICHA CATALOGRÁFICA ELABORADA PELO SISTEMA DE BIBLIOTECAS/UFPR
BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

EG29e

Egido, Sidnéia Valero

Educação matemática e desenvolvimento do pensamento computacional no 3º ano do ensino fundamental: crianças programando jogos com scratch / Sidnéia Valero Egido. – Curitiba, 2018. 138 p. : il. color. ; 30 cm.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Educação em Ciências e em Matemática, 2018.

Orientadora: Luciane Mulazani dos Santos.

1. Educação matemática. 2. Scratch. 3. Pensamento computacional. 4. Ensino de programação.
I. Universidade Federal do Paraná. II. Santos, Luciane Mulazani dos. III. Título.

CDD: 372.3

Bibliotecária: Romilda Santos - CRB-9/1214

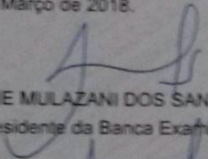


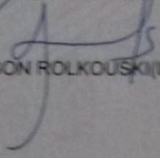
MINISTÉRIO DA EDUCAÇÃO
SETOR CIÊNCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EDUCAÇÃO EM
CIÊNCIAS E EM MATEMÁTICA

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em EDUCAÇÃO EM CIÊNCIAS E EM MATEMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **SIDNÉIA VALERO EGIDO**, intitulada: **EDUCAÇÃO MATEMÁTICA E DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL NO 3º ANO DO ENSINO FUNDAMENTAL: CRIANÇAS PROGRAMANDO JOGOS COM SCRATCH**, após terem inquirido a aluna e realizado a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa. A outorga do título de Mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 23 de Março de 2018.


LUCIANE MULAZANI DOS SANTOS(UFPR)
(Presidente da Banca Examinadora)


EMERSON ROLKOÚSKI(UFPR)


IVANETE ZUCHI SIPLE(UDESC)



Dedico essa dissertação a Deus, aos meus pais (Aparecida Valero e João Egido), que sempre me incentivaram e acreditaram em meu potencial, ao meu amado marido, que foi companheiro durante todo o processo, abrindo mão de suas vontades e para me acompanhar durante cada dia, aos meus irmãos pelas palavras de incentivo e apoio, a minha irmã de mestrado e coração pelo ombro amigo, aos meus professores e colegas, pessoas inspiradoras conversas e orientações e a meus alunos, razão de minha dedicação, busca pelo conhecimento e sempre aprimoramento, gratidão por todo apoio e carinho.

AGRADECIMENTOS

À minha orientadora, Profa. Dra. Luciane Mulazani dos Santos, pelo acompanhamento, orientação e amizade.

Ao Programa de Pós-Graduação em Educação em Ciências e em Matemática (PPGECM), do Setor de Ciências Exatas, da Universidade Federal do Paraná, na pessoa de seu coordenador, Prof. Dr. Sérgio Camargo, pelo apoio recebido.

Ao Colegiado do Curso de Pós-Graduação em Educação em Ciências e em Matemática, por todo apoio e compreensão aos momentos difíceis. Aos Professores Dr. Carlos Roberto Vianna, Dr. Emerson Rolkouski, Dra. Luciane Ferreira Mocrosky, Dr. Marco Aurélio Kalinke, Dr. Marcos Aurélio Zanlorenzi, pela maestria com que conduziram as disciplinas ofertadas no curso e por suas contribuições e sugestões no trabalho. À querida Antonyhella Santini, Secretária do PPGECM, por toda atenção e paciência.

Aos meus colegas de curso da turma de 2016, por todo companheirismo e amizade.

Aos meus colegas membros do Grupo de Pesquisas sobre Tecnologias na Educação Matemática (GPTEM), por estarem sempre unidos na causa do desenvolvimento de conhecimento no campo do uso das Tecnologias na Educação Matemática.

Aprender é uma grandeza que não ocupa espaço.

Autor desconhecido.

RESUMO

Este trabalho apresenta uma pesquisa qualitativa realizada no âmbito da Educação Matemática, no campo da tecnologia educacional. Seu objetivo foi promover e investigar situações de ensino de programação para estudantes dos anos iniciais Ensino Fundamental, observando o desenvolvimento do seu pensamento computacional. Para tal, foram realizadas aulas em contraturno, no formato de minicurso, utilizando o software Scratch com alunos do terceiro ano do Ensino Fundamental em uma escola da Região Metropolitana de Curitiba. Como resultados, observou-se que os alunos se mostraram satisfeitos com as atividades realizadas e que foram identificadas situações de aprendizagem nas quais houve desenvolvimento do pensamento computacional das crianças, o que evidencia a potencialidade da inserção do ensino de programação nos anos iniciais do Ensino Fundamental.

Palavras-chave: Educação Matemática. Scratch. Pensamento Computacional. Ensino de programação.

ABSTRACT

This work presents a qualitative research carried out in the Mathematics Education field, in the scope of educational technology. Its purpose was to promote and investigate instances of programming teaching for young learners from the Basic Education, observing students' computational thinking. In order to do so, classes were carried out in the model of workshop, using the software Scratch with students of third year of Fundamental Education in a school located in Curitiba metropolitan area. As for the findings, it was possible to observe students' satisfaction with the activities done and it was equally possible to identify instances of learning in which students developed computational thinking. It illustrates the affordance of implementing programming teaching in the first years of the Basic Education.

Key-words: Mathematics Education. Scratch. Computational Thinking. Programming Teaching.

LISTA DE FIGURAS

FIGURA 1: ETAPAS DA RESOLUÇÃO DE PROPLEMAS	27
FIGURA 2: TELA INICIAL DO SCRATCH 1.4	38
FIGURA 3: TELA INICIAL DO SCRATCH 2.0	38
FIGURA 4: TELA INICIAL DO SCRATCH 1.4	39
FIGURA 5: CATEGORIA MOVIMENTO E COMANDO GIRAR	41
FIGURA 6: BLOCO EMPILHADO COM CHAPÉU	43
FIGURA 7: BLOCO EMPILHADO COM CHAPÉU E REPÓRTERES/VALORES	44
FIGURA 8: BLOCOS REPÓRTER NO PALCO	44
FIGURA 9: LABORATÓRIO DE INFORMÁTICA	57
FIGURA 10: EXEMPLO 1 APRESENTADO AOS ALUNOS	63
FIGURA 11: EXEMPLO 2 APRESENTADO AOS ALUNOS	63
FIGURA 12: EXEMPLO 3 APRESENTADO AOS ALUNOS	64
FIGURA 13: TELA INICIAL DO SITE DO SCRATCH DISPONÍVEL NA INTERNET	64
FIGURA 14: SITE DO COLÉGIO	66
FIGURA 15: AULA 1 - APRESENTAÇÃO.....	67
FIGURA 16: AULA 1 – ÁREA DO SCRATCH	68
FIGURA 17: AULA 1 – EXPLORANDO O SCRATCH	68
FIGURA 18: AULA 1 – EFEITOS	69
FIGURA 19: SPRITE E PROPRIEDADES	70
FIGURA 20: COMANDOS E GRUPOS	71
FIGURA 21: CONCEITOS GERAIS	71
FIGURA 22: COMANDOS INICIAIS SCRATCH APRESENTADOS AOS ALUNOS ..	72
FIGURA 23: ATIVIDADE 1	73
FIGURA 24: ATIVIDADE DESENVOLVIDA POR UM ALUNO - PROGRAMAÇÃO. 73	
FIGURA 25: ATIVIDADE DESENVOLVIDA POR UM ALUNO - CENÁRIO	74
FIGURA 26: RESUMO DAS DIMENSÕES DO PENSAMENTO COMPUTACIONAL PRESENTES NA ATIVIDADE ANTERIOR DE ACORDO COM BRENNAN & RESNICK (2012)	76
FIGURA 27: CENÁRIO E PERSONAGEM – DJ MALUCO.....	78
FIGURA 28: PROGRAMÇÃO – DJ MALUCO	78
FIGURA 29: PROGRAMAÇÃO – CENÁRIO DJ MALUCO	78
FIGURA 30: DANÇANDO NA AREIA – ATIVIDADE DE UM ALUNO	79

FIGURA 31: RESUMO DAS DIMENSÕES DO PENSAMENTO COMPUTACIONAL PRESENTES NA ATIVIDADE ANTERIOR	81
FIGURA 32: CENÁRIO E PERSONAGENS – 1	82
FIGURA 33: PROGRAMAÇÃO 1	82
FIGURA 34: PROGRAMAÇÃO 2	83
FIGURA 35: FIGURAS PLANAS – ATIVIDADE 2	83
FIGURA 36: PROGRAMAÇÃO	84
FIGURA 37: RESUMO DAS DIMENSÕES DO PENSAMENTO COMPUTACIONAL PRESENTES NA ATIVIDADE ANTERIOR	87
FIGURA 38: CENÁRIO E PERSONAGEM DA AULA	87
FIGURA 39: COMANDOS DA AULA	88
FIGURA 40: RESUMO DAS DIMENSÕES DO PENSAMENTO COMPUTACIONAL PRESENTES NA ATIVIDADE ANTERIOR	89
FIGURA 41: EXEMPLO UTILIZADO NA AULA	90
FIGURA 42: RESUMO DAS DIMENSÕES DO PENSAMENTO COMPUTACIONAL PRESENTES NA ATIVIDADE ANTERIOR	91
FIGURA 43: SOMA DE DOIS NÚMEROS	92
FIGURA 44: SOMA DE DOIS NÚMEROS COM VERIFICAÇÃO DE ACERTOS	93
FIGURA 45: SOMA DE DOIS NÚMEROS COM CONTADOR	93
FIGURA 46: RESUMO DAS DIMENSÕES DO PENSAMENTO COMPUTACIONAL PRESENTES NA ATIVIDADE ANTERIOR	95
FIGURA 47: PROJETO – AS OPERAÇÕES MATEMÁTICAS	95
FIGURA 48: ABERTURA	96
FIGURA 49: PERSONAGEM ADIÇÃO	97
FIGURA 50: PERSONAGEM SUBTRAÇÃO.....	97
FIGURA 51: PERSONAGEM MULTIPLICAÇÃO	98
FIGURA 52: RESUMO DAS DIMENSÕES DO PENSAMENTO COMPUTACIONAL PRESENTES NA ATIVIDADE ANTERIOR	99
FIGURA 53: PACMAN – CENÁRIO E PERSONAGENS	100
FIGURA 54: PROGRAMAÇÃO DOS PERSONAGENS - MONSTROS	101
FIGURA 55: PROGRAMAÇÃO DOS PERSONAGENS – PACMAN – PARTE 1 ...	101
FIGURA 56: PROGRAMAÇÃO DOS PERSONAGENS – PACMAN – PARTE 2 ...	102
FIGURA 57: PROGRAMAÇÃO DOS PERSONAGENS – PACMAN – PARTE 3 ...	102
FIGURA 58: RESUMO DAS DIMENSÕES DO PENSAMENTO COMPUTACIONAL PRESENTES NA ATIVIDADE ANTERIOR	104

FIGURA 59: SQUARE DUDE	104
FIGURA 60: COLOR MEMORY	105
FIGURA 61: RESUMO DAS DIMENSÕES DO PENSAMENTO COMPUTACIONAL PRESENTES NA ATIVIDADE ANTERIOR	110
FIGURA 62: ESQUEMA DE DESARTICULAÇÃO DE ATIVIDADE	102
FIGURA 63: PRÁTICAS COMPUTACIONAIS PRESENTES NAS ATIVIDADES ...	116
FIGURA 64: PERSPECTIVAS COMPUTACIONAIS	116
FIGURA 65: RESUMO DAS DIMENSÕES DO PENSAMENTO COMPUTACIONAL	117

LISTA DE GRÁFICOS

GRÁFICO 1: INTERESSES DOS PARTICIPANTES EM PARTICIPAR DO MINICURSO	51
GRÁFICO 2: FREQUÊNCIA COM QUE OS PARTICIPANTES UTILIZAM TECNOLOGIA DURANTE SUAS AULAS	52
GRÁFICO 3: QUANTIDADE DE ALUNOS POR ANO	55
GRÁFICO 4: ATIVIDADES REALIZADAS NO COMPUTADOR EM CASA	61
GRÁFICO 5: ATIVIDADES REALIZADAS NA AULA DE INFORMÁTICA NA ESCOLA	61
GRÁFICO 6: OPINIÃO DOS ALUNOS QUANTO A CRIAR SEUS JOGOS NO COMPUTADOR ENVOLVENDO AS ATIVIDADES E CONTEÚDOS ESCOLARES ..	62
GRÁFICO 7: ATIVIDADE 1 – AULA 2	75
GRÁFICO 8: QUANTIDADE DE ALUNOS QUE REALIZARAM O SOLICITADO – AULA 3	80
GRÁFICO 9: QUANTIDADE DE ALUNOS QUE CONCLUÍRAM A ATIVIDADE 1 – AULA 4	85
GRÁFICO 10: QUANTIDADE DE ALUNOS QUE CONCLUÍRAM A ATIVIDADE 2 – AULA 4	86
GRÁFICO 11: ATIVIDADE MAIOR E MENOR – AULA 5	89
GRÁFICO 12: ATIVIDADE MAIOR, MENOR E IGUAL – AULA 06	91
GRÁFICO 13: DESEMPENHO DOS ALUNOS NA ATIVIDADE	94
GRÁFICO 14: PROJETO OPERAÇÕES	99
GRÁFICO 15: ATIVIDADE PACMAN	103
GRÁFICO 16: EXCELÊNCIA DOS PROJETOS	110
GRÁFICO 17: QUANTIDADE DE ALUNOS QUE DERAM CADA NOTA	111
LISTA DE TABELAS	

TABELA 1: NOTAS E MÉDIA	111
--------------------------------------	-----

TABELA 2: RESUMO DOS CONCEITOS COMPUTACIONAIS PRESENTES NAS DIVERSAS ATIVIDADES	114
--	-----

LISTA DE ABREVIATURAS E SIGLAS

BNCC	- Base Nacional Comum Curricular
CSTA	- Computer Science Teacher Association
GPTEM	- Grupo de Pesquisas sobre Tecnologias na Educação Matemática
ISTE	- International Society for Technology in Education
LLK	- Lifelong Kindergarten
MIT	- Massachusetts Institute of Technology
NCTM	- Normas para o currículo e avaliação em Matemática escolar (Portugal)
PPGECM	- Programa de Pós-Graduação em Educação em Ciências e em Matemática
TIC	- Tecnologias da Informação e Comunicação
TPACK	- Technological Pedagogical Content Knowledge
UFPR	- Universidade Federal do Paraná

TABELAS

TABELA 1: NOTAS E MÉDIA	111
TABELA 2: RESUMO DOS CONCEITOS COMPUTACIONAIS PRESENTES NAS DIVERSAS ATIVIDADES	114

SUMÁRIO

1	SOBRE A PESQUISA: CONTEXTO, OBJETIVOS, MÉTODOS	18
1.1	CONTEXTO	18
1.2	OBJETIVOS	21
1.2.1	Objetivo geral.....	21
1.2.2	Objetivos específicos.....	21
1.3	MÉTODOS.....	22
2	ESTUDOS QUE EMBASARAM A PESQUISA	24
2.1	SOBRE O CONSTRUTIVISMO	24
2.2	SOBRE O CONSTRUCIONISMO	26
2.3	SOBRE O PENSAMENTO COMPUTACIONAL	28
2.4	RESOLUÇÃO DE PROBLEMAS	32
2.5	TRABALHOS CORRELATOS À PESQUISA.	33
2.6	O SCRATCH	37
2.6.1	Uso do Scratch como tecnologia educacional.....	45
3	A PESQUISA: DOS PASSOS INICIAIS AOS RESULTADOS	49
3.1	A ESCOLA	49
3.2	O MINICURSO PARA OS PROFESSORES.....	53
3.3	O MINICURSO PARA OS ALUNOS	54
3.3.1	O minicurso para os alunos do 3º ano do Ensino Fundamental	57
3.3.2	A escolha de conteúdo.....	57
3.3.3	Geração de dados	58
3.3.4	Análise de algumas aulas	60
3.3.4.1	Primeira aula	66
3.3.4.2	Segunda aula	79
3.3.4.3	Terceira aula	77
3.3.4.4	Quarta aula	81
3.3.4.5	Quinta aula	86
3.3.4.6	Sexta aula	89
3.3.4.7	Sétima aula	91

3.3.4.8	Oitava e nona aulas.....	94
3.3.4.9	Décima e Décima primeira aulas.....	99
3.3.4.10	Décima segunda aula.....	103
3.3.4.11	Décima terceira a Vigésima aulas.....	105
3.3.4.12	Vigésima primeira e Vigésima segunda aulas.....	107
3.3.4.13	Vigésima terceira e Vigésima quarta aulas.....	108
3.3.4.14	Vigésima quinta aula.....	109
4	RESULTADOS DA PESQUISA.....	112
4.1	APRENDIZADOS E DESAFIOS.....	112
4.2	CONSIDERAÇÕES E ANSEIOS.....	112
	REFERÊNCIAS.....	118
	APÊNDICE A - FICHA DE INSCRIÇÃO PARA O CURSO “APRENDENDO E ENSINANDO A PROGRAMAR COM O SCRATCH”.....	122
	APÊNDICE B - TERMO DE AUTORIZAÇÃO DE USO DE IMAGEM, VOZ E DADOS DIGITAIS E RESPECTIVA CESSÃO DE DIREITOS.....	123
	APÊNDICE C - APRENDENDO A PROGRAMAR COM O SCRATCH.....	124
	APÊNDICE D - APRENDENDO A PROGRAMAR COM O SCRATCH.....	125
	APÊNDICE E - TERMO DE AUTORIZAÇÃO DE USO DE IMAGEM, VOZ E DADOS DIGITAIS E RESPECTIVA CESSÃO DE DIREITOS.....	126
	APÊNDICE F - QUESTIONÁRIO INTRODUTÓRIO AOS ALUNOS: USO DO COMPUTADOR NO SEU COTIDIANO.....	127
	APÊNDICE G - PLANO DE AULA.....	128
	APÊNDICE H - GRADES DE AVALIAÇÕES DAS AULAS.....	129
	APÊNDICE I - MODELO DE PROJETO.....	130
	APÊNDICE J - ESTRUTURA DE APRESENTAÇÃO DO PROJETO/JOGO..	132
	APÊNDICE K – QUESTIONÁRIO FINAL.....	133

1 SOBRE A PESQUISA: CONTEXTO, OBJETIVOS E MÉTODOS

1.1 CONTEXTO

Vivemos, na contemporaneidade, situações de evolução tecnológica que têm modificado nossas formas de lidar com o mundo em diferentes esferas, a saber: profissional, escolar, social, dentre outras. Tais mudanças vêm acompanhadas de benefícios e desafios decorrentes das novas possibilidades de equipamentos, dispositivos, programas e aplicativos com os quais interagimos em nosso dia a dia. Tanto professores quanto pesquisadores da área da Educação Matemática têm visado a entender os fenômenos de inserção e de uso da tecnologia nos processos de ensino e aprendizagem, considerando pontos favoráveis e aqueles que precisam ser aprimorados. Isso demonstra que o processo de inovação tecnológica na educação merece nossa atenção e nossos estudos.

Uma das questões que vêm sendo discutidas diz respeito à defesa da inserção do ensino de programação na Educação Básica (VALENTE, 1993; 2013). Nesse sentido, pesquisadores têm discutido mudanças de papéis às quais podemos nos colocar – professores – e dos nossos alunos. Partimos do pressuposto de que nos tornamos produtores de conhecimento de conteúdo, ou seja, como dito em Resnick et al. (2009) e Resnick (2012), passar da fase de consumidores à de criadores. Acreditamos que essa mudança vai ao encontro das ideias de Valente (2003) ao afirmar que o conhecimento não está mais associado à memorização de informações, mas sim, ao saber buscar, selecionar e usar as informações.

Em termos da Educação Matemática, entendemos que essa pode ser uma interessante questão quando se deseja o ensino voltado a algumas destas tendências as quais discutimos, tais como a resolução de problemas (ONUCHIC, 2010) e a inserção de tecnologias no ensino e na aprendizagem (MALTEMPI, 2012). Há uma ligação entre esses enfoques e os recursos aos quais hoje temos acesso graças à crescente e contínua inovação tecnológica. À título de ilustração, elencamos (i) o uso dos jogos digitais (*games*) como forma de aproximação com o universo da maioria dos estudantes por entendermos que este é um mundo que eles conhecem e lhes atrai e (ii) o pretexto/contexto para colocá-los em posição de produtores de conteúdo. Em síntese, é uma situação que dá ao professor a possibilidade de convidar e propor aos estudantes que eles sejam criadores e programadores de jogos.

Para programar jogos, é preciso conhecer e utilizar códigos de programação, lidar com situações de resolução de problemas e tomar decisões, ações estas que desenvolvem o pensamento computacional. Para a Educação Matemática, acreditamos ser uma importante alternativa, em razão de incorporar elementos debatidos e exercitados por professores e pesquisadores que almejam melhorar os processos de ensino e aprendizagem de matemática nas escolas.

A respeito do pensamento computacional, há as ideias de Wing (2007), as quais indicam, em síntese, que o pensamento computacional é a habilidade de reformulação e resolução de problemas do mundo real. De acordo com Phillips (2009) e Resnick (2012), ao se fomentar o pensamento computacional nas pessoas, elas ficam mais próximas de compreender as informações tecnológicas passando a ser criadoras e não somente utilizadoras de elementos da tecnologia. Além disso, não somente aumenta a habilidade de desenvolver programas, mas também desenvolve capacidades, tais como: pensamento abstrato, pensamento algorítmico, pensamento lógico e pensamento dimensionável (PHILLIPS, 2009; WING, 2006). Salientamos as ideias de Wing (2006) e Resnick (2012) no que diz respeito ao desenvolvimento do pensamento computacional que permite aumentar a capacidade analítica nas diversas áreas do conhecimento. Dito de outro modo, essa expansão de conhecimentos extrapola o ambiente escolar, ou seja, pode ajudar as pessoas nos mais diferentes contextos.

Por termos um envolvimento pessoal e profissional com a tecnologia e por acreditarmos no seu potencial positivo na Educação Matemática, escolhemos observar o Conhecimento Tecnológico e Pedagógico do Conteúdo, conhecido na literatura internacional como TPACK (*Technological Pedagogical Content Knowledge*), que pode contribuir para o enfrentamento do desafio de aproximar o ensino e a tecnologia pelos professores. O TPACK refere-se à forma sintetizada de conhecimento com a finalidade de integrar as TIC e tecnologias educacionais para o ensino e aprendizagem em sala de aula (CHAI; KOH; TSAI, 2013). Dessa forma, iniciamos a pesquisa com 22 professores que trabalham no Ensino Fundamental I da Educação Básica de uma escola particular da região metropolitana de Curitiba. O trabalho foi desenvolvido com a autorização da direção. Nosso objetivo era entender e descrever os tipos de conhecimentos necessários a um professor para a prática pedagógica efetiva em um ambiente de aprendizagem equipado com tecnologia, segundo o modelo teórico formulado.

Em contrapartida, a direção da escola solicitou um trabalho semelhante com os alunos de 3º à 9º anos dos Ensinos Fundamentais I e II, nos quais decidimos que as ferramentas de aplicação seriam as mesmas, porém neste contexto verificaríamos o desenvolvimento do pensamento computacional.

O primeiro manuscrito deste relato de pesquisa foi baseado no curso ministrado aos professores e a observação realizada no contexto do TPACK. Contudo, alguns percalços ocorreram e a banca de qualificação sugeriu que analisássemos dados observados no curso ministrado aos alunos. Desse modo, restringimos nossos olhares ao desenvolvimento do pensamento computacional de estudantes do Ensino Fundamental, quando envolvidos em atividades de programação de jogos digitais. Especificamente, propor e realizar atividades de ensino de programação de jogos com o *software* Scratch¹ para estudantes do terceiro ano do Ensino Fundamental. Assim, nosso problema de pesquisa do mestrado tornou-se, assim, discutir o ensino de programação nos anos iniciais como uma possibilidade para desenvolvimento do pensamento computacional das crianças no contexto da Educação Matemática.

Escolhemos o *software* Scratch para criar os jogos por meio de códigos de programação. Desenvolvido em 2007 pelo *Media Lab* do *Massachusetts Institute of Technology* (MIT), o programa é utilizado e estudado desde então como uma plataforma ideal para uso por pessoas que estão começando a programar, visto que não exige conhecimentos prévios e dispõe de interface bastante amigável. Adicionalmente, o sistema não exige conhecimento de escrita de códigos de programação, mas sim, de montagem por blocos, como se fossem peças de um quebra-cabeça. Apresentaremos mais detalhes do Scratch em outro capítulo dessa dissertação, mas reforçamos aqui a razão da escolha desse software: gratuidade, facilidade de acesso e ideal para uso com crianças a partir dos 8 anos de idade.

Em outras palavras a plataforma que nos ajudaria no desenvolvimento de nossa pesquisa. Resnick, Kafai e Maeda (2003) também relataram vantagens em sua pesquisa, ao apontarem que o Scratch é um ambiente visual de programação em blocos, de fácil aprendizado que desperta interesse de crianças, adolescentes e jovens.

No que concerne ao escopo nacional, Araújo et al. (2015) afirmam que conteúdos de programação ainda não fazem parte dos currículos das escolas brasileiras. No entanto, eles citam algumas iniciativas a respeito desta prática como,

¹ Disponível para uso e download em <https://scratch.mit.edu>.

por exemplo, as escolas de programação *SuperGeek* e *MadCod*, sediadas em São Paulo, e programas como Robótica na Escola, promovido no Estado do Recife.

Sobre o cenário internacional, Valente (2003) relata ações importantes que, realizadas em diferentes localidades, tratam da importância da programação para todos. Tal proposta de inclusão curricular implica, em muitos casos, mudanças nos currículos escolares da Educação Básica. Exemplificamos essa tendência com dados de 13 países europeus que, desde 2014, inseriram a programação como conteúdo das disciplinas obrigatórias do ensino infantil ao nono ano.

Pereira (2013) defende que a Lógica de Programação deveria andar junto com outras disciplinas do ensino básico, tais como: Biologia, Química e Física. Neste contexto, o ensino de programação para crianças poderia desenvolver o pensamento computacional e passos lógicos para a resolução de problemas utilizando a tecnologia (KAFAI; BURKE, 2013).

1.2 OBJETIVOS

1.2.1 Objetivo geral

Discutir o desenvolvimento do pensamento computacional de estudantes dos anos iniciais do Ensino Fundamental participantes de um curso de programação de jogos.

1.2.2 Objetivos específicos

1. Apresentar uma discussão sobre o ensino de programação como uma alternativa para a implantação de uma metodologia ativa de ensino que, por suas características, coloca os estudantes como protagonistas do processo de construção de conhecimento e de produção intelectual;
2. Apresentar possibilidades de utilização do software Scratch para ensinar programação para crianças;
3. Apresentar possibilidades de trabalho com a resolução de problemas por meio do ensino de programação;
4. Evidenciar a produção de estudantes dos anos iniciais do Ensino Fundamental participantes de atividades de ensino de programação.

1.3 MÉTODOS

Para a realização da investigação relatada nesta dissertação, utilizamos métodos qualitativos de pesquisa. Conforme Richardson (2011), pesquisar qualitativamente é buscar uma compreensão de significados e características de fenômenos em contextos que envolvem aspectos subjetivos e motivações não explícitas de comportamentos. Nesse mesmo sentido, Silva e Menezes (2005) consideram a existência de uma relação dinâmica entre o mundo real e o sujeito que não pode ser traduzida em números, o que vai ao encontro de Richardson (2011, p.79) quando o autor diz que uma pesquisa qualitativa “não pretende numerar ou medir unidades ou categorias homogêneas”.

O desenho da pesquisa realizada pode ser interpretado por algumas características de pesquisa de natureza qualitativa apontadas por Triviños (1987), a saber: (i) o ambiente natural como fonte direta dos dados e o pesquisador como instrumento chave, (ii) o objetivo descritivo, (iii) há preocupação com o processo e não simplesmente com os resultados e o produto, (iv) a análise dos dados é indutiva e (v) a preocupação essencial é o significado.

O ambiente de investigação foi um minicurso de programação de jogos com Scratch. Essa experiência educacional teve duração de 25 aulas de 50 minutos cada, realizado com 29 estudantes do terceiro ano do Ensino Fundamental de uma escola do Município de Araucária/PR. Para reconhecimento dos estudantes e de seus conhecimentos prévios acerca do uso do computador e do Scratch, aplicamos um questionário. A análise dos dados coletados com a aplicação do questionário nos auxiliou no processo de planejamento das aulas. Nestas, desenvolvemos atividades em etapas cujo objetivo final era que os alunos desenvolvessem um jogo por eles programados.

Fizemos todos os registros das aulas em um diário de bordo. Analisamos a observação da atuação dos alunos e de suas produções após a finalização do minicurso, segundo os conceitos computacionais definidos por Brennan e Resnick (2012) e as questões ligadas ao desenvolvimento do pensamento computacional (WING, 2006; VALENTE, 2016; ONUCHIC, 2010). Para conhecer a satisfação dos estudantes com o ensino de programação com o Scratch após participarem do minicurso, aplicamos um segundo questionário aos alunos.

2 ESTUDOS QUE EMBASARAM A PESQUISA

Focalizamos o desenvolvimento da pesquisa em aspectos relacionados à aprendizagem dos estudantes quando eles estavam em contato com atividades de programação de jogos e ligadas ao desenvolvimento dos seus pensamentos computacionais. Por conta disso, para tratar dos temas envolvidos na investigação, necessitamos discutir alguns enfoques teóricos que se referem à aprendizagem. Podemos entender tais recortes como resultados de estudos que se interessaram em debater as maneiras pelas quais as pessoas constroem conhecimento. Tais resultados se constituem em formas sistematizadas de organizar e definir os modos de ensino e de aprendizagem em determinada época. Como os estudos sobre o pensamento computacional baseiam-se em teorias construídas anteriormente no que se refere ao tempo histórico, iniciamos a discussão apresentando aspectos que caracterizam o construtivismo e o construcionismo. Na sequência, discutimos características e desdobramentos do desenvolvimento do pensamento computacional.

Aproximando a discussão do âmbito da Educação Matemática, realizamos um estudo sobre a abordagem da resolução de problemas.

2.1 SOBRE O CONSTRUTIVISMO

O Construtivismo é um enfoque teórico ligado ao ensino e à aprendizagem. Nessa condição, este paradigma se ampara em um conjunto de pressupostos que definem como se dão os processos de ensino e aprendizagem.

Os estudos e as teorias elaboradas pelo biólogo Jean Piaget² abriram caminho para a sistematização do construtivismo. O trabalho de Piaget abordou o desenvolvimento mental da criança e seu desenvolvimento cognitivo, tratando de temas como pensamento, linguagem, inteligência, fases de desenvolvimento, equilíbrio e percepção (GOMES; GHEDIN, 2012).

A forma como Piaget concebe a inteligência deixa evidente que ela modifica-se nas relações com o meio, isto é, sujeitos numa ação sobre os objetos, sujeitos participando ativamente no processo de

² Jean William Fritz Piaget era suíço, nascido em Genebra em 09/08/1896. Faleceu na mesma cidade, aos 80 anos, em 16/09/1980. Sua formação inicial foi em biologia e depois dedicou-se às áreas de psicologia, epistemologia e educação.

aprendizagem e interagindo na construção de conhecimentos mediados pelo professor. [...] Para Piaget, o conhecimento não é algo acabado e estável, mas está em constante transformação pelo sujeito que através da sua ação constrói conhecimentos indispensáveis na sua adaptação ao meio. (GOMES; GHEDIN, 2012, p. 214).

Ainda segundo esses autores, que estudaram a obra e a vida de Piaget,

[o] professor que faz uso das ideias desse autor [Piaget] pode compreender como seu aluno assimila e acomoda informações e consegue transformá-las em conhecimentos. O professor que entende e abraça essa teoria permite ao aluno a oportunidade de ser sujeito ativo no processo de aprendizagem. (GOMES; GHEDIN, 2012, p. 214).

As teorias de Piaget foram interpretadas como solução de alguns dos problemas enfrentados na educação. Estas problemáticas estão ligadas a um tempo em que alguns pensavam que os alunos aprendiam por meio da assimilação dos conteúdos que a eles eram transmitidos pelos professores. Em outras palavras, situações em que os alunos eram entendidos como passivos, receptores dos processos de ensinamentos, enquanto os professores eram os detentores dos conhecimentos a serem depositados nos alunos. Nessa perspectiva, professores não abriam espaços de diálogo sobre o processo de construção do conhecimento com seus alunos.

Dentre as contribuições possíveis dos estudos piagetianos, acreditamos interessar, para a construção desta pesquisa, o fato de que, para Piaget, o papel do professor não é apenas ensinar, sinônimo de transmissão, mas sim, orientar o aluno em um processo de aprendizagem autônoma que permita que ele seja agente da construção de seu próprio conhecimento, o que pode ser retratado pela seguinte reflexão:

Piaget contribui no sentido de explicar a importância da ação do sujeito mediante o meio e essa ação é indispensável na construção de conhecimentos, ou seja, o aluno precisa participar ativamente do processo de aprender. Dessa forma, ele passa a construir coisas novas ao invés de repetir ou reproduzir algo que lhes foi transmitido. (GOMES; GHEDIN, 2012, p. 228).

Pesquisadores que se aproximam das ideias de Piaget entendem que o enfoque construtivista defende um modelo de aprendizagem no qual o professor estimula que o aluno construa conhecimentos a partir de uma busca autônoma de respostas em sintonia com a realidade que o cerca e de forma interativa com seus colegas. Ademais, a aprendizagem acontece na realização de um trabalho em

conjunto entre o professor e o aluno, no qual aquele proporciona a estas novas experiências, que está inserido em um ambiente favorável para a construção de novos significados.

Coll e colaboradores (2001) discutem os pressupostos do construtivismo apontando elementos que o caracterizam. Ressaltam que é preciso considerar que aprender não é copiar ou reproduzir a realidade e que, assim, a aprendizagem resulta de um processo de impulsão cognitiva desencadeada por conveniência, indispensabilidade ou conhecer, colaborando para o desenvolvimento. A teoria construtivista enxerga a pessoa que aprende como ativa e agente do próprio conhecimento, por meio da elaboração e integração de novas informações às suas estruturas de saber, ligando-as a saberes existentes e ou formando novos (PINTO, 2002). Nesse sentido, o conhecimento é construído, a partir da interação, em um processo no qual os saberes anteriores são importantes para construção de saberes posteriores, no qual cada conhecimento construído se integra aos conhecimentos prévios.

2.2 SOBRE O CONSTRUCIONISMO

O Construcionismo também é considerado um enfoque teórico ligado ao ensino e à aprendizagem. Credita-se o termo ao matemático Seymour Papert³, nas décadas de 1970 e 1980. Assim como para Piaget, Papert também entendia o conhecimento como ativamente construído pelas pessoas, sendo que o tempo de aprendizagem é diferente de um para o outro.

Papert sistematizou os pressupostos do construcionismo como uma abordagem do construtivismo. Ao fazê-lo, ele acrescentou dois novos aspectos principais: a) construção do conhecimento de forma partilhada e em conjunto e b) a produção de artefatos, já que para aprender algo é necessário construir algo (PAPERT, 1993). Exemplificamos que o artefato citado por Papert pode ser, por exemplo, o computador. Assim, introduzimos a ideia de que os alunos podem ser responsáveis pela construção de seus conhecimentos, individualmente ou em grupo, utilizando a tecnologia do computador.

³ Seymour Papert nasceu na África do Sul em 01/03/1928. Estudou e fez carreira acadêmica nos Estados Unidos, dedicando-se a pesquisar o uso dos computadores na educação e a inteligência artificial.

Papert (1998) critica os rótulos que, na maioria das vezes, são atribuídos às crianças pela sociedade.

Nossas crianças crescem [em uma] *sic.* sociedade permeada pela ideia de que há “pessoas espertas” e “pessoas estúpidas” ... “tudo é preparado para as crianças atribuírem suas primeiras experiências de aprendizagem desagradáveis ou mal sucedidas à sua própria inabilidade (PAPERT, 1998, p. 63).

Nessa visão, Papert (1998) considera o computador uma ferramenta poderosa que permite transpor a barreira do pensamento concreto para o abstrato.

O computador não é somente mais um instrumento educacional poderoso. Ele é o único a nos permitir os meios para abordar o que Piaget e muitos outros identificam como o obstáculo que deve ser transposto para a passagem do pensamento infantil para o pensamento adulto. [...] Conhecimentos que só eram acessíveis através de processos formais podem agora ser abordados concretamente. (PAPERT, 1998, p. 37).

Pelo enfoque do construcionismo de Papert (1985, 1993) ao usar o computador em atividades de aprendizagem, o aluno liga o concreto ao abstrato. Essa ponte acontece em um processo participativo que contribui na construção de conhecimento sobre o conteúdo que é ensinado. Uma das concepções da teoria de Papert (1985) trata da elaboração de cenários práticos e objetivos, por parte dos professores, que permitam que o aluno teste suas convicções e teses nos momentos de aprendizagem. Papert (1985) compreendeu na tecnologia digital uma chance de realização de mudanças nos processos de ensino e aprendizagem. As tecnologias digitais desenvolvem, em muitos alunos, a motivação que permite uma interação aluno-objeto, aluno-aluno e aluno-professor, baseando-se nos desafios e trocas de experiências.

Em 1967, Papert desenvolveu uma linguagem de programação chamada LOGO. Por ser uma linguagem interpretada e interativa, o resultado da ação dos usuários, ou seja, dos comandos dados, é mostrado na tela do computador imediatamente após ter sido digitado. Essa rapidez auxilia no processo de ensino e aprendizagem, pois podem ser exercitados e estimulados os testes e hipóteses em raciocínios para resolução de problemas. O aluno verifica seus erros e busca soluções, em um processo que vai ao encontro do enfoque construcionista.

A partir dos estudos desenvolvidos com o ambiente LOGO, Papert (1985) elaborou dimensões que servem de suporte à criação de ambientes de aprendizagem que estejam em consonância com o enfoque do construcionismo. A primeira dimensão é a pragmática, na qual trabalhamos a sensação do aluno estar aprendendo algo que poderá ser utilizado imediatamente. Na segunda dimensão, sintônica, consideramos a construção de projetos contextualizados em sintonia com o que o aluno considera importante. Já na dimensão semântica, terceira, o aluno manipula elementos que carregam significados para si e descobre novos conceitos. Na quarta e última dimensão, social, o aluno relaciona a atividade com a sua cultura e o seu social.

2.3 SOBRE O PENSAMENTO COMPUTACIONAL

No final dos anos 1960, quando foi criada a linguagem LOGO, Papert defendia a ideia de que a utilização da programação ajudava a pensar melhor. Assim, era algo importante para o processo de construção de conhecimento e de desenvolvimento do pensamento. Segundo este autor, a computação poderia ter "um impacto profundo por concretizar e elucidar muitos conceitos anteriormente sutis em psicologia, linguística, biologia, e os fundamentos da lógica e da matemática", pois permite à criança a possibilidade "de articular o trabalho de sua própria mente e, particularmente e a interação entre ela e a realidade no decurso da aprendizagem e do pensamento" (PAPERT, 1985, p. 2). Nesse contexto, para o autor, os computadores poderiam ser utilizados para que as pessoas pudessem pensar com as máquinas e pensar sobre o próprio pensar (PAPERT, 1985).

Na continuidade do desenvolvimento das pesquisas sobre o tema e após quatro décadas, Wing (2006) propõe um agrupamento dos mecanismos de raciocínio que podem ser utilizados para resolução de problemas por meio da computação. A pesquisadora chama esses mecanismos de competências e as sistematiza sob o termo de pensamento computacional. Ademais, defende que tais mecanismos podem ser estendidos a diferentes aplicações, inclusive àquelas que ultrapassam as fronteiras da área da computação. Nesse sentido, o pensamento computacional pode ser caracterizado como sendo uma abordagem de resolução de problemas que utiliza processos que exploram a organização e análise de dados, criação de modelos, simulação, construção de algoritmos e automatização de soluções.

Barcelos e Silveira (2012) apresentam síntese de conceitos discutidos por Wing (2006) sobre o pensamento computacional, apresentada nos seguintes pontos:

1. *Conceituar ao invés de programar.* Resolver um problema aplicando o pensamento computacional significa reduzir problemas grandes e aparentemente insolúveis em problemas menores e mais simples de resolver. Isso exige a capacidade de pensar de forma abstrata e em múltiplos níveis, e não a mera aplicação de técnicas de programação;
2. *É uma habilidade fundamental e não utilitária.* O pensamento computacional não é uma habilidade mecânica ou utilitária, mas algo que permite a resolução de problemas diversos utilizando um recurso ubíquo na sociedade atual – os computadores – e por isso deveria ser desenvolvido por todos os estudantes;
3. *É a maneira na qual pessoas pensam, e não os computadores.* A resolução de problemas através do pensamento computacional é um tratamento específico do problema de forma que ele possa ser resolvido por computadores, e não uma redução do raciocínio para simular o processamento do computador;
4. *Complementa e combina a Matemática e a Engenharia.* A definição de Wing considera o aporte da Matemática e da Engenharia para a Computação, conforme mencionamos anteriormente, e reconhece as particularidades trazidas pelo enfoque computacional;
5. *Gera ideias e não artefatos.* O pensamento computacional não deve ter necessariamente como resultado final a produção de software e hardware e reconhece que os conceitos fundamentais da Computação estarão presentes para resolver problemas em vários contextos do cotidiano;
6. *Para todos, em qualquer lugar.* Por fim, o pensamento computacional pode ser útil para todas as pessoas, em diversas aplicações. (BARCELOS; SILVEIRA, 2012, p.3-4).

Pesquisadores e organizações de diversas áreas interessadas no assunto foram reunidos pela *National Academy of Sciences* dos Estados Unidos da América em dois *workshops* (2009 e em 2011) para debaterem os enfoques do pensamento computacional.

Debates realizados em fevereiro de 2009 não chegaram a um acordo geral entre os participantes sobre o conteúdo preciso de pensamento computacional, e muito menos a sua estrutura. (USA NATIONAL RESEARCH COUNCIL, 2010, p. 65).

Embora os participantes do workshop não tenham concordado explicitamente sobre a definição de pensamento computacional, os exemplos que eles forneceram durante o workshop são valiosos como indicadores das maneiras como as pessoas veem a intersecção da computação, conhecimento disciplinar e algoritmos. (USA NATIONAL RESEARCH COUNCIL, 2011, p. 5).

A *International Society for Technology in Education* (ISTE) e a *American Computer Science Teachers Association* (CSTA) tentaram identificar conceitos do pensamento computacional e propuseram uma definição para direcionar as atividades realizadas na Educação Básica (K-12⁴). As organizações apontaram nove conceitos, a saber: coleta, análise e representação de dados, decomposição de problema, abstração, algoritmos, automação, paralelização e simulação. Em síntese, ressaltaram que as aptidões relacionadas aos conceitos não estão restritas às áreas das Ciências exatas ou tecnológicas.

Vários autores têm buscado reconhecer os componentes do pensamento computacional. Isbell et al. (2010) consideram que os conceitos principais que envolvem o pensamento computacional são: modelos, abstração, compreensão, escala e limites para o desenvolvimento, além de simulação a partir de modelos e possibilidade.

Segundo Zapata-Ros (2015), o pensamento computacional é composto de 14 componentes, a saber: análise, heurística, pensamento divergente, criatividade, resolução de problema, abstração, interação, recursividade, métodos colaborativos e metacognição. Alguns desses componentes estão relacionados não somente à resolução de problemas usando tecnologia, mas também sem o uso das tecnologias, sendo importante a distinção dos componentes que pertencem ao pensamento geral e ao pensamento computacional. A diferença é que quando utilizamos as tecnologias digitais, adicionamos possibilidades à resolução de problemas.

O CSTA K-12 (2011) ressalta a necessidade de desenvolvimento de habilidades computacionais e define o pensamento computacional como um processo de resolução de problemas, com as seguintes características: elaboração de problemas em que o computador possa ser utilizado como ferramenta de apoio na resolução; sistematização lógica e análise de dados; exposição de dados por modelos e simulações; automação de soluções por meio do pensamento algorítmico; reconhecer, analisar e implementar possíveis soluções e generalizar as soluções para uma variedade de problemas.

Karen Brennan e Mitchel Resnick, investigadores do MIT - *Massachusetts Institute of Technology* -, têm realizado, nos últimos anos, diversos estudos

⁴ K-12 é uma designação para a educação primária e a educação secundária como um todo. É usada nos Estados Unidos, em algumas partes da Austrália e no Canadá.

relacionados com o pensamento computacional. Os resultados desses estudos deram origem a um quadro de referências para estudar e avaliar o desenvolvimento do pensamento computacional, englobando três dimensões: conceitos computacionais; práticas computacionais e perspectivas computacionais.

Os conceitos computacionais são sete e englobam sequências (identificar uma série de etapas de uma tarefa), ciclos (executar a mesma sequência várias vezes), execução em paralelo (fazer as ações decorrerem ao mesmo tempo), eventos (fazer um acontecimento provocar outro), condições (tomar decisões com base em condições), operadores (expressar operações matemáticas e lógicas), e dados (armazenar, recuperar e atualizar valores) (BRENNAN; RESNICK, 2012).

As práticas computacionais estão focadas no processo de pensar e aprender, indo além do que se aprende, ou seja, para o como se aprende (BRENNAN; RESNICK, 2012). Assim, foram definidos quatro conjuntos de práticas: ação iterativa e incremental (desenvolver, verificar se funciona e, em seguida, continuar a desenvolver), teste e depuração (verificar se tudo funciona e encontrar e corrigir erros, se for o caso), reutilização e reformulação (fazer algo utilizando o que já foi feito, ou o que outros fizeram), abstração e modulação (construir algo grande unindo conjuntos de partes mais pequenas) (MIT, 2011).

As perspectivas computacionais são três: expressão (a computação é um meio de criação), conexão (criar com e para outros), questionamento (a tecnologia e com a tecnologia) (MIT, 2011). Para estes autores, a avaliação do desenvolvimento do pensamento computacional em jovens, pode ser feita através da análise dos projetos tendo por base as três dimensões por estes definidas.

A definição adotada neste trabalho para pensamento computacional consiste em uma abordagem de resolução de problemas anexando processos mentais e ferramentas que utilizam habilidades de organização e análise de dados, construção de algoritmos, abstração, criação de modelos, simulação, automatização de soluções e paralelismo. A definição foi escolhida por ser a definição adotada pela *International Society for Technology in Education* (ISTE) e pela *Computer Science Teachers Association* (CSTA).

De acordo com Nunes (2011), o pensamento computacional manifesta-se ainda na idade infantil, uma vez que as crianças raciocinam de forma computacional. Contudo, esta capacidade, infelizmente, não é explorada no ensino básico em todo seu potencial.

2.4 RESOLUÇÃO DE PROBLEMAS

Atividades de programação de computadores envolvem, de diferentes formas, o trabalho com resolução de problemas. Assim, pode-se dizer que desenvolver o pensamento computacional também é aprender a resolver problemas. Sendo o pensamento computacional um processo de formulação de problemas, trazemos, aqui algumas considerações sobre a resolução de problemas, algumas delas ancoradas na Educação Matemática.

A resolução de problemas pode ser uma importante habilidade desenvolvida pelos estudantes em diferentes contextos, quando a aprendizagem decorre de problemas que necessitam ser resolvidos; os estudantes aprendem enquanto resolvem problemas (JONASSEN, 2004). Essa ideia de aprendizagem fundamentada na resolução de problemas teve sua origem no trabalho de Dewey (1916), quando descreve as salas de aula como laboratórios de investigação de situações. Para D'Ambrosio (2010, p.1), "problema é uma situação, real ou abstrata, ainda não resolvida, em qualquer campo do conhecimento e de ação". Onuchic (2007, p. 9) conceitua "problema" como "[...] tudo aquilo que não sabemos fazer, mas que estamos interessados em fazer".

Echeverría e Pozo (1998) definem a resolução de problemas como sendo uma apresentação de circunstâncias abertas que requer dos alunos atitudes dinâmicas para encontro das resoluções. Dessa forma, a ação de ensinar pressupõe proporcionar aos alunos o entendimento de métodos, bem como o aproveitamento dos conhecimentos já disponíveis. Nessa proposta de ensino e aprendizagem, temos o propósito de responder às mais diferentes situações. Para que tal processo não seja confundido com o de resolução de exercício, os autores o definem como sendo um processo de ponderação dos diversos passos a se seguir, necessitando de análise para ser resolvido. Nesse sentido, Pozo (1998) acredita que na resolução de exercício, o aluno não necessariamente precisa decidir sobre o procedimento a ser utilizado para se chegar à solução. Contudo, a autora enfatiza que a reflexão é condição essencial para a resolução, assim como indagações e possíveis tomadas de decisões.

Dante (2009, p. 48) também faz semelhante diferenciação, ao dizer que exercício "[...] serve para exercitar, para praticar um determinado algoritmo ou processo." Além do mais, o problema "[...] é a descrição de uma situação [em que] se

procura algo desconhecido e não temos previamente nenhum algoritmo que garanta a sua solução." Para o autor, a resolução de um problema "[...] exige uma certa dose de iniciativa e criatividade aliada ao conhecimento de algumas estratégias."

Papert (1993) afirma que o método de resolução de problemas ao ser utilizado, auxilia os alunos a aprenderem a aprender, que faz com que o aluno busque uma resposta e não espere uma resposta do seu professor, desenvolvendo habilidades e domínio dos procedimentos.

Valente (1993) defendia as possibilidades de os alunos utilizarem os computadores em atividades de busca de informações, para processá-las e utilizá-las na resolução de problemas, o que permitiria uma compreensão daquilo que eles fazem e a construção de seus próprios conhecimentos.

Para proceder a situações que envolvam a resolução de problemas, várias habilidades são necessárias. Além de compreenderem as palavras, frases e símbolos, é importante a busca por solução, superando os obstáculos e dificuldades apresentados (POLYA, 2003). Nesse sentido, Pólya (2003) corrobora que a resolução de problemas engloba quatro etapas, a saber: (i) entender o problema; (ii) elaborar um plano, logo que se conheça a estratégia a ser utilizada; (iii) executar o plano para alcançar os resultados e, se necessário, reformulá-lo; e (iv) verificar os resultados, fazendo sua validação de acordo com o problema inicial.

De acordo com Onuchic (1999), na resolução de problemas o ensino está centrado no aluno, que constrói seus conceitos. Assim, a apresentação de problemas é caminho para se ensinar de forma diferenciada do tradicional. Leite e Esteves (2005) relatam que a aprendizagem baseada na resolução de problemas leva os alunos ao desenvolvimento de competências relacionadas ao trabalho em grupo, comunicação, cooperação e respeito mútuo. Contrapondo à estratégia do uso de problemas como meros exercícios, Onuchic (2009) sugere a implantação da resolução de problemas como metodologia de ensino.

2.5 TRABALHOS CORRELATADOS À PESQUISA

A história dos avanços tecnológicos ao longo dos séculos XX e XXI tem relação com cálculos numéricos. Alan Turing⁵, em *On computable numbers, with an application to the Entscheidungsproblem*, definiu as estruturas básicas da computação e a viabilidade dos cálculos numéricos, criando um artefato que ficou conhecido como

Máquina Universal de Turing. Para proporcionar a construção de computadores e o desenvolvimento de linguagens de programação e compiladores, outras áreas do conhecimento foram incorporadas à computação de forma a estruturar o seu corpo de conhecimento (*Body of Knowledge*). Segundo Denning (2005), a computação buscou suporte nas engenharias, na matemática e nas ciências naturais. O desenvolvimento de algoritmos e *softwares* é baseado no princípio da experimentação, como nas atividades de engenharia. Da matemática, utiliza a representação simbólica e o sistema de dedução fundamentado em axiomas, o qual é fundamental para o estudo da complexidade de algoritmos e da análise numérica.

Medina (2005) ressalta que para o computador executar um algoritmo, este deve ser escrito de maneira que o processador esteja habilitado a interpretá-lo. Gudwin (1997) menciona que as primeiras linguagens utilizadas no processo de desenvolvimento de software foi a linguagem *Assembly*, também conhecida como linguagem de baixo nível devido a sua proximidade com a linguagem de máquina. Com o passar dos anos, surgiu a necessidade de linguagens de mais alto nível, que

5 Alan Turing foi um matemático, lógico e cientista da computação. Britânico, nasceu em Londres em 23 de junho de 1912 e morreu em Cheshire em 7 de junho de 1954. É considerado o “pai da computação” devido às suas contribuições para a área.

tivessem outros recursos. Surgem, então as linguagens procedurais ou estruturadas, também consideradas de alto nível por se assemelharem com a linguagem natural. Elas representaram um salto no processo de desenvolvimento de software devido aos recursos oferecidos (GUDWIN, 1997). Três exemplos de linguagens procedurais são: C, Pascal e Fortan.

Como a busca pelo aperfeiçoamento da tecnologia da computação é um processo constante, as linguagens orientadas ao objeto foram desenvolvidas. Elas surgiram a partir do esforço de várias profissionais e foram sendo aperfeiçoadas ao longo dos anos pelo trabalho de inúmeros profissionais da área (PAGE-JONES, 1997). A proposta deste tipo de linguagem de programação é representar as situações do mundo o mais fielmente possível nos sistemas computacionais. Nelas, os *softwares* são considerados, analogamente, a uma coleção de objetos que interagem entre si. Essa percepção difere de outras aceções que os entendem como uma coleção estruturada de processos (FARINELLI, 2007).

Apesar das inúmeras vantagens, as linguagens orientadas ao objeto não são totalmente eficientes em todas as situações. Para tentar solucionar esse problema, surgem, na década de 1990, as linguagens orientadas ao aspecto, cujo objetivo era

aperfeiçoar as linguagens orientadas ao objeto. A orientação a aspecto é um paradigma que estende a orientação a objeto, implementando e acrescentando novas abstrações ou recursos a fim de suprir deficiências na capacidade de representação de algumas situações. (GOETTEN, 2006).

Entendemos haver várias linguagens de programação existentes na área da computação e é fato que cada uma possui características e particularidades, mas independentemente da linguagem utilizada, a programação também pode ser considerada um caminho para resolução de problemas.

Segunda esta linha de pensamento, Ponte (1991) salienta que os alunos, ao programarem, são levados a verificar a funcionalidade de seus programas, examinando as mais diferentes e possíveis soluções:

A construção de programas exige do aluno um esforço suplementar de compreensão dos conceitos, muitas vezes obrigando a encará-los sob novas formas, e exige a elaboração de uma estratégia semelhante, em muitos aspectos, às que se usam para enfrentar situações problemáticas. (PONTE, 1991, p. 78).

No entendimento deste autor, não há uma linguagem de programação melhor, assim como também não há necessidade de os alunos terem pleno conhecimento de programação para desenvolverem programas que solucionam problemas de diferentes áreas. À época de suas considerações sobre o tema, ele citou como exemplo a linguagem LOGO de Papert para dizer que com “[...] um número reduzido de instruções fundamentais de LOGO, podem ser feitos programas, resolvendo muito satisfatoriamente problemas já de grande complexidade” (PONTE, 1991, p. 79). Desde então, em razão das rápidas mudanças na área da computação e do desenvolvimento da tecnologia digital, outras linguagens têm ações semelhantes como, por exemplo, o Scratch, desenvolvido no Instituto de Tecnologia de Massachusetts (MIT), Estados Unidos, por membros do grupo de pesquisa *Lifelong Kindergarten (LLK)*.

No que concerne as críticas sobre os prováveis efeitos prejudiciais da introdução da programação prematuramente no desenvolvimento das crianças, Ponte (1991) relatou que

Seria um tremendo erro pedagógico estar permanentemente à espera que a mente amadurecesse para se lhes poder ensinar as coisas em

estado puro. A inteligência não se desenvolve por um simples processo de maturação interna, mas sim no confronto com as dificuldades de toda a espécie que lhe proporciona o mundo exterior. Quanto mais rápido e estimulante for este mundo, mais rápido e profundo será o seu desenvolvimento (PONTE, 1991, p. 80).

Nessa concepção de ensino, a idade não deveria ser um fator decisivo que limitasse o contato pedagógico da criança com a programação de computadores. Em razão de sua pertinência, essa questão tem sido constantemente discutida. Por exemplo, há pesquisas que salientam o crescente número de redes de ensino na Educação Básica que estão implementando o ensino da programação em seus currículos. Segundo Alvarez (2014, p.2), aprender a programar “[...] é um tipo de aprendizado que permite que elas [as crianças] sejam não apenas consumidoras de tecnologia, mas também produtoras”. Desse modo, os alunos podem utilizar a tecnologia sob outras perspectivas, que não somente como consumidores de soluções prontas.

Papert (1985) defendia que a criança deve aprender a programar o computador para sentir-se superior à tecnologia e, ao fazê-lo, obter uma proximidade com ideias mais significativas da ciência. Para o autor, ao usar o computador, as pessoas podem estabelecer novas relações com o conhecimento. Esta é uma poderosa ferramenta na mudança cultural das pessoas e é possível aprender de maneira natural, como se estivesse aprendendo a língua materna. Papert desenvolveu o LOGO para servir de suporte ao pensamento, pois quando a criança programa a tartaruga – símbolo e personagem do *software* LOGO –, controla seu mundo. Na visão de Papert, quando a criança tem o computador como ferramenta disponível, há a possibilidade de alterar o seu meio, permitindo-lhe novas aprendizagens. A programação de computadores oportuniza, assim, a construção do conhecimento. Em síntese, entendemos que a programação de computadores pode transformar o conhecimento das pessoas, uma vez que permite uma maneira diferente de trabalhar com a resolução de problemas.

Um aspecto do LOGO que também está presente no ensino da programação é o aspecto construtivista de construção do conhecimento e o ensino participativo. Pesquisas (ALVAREZ, 2014; BARCELOS; SILVEIRA, 2012) afirmam, por base empírica, que alunos em contato com ferramentas tecnológicas desenvolvem mais sua agência, ou seja, pró atividade, e estão mais abertos à interação e ajuda aos colegas. Papert não apresentou o LOGO como uma linguagem acabada, o que abriu caminho para o desenvolvimento de novas linguagens. O *Scratch* é uma delas.

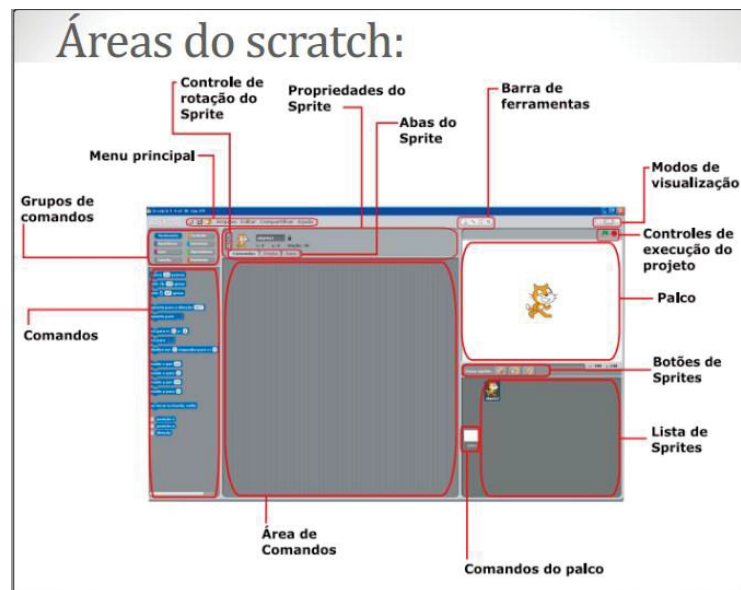
2.6 O SCRATCH

O *Scratch*⁵ é uma linguagem gráfica de programação que, de acordo com Andrade, Silva e Oliveira (2013, p. 261), é a “[...] mais recente de uma longa linhagem de ferramentas que se iniciou com a criação da linguagem de programação LOGO por Seymour Papert”. Ela foi desenvolvida no Instituto de Tecnologia de Massachusetts, Estados Unidos, por membros do grupo de pesquisa *Lifelong Kindergarten (LLK)*. Utilizando o conceito de arrastar-soltar, o Scratch possibilita programar por meio do arrastamento de blocos de construção (*building blocks*) que formam pilhas ordenadas (*stacks*), em um processo de estruturação dos códigos que se assemelha à montagem de um quebra-cabeças. O propósito do projeto foi criar um *software* simples e fácil de compreender por aqueles não habituados a programar. Em outras palavras, àqueles que poderiam criar seus projetos sem a necessidade de escrever os códigos de programação. Com o auxílio do Scratch, é possível criar histórias interativas, animações, simulações, jogos e músicas, desenvolvendo capacidades como criatividade, comunicação, colaboração, aprendizagem contextualizada e manipulação de diferentes tipos de mídia digital como, por exemplo, texto, imagem, áudio e vídeo.

Em nossa pesquisa, utilizamos a versão Scratch 1.4, disponível para Windows, Linux Ubuntu e Mac OSX. Na figura 2, ilustramos a área de programação do Scratch, indicando os recursos e as ferramentas disponíveis.

Figura 2: Tela inicial do *Scratch* 1.4

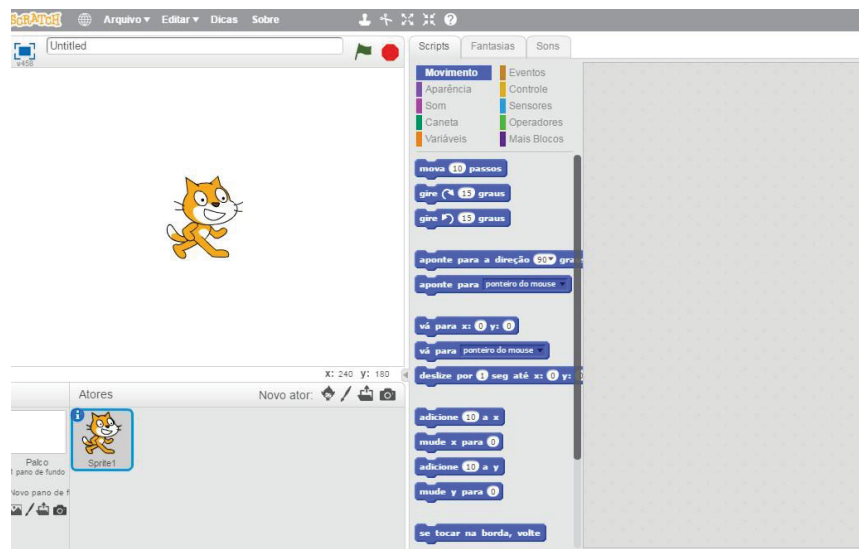
⁵ O site do projeto é o <http://scratch.mit.edu>, uma plataforma gratuita para desenvolvimento dos projetos.



Fonte: A autora.

Para ser utilizada, a versão 1.4 precisa ser baixada no computador. Contudo, já está disponível a versão 2.0, (tela inicial ilustrada na Figura 3) que permite, exclusivamente, seu uso *online*, sem a necessidade de instalação na máquina local.

Figura 3: Tela inicial do Scratch 2.0

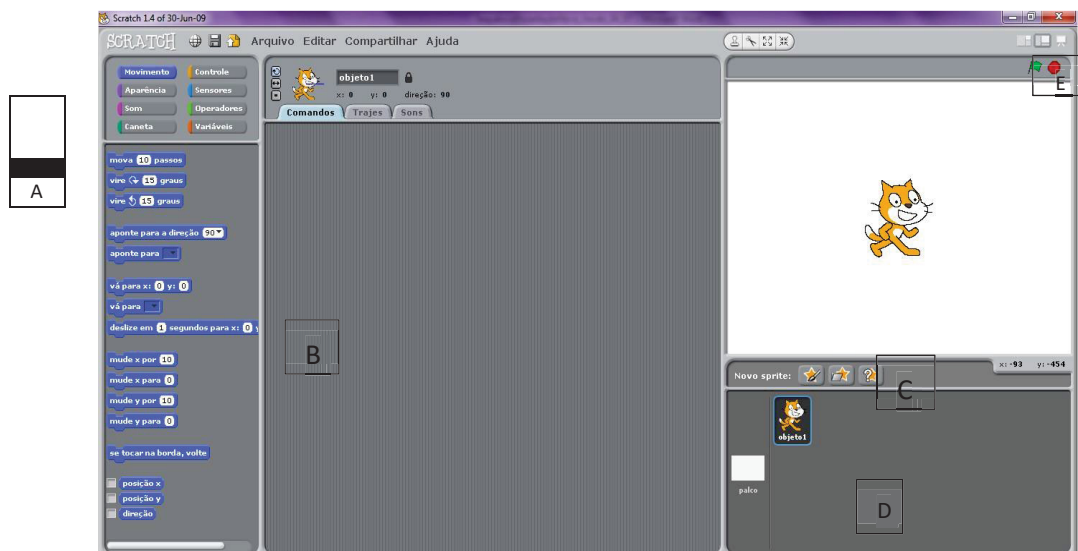


Fonte: A autora.

Todos os projetos criados com o Scratch podem ser compartilhados gratuitamente pela internet no site do projeto. Isso permite a divulgação de recursos e de ideias que abrem oportunidades à realização de trabalhos colaborativos de programação entre professores, por exemplo.

Explicamos, na figura 4, a divisão da área de trabalho do Scratch versão 1.4, que foi utilizada nas atividades realizadas com as crianças ao longo da pesquisa. Ao executarmos o Scratch, a tela inicial mostra uma janela como a ilustrada na Figura 4. Essa tela é composta por (i) uma área que apresenta e possibilita a escolha dos grupos de comandos desta linguagem de programação, (ii) uma área de edição que possibilita a criação do projeto ou a programação de eventos (ou “scripts”), (iii) uma área de definição dos objetos (ou “sprites”) e cenários (ou “palcos”) que integram um dado projeto, (iv) uma área que lista miniaturas dos “sprites” utilizados no projeto e (v) uma área de apresentação, que viabiliza a execução do projeto criado. Assim, a criação de um projeto no programa Scratch requer a escolha de comandos da linguagem de programação. A edição de um projeto que envolve a programação utilizando elementos gráficos para compor o “palco” da história, a definição de scripts ou “rotinas de ações” a partir do uso de comandos, especificação de parâmetros, sprites (objetos), trajés e sons.

Figura 4: Tela inicial do Scratch 1.4



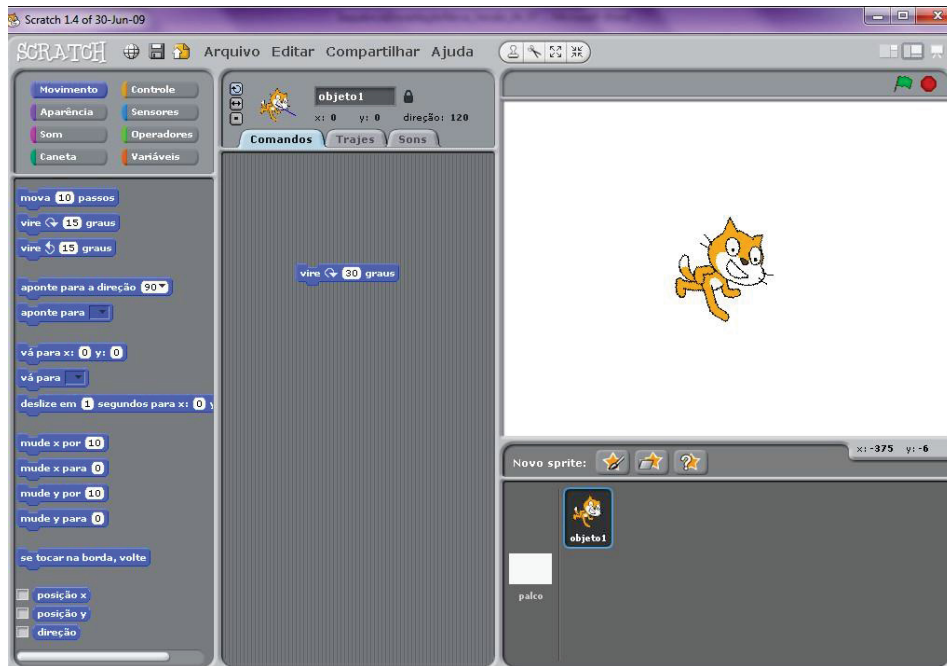
Fonte: A autora.

Na área de animação, são inseridos os sprites. O Sprite padrão do Scratch é um gato, símbolo do *software*. Para programar um Sprite, devemos selecioná-lo e, em seguida, arrastar para a área de programação dos blocos de programação que contêm os comandos que se quer associar ao Sprite. Quando quiser saber os efeitos de sua programação, clique na bandeira verde para visualizar. Os botões de programação estão divididos nas seguintes categorias: Movimento, Aparência, Som, Caneta,

Controle, Sensores, Variáveis e Operadores. Ao escolher uma categoria, encontram-se várias opções de comandos. Ainda na tela inicial, têm-se os botões de atalho: “selecionar idioma”, “salvar este projeto” e “compartilhar este projeto”. Acima da área de edição, a qual pode variar de acordo com a aba selecionada, há abas para opções comandos, trajes e sons. O palco é o local em que visualizamos o Sprite. Na parte superior do palco, há os botões para editar o objeto no palco, os quais são: Duplicar, Apagar, Crescer objeto e Encolher objeto. Na parte superior direita do palco, há os botões para alterar o tamanho de visualização do palco e, logo abaixo, há os botões para iniciar (bandeira verde) e parar o script (círculo vermelho) e, por fim, existe a área que disponibiliza a seleção/criação dos Sprites que figura abaixo do palco. Em Arquivos de Programas/Scratch/Projetos (se o Scratch estiver instalado no computador), há várias animações que o próprio programa oferece, as quais ajudam um novo usuário a conhecer o programa.

Para exemplificar um comando, na Figura 5, mostramos o código construído para a categoria “movimento” e o comando “girar” com o objetivo de mudar a posição do Sprite (gato). As ações que levaram à programação do código apresentado são: aparece um gato (Sprite) no centro do palco e dispomos da opção **movimento**, da categoria de comandos. Utilizando o cursor do mouse, podemos arrastar um dos comandos para a área de “Edição”. A fim de executar este comando, clicamos duas vezes com o botão esquerdo do mouse sobre o botão que arrastamos. O Objeto (Gato) gira/ move-se de acordo com o comando utilizado. É possível experimentar outros números como parâmetro do comando. Para alterar, precisamos clicar apenas uma vez com o botão esquerdo do mouse sob o número 15° e alterar o valor do parâmetro de entrada do comando e, em seguida, apertar “enter”. Outra forma de verificar a ação do comando é clicando no ícone da bandeira verde, que é o iniciar comandos.

Figura 5: Categoria movimento e comando girar



Fonte: A autora.

Ao abrir o Scratch na tela inicial, não existem comandos na “Área de Edição” e o Objeto (Gato) está localizado no centro do palco, ou seja, encontra-se na origem do plano cartesiano. O palco é onde ocorre a execução dos eventos que podem programar a movimentação de objetos na tela. Este dispõe de 480 (que vai de -240 à 240) unidades de largura por 360 (que vai de -180 à 180) de altura e é associado a um sistema de eixos coordenados cartesianos x-y.

Ainda na categoria **movimento** tem-se:

Categoria: Movimento / Comandos

Descrição dos comandos



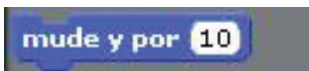
Movimenta o sprite para frente ou para trás (nº positivo ou negativo).



Muda a abscissa usando o incremento especificado.



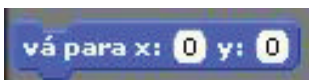
Leva o sprite para a abscissa especificada.



Muda a ordenada y usando o incremento especificado.



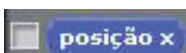
Leva o sprite para a ordenada especificada.



Move o sprite para a posição x e y especificada (posição no Palco).



Leva p/posição especificada no tempo determinado.



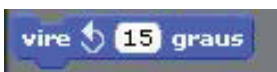
Reporta a abscissa do sprite (-240 a 240).



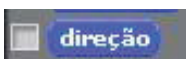
Reporta a ordenada do sprite (-180 a 180).



Gira o sprite no sentido horário.



Gira o sprite no sentido anti-horário.



Aponta o sprite para uma direção especificada 0=p/cima, 90=direita, 180=p/baixo, -90=esquerda.

Reporta a direção do sprite.

Muda a direção do sprite em 180º quando tocar na borda.

Movimenta o Sprite para a posição do cursor do mouse ou de outro sprite.



Aponta o sprite para o mouse ou para outro sprite.

Esses comandos só aparecem se for em Editar/Exibir blocos de Motor e funcionarão com um motor LEGO WeDo



Faz o motor funcionar pelo tempo designado.

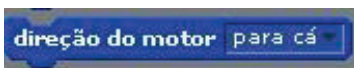


Faz o motor funcionar.

Interrompe o funcionamento do motor.



Estabelece a potência indicada e liga (varia de 0 a 100).



Muda o sentido mas não liga. (Para cá = sentido horário; para lá = sentido anti-horário;

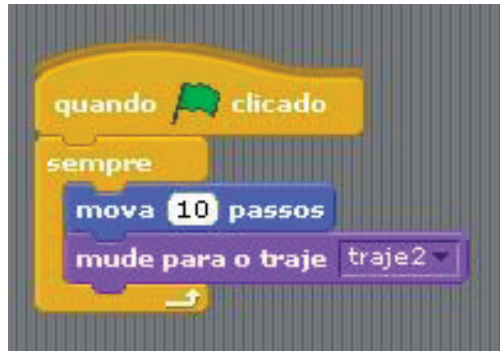
reverso = inverte o sentido)

As outras categorias e seus comandos serão descritos no decorrer dos relatos do capítulo 5.

Dentre as oito categorias existentes, há três tipos principais de blocos de comando, a saber:

1. Blocos Empilháveis (Simples): Estes blocos têm uma cavidade na parte superior e uma saliência na inferior. Podem ser conectados e reunidos em pilhas. Enquanto alguns blocos têm uma área de entrada na qual pode ser digitado um valor ou escolhido um item de um “menu”, outros têm a forma de “boca” em C, como este, permitindo a inserção de outro Bloco de comando.
2. Chapéus (Figura 6): Estes blocos têm a parte superior arredondada. Eles são colocados no topo das pilhas e esperam que ocorra algum evento como, por exemplo, uma tecla pressionada ou o clicar na bandeira verde, para fazer rodar os outros blocos da pilha.

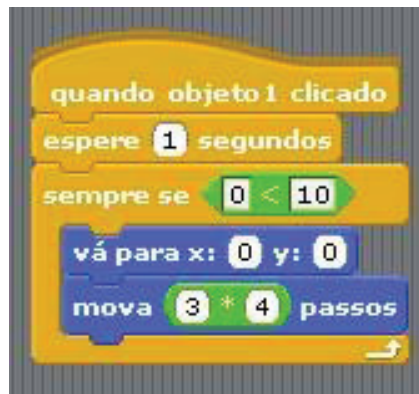
Figura 6: Bloco empilhado com chapéu



Fonte: A autora.

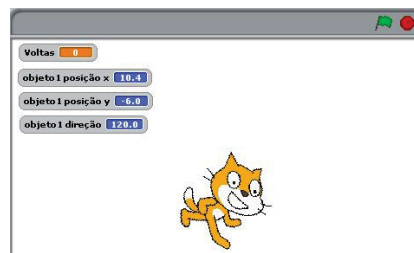
3. Repórteres/Valores (Figura 7). Estes blocos são criados para preencher a área de input de outros blocos. Podem ter 2 formatos e apenas se encaixam nas cavidades de mesmo formato. Repórteres com extremidades arredondadas devolvem valores numéricos e repórteres com extremidades pontudas devolvem valores “booleanos” (verdadeiro ou falso) e cabem nos que têm, dentro, reentrâncias com extremidades em triangulares ou retangulares.

Figura 7: Bloco empilhado com chapéu e Repórteres/valores



Fonte: A autora.

Alguns blocos do tipo repórter têm uma caixa de checagem que, se estiver selecionada, aparecerá no palco um monitor, mostrando o valor do repórter na ocasião e atualizando esse valor, à medida que for mudando. Um monitor pode mostrar o valor em 3 formatos: uma pequena janela com o nome do repórter, uma grande janela sem qualquer nome ou um *slide* que permite a manipulação do valor do repórter e que só está disponível para variáveis criadas pelo usuário.

Figura 8: Blocos repórter no palco

Fonte: A autora.

No Scratch, podemos criar e manipular listas formadas por números, letras, frases ou outros caracteres. As listas ficarão visíveis no palco, assim como os blocos repórter que possuem a opção de checagem.

2.6.1 Uso do Scratch como tecnologia educacional

Algumas pesquisas discorrem sobre o uso e a análise do Scratch. A pesquisa conduzida por Andrade, Silva e Oliveira (2013) apresentou resultados de jogos desenvolvidos por alunos para compreender como o processo influenciou a participação e interesse em sala de aula de matemática. Em razão deste propósito, os autores analisaram interações de alunos da Educação Básica durante as aulas e os produtos finais de suas participações. Os pesquisadores notaram um acréscimo de interesse dos alunos pelas aulas de matemática enquanto desenvolviam suas propostas de jogos. Os autores relatam que

[N]o início das aulas percebemos um pouco de resistência por parte dos alunos. Quando foi falado sobre produção de jogos eles tinham em mente o desenvolvimento de grandes games comerciais que eles estão acostumados a jogar em consoles ou no computador. No entanto, quando os primeiros projetos começaram a serem desenvolvidos os alunos se entusiasmaram e rapidamente dominaram a ferramenta. Segundo um dos alunos: “Eu ficava imaginando como o gatinho sabia que tinha que se movimentar. Gostei de fazer um joguinho com ele mexendo quando eu acertava a tabuada”. (Aluno B). (ANDRADE; SILVA; OLIVEIRA, 2013, p.263)

Na interação apresentada do professor com o aluno, pudemos perceber que mesmo os alunos, cuja maioria está frequentemente em contato com jogos fora do contexto escolar, sentiram-se surpresos quando esta proposta chegou à sala de aula pelo professor.

Em seu estudo, Aureliano e Tedesco (2013) buscaram responder à seguinte pergunta de pesquisa “Qual a opinião dos alunos sobre as suas experiências ao utilizar o Scratch como ambiente de apoio ao aprendizado de programação?”. Tendo como participantes alunos de um curso sobre programação desenvolvido no decorrer de duas semanas, os autores analisaram as interações em ambientes virtuais de aprendizagem em que ocorreram as interações e questionários aplicados aos participantes. Eles, pesquisadores, puderam perceber que as opiniões dos participantes estavam divididas entre positivas e negativas, ambas com argumentos para sustentação de suas respostas.

Na visão dos que avaliaram positivamente o uso do Scratch durante algumas aulas, eles mencionaram “[...] a facilidade de entendimento e utilização do ambiente por iniciantes na disciplina e a exibição do passo a passo de execução dos programas como vantagens de sua utilização”. O outro grupo de participantes, ou seja, aqueles que compreenderam como improfícua a utilização do Scratch, ressaltaram que “[...] a sua utilização havia sido desnecessária e que preferiam que a disciplina tivesse iniciado com o Portugol⁶ por ser mais semelhante a uma linguagem de programação escrita” (AURELIANO; TEDESCO, 2013, p.639).

Com base no mesmo aporte teórico-metodológico, Batista e Baptista (2013) desenvolveram um jogo com o programa Scratch e convidaram seis ex-alunos de uma disciplina de programação ofertada no curso de Bacharel em Sistemas de Informação para avaliar o produto final. O objetivo era de tecerem comentários sobre o desempenho do programa e sua aplicabilidade. Em linhas gerais, as avaliações foram positivas, mas os participantes-avaliadores também ressaltaram aspectos que precisam ser aprimorados.

Bressan e Amaral (2015) desenvolveram uma pesquisa etnográfica com alunos participantes de uma oficina ofertada no período do contraturno. No decorrer de dez encontros, as pesquisadoras geraram dados por meio de observações relatadas, entrevistas semiestruturadas e questionários. O objetivo foi “[...] verificar se a utilização do *Scratch* pode contribuir para o desenvolvimento do pensamento criativo por meio da aprendizagem baseada em problemas” (BRESSAN; AMARAL, 2015, p.509). Após proceder à análise dos dados, as autoras ressaltaram em suas considerações finais que

⁶ Segundo os autores, Portugol é a utilização de uma linguagem de programação escrita em Língua Portuguesa.

Verificou-se que o Scratch parece ajudar a resolver os conflitos cognitivos introduzidos pelos problemas encontrados pelos sujeitos discentes na construção de seus próprios projetos. O Scratch, pode desta forma, contribuir para o desenvolvimento do pensamento criativo evidenciado na capacidade de solução de problemas, estimulando a (re)formulação, reflexão, (re)organização de estratégias para atingir o objetivo proposto pelo próprio estudante em seu projeto. (BRESSAN; AMARAL, 2015, p.509).

Com aspectos e resultados similares aos de Andrade, Silva e Oliveira (2013), Cabral (2015) objetivou “investigar experiências do uso da informática no ensino de geometria e desenvolver uma atividade de intervenção com educandos no ensino fundamental, para verificação acerca de suas potencialidades” (CABRAL, 2015, p.8). O pesquisador ministrou 5 aulas, com duração de 50 minutos cada, nas quais 18 alunos participaram. No que diz respeito aos resultados, o autor afirma que “os educandos se mostraram participativos nas aulas no laboratório”, e que “passaram a atribuir novos significados aos conceitos da geometria, principalmente no tocante aos polígonos, visto que as atividades realizadas proporcionaram uma maior interação entre a teoria e prática, pois aconteceram de forma mais criativa e prazerosa ao educando” (CABRAL, 2015, p.84).

Sápiras, Vecchia e Maltempi (2015, p.973) buscaram “[...] indícios da formação de habilidades relacionadas à literacia⁷ digital e sua associação com aspectos lógicos matemáticos na construção de jogos eletrônicos feitos por meio do *software* Scratch”. A fim de cumprir este propósito, consideraram interações de alunos de 7º e 8º anos do Ensino Fundamental. Respaldados na literatura profissional e nas análises dos dados, os autores corroboram o que já conhecemos segundo a literatura da área, ou seja, visto que “consideram que o desenvolvimento de habilidades relacionadas à simulação pode contribuir para a capacidade de lidar e interpretar as mídias digitais, envolvendo assim o conceito conhecido como literacia digital” (SÁPIRAS; VECCHIA; MALTEMPI, 2015, p.987).

No estudo conduzido por Wangenheim, Nunes e Santos (2014), os pesquisadores buscaram entender o desenvolvimento de alunos da Educação Básica quanto ao uso do programa Scratch para a criação de animações. Os autores relataram os avanços dos alunos respondendo às quatro seguintes questões. A primeira era: “Quais conceitos de programação ou recursos do ambiente foram

⁷ Segundo autores, literacia digital é a capacidade de lidar e interpretar as mídias digitais.

aprendidos pelos alunos ao final da unidade instrucional?” (WANGENHEIM; NUNES; SANTOS, 2014, p.120); dentre os comandos que os alunos realizaram, os autores listaram desenhar um personagem, controlar o movimento via teclas, mudar o fundo durante a história, fazer uma personagem desaparecer quando toca um outro personagem, dentre outros. Um total de 15 comandos foram realizados pelos participantes e observados pelos pesquisadores. A segunda pergunta de pesquisa era: “Os objetivos de aprendizagem foram atingidos usando a unidade instrucional?” (WANGENHEIM; NUNES; SANTOS, 2014, p.121). Como respostas, os autores consideraram quatro categorias, a saber: pensamento computacional, colaboração, práticas computacionais e de programação e computadores e dispositivos de comunicação. Nestes aspectos todos os alunos tiveram êxito, exceto na categoria “pensamento computacional” que consideraram “a maioria”. A fim de compreender sobre a terceira questão “O ambiente SCRATCH facilita a aprendizagem?” (WANGENHEIM; NUNES; SANTOS, 2014, p.122), os proponentes do estudo mencionaram que “formatos e encaixes diferenciados das instruções e as cores distintas para cada categoria de comandos” e dentre outros. No que concerne à última pergunta de pesquisa, “O ambiente SCRATCH motiva os alunos a aprender computação” (p.122), os autores discorreram sobre o aspecto da motivação. Na resposta, foram afirmativos quanto a característica supracitada. Eles ressaltaram que

A impressão observada durante as aulas era que as crianças estavam aprendendo a programar sem perceber – focado na criação das histórias. Em muitos momentos foi possível observar um sentimento de satisfação/orgulho das crianças sempre que conseguiram fazer as personagens interagir (p.ex. movê-los ou fazelos falar). (WANGENHEIM; NUNES; SANTOS, 2014, p.122).

Esta visão sobre a experiência dos alunos corrobora a resposta dos pesquisadores quanto a última questão, que mencionava a criação de um ambiente agradável durante o desenvolvimento das aulas. As pesquisas apresentadas evidenciaram as potencialidades do uso do Scratch para ensino de programação.

3 A PESQUISA: DOS PASSOS INICIAIS AOS RESULTADOS

Neste capítulo, apresentamos a pesquisa realizada. A estruturação do capítulo contempla o planejamento e execução da agenda de pesquisa. Ademais, indicamos os métodos utilizados na geração dos dados e em suas análises.

3.1 A ESCOLA

Realizamos as atividades de ensino de programação na Escola Vicentina Sagrado Coração de Jesus localizada no Município de Araucária, na região metropolitana de Curitiba/PR. Esta é uma escola confessional de Educação Infantil e Ensino Fundamental e fundada em 1928. Atualmente, a instituição oferece aulas nos períodos matutino e vespertino. Segundo informação disposta na página eletrônica da escola⁸, ela adota como proposta pedagógica a metodologia interacionista e “visa a formação de sujeitos capazes de imaginar e elaborar ideias novas, de progredir e aperfeiçoar conceitos e tecnologias, desenvolvendo a criatividade em sintonia com a consciência social”. A respeito de sua estrutura física, além das salas de aula e demais espaços de ensino e aprendizagem, tais como: biblioteca e ginásio, a escola também dispõe de um laboratório de informática. As aulas de ensino de programação para os alunos que participaram da pesquisa foram realizadas neste espaço físico. Escolhemos a referida escola a partir da indicação de uma colega de trabalho, que lá atuava também. Segundo ela, caso apresentássemos o projeto à direção da instituição educacional, a escola poderia se interessar pelas atividades propostas, o que realmente aconteceu.

Nosso primeiro contato com o contexto supracitado aconteceu em 4 de abril de 2016, quando fomos conversar sobre a possibilidade de tê-la como campo da nossa pesquisa de mestrado. Fomos recebidas pela diretora, Irmã Amélia, e pela professora Adriana, que leciona no 5º ano, a quem explicamos nossa proposta de pesquisa e esclarecemos qual seria o envolvimento da escola. Como, até então, a ideia era trabalhar com a formação de professores para ensino de programação a alunos do Ensino Fundamental, explicamos a elas que estávamos propondo a realização de um minicurso, gratuito e com carga horária de 25 horas, sobre o uso do *software* Scratch para professores da escola.

⁸ <http://www.escolasagrado.com.br/institucional/proposta-pedagogica/20>

Sobre a contrapartida da instituição educacional, solicitamos a participação do grupo de professores e a infraestrutura do laboratório de informática. A diretora foi receptiva e concordou com a proposta. Contudo, solicitou que fizéssemos adaptações nas atividades para também oferecer um minicurso de programação aos alunos do terceiro ao nono ano do Ensino Fundamental. Adicionalmente, nos convidou para apresentar a proposta do minicurso aos professores da escola em um período de dois dias, quando da reunião pedagógica. A respeito do pedido da diretora, de também oferecer um minicurso de programação para os alunos, conversamos, mestrande e orientadora, e decidimos aceitar essa proposta, porém continuando com o foco de pesquisa na prática com os professores.

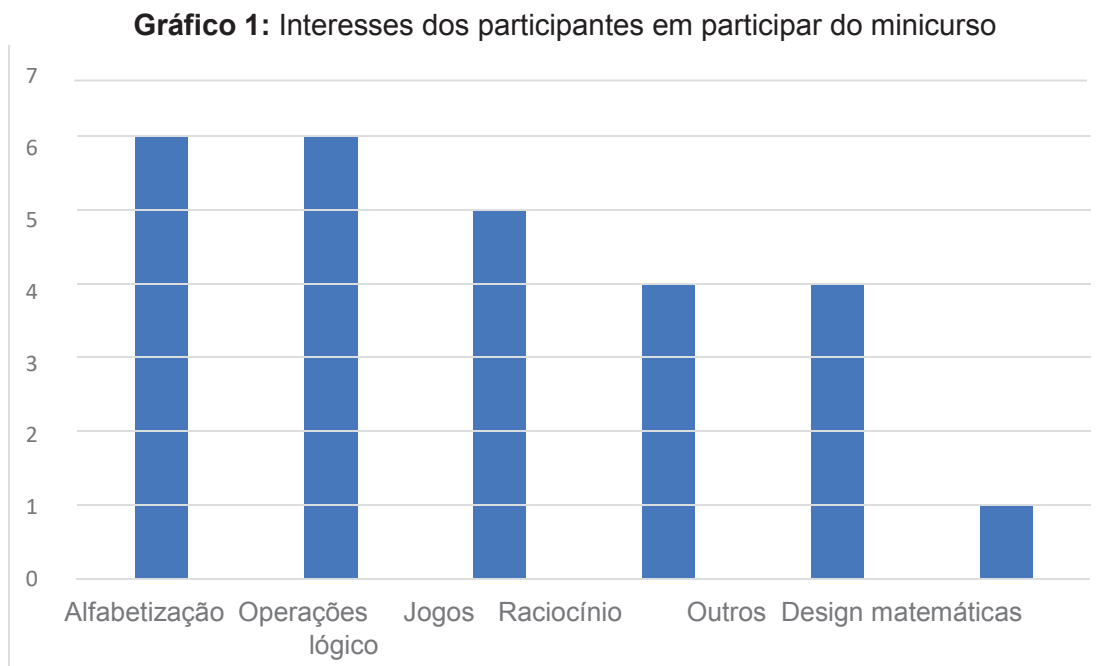
No dia 6 de abril de 2016, como combinado com a diretora, retornamos à escola para conversar com os professores e com a equipe pedagógica e lhes apresentar o projeto. Nessa oportunidade, em uma breve conversa, os presentes na reunião relataram suas experiências com o uso da tecnologia, dentre as quais, as dificuldades enfrentadas como, por exemplo, o receio em utilizar a tecnologia em atividades com os alunos por acreditarem que eles apresentam mais conhecimento nessa área do que os professores.

Outros relataram que utilizam as ferramentas tecnológicas somente para prepararem suas aulas, mas pouco usam com seus alunos. Quando mencionamos que o curso seria sobre o ensino de programação usando o *software* Scratch, nenhum dos presentes disse ter ouvido falar sobre ele. Nesse momento, solicitamos ao técnico do laboratório de informática da escola, que instalasse o Scratch nos computadores que seriam utilizados no curso. No mesmo dia, a proposta também foi apresentada à coordenadora do Ensino Fundamental II, Rosângela, que falou sobre o projeto com o grupo de professores que não pode participar da reunião. Todos os professores que se interessaram em participar do curso preencheram uma ficha de inscrição (Apêndice A), que continha campos que identificavam o perfil dos interessados apontando a disciplina que ministram, os anos com que trabalham e seus interesses relacionados à aplicação da tecnologia no ensino. Conhecer tais informações nos possibilitou entender, inicialmente, quem eram os interessados e o que os motivava para participar do curso.

Como resultado do convite à participação no minicurso, um total de 23 interessados preencheram a ficha de inscrição. Houve uma pessoa que disse que não tinha interesse em participar das atividades, mas que gostaria de observar os encontros. Dos 23 interessados que preencheram a ficha, 16 deles informaram quais

eram seus interesses ao participar do minicurso. Dos 22 participantes do minicurso voltado aos professores, 20 eram docentes, 1 era coordenador e 1 era técnico do laboratório. Todos preencheram um termo de autorização de uso de imagem, voz e dados digitais e respectiva cessão de direitos (Apêndice B).

No gráfico 1, apresentamos os dados a respeito dos interesses manifestados pelos participantes. Consideramos importante ressaltar que houve situações em que um participante expressou mais de um interesse.



Fonte: A autora.

Os temas mais citados pelos professores foram alfabetização e operações matemáticas. Jogos e raciocínio lógico foram outros interesses também citados. Na categoria “Outros” foram mencionados os temas: pesquisa, ortografia, nome e alimentação saudável. Houve um participante que disse se interessar pelo tema design.

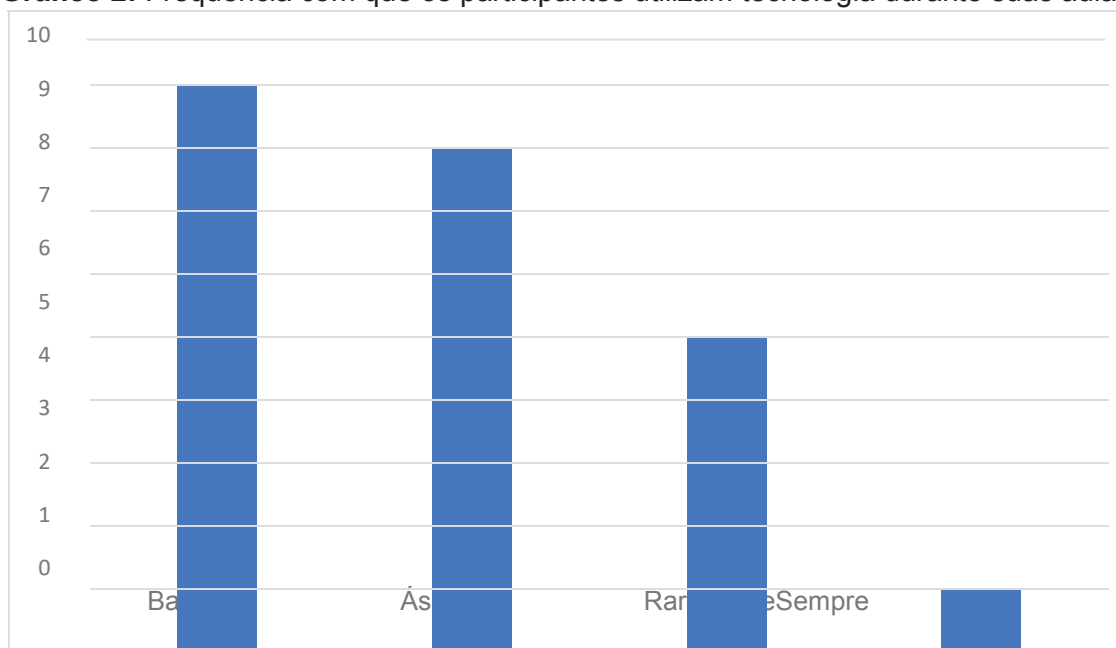
Com relação ao minicurso para os alunos, cuja realização foi combinada com a diretora da escola, as coordenadoras fizeram a divulgação para eles durante a semana de 10 a 16 de abril de 2016. No que diz respeito às ferramentas usadas, a diretoria divulgou cartazes (Apêndices C e D) e pediu que os interessados preenchessem fichas de inscrição com autorização do responsável (Apêndice E). Os minicursos, tanto dos professores quanto dos alunos, iniciariam no dia 10 de maio de 2016.

No primeiro encontro do minicurso voltado aos professores, iniciamos conversando com os participantes sobre seus interesses relacionados ao conteúdo do minicurso e sobre seus conhecimentos de informática, programação e uso de computador. Para guiar a conversa, fizemos seguintes perguntas ao grupo:

- 1) Usa tecnologia para elaborar suas aulas/trabalho?;
- 2) Faz uso da tecnologia durante suas aulas?;
- 3) Conhece algum jogo?;
- 4) Conhece/ sabe alguma linguagem de programação?
- 5) Gostaria de elaborar seus próprios jogos?

Todos os participantes responderam afirmativamente às perguntas 1, 3 e 5. Todos os participantes, exceto um, responderam negativamente à quarta pergunta. Quanto à segunda questão, os participantes responderam em termos de frequência com que a utilizam, o que é mostrado no Gráfico 2.

Gráfico 2: Frequência com que os participantes utilizam tecnologia durante suas aulas



Fonte: A autora.

Todos os participantes já utilizavam ferramentas tecnológicas em suas aulas, independentemente da frequência. No entanto, ao considerar suas respostas às questões, pudemos perceber que esse conhecimento tecnológico não se estende aos jogos e aplicativos recorrentemente usados pelos alunos fora do contexto escolar.

Completamos o minicurso com encontros que totalizaram as 25 horas propostas, em horários que foram definidos conforme a disponibilidade em comum dos participantes. Dos 22 docentes e funcionários que iniciaram o minicurso, somente

dois o concluíram. Quando perguntadas as razões das desistências, alguns a ligaram ao horário do curso, ao acúmulo de atividades do trabalho e a não remuneração, por parte da escola, pela participação fora do horário de trabalho.

Conduzimos o minicurso para os alunos no mesmo período do curso voltado aos professores, com a mesma carga horária e no contra turno escolar. Neste ponto desta narrativa, fazemos o que acreditamos ser uma importante consideração sobre uma mudança de rumo da pesquisa que a levou a ser concluída de forma diferente da inicialmente concebida: após a realização dos dois minicursos, houve mudança total de foco, o que se deveu ao encontro de poucas evidências no curso realizado com o professores – com pouca participação e conseqüente diminuta conclusão das atividades – em comparação com a riqueza das contribuições das crianças na participação no curso com os alunos. Decisivas para esse ajuste de foco foram as contribuições dos professores membros da qualificação da pesquisa que sugeriram a mudança da pesquisa, que passaria, então, a apresentar e discutir a produção dos alunos participantes do curso.

3.2 O MINICURSO PARA OS PROFESSORES

Utilizamos o laboratório de informática (LI) para a aplicação do projeto de pesquisa, local onde aconteceram as reuniões, aulas e conversas do grupo. Este espaço é amplo, arejado e conta com ar condicionado. A iluminação é de boa qualidade e também contém janelas enormes. Há 35 computadores interligados e conectados à internet. No decorrer do minicurso foi utilizado o Kit multimídia (computador, projeto, tv).

O minicurso aconteceu na escola, às sextas feiras após o turno vespertino de abril a julho de 2016, o qual foi intensificado nas férias, ocorrendo em horários diversificados e extras. Com uma hora de duração, das 17h15min às 18h15min, este foi o dia e horário escolhido pelos participantes como o mais apropriado. Sua carga horária total foi de 50h, sendo dividida entre 25 presencial e 25 a distância.

Nos encontros presenciais, discutíamos os projetos propostos e encaminhamentos com os professores-participantes, bem como lhes proporcionávamos conhecimentos específicos relativos à ferramenta *Scratch* para que pudessem manuseá-la. A carga horária a ser cumprida remotamente destinava-se à realização, ou continuação, do projeto iniciado em sala, podendo cada um terminar

ou não. O projeto modelo podia ser copiado pelo *pendrive* ou enviado por email. O material utilizado foi o *Software Scratch 1.4*, computadores, internet, *slides* com exemplos e passo a passo, baseado no livro “Aprenda a programar com Scratch” do autor Majed Marji (2014).

Os dados foram gerados durante todo o processo, por meio de conversas, relatos e expressões. Foi elaborado um diário de bordo com todas as informações. Este diário está sob minha posse.

A escrita do diário aconteceu simultaneamente ao minicurso. Eu, mestranda, escrevia o diário muitas vezes na frente dos participantes ou após o término das aulas. Os participantes não colaboraram com o processo de escrita para não ocorrer mudanças na fala. Pela mesma razão, eles não leram o diário por fazerem parte da pesquisa em andamento e para não tentarem influenciar e ou alterar os relatos e suas falas.

As conversas aconteciam no término de cada aula, onde cada um colocava seu ponto de vista sobre o conteúdo abordado na aula do dia. Durante as discussões, eram mencionados, por exemplo, possíveis formas dos participantes aplicarem as atividades desenvolvidas em nossos encontros em suas aulas.

Em relação às expressões também observadas por mim enquanto pesquisadora, compreendo-as como gestos, fisionomia (caras) e frases que ocorriam durante a aula/minicurso, tais como: *gostei!*, *não sei como aplicar*, entre outros.

Os relatos aconteciam no início de cada aula, em que os participantes narravam como foi a semana, o que fizeram, se aplicaram, ou não, o conteúdo trabalhado em nossos encontros e como o aplicaram.

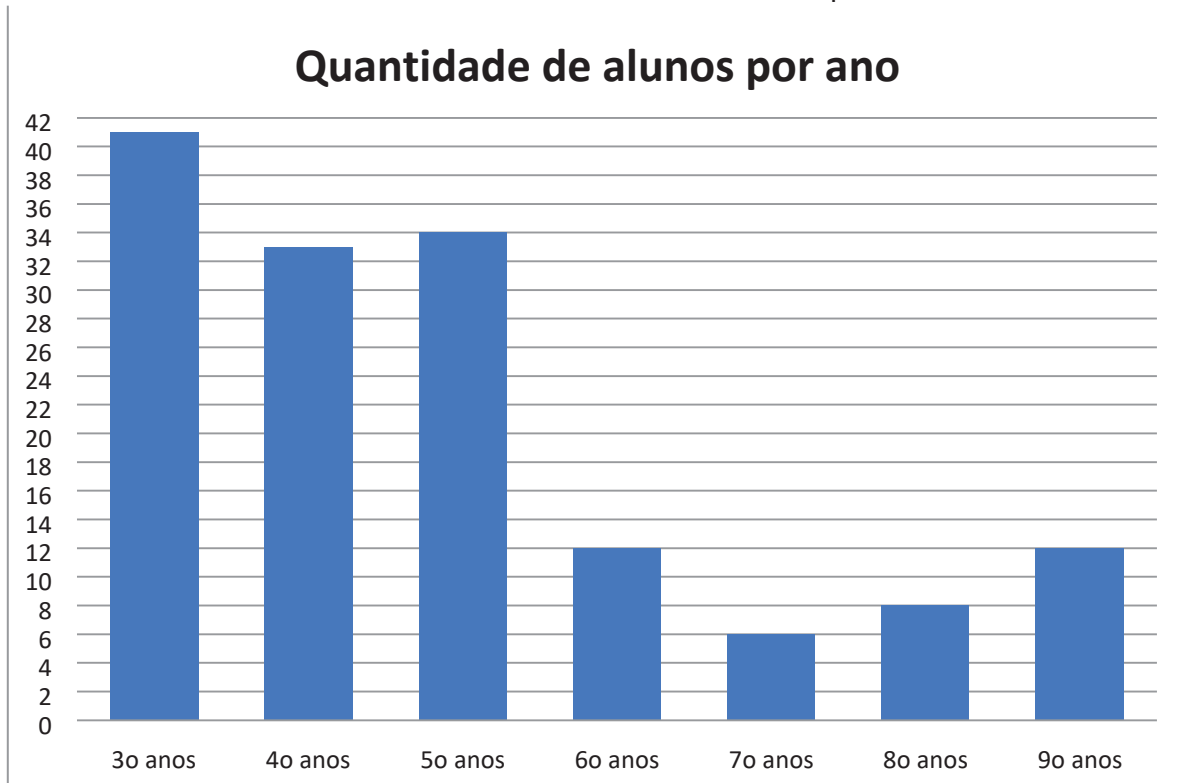
3.3 O MINICUROS PARA OS ALUNOS

Foi devido ao pedido da diretora da escola que planejamos e ministramos o minicurso de Scratch oferecido na Escola Vicentina Sagrado Coração de Jesus, no Município de Araucária, Estado do Paraná. Os alunos do terceiro ao nono ano do Ensino Fundamental que manifestaram interesse em participar após a divulgação feita pela escola, compareceram aos encontros do minicurso no período vespertino. As turmas da tarde frequentaram o minicurso durante as aulas regulares de informática. As turmas da manhã, de forma opcional, participaram das atividades no contra turno

escolar. Os responsáveis pelos alunos foram informados sobre o minicurso (Apêndices C e D) e preencheram termos de autorização para participação e de autorização de uso da imagem das crianças (Apêndice E).

No total, tivemos a participação de 146 alunos distribuídos nas turmas, conforme apresentamos no Gráfico 3.

Gráfico 3: Quantidade de alunos por ano



Fonte: A autora.

Dos 146 alunos que iniciaram o minicurso, apenas 4 alunos não o concluíram. Isso corresponde a uma desistência de cerca de 2,74%. Ao compararmos com o percentual de desistência do curso dos professores, de cerca de 91%, podemos confirmar que foi acertada a decisão de mudar o foco de pesquisa. Além disso, podemos nos questionar a respeito das razões de tamanha diferença, questionamentos esses que serão retomados na etapa de análise da pesquisa. Com relação ao público de alunos desistentes, foram dois alunos do 6º ano, um aluno do 8º ano e um aluno do 9º ano. Como justificativa, um deles disse não ter gostado do minicurso e três informaram que tiveram dificuldade em comparecerem nas aulas que eram dadas em contra turno.

Assim, o minicurso para todas as turmas aconteceu no turno vespertino, de abril a outubro de 2016, em dias e horários que respeitaram o calendário escolar das turmas: sexta-feira, das 14h45 às 15h20 (3º e 4º anos); sexta-feira, das 15h20 às 16h05 (5º e 6º anos); sexta-feira, das 16h05 às 16h45 (7º e 8º anos) e segunda-feira, das 16h45 às 17h25 (9º ano). Entendemos, então, que cada turma realizou as atividades dispondo de 1 hora/aula por semana, perfazendo uma carga horária total de 25h. Ministramos, portanto, 25 encontros presenciais com alunos do 3º ao 9º ano do Ensino Fundamental realizados no laboratório da escola.

O laboratório de informática da escola (Figura 9) é um espaço amplo, arejado e climatizado. A sua iluminação é de boa qualidade, com grandes janelas. Nele, há 35 computadores conectados à internet, organizados em mesas dispostas em U, além de um conjunto no centro da sala com as mesas uma virada para a outra, onde também fica a mesa do professor. A sala também possui projetor e uma TV ligada ao computador do professor. No decorrer do minicurso foi utilizado o Kit multimídia (computador, projetor e TV).

Figura 9: Laboratório de Informática



Fonte: <http://www.escolasagrado.com.br/institucional/estrutura-fisica/64>, acesso:

25/01/2017.

Durante as aulas do minicurso, cada aluno de cada turma utilizou um computador. Considerando que conduzimos o minicurso para sete turmas, escolhemos estudar, nesta pesquisa, as atividades que foram realizadas com as turmas mais jovens, ou seja, do 3º ano do Ensino Fundamental (turmas B e C da escola, que realizaram juntas o minicurso, formando uma única turma), crianças na faixa etária dos 8 anos. Essa escolha foi feita para discutir temas relacionados ao desenvolvimento do pensamento computacional, na perspectiva da Educação Matemática, de crianças que estão em fase de alfabetização. Além disso, foi uma oportunidade de debater um tema que acreditamos que merece ter sua investigação ampliada: o ensino de programação nos anos iniciais do Ensino Fundamental.

3.3.1 O minicurso para os alunos do 3º ano do Ensino Fundamental

De acordo com informações levantadas junto às professoras regentes das turmas de 3º ano da escola e à coordenadora dos anos iniciais do Ensino Fundamental I, a turma B era constituída por 14 alunos, sendo 6 meninos e 8 meninas, com uma média de idades de 8 anos. Nesta turma, não existiam alunos com retenções e nenhum aluno de inclusão. A turma C era constituída por 15 alunos, sendo 7 meninos e 8 meninas com 8 anos de idade em média. Assim como na turma anterior, nesta também não existiam retenções.

As professoras regentes ressaltaram que os alunos são receptivos às novidades e a novos desafios, sendo participativos, comportados e comprometidos com sua aprendizagem. Entre os meses de abril e outubro de 2016, ministramos o minicurso às duas turmas do 3º ano durante as suas aulas de informática, acompanhadas pelo professor de Informática da escola e pela professora regente da turma, que participaram de todas as aulas. Durante esse período, tivemos a oportunidade de observar e registrar informações fundamentais por meio de observação direta.

3.3.2 A escolha de conteúdo

Nos anos iniciais do Ensino Fundamental, entendemos que precisamos retomar as vivências cotidianas das crianças com números, formas e espaço, e também as

experiências desenvolvidas na Educação Infantil a fim de iniciar sistematização dessas noções. Nessa fase, as habilidades matemáticas que os alunos devem desenvolver não podem ficar restritas à aprendizagem dos algoritmos das chamadas “quatro operações”.

Compartilhamos da compreensão de que a aprendizagem em Matemática está relacionada à compreensão, ou seja, à apreensão de significados dos objetos matemáticos, sem deixar de lado suas aplicações. Os significados desses objetos resultam das conexões que os alunos estabelecem entre eles e os demais componentes, entre eles e seu cotidiano e entre os diferentes temas matemáticos.

Nessa perspectiva, trabalhamos com os alunos os mesmos conteúdos que as professoras tinham trabalhado ou estavam trabalhando com eles em sala de aula, porém com uma abordagem diferente. Nosso propósito ao focar os mesmos conteúdos curriculares tem respaldo na possibilidade de os alunos formularem problemas em outros contextos. Assim, os conteúdos pré-selecionados foram:

Números: Ordenação, composição, reta numérica, operações (adição, subtração, multiplicação e divisão) e problemas.

Álgebra: Sequências.

Geometria: Congruência, figuras geométricas planas e espaciais.

Grandezas e medidas: Perímetro, área, volume, medidas de massa e tempo, sistema monetário.

Probabilidade e estatística: Coleta, classificação e interpretação dos dados, espaço amostral, gráfico de barras.

Alguns conteúdos foram trabalhados mais de uma vez e vários conteúdos foram abordados em conjunto com outro.

3.3.3 Geração de dados

Optamos por adotar os seguintes métodos e técnicas de geração dos dados oriundos da pesquisa, que foram obtidos ao longo das aulas do minicurso:

1. Observação direta durante o minicurso, na qual procuramos conhecer os alunos, as suas motivações, aptidões, características, competências e limitações para ajudar na caracterização de seus perfis;
2. Consulta aos documentos escolares para caracterização das turmas, como planos de aula, atas de conselhos e relatórios de turmas;

3. Grade de observação dos alunos, na qual foram marcados a frequência deles, a utilização do material escolar e aspectos concernentes ao comportamento;
4. Grade de avaliação do minicurso, na qual ao término de cada aula o aluno tinha uma classificação final de acordo com o seu desenvolvimento;
5. Grade de avaliação de usabilidade do conteúdo do minicurso e um questionário de satisfação para verificar como os alunos se sentiam ao utilizar o *Scratch* para desenvolver os seus próprios jogos;
6. Conversa informal com os regentes das turmas, na qual procuramos capturar dados dos alunos, antes, durante e após do minicurso, de forma a perceber se houve mudanças deles com relação ao uso da tecnologia envolvida no minicurso.

Dessa forma, os dados foram gerados durante todo o processo a partir de conversas, relatos e observações. Elaboramos um diário de bordo com todas as informações. A escrita do diário aconteceu simultaneamente ao minicurso.

As conversas aconteciam no término de cada aula, quando cada aluno colocava seu ponto de vista sobre o conteúdo abordado na aula do dia. Durante as discussões, eram mencionadas, por exemplo, possíveis formas dos participantes aplicarem as atividades desenvolvidas em nossos encontros em suas aulas de Matemática, nas resoluções de problemas, das atividades e exercícios propostos por seus regentes.

Em relação à observação, também consideramos as expressões dos alunos ao longo do minicurso, que foram materializadas na forma de gestos, mudança na fisionomia (caras) e falas que ocorriam durante a aula/minicurso, tais como: *gostei! não sei como aplicar*, entre outros.

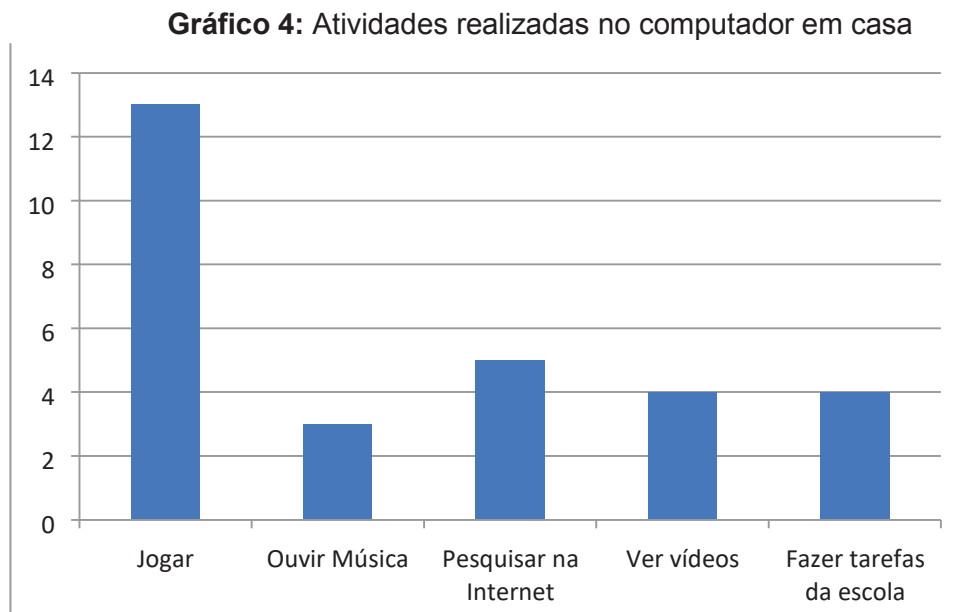
Os relatos aconteciam no início de cada aula, quando os participantes narravam como foi a semana, o que fizeram, se aplicaram ou não o conteúdo trabalhado em nossos encontros e como o aplicaram. Os relatos eram marcados por exemplos e experiências tanto positivos quanto negativos.

3.3.4 Análise de algumas aulas

Esta fase do projeto teve início com a análise do contexto com a finalidade de caracterizar os perfis dos alunos e identificar as necessidades e interesses da turma.

No primeiro dia de contato com os alunos, pedimos que eles respondessem a um questionário (Apêndice F) para obter dados relativos à utilização que eles faziam e gostariam de fazer do computador. O questionário foi respondido durante o tempo de aula pelos 29 alunos presentes.

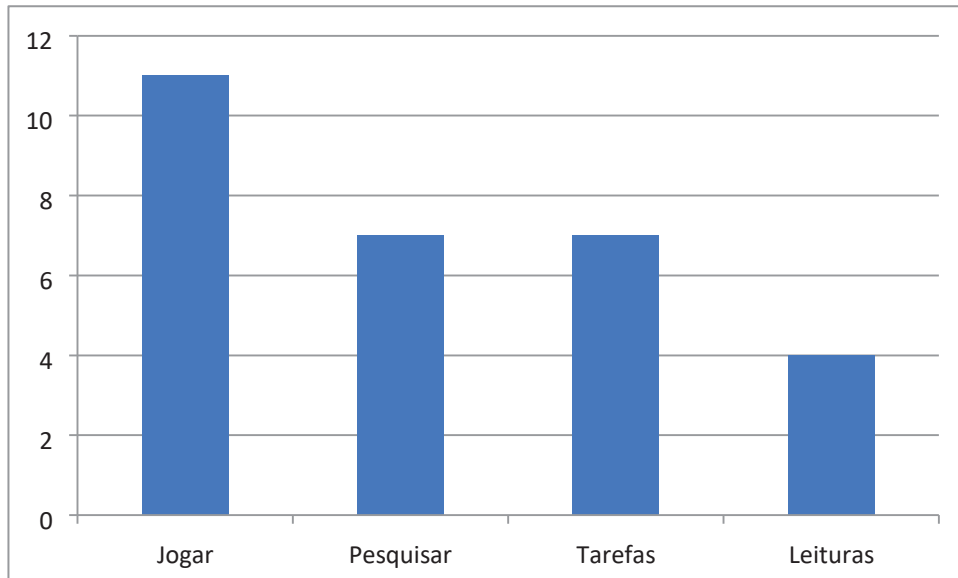
Quando questionados a respeito das atividades que costumavam realizar no computador em casa, a atividade mais mencionada pelos alunos foi jogar no computador, conforme representamos no Gráfico 4.



Fonte: A autora.

Já sobre as atividades que mais realizam no computador nas aulas de informática na escola, ilustramos no Gráfico 5 as opções escolhidas, indicando que, também, segundo os alunos, o que eles mais fazem é jogar.

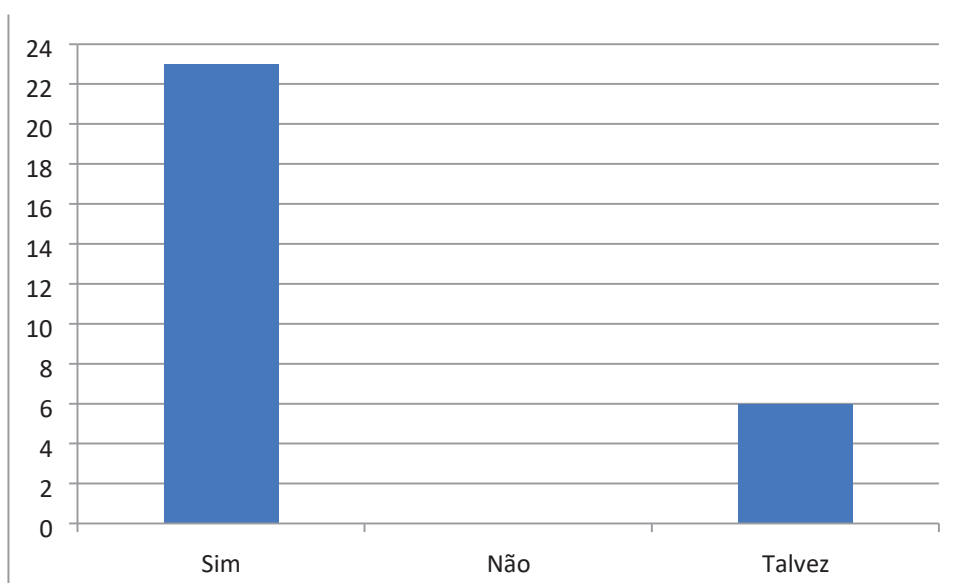
Gráfico 5: Atividades realizadas na aula de informática na escola



Fonte: A autora.

Sobre a questão de utilização de jogos no computador pelas crianças, como a proposta e pesquisa envolve o ensino de programação, nos interessava saber se os alunos tinham interesse na criação de jogos digitais, o que poderia ser alcançado nas aulas do minicurso de *Scratch* proposto. As respostas dos alunos, que representamos no Gráfico 6, indicaram que eles se interessaram pela ideia de criarem os jogos, o que ia ao encontro da proposta do minicurso.

Gráfico 6: Opinião dos alunos quanto a criar seus jogos no computador envolvendo as atividades e conteúdos escolares



Fonte: A autora.

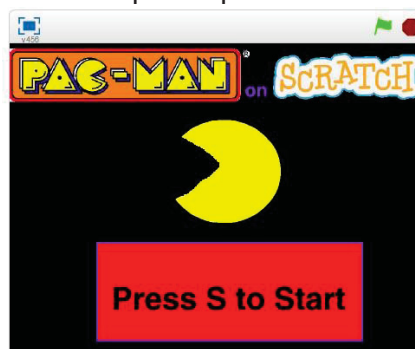
Esses dados nos despertou a reflexão sobre o maior uso do computador para jogar, mesmo nas atividades feitas na escola. Acreditamos que isso pode indicar tanto a utilização do jogo por si só, como diversão, mas também pode ser encarado como instrumento para abordar conteúdos escolares pelos professores. Entendemos que independentemente do enfoque (diversão ou pedagógico), já que as crianças indicaram não terem trabalhado na criação de seus próprios jogos, ou seja, utilizam jogos prontos, na maioria das vezes, poderiam desenvolver trabalhos de autoria por meio da tecnologia. Observando as considerações de Papert (1993), concluímos que era necessário adotar uma estratégia que os motivasse, os desafiasse e os envolvesse na concepção de algo significativo. Nesta nova possibilidade de motivá-los, decidimos utilizar a linguagem de programação do *Scratch* tanto para tornar os alunos autores de jogos quanto para trabalhar com os conteúdos escolares, ao contextualizar e dar significado ao uso da tecnologia dos computadores.

Quando questionamos aos alunos se eles gostariam de aprender a utilizar novos programas de computador, todos afirmaram que sim. Nenhum dos 29 alunos tinham conhecimentos de programação e também não conheciam a ferramenta Scratch. Isso nos apontou a pré-disposição dos alunos para aprenderem algo novo.

Os dados que obtemos com a aplicação do questionário nos permitiram verificar se agradava a maioria dos alunos utilizar o computador para jogar e também para aprender a utilizar novos programas de computador e a criar jogos.

Depois de aplicar o questionário aos alunos, apresentamos a eles alguns jogos prontos que foram construídos com o *Scratch*, para que tivessem uma ideia do que eles poderiam criar ao aprenderem a programar. Observamos a reação deles aos exemplos apresentados, ilustrados nas Figuras 10, 11 e 12.

Figura 10: Exemplo 1 apresentado aos alunos



Fonte: <https://scratch.mit.edu/projects/125751986/> acesso em: 03/04/2016.

O primeiro exemplo é semelhante ao *PacMan*, porém uma versão adaptada e elaborada utilizando o *Scratch*.

Figura 11: Exemplo 2 apresentado aos alunos



Fonte: <https://scratch.mit.edu/projects/119347416/> acesso em: 03/04/2016.

Neste exemplo, o personagem cachorro é um professor de matemática que apresenta problemas aos alunos que devem resolvê-los. Conforme as respostas corretas são dadas pelo jogador, ele sobe de nível no jogo.

Figura 12: Exemplo 3 apresentado aos alunos



Fonte: <https://scratch.mit.edu/projects/72008980/> acesso em: 03/04/2016.

Esse é o exemplo de um jogo no qual é necessário que o jogador tenha boa concentração e coordenação motora para alterar as cores de acordo com as setas que aparecem. Conforme o jogador altera as cores, ele sobe de nível e a entrada das setas ficam mais rápidas.

Após apresentar esses exemplos, mostramos o site do Scratch na internet⁹ (Figura 13) e informamos aos alunos que no site, especificamente na opção “projetos”, eles poderiam encontrar outros jogos e aplicativos, que podiam acessar e jogar

⁹ <https://scratch.mit.edu/>

gratuitamente. Dissemos, ainda, que depois de aprenderem a programar com o Scratch, eles estariam aptos a se cadastrarem no site e criarem seus próprios jogos e aplicativos, além de alterarem os existentes.

Figura 13: Tela inicial do site do Scratch disponível na internet



Fonte: <https://scratch.mit.edu>

Depois de apresentarmos todos os exemplos, fizemos as seguintes perguntas à turma a fim de perceber a aceitação da proposta:

- Acharam interessantes os exemplos demonstrados?;
- Gostariam de aprender um pouco sobre a ferramenta *Scratch*?;
- Poderiam utilizar a ferramenta para auxiliá-los nas tarefas da escola?

Todas as respostas a esses questionamentos foram verbais. Com a ajuda do laboratorista e das professoras regentes para identificar as respostas dos alunos, verifiquei que a turma foi bastante receptiva à possibilidade de utilizar a ferramenta para aprenderem a programar e criar jogos. Todos mostraram muito entusiasmo ao verem as suas potencialidades. Assim, percebemos que os alunos consideraram a ferramenta interessante e que demonstraram vontade em trabalhar com ela.

Então, apresentamos aos alunos a proposta do minicurso: programação de jogos sobre conteúdos escolares das diferentes disciplinas que abrangiam o seu ano escolar, ou seja, o 3º ano do Ensino Fundamental. A maioria demonstrou gostar dessa ideia, mas alguns disseram preferirem lidar com jogos que não abrangessem conteúdos escolares, o que era uma reação esperada considerando o interesse já identificado quando foi analisamos o perfil da turma com relação ao uso do computador em casa e na escola.

Todavia, deixei claro que não descartaríamos a possibilidade de criarmos outros tipos de jogos para, assim, não tirar o interesse e a criatividade dos alunos. Ao

final da apresentação dos exemplos, alguns alunos solicitaram uma cópia desses jogos para gravarem em seus *pendrives*, dispositivos de armazenamento pessoais, para que pudessem mostrar aos seus pais e experimentar em casa, sem necessariamente ter que realizar o cadastro no site. Essa atitude nos surpreendeu, considerando que era a primeira vez que eles entraram e contato com o programa.

Em comum acordo com as regentes dos alunos do terceiro ano, verificamos o planejamento das aulas regulares, a lista dos conteúdos já estudados e dos que ainda seriam para, assim, estabelecer com as professoras das turmas um processo de construção em conjunto e de colaboração para escolha dos conteúdos que seriam trabalhados por meio dos jogos ou que seriam necessários para a programação dos mesmos.

Seguindo a sugestão do laboratorista da escola e a pedido dos alunos, após cada aula do minicurso eu disponibilizei os slides com os exemplos do dia para serem publicados no site da escola, no ambiente online, na sessão tarefas da turma. Assim, criamos um ambiente virtual de aprendizagem organizado e promotor de um clima de aprendizagem, em sentido conforme exposto em Valente e Dias (2009). O ambiente da escola, costumeiramente utilizado pelos professores, é mostrado na Figura 14.

Figura 14: Site do colégio



Fonte: <http://www.escolasagrado.com.br/>

Na página do colégio é possível acessar, remotamente, os conteúdos disponíveis pela instituição. Ao acessar o ícone aluno on-line, cada estudante com seu *login* e senha tem acesso a outra página que lista as circulares, boletos e tarefas da turma. Assim, após a inserção dos slides do minicurso nesse ambiente, os alunos tiveram acesso remoto aos conteúdos estudados em cada dia.

Elaboramos o planejamento das aulas (Apêndice G) antes do início do minicurso e, posteriormente, o adaptamos de acordo com o andamento das atividades, levando em conta o tempo de cada aula, as características e aproveitamento dos alunos.

Todos os materiais utilizados nas aulas (apresentações, enunciados, demonstrações) foram adequados às características da faixa etária dos alunos, ou seja, o tipo de linguagem era simples e as explicações de conceitos eram curtas e os enunciados claros, de forma que todos os alunos pudessem acompanhar as aulas sem grandes dúvidas.

De acordo com as informações obtidas durante o período inicial de observação e levantamento dos interesses dos alunos, foi possível constatarmos que as duas turmas eram muito similares no que diz respeito ao comportamento, aproveitamento e interesse, o que facilitou a realização do minicurso quando as duas turmas participaram juntas das atividades.

3.3.4.1 Primeira aula

Preparamos a primeira aula considerando a avaliação e a reflexão sobre as observações feitas na primeira conversa com os alunos. Ressaltamos que fizemos alterações estratégicas ou metodológicas sempre que as consideramos necessárias ao longo da realização do minicurso.

Ministramos a primeira aula na terça-feira da terceira semana do mês de abril/2016, na qual participaram alunos das duas turmas de terceiro ano. Fizemos a aula neste formato com os objetivos de permitir aos alunos: (i) o primeiro contato com a plataforma do Scratch, e (ii) conhecerem e compreenderem funções básicas do software (abrir, salvar, novo, executar, depurar).

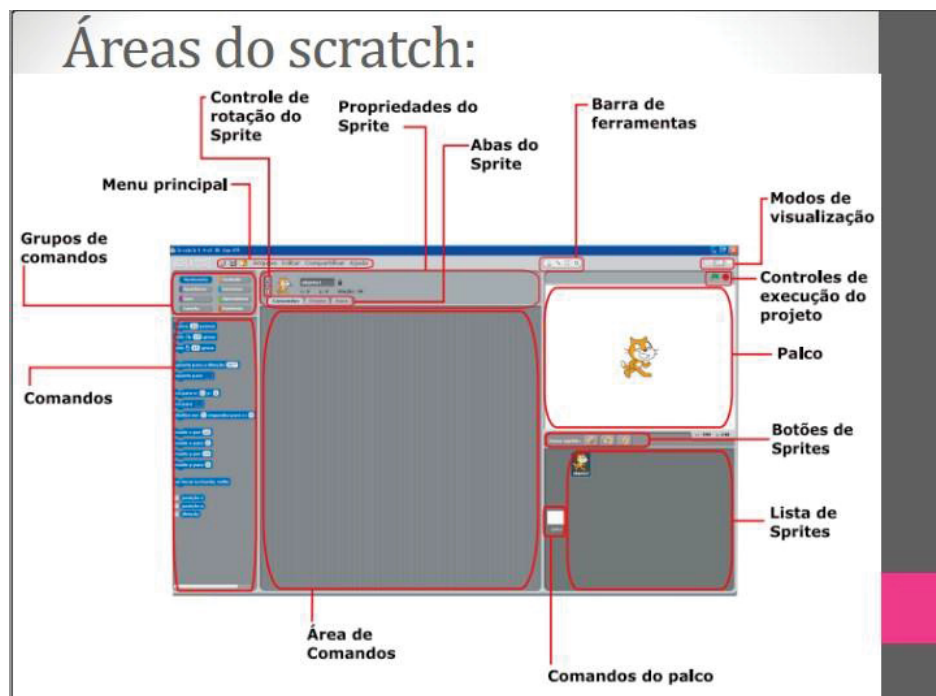
Inicialmente, apresentamos aos alunos os objetivos supracitados. Em seguida, planejamos uma apresentação de slides, como ilustramos as Figuras de 15 a 18, com o intuito de os alunos acompanharem os passos nela indicados, e, assim, concretizarem os objetivos da aula.

Figura 15: Aula 1 - Apresentação



Fonte: A autora.

Figura 16: Aula 1 – Área do Scratch

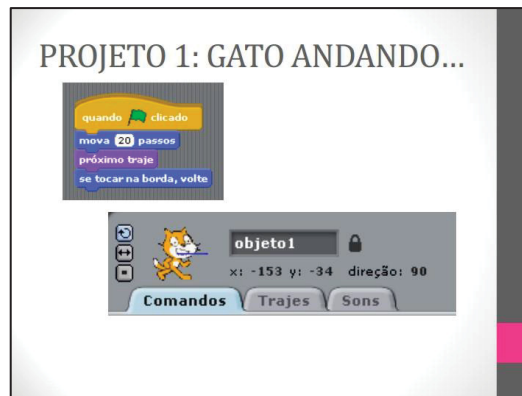


Fonte: A autora.

Nesta apresentação, fizemos uma breve explicação das potencialidades do *software*, sua história e mostramos a sua tela inicial com os nomes das áreas de trabalho. No final da aula, planejamos tempo livre para que os alunos explorassem o

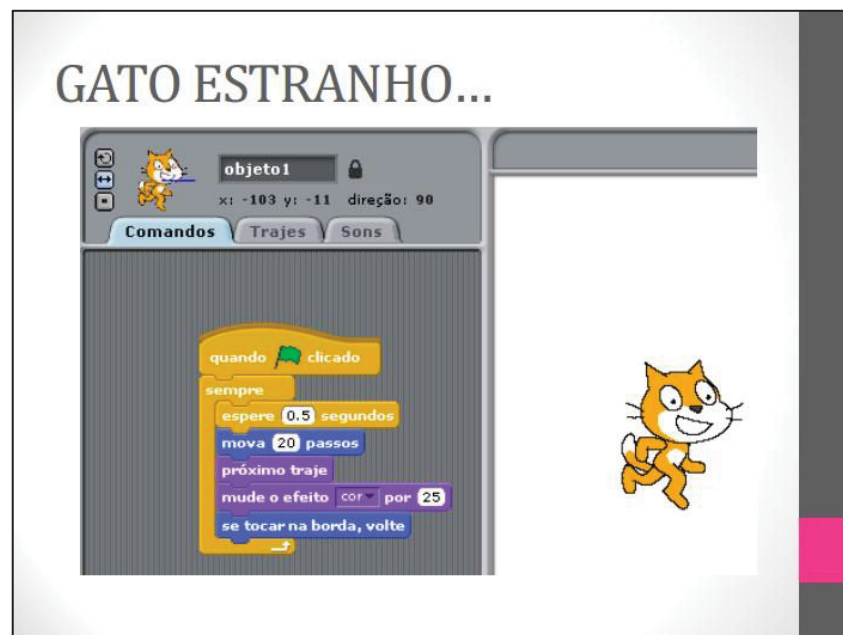
Scratch, o que foi importante para eles percebessem outros elementos, saciassem algumas curiosidades e iniciassem atividades autônomas e investigativas daquele espaço que desejavam conhecer.

Figura 17: Aula 1 – Explorando o Scratch



Fonte: A autora.

Figura 18: Aula 1 – Efeitos



Fonte: A autora.

Consideramos que a estratégia que foi implementada nesta aula funcionou da forma como a havíamos planejado. O apoio do laboratorista e das professoras regentes, que prestaram auxílio individualmente a cada aluno enquanto ministrávamos a aula, foi importante para o cumprimento dos objetivos propostos para aquele dia. Percebemos que é essencial contar com esse tipo de suporte para

conseguir manter o ritmo das aulas sem deixar de prestar assistência individual aos alunos que precisam, visto que nem todos acompanham as atividades da mesma forma.

Os alunos acompanharam o desenvolvimento da aula, reproduziram as programações apresentadas e, no término, criam suas próprias programações utilizando os conceitos compreendidos até aquele momento, com uso de outros *Sprites*.

3.3.4.2 Segunda aula

Como a estratégia utilizada na aula anterior obteve um bom resultado, optamos por seguir o mesmo padrão, porém com adequação ao conteúdo então trabalhado.

Nesta aula, ensinamos aos alunos algumas opções disponíveis na ferramenta Scratch. Como objetivos à aula, propomos: (i) apresentar noções básicas da ferramenta; (ii) identificar seus componentes e (iii) explorá-la.

Inicialmente, apresentamos tais objetivos da aula aos alunos. Em seguida, prosseguimos à apresentação dos conteúdos recorrendo a uma sequência de slides (Figuras 19 e 20) para que eles pudessem acompanhar nossa lógica. À medida que apresentávamos os slides com as explicações, os alunos utilizavam a plataforma do Scratch para localizarem e experimentarem aquilo que era apresentado.

Figura 19: Sprite e propriedades

Propriedades do sprite:

Nome do sprite

Permite ou não, mover o Sprite com o mouse quando o projeto estiver on-line.

Linha azul: Direção que o sprite está apontando.

Localização do sprite na tela. Coordenada x (horizontal) e coordenada y (vertical)

Direção que o sprite está apontando (em graus)


CONTROLES DE ROTAÇÃO DO SPRITE:

- O sprite pode girar livremente
- O sprite pode virar apenas na horizontal.
- O sprite não pode virar nem girar.

Fonte: A autora.

Figura 20: Grupos de comandos

Comandos e Grupos de Comandos:



Os comandos são agrupados por cor e tipo, ou seja, comandos que se referem a movimentos do Sprite, estão na cor azul, comandos que se referem aos controles estão na cor laranja, e assim por diante.

Alguns comandos tem campos de texto editável no interior, tais como:

Para alterar o valor, clique dentro da área branca e digite o número:

Fonte: A autora.

Depois de apresentar essas características do Scratch, explicamos aos alunos alguns conceitos básicos envolvidos com as atividades que faríamos. Assim, ressaltamos a eles o que eram programação, pensamento computacional e algoritmos (Figura 21). Damos ênfase à ideia de que programar é descrever passo a passo como o computador irá executar uma operação específica, ou seja, dar instruções ao computador para que ele realize alguma tarefa.

Figura 21: Conceitos Gerais

Conceitos gerais

- **Algoritmo:** É uma sequência de dados/informações/instruções bem definidas. É como uma receita de bolo que deve ser seguida.
- **Pensamento computacional:** Formado por uma sequência de ações – organização de dados e análise, construção de algoritmo, abstração, criação de modelos, simulação, automatização e paralelismo.
- **Programação:** Ação que coloca em prática o pensamento computacional.

Fonte: A autora.

Depois, voltamos ao modo de funcionamento e organização do Scratch para a aprendizagem dos primeiros comandos da ferramenta. Esses eram necessários para iniciar uma primeira programação do computador que faríamos: *quando clicado*, *diga por*, *diga* e *espere*, mostrados na Figura 22.

Figura 22: Comandos iniciais Scratch apresentados aos alunos



Fonte: A autora.

Como o Scratch funciona como blocos de arrastar que se juntam como se fossem peças de quebra-cabeças ou brinquedos do tipo lego, os alunos conseguiram realizar as atividades de montagem desse primeiro código de programação. Quando juntaram as peças e testaram a programação, viram o computador realizando a ação. Os alunos se mostraram bastante empolgados por terem conseguido realizar a atividade e iniciado seus papéis como programadores.

Propusemos para eles, então, a primeira atividade, que deveria ser feita individualmente baseada nos conceitos e comandos apresentados até aquele momento. A atividade consistia no problema apresentado na Figura 23.

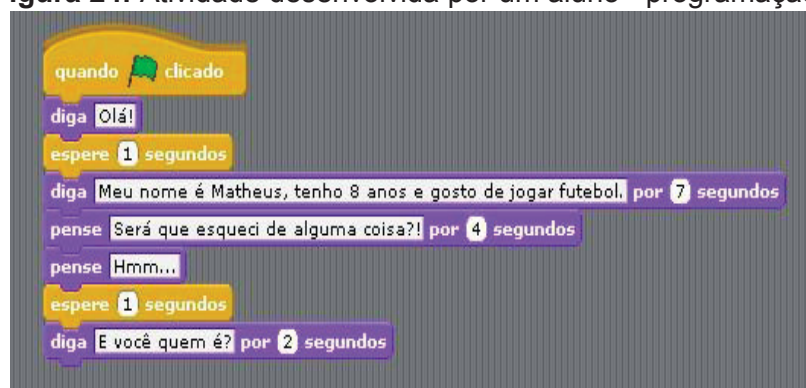
Figura 23: Atividade 1

Atividade

- Faça sua apresentação pessoal utilizando um Sprite (personagem) para te representar. Você deve informar nome, idade e interesses. A informações devem aparecer em balões de diferentes formatos.
- Salve em sua pasta com o nome **MeuSprite**.
- Se quiser acrescentar efeitos, fique a vontade.

Fonte: A autora.

Os alunos tiveram um tempo para pensar sobre a atividade, planejar, montar e executar seus códigos de programação. Tudo foi feito na plataforma do Scratch. Tanto nós quanto o laboratorista e as professoras regentes prestamos apoio individual aos alunos que nos pediam. Alguns alunos me surpreenderam ao escreverem códigos com funções além das apresentadas. Na figura 24, mostramos um exemplo de um desses códigos, no qual o aluno incluiu a função *pense* que, provavelmente, encontrou na lista de ferramentas do programa e decidiu utilizar. Já na Figura 25, ilustramos cenário que foi criado por um outro aluno, que utilizou elementos de programação que não tinham sido apresentados e de uma forma bastante rica e criativa.

Figura 24: Atividade desenvolvida por um aluno - programação

Fonte: A autora.

Figura 25: Atividade desenvolvida por um aluno - cenário

Fonte: A autora.

Vimos, assim, como alunos conseguiram ir além do proposto. Adicionaram outros elementos e até desafiaram seus colegas a fazerem diferente, o que tornou a aula mais dinâmica. Consideramos que a estratégia desta aula funcionou adequadamente. Os alunos prestaram atenção, participaram ativamente da aula e desenvolveram a atividade proposta. Tal participação resultou em produções além do esperado. Notamos que a maioria dos alunos demonstrou entusiasmo e vontade de aprender mais. Igualmente, percebemos que igual número, no tempo em que exploravam a ferramenta para realização da atividade, aproveitou para navegar em opções que eu ainda não havia sido apresentadas. Por exemplo, ao testar novas ferramentas e comandos e perguntar o que eles faziam, essas ocorrências nos indicam a forma autônoma como os alunos começavam a lidar com aquele ambiente.

Novamente, destacamos a importância do apoio individual prestado aos alunos ao longo da aula, tanto para ajudar como para incentivar e valorizar o que estava sendo feito como também para dar suporte aos alunos que têm dificuldades específicas, principalmente àquelas ligadas ao uso do computador. A título de exemplificação, (i) digitação no teclado, pois não podemos esquecer que são alunos

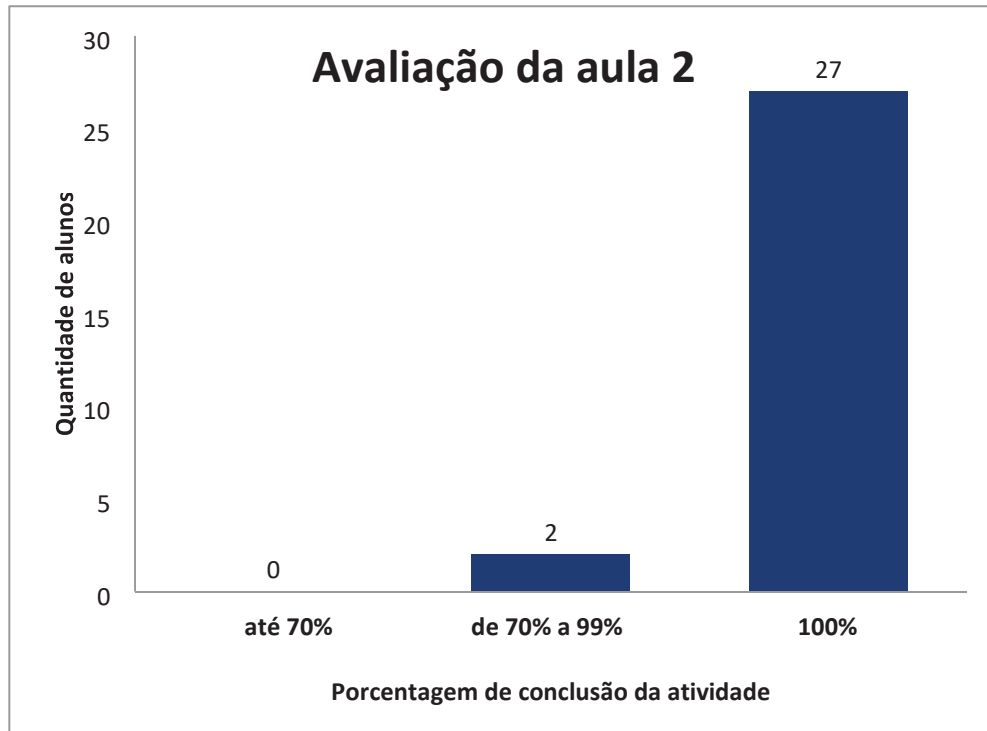
com idade média de 8 anos em fase de alfabetização e; (ii) alguns alunos desconhecem como usar a tecnologia disponível no computador.

Em razão da atividade proposta ter sido simples – e não poderia ser diferente já que era a primeira atividade –, ela envolveu conceitos computacionais bastante iniciais. Contudo, acreditamos que apesar do baixo nível de complexidade ela permitiu aos alunos entenderem alguns aspectos centrais do programa, a saber: (i) as formas de escrita do código de programação e (ii) a ideia dos blocos e dos encaixes, em que importa a sequência para a execução dos comandos, ou seja, para a elaboração de instruções de programação que façam com que o computador execute exatamente aquilo que esperamos. Pudemos perceber nesta primeira da atividade como os alunos se envolveram com as propostas de escrita, teste e ajuste dos códigos de programação. Além disso, eles compartilhavam o que faziam com os colegas, tiravam dúvidas entre si e questionavam as funcionalidades da ferramenta.

Os alunos elaboraram suas próprias apresentações, nas quais utilizaram componentes importantes da programação e do desenvolvimento do pensamento computacional. Nesta atividade, de forma implícita, os conteúdos matemáticos que estão presentes são: Ordenação, composição, reta numérica, operações (adição, subtração, multiplicação e divisão), problemas e sequências.

Para avaliação das atividades foi elaborada uma grade de avaliação (Apêndice H), para a qual nos baseamos em Brennan e Resnick (2012). Mostramos o resultado da avaliação da primeira atividade da segunda aula no Gráfico 7, apontando que a maioria dos alunos concluiu a atividade que foi proposta.

Gráfico 7: Atividade 1 – Aula 2

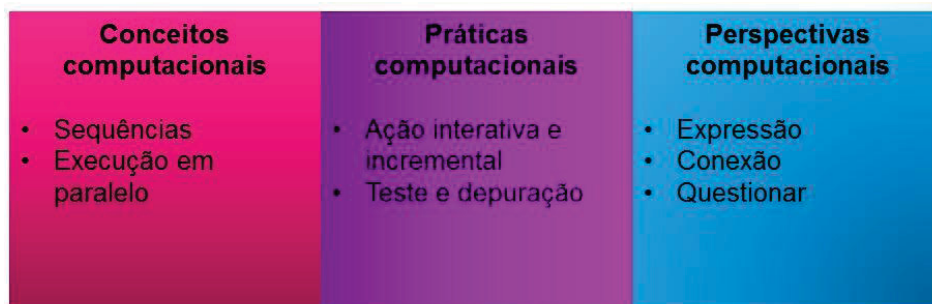


Fonte: A autora.

Examinamos as atividades desenvolvidas pelos alunos conforme as variáveis de avaliação sugeridas por Brennan e Resnick (2012). Dessa forma, a partir da análise de algumas atividades / projetos realizados pelos alunos e da observação feita durante a aula, nos foi possível detectar as três dimensões do pensamento computacional, a saber: (i) conceitos computacionais, (ii) práticas computacionais e (iii) perspectivas computacionais.

Na Figura 26, apresentamos um esquema das dimensões do pensamento computacional que estiveram envolvidas na atividade realizada.

Figura 26: Resumo das dimensões do Pensamento Computacional presentes na atividade anterior de acordo.



Fonte: Adaptado de Brennan e Resnick (2012).

3.3.4.3 Terceira aula

Para preparar a terceira aula, levamos em consideração o resultado das nossas observações e avaliações da aula anterior. Dentre elas, consideramos tanto os aspectos positivos quanto os negativos. Nossa reflexão sobre o trabalho realizado na segunda aula nos levou a alterar a estratégia de apresentação e explicação dos conteúdos aos alunos. Nesta terceira aula, incluímos momentos expositivos no início da aula porque percebemos que explicar os conteúdos enquanto os alunos estavam desenvolvendo as atividades no ambiente do Scratch causava-lhes uma ansiedade que consideramos difícil de conter.

Por um lado, a ansiedade demonstrada pelos alunos foi um ponto positivo relativo à sua interação com o software, visto o que eles queriam era explorar e conhecer toda a ferramenta para descobrirem o que poderiam construir com ela. No entanto, tal momento dificultou a organização da aula para o grupo. A atividade que propomos para esta aula foi mais complexa. Desse modo, construímos com os alunos, passo a passo, um exemplo daquilo que estávamos propondo. Em seguida, cada um deles teria um tempo para executar seu projeto de forma autônoma. Outro diagnóstico que fizemos na aula anterior, constatamos que muitos alunos terminaram a tarefa mais cedo do que o tempo que havíamos previsto em nosso planejamento. Assim, decidimos acrescentar à tarefa desta terceira aula um desafio extra com um grau de complexidade mais elevado. Nosso objetivo era desafiar os alunos que quisessem avançar ainda mais. Além disso, adicionamos outro elemento para a aula, a saber: a demonstração da tarefa resolvida, sem a visualização dos blocos de programação.

Assim, para a terceira aula propomos os seguintes objetivos: (i) adicionar movimento aos *sprites*, (ii) adicionar sons aos *sprites*, (iii) editar trajes e (iv) conhecer o conceito de coordenadas. Iniciamos a aula com a apresentação dos objetivos do dia e da proposta de atividade aos alunos. Em seguida, ilustramos o que deles esperaríamos, o que foi importante para que percebessem qual era o resultado final almejado. Nas figuras de 27 a 29, mostramos a proposta de projeto DJ Maluco, que apresentamos como exemplo, em que são mostramos o resultado final pretendido e o código de programação envolvido.

Figura 27: Cenário e personagem – Dj Maluco



Fonte: A autora.

Figura 28: Programção – Dj Maluco



Fonte: A autora.

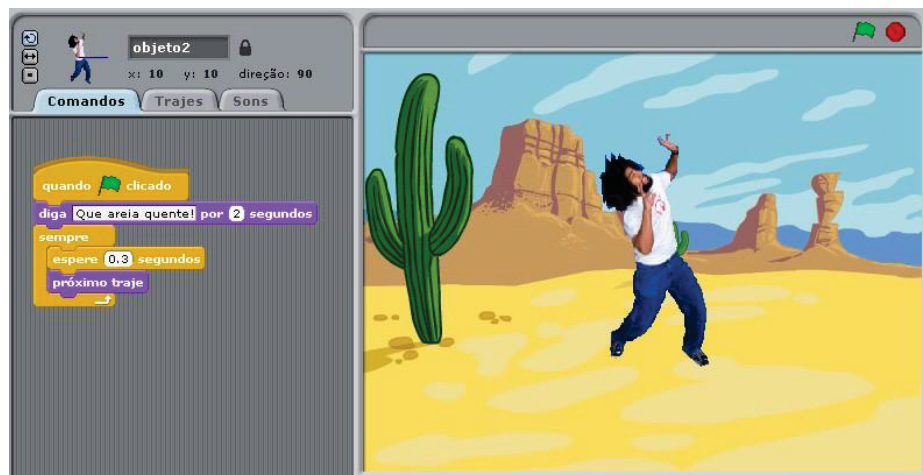
Figura 29: Programação – Cenário Dj Maluco



Fonte: A autora.

Depois de nossa apresentação do exemplo, os alunos iniciaram a construção de seus próprios projetos. A atividade foi realizada individualmente. Nós e o laboratorista auxiliamos aos alunos sempre que eles nos solicitavam. Na Figura 30, apresentamos o projeto desenvolvido por um aluno, baseado no exemplo.

Figura 30: Dançando na areia – atividade de um aluno



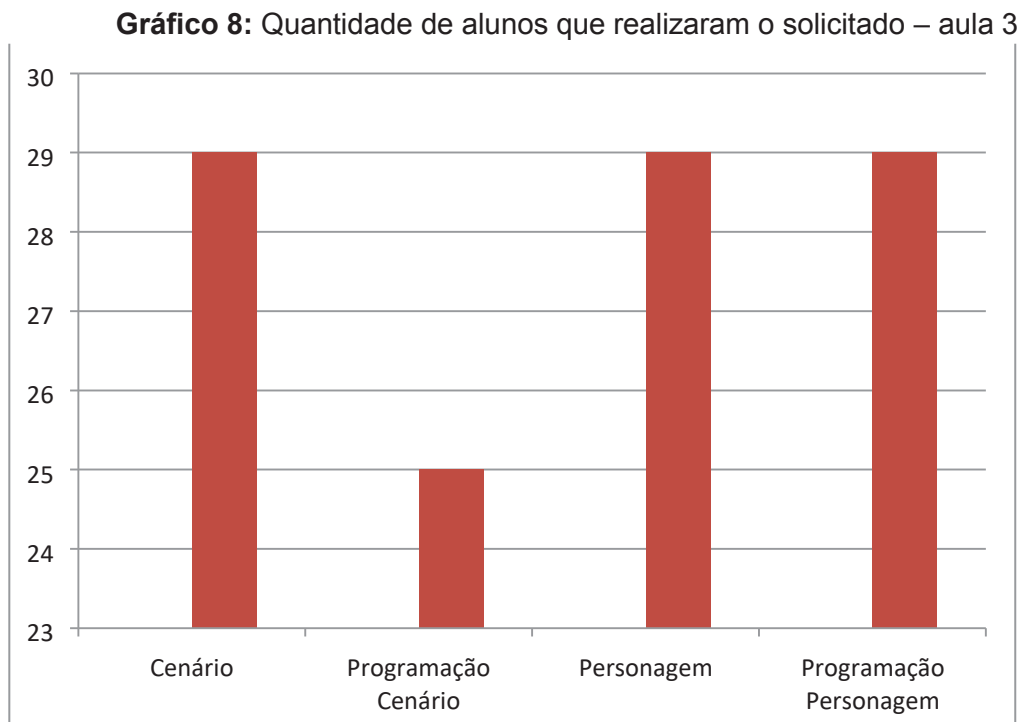
Fonte: A autora.

Os alunos concluíram a tarefa no tempo previsto. Observando a forma como trabalharam, percebemos um maior grau de atenção e motivação para o cumprimento dos objetivos da aula em relação à aula anterior. Concluímos que foram válidas as mudanças de estratégia que realizamos. O método de demonstração prévia da tarefa resolvida acompanhada com a explicação oral, permitiu que os alunos entendessem de forma mais clara os objetivos e a concluíssem com sucesso.

Nesta atividade, os alunos precisavam e conseguiram programar os sprite e o palco ao mesmo tempo, além da necessidade da inserção de som. Os conteúdos matemáticos abordados nesta atividade foram: Ordenação, composição, reta numérica, operações (adição, subtração, multiplicação e divisão), problemas, seqüências e medidas de tempo.

Para a aula seguinte, procuramos reduzir o tempo de exposição de conteúdos para dar mais oportunidades aos alunos para aprenderem a aprender autonomamente e aprenderem uns com os outros (PAPERT, 1993). Com a avaliação da atividade com base nos conceitos computacionais de Brennan e Resnick (2012), nos foi possível

recolher mais evidências importantes para a investigação, as quais mostramos no Gráfico 8.



Fonte: A autora.

Nessa tarefa, os conceitos, práticas e perspectivas computacionais em evidência foram as mesmas da aula anterior, o que é mostrado na Figura 31.

Figura 31: Resumo das dimensões do Pensamento Computacional presentes na atividade anterior.



Fonte: Adaptado de Brennan e Resnick (2012).

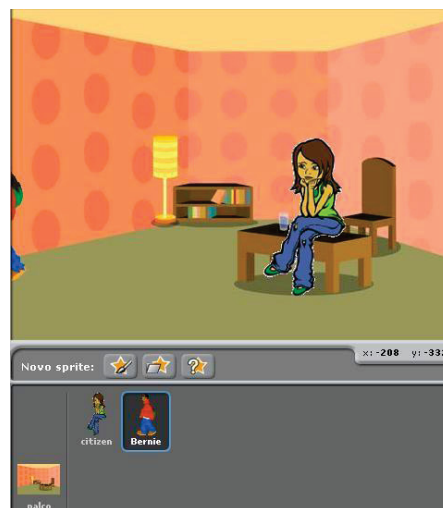
3.3.4.4 Quarta aula

Por considerarmos positivos os resultados decorrentes da estratégia empregada na aula anterior, optamos por manter o momento expositivo no início da aula e o incentivo pelo desafio extra na tarefa.

No início da aula, apresentamos aos alunos os seguintes objetivos: (i) adicionar movimento aos *sprites*, (ii) adicionar sons aos *sprites*, (iii) editar trajés, (iv) aplicar o conceito de coordenadas, e (v) conhecer e aplicar conceitos de direção e ângulos.

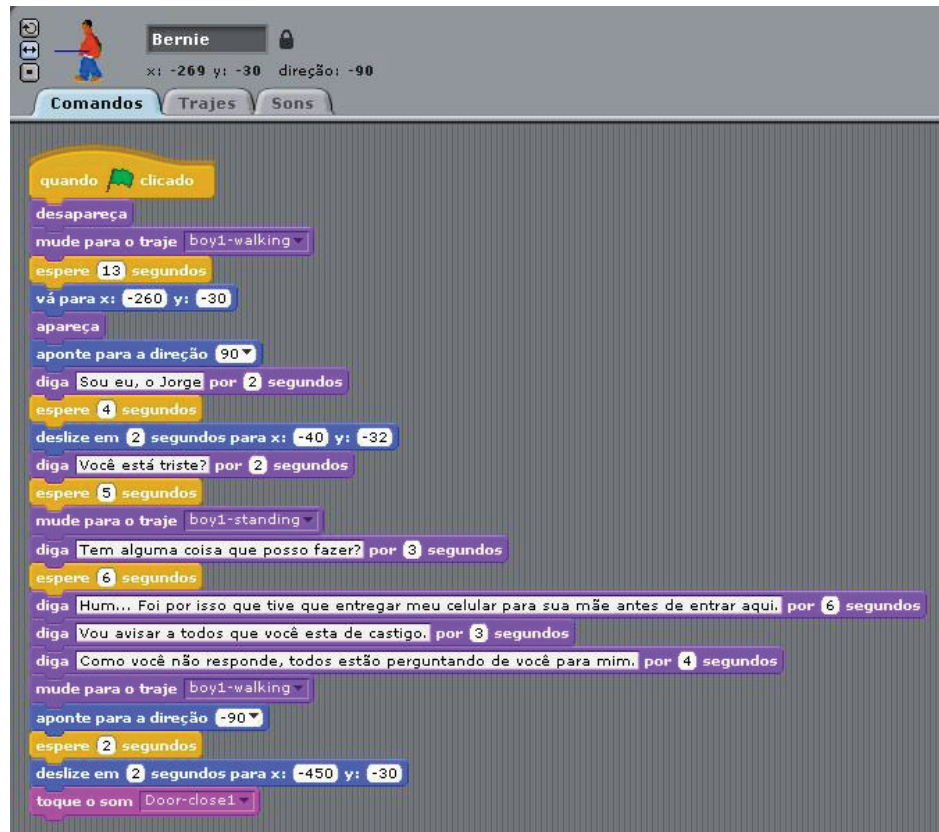
Apresentamos conteúdos aos alunos no início da aula. Em seguida, passamos a exemplificação da atividade (como mostramos nas Figuras 32, 33, 34, 35 e 36) da tarefa e do desafio proposto.

Figura 32: Cenário e Personagens - 1



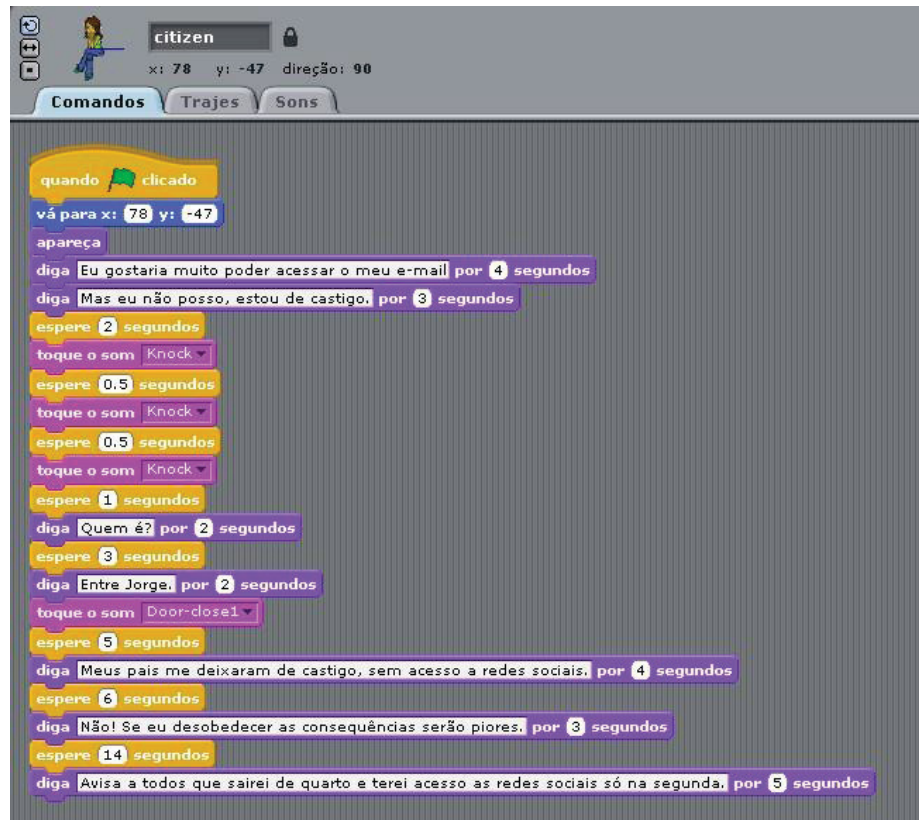
Fonte: A autora.

Figura 33: Programação 1



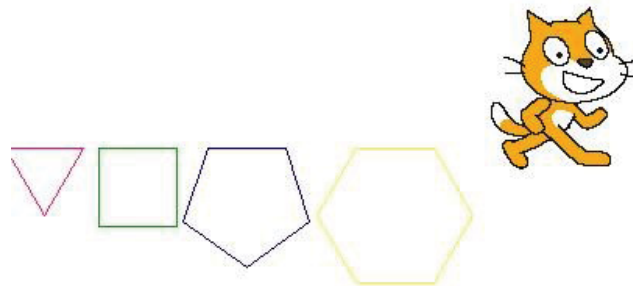
Fonte: A autora.

Figura 34: Programação 2



Fonte: A autora.

Figura 35: Figuras planas – Atividade 2



Fonte: A autora.

Figura 36: Programação



Fonte: A autora.

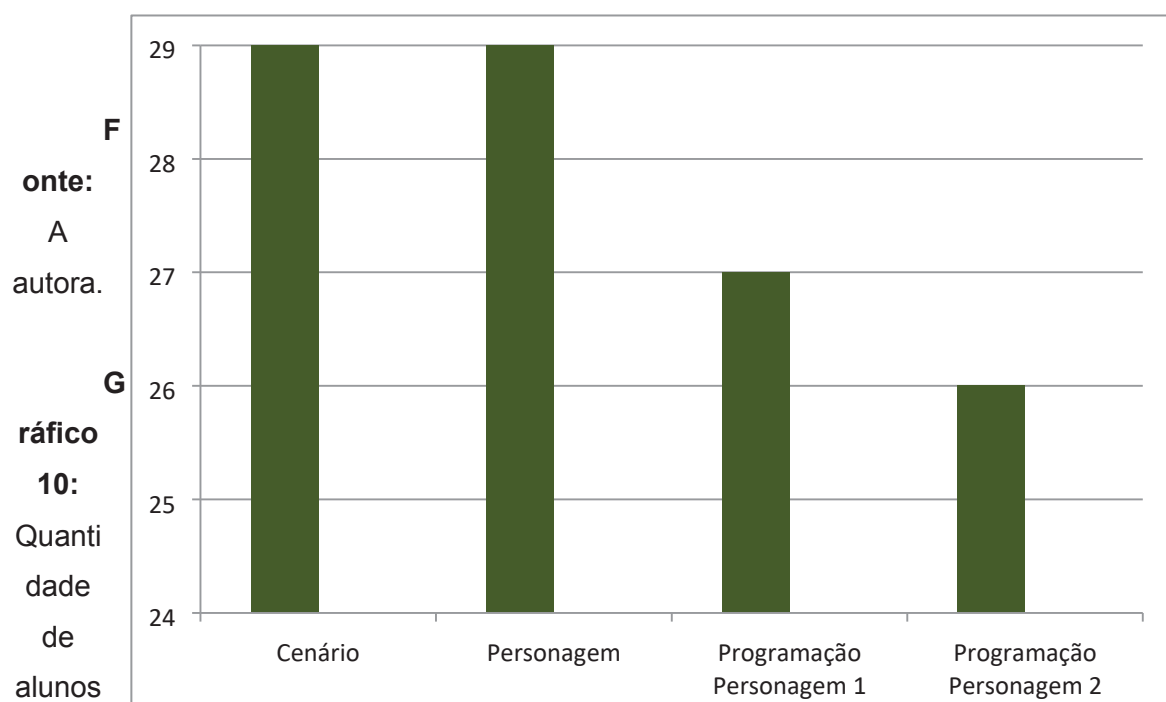
A aula transcorreu como a havíamos planejado. Os alunos participaram ativamente, produzindo seus projetos, questionando e demonstrando interesse pela aula e pelos conteúdos propostos.

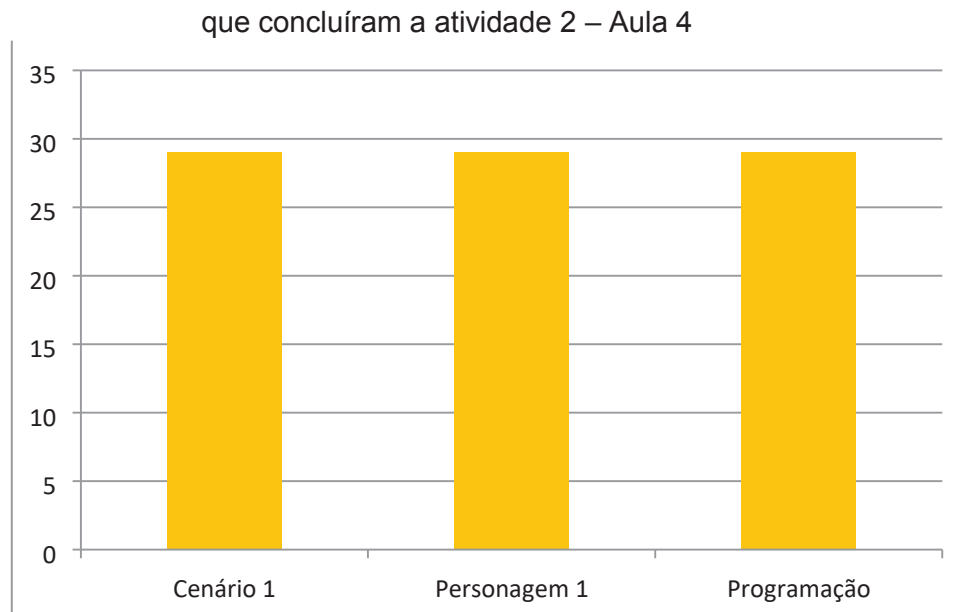
Apesar de termos aumentado o nível de complexidade dos conteúdos de programação envolvidos, percebemos que os alunos analisaram e resolveram os problemas por partes, encontrando as melhores soluções pessoais para conclusão da atividade. Alguns alunos resolveram o desafio extra. Novamente, a proposta do desafio mostrou-se importante tanto para estimular os alunos a continuarem participando da aula quanto aos colegas que realizavam suas tarefas

Durante a aula, os alunos tinham que planejar e desenvolver um diálogo mais extenso entre dois colegas, em que exigíamos movimentação dos personagens e cronometragem dos tempos para aparição e ocultação dos personagens. Na sequência, precisaram desenhar figuras planas, mas sendo necessário um espaço entre elas, para que não ficassem uma sobre a outra, além da complexidade na alteração da quantidade de lados das figuras planas. Nesta aula, os conteúdos matemáticos abordados de maneira implícita e explícita foram: Ordenação, composição, reta numérica, operações (adição, subtração, multiplicação e divisão), problemas, sequências, congruência, figuras geométricas planas, perímetro, área e medidas de tempo.

Mediante a análise dos resultados da atividade, os quais indicamos no Gráfico 9, conseguimos verificar que os alunos apresentaram resultados satisfatórios em relação ao que estávamos propondo a eles.

Gráfico 9: Quantidade de alunos que concluíram a atividade 1 – Aula 4

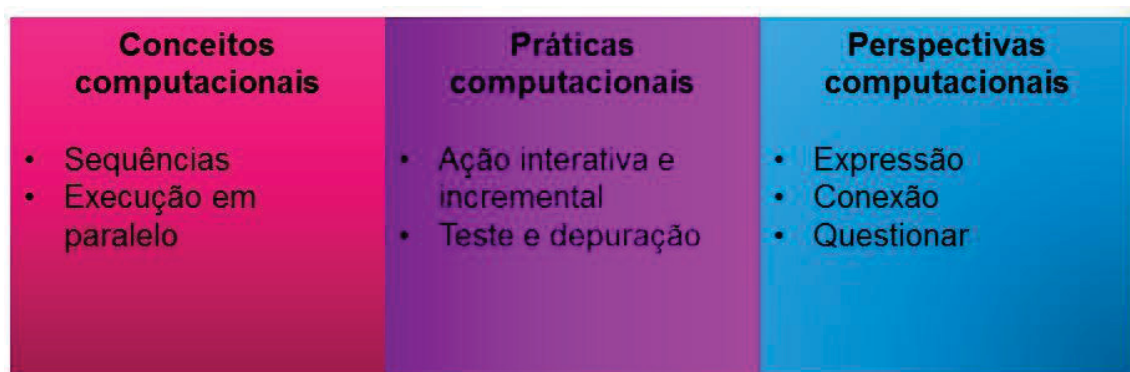




Fonte: A autora.

A partir da nossa observação, percebemos que os conceitos, práticas e perspectivas computacionais foram perceptíveis nesta tarefa. Identificamos as várias etapas do pensamento computacional e também as operações matemáticas e lógicas (BRENNAN; RESNICK, 2012).

Figura 37: Resumo das dimensões do Pensamento Computacional presentes na atividade anterior.



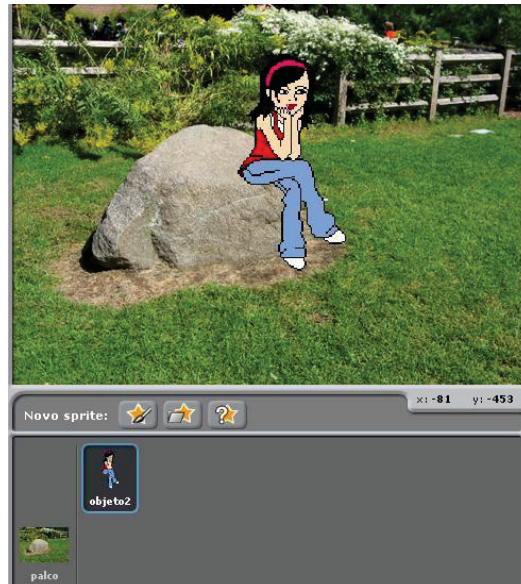
Fonte: Adaptado de Brennan e Resnick (2012).

3.3.4.5 Quinta aula

Realizamos a aula com a mesma metodologia da aula anterior e tivemos os seguintes objetivos: (i) adicionar comandos de repetição e condição, (ii) utilizar operadores e (iii) input de informação. Nas Figuras 38 e 39 ilustramos o exemplo do

projeto que solicitamos na aula, apresentado aos alunos juntamente com a explicação dos conteúdos envolvidos.

Figura 38: Cenário e personagem da aula



Fonte: A autora.

Figura 39: Comandos da aula



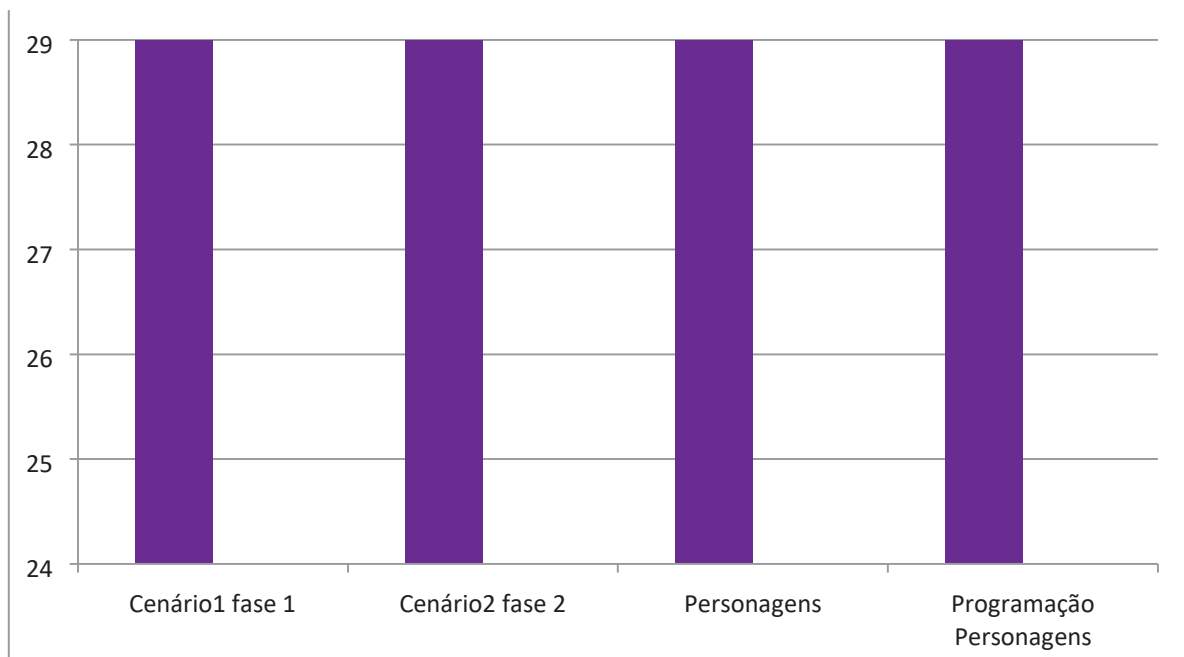
Fonte: A autora.

Após avaliarmos a aula, consideramos que foi adequada a estratégia de ensino que utilizamos, pois percebemos que os alunos participaram das aulas com atenção e interesse.

Os alunos, de imediato, acharam fácil a atividade, mas logo perceberam que era necessário pensar como fazer a comparação e a inserção dos valores, o que trouxe um grau de dificuldade maior. Os conteúdos matemáticos abordados nesta atividade foram: Ordenação, composição, reta numérica, problemas e sequências.

Ainda que a atividade/desafio que propusemos nesta aula apresentassem um maior grau de dificuldade se comparado com as aulas anteriores, os alunos obtiveram bons resultados finais, conforme evidenciados no Gráfico 11.

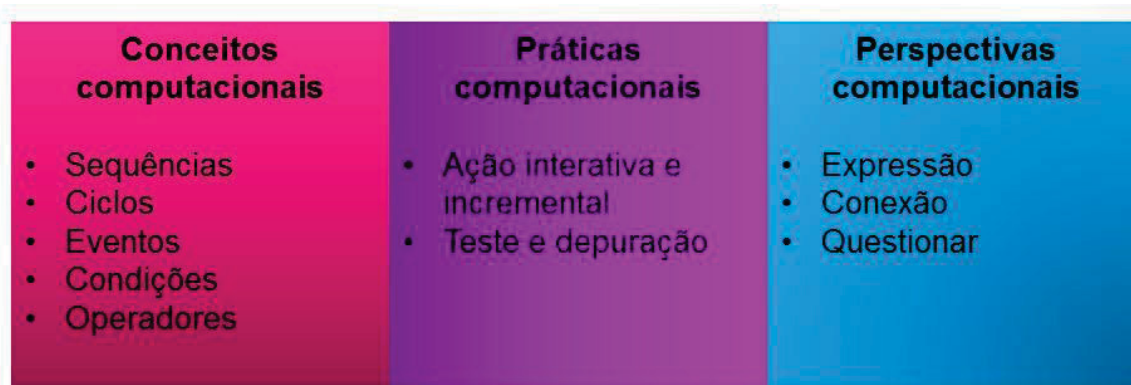
Gráfico 11: Atividade maior e menor – Aula 5



Fonte: A autora.

Identificadas as várias etapas do pensamento computacional e também as operações matemáticas e lógicas (BRENNAN; RESNICK, 2012), assim como os conceitos, práticas e perspectivas computacionais.

Figura 40: Resumo das dimensões do Pensamento Computacional presentes na atividade anterior.



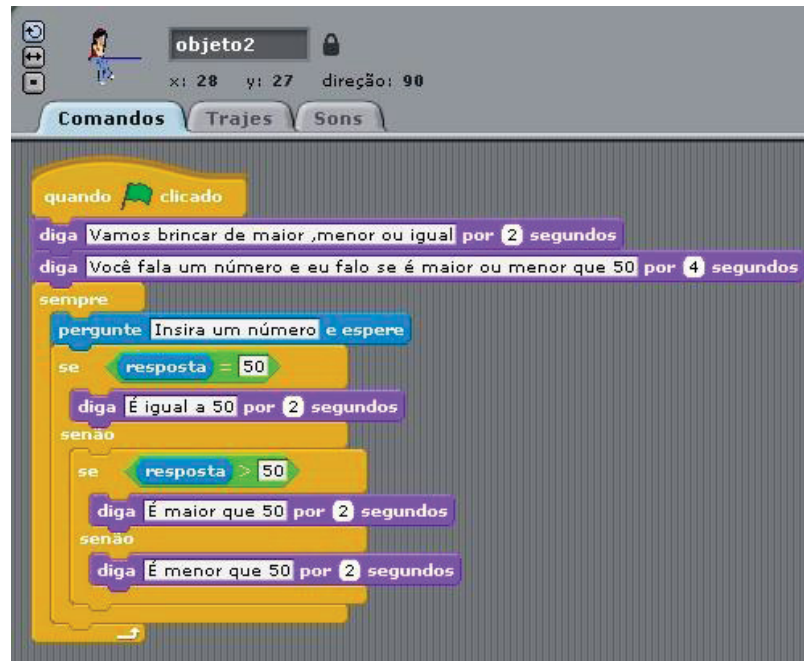
Fonte: Adaptado de Brennan e Resnick (2012).

3.3.4.6 Sexta aula

Assim como nas anteriores, também a iniciamos apresentando aos alunos os objetivos da aula: (i) adicionar comandos de repetição e condição; (ii) utilizar operadores; (iii) input de informação; e (iv) fazer uso de variáveis. Em seguida, esclarecemos sobre a aplicação dos conteúdos e do exemplo da atividade que eles realizariam (Figura 41), além da apresentação do desafio.

Pedimos aos alunos que abrissem o arquivo do projeto no qual estavam trabalhando na aula anterior (sobre o assunto “maior do que” e “menor do que”). Aproveitando o que já tinham feito, eles deveriam elaborar programações para comparações de igualdade, “maior do que” e “menor do que” entre os números. Os conteúdos matemáticos abordados nesta atividade foram: Ordenação, composição, reta numérica, operações (adição, subtração, multiplicação e divisão), problemas, sequências e medidas de tempo, os mesmos da aula anterior, porém com um grau maior de complexidade.

Figura 41: Exemplo utilizado na aula

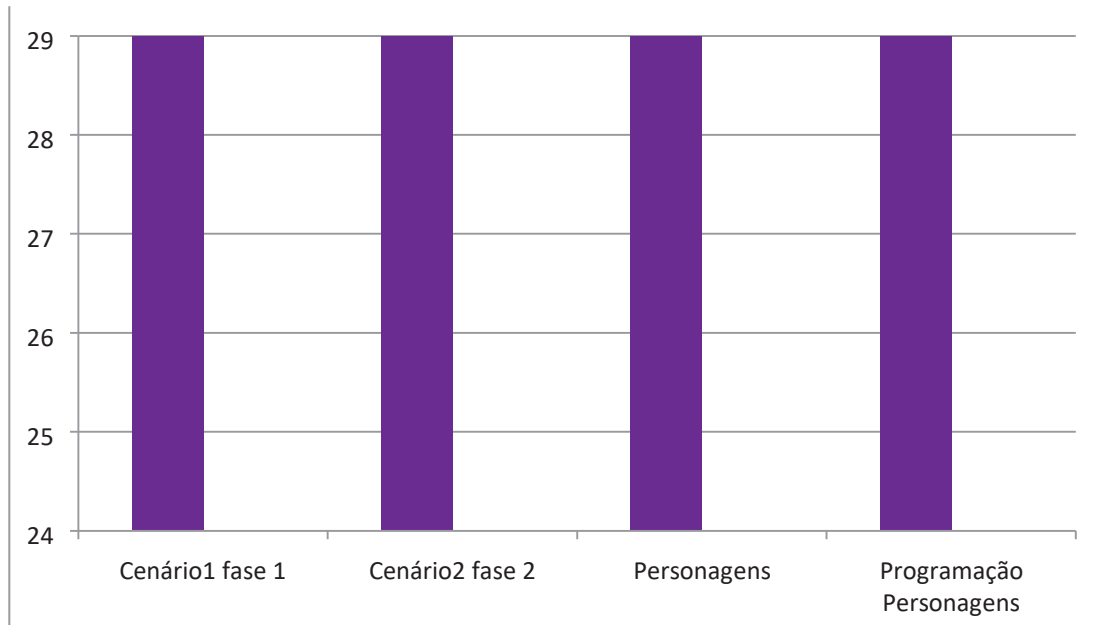


Fonte: A autora.

Avaliamos que os alunos participaram ativamente da aula e que ficaram empolgados com as atividades realizadas. A atividade/desafio reuniu um grau de dificuldade bem maior que as aulas anteriores e os alunos conseguiram atingir os objetivos que propusemos. A atividade foi relevante em virtude de ter possibilitado o trabalho com conteúdos do cotidiano deles. Ademais, eles começaram a compreender que é possível implementar programação dentro de programação, ou seja, que é possível ampliar, em etapas, códigos desenvolvidos anteriormente.

No apuramento dos resultados parciais da atividade, verificamos que os resultados apresentados pelos alunos foram coerentes com o tempo da aula e seu domínio da ferramenta.

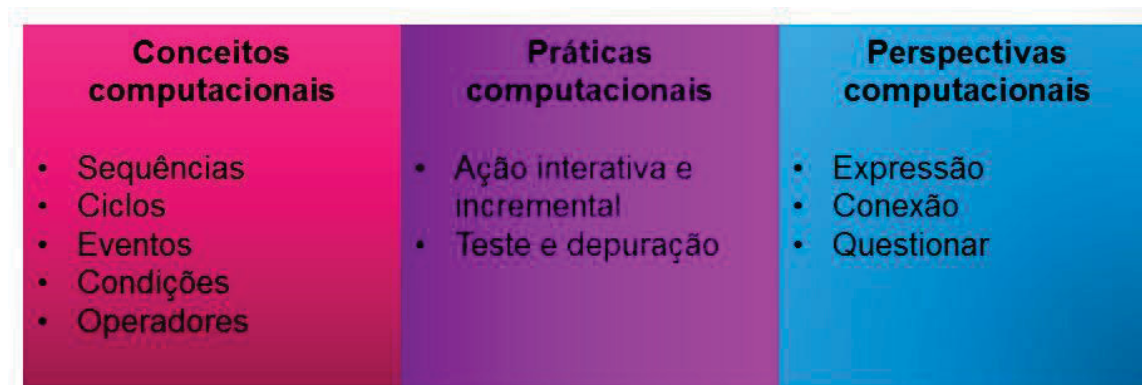
Gráfico 12: Atividade maior, menor e igual – Aula 06



Fonte: A autora.

A lógica, as operações matemáticas e as diversas etapas do pensamento computacional foram observadas no decorrer da atividade (BRENNAN; RESNICK, 2012), assim como os conceitos, práticas e perspectivas computacionais.

Figura 42: Resumo das dimensões do Pensamento Computacional presentes na atividade anterior.



Fonte: Adaptado de Brennan e Resnick (2012).

3.3.4.7 Sétima aula

Na sétima aula, tivemos como objetivos: (i) adicionar comandos de repetição e condição, (ii) utilizar operadores, (iii) input de informação, e (iv) fazer uso de variáveis. Os procedimentos adotados nas aulas anteriores foram aqui mantidos.

A proposta da aula foi utilizar a programação para realizar a multiplicação de dois números aleatórios. Apresentamos os conteúdos e as orientações para o desenvolvimento aos alunos no início da aula por meio de uma sequência de slides, na qual incluímos o exemplo da atividade/desafio proposto (Figuras 43, 44 e 45).

Figura 43: Multiplicação de dois números



Fonte: A autora.

Figura 44: Multiplicação de dois números com verificação de acertos



Fonte: A autora.

Figura 45: Multiplicação de dois números com contador

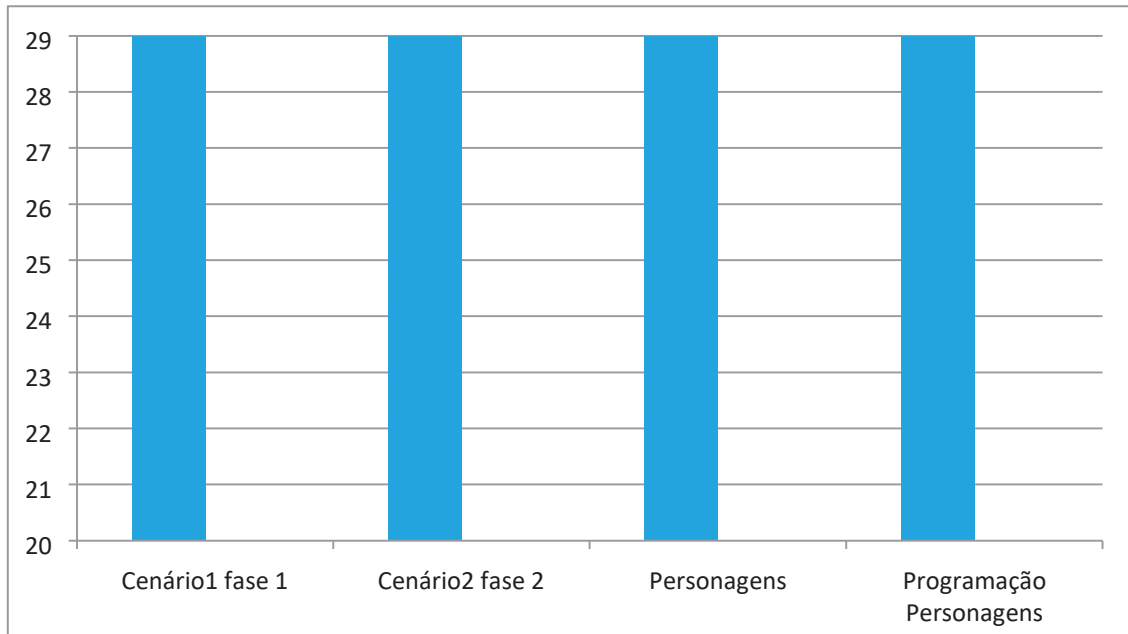


Fonte: A autora.

Esta atividade envolveu conteúdos com nível de dificuldade maior do que as anteriores. Neste dia, os alunos conseguiram atingir os objetivos, porém conseguiram finalizar a atividade somente com a ajuda do laboratorista e da professora regente. Entendemos que a atividade foi relevante em virtude de ter trabalhado com outros conteúdos do cotidiano dos alunos, que começaram a compreender como seria complexa a elaboração de um jogo/game e/ou objeto de aprendizagem.

Na conferência dos resultados da atividade, pudemos verificar que os alunos obtiveram efeitos coerentes com o grau de dificuldade da atividade, tempo de aula e domínio individual da ferramenta. Nesta proposta, os alunos precisavam pensar e agir como se estivessem em uma gincana, em que o organizador sorteia dois valores (computador/o programa) e ele precisa agir como uma “calculadora”, onde se digita o resultado da multiplicação das parcelas. Caso o valor digitado pelo usuário fosse correto, ele ganharia dois pontos e, se fosse errado, perderia um. Os alunos definiram que as parcelas das multiplicações sorteadas seriam valores numéricos inteiros entre zero e dez. Os conteúdos matemáticos nesta atividade foram: Ordenação, composição, operações, problemas, sequências, Coleta, classificação, interpretação dos dados e espaço amostral.

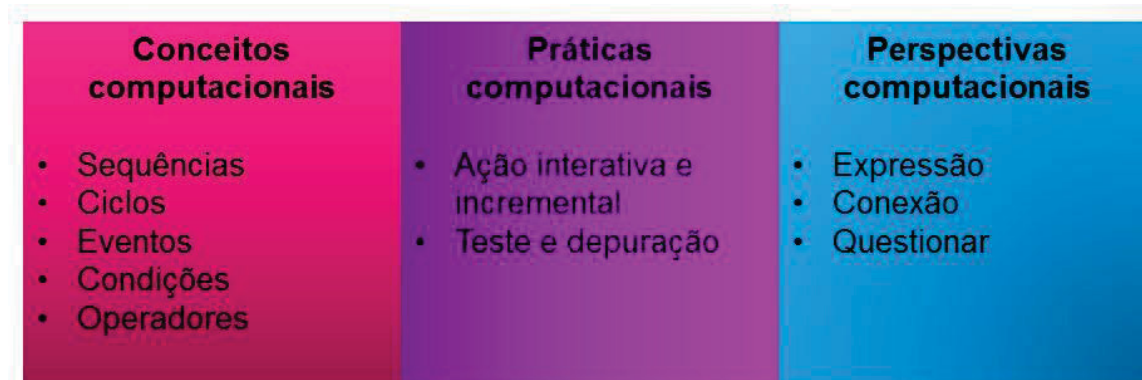
Gráfico 13: Desempenho dos alunos na atividade



Fonte: A autora.

A lógica, as operações matemáticas e as diversas etapas do pensamento computacional puderam ser observadas no decorrer da atividade (BRENNAN; RESNICK, 2012), assim como os conceitos, práticas e perspectivas computacionais.

Figura 46: Resumo das dimensões do Pensamento Computacional presentes na atividade anterior.



Fonte: Adaptado de Brennan e Resnick (2012).

3.3.4.8 Oitava e nona aulas

Solicitamos aos alunos que abrissem o projeto desenvolvido na aula anterior (multiplicação de dois números aleatórios) e que, baseados nele, elaborassem programações para as outras operações de subtração e de adição. Cada programação deveria ser feita para um *Sprite* em um cenário diferente. Apresentamos,

no início da aula, os conteúdos e dicas para a produção, bem como um exemplo da atividade/desafio proposto.

Ao dar continuidade na aula anterior, a proposta que fizemos para esta aula a foi dar sequência na realização da programação para conectar os palcos. Assim, tornar as programações antes soltas/isoladas em um jogo de quatro fases, ocultando e mostrando palcos e *Sprites*, de acordo com a pontuação do jogador. O fato de acrescentar um contador para marcar os pontos de acordo com erros e acertos, estimulou os alunos e começaram a aula sem compreender muito bem as etapas.

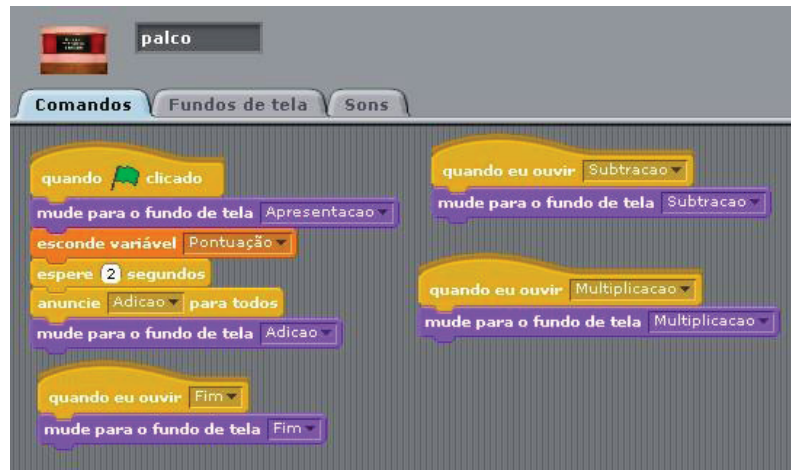
Figura 47: Projeto – As operações matemáticas



Fonte: A autora.

O primeiro passo que fizemos foi criar todos os cenários e inserir todos os personagens. Em seguida, definimos a sequência e programamos a abertura.

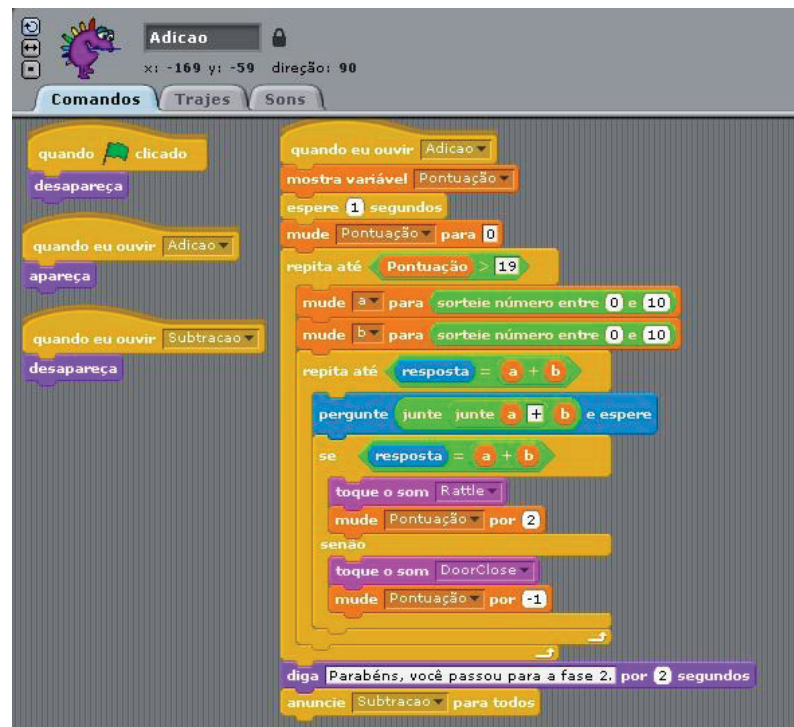
Figura 48: Abertura



Fonte: A autora.

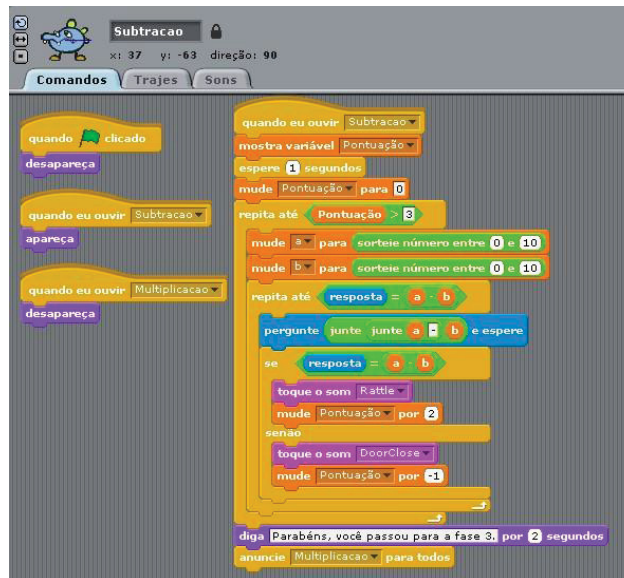
Como próximo passo, renomeamos os personagens para cada uma das operações que seriam responsáveis, para não nos perdermos nas programações. Passamos, então, à programação de cada personagem.

Figura 49: Personagem adição



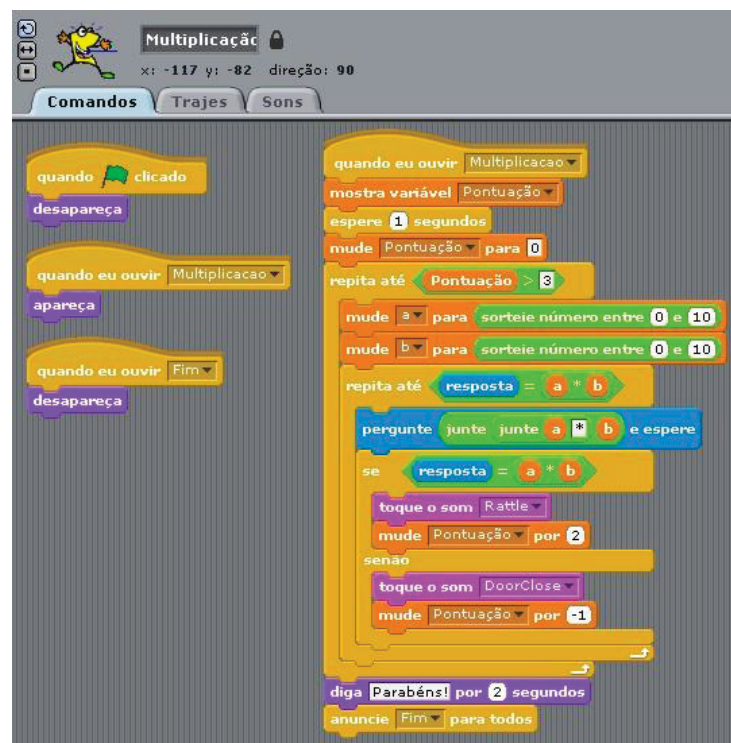
Fonte: A autora.

Figura 50: Personagem Subtração



Fonte: A autora.

Figura 51: Personagem Multiplicação



Fonte: A autora.

Os alunos ficaram empolgados, o que nos demonstrou que as estratégias utilizadas nas aulas funcionaram.

O grau de dificuldade da atividade/desafio foi maior que nas aulas anteriores, pois era necessário trabalhar em diferentes palcos/cenários e com vários personagens. Os alunos conseguiram atingir os objetivos, finalizando o teste e as alterações de ajustes de pontuação. Entendemos que a atividade foi significativa para

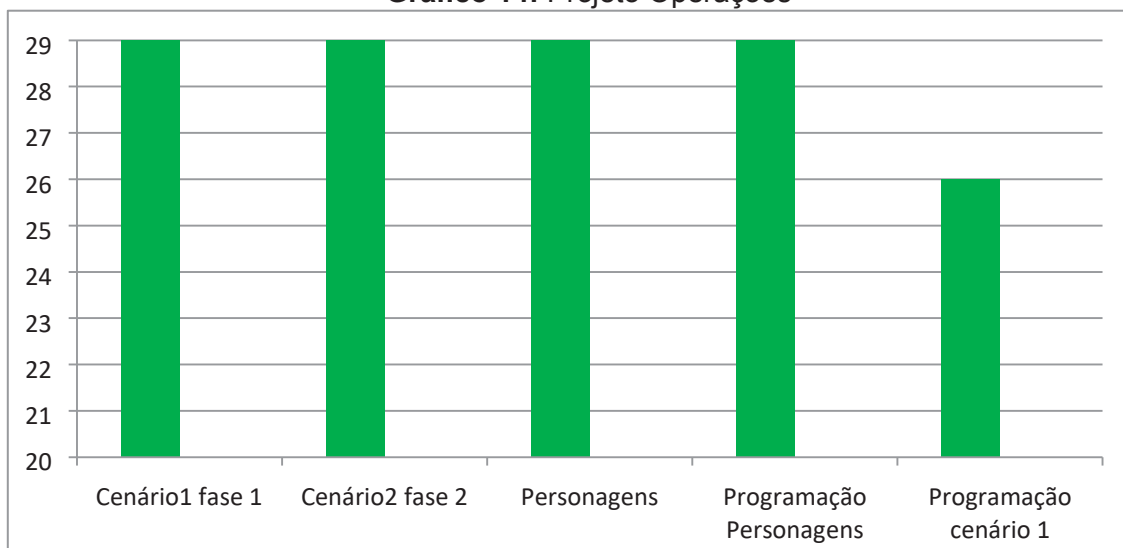
eles por trabalhar com conteúdos da rotina, além de fazê-los compreender a organização e complexidade na elaboração de um jogo e ou objeto de aprendizagem.

A complexidade da programação não aumentou, mas com relação a organização e apresentação, sim. Os alunos precisaram criar cenários de apresentação geral do jogo e, depois, de cada operação. Na apresentação geral, tiveram que colocar *link* para cada cenário das operações. No que diz respeito ao cenário específico da operação, precisaram implementar um botão para retornar ao cenário principal. Além de conteúdos matemáticos, eles desenvolveram muito raciocínio lógico, organização e concentração para o desenvolvimento da atividade.

Os alunos estavam ansiosos para verem funcionando o jogo que desenvolveram. Ficaram entusiasmados quando mencionamos que na aula seguinte faríamos um jogo no estilo *PacMan*¹⁰.

A atividade/desafio englobou um nível alto de dificuldade, haja vista os diversos cenários e personagens/*Sprites* com os quais tiveram contato. A maioria dos alunos conseguiram atingir os objetivos, porém alguns precisaram da ajuda do laboratorista para finalizar as atividades. Mesmo não tendo ocorrido tudo como o previsto, consideramos que a atividade foi relevante. Dentre os fatores que propiciaram aprendizagem, elencamos o alto nível de complexidade e a parceria entre os colegas.

Gráfico 14: Projeto Operações



Fonte: A autora.

¹⁰ *Pac-Man* é uma série de jogos de videogame cujo objetivo, em geral, é percorrer um labirinto, comendo pontos e fugindo de fantasmas. Foi desenvolvido comercialmente pela empresa japonesa Namco, no início dos anos 1980, sendo ainda popular atualmente, jogado em diferentes plataformas, com diversas adaptações.

A lógica, as operações matemáticas e as diversas etapas do pensamento computacional puderam ser apreciadas no decorrer da atividade (BRENNAN; RESNICK, 2012), assim como os conceitos, práticas e perspectivas computacionais.

Figura 52: Resumo das dimensões do Pensamento Computacional presentes na atividade anterior.

Conceitos computacionais	Práticas computacionais	Perspectivas computacionais
<ul style="list-style-type: none"> • Sequências • Ciclos • Execução em paralelo • Eventos • Condições • Operadores • Dados 	<ul style="list-style-type: none"> • Ação interativa e incremental • Teste e depuração 	<ul style="list-style-type: none"> • Expressão • Conexão • Questionar

Fonte: Adaptado de Brennan e Resnick (2012).

3.3.4.9 Décima e décima primeira aulas

O procedimento didático de iniciarmos as aulas expondo aos alunos os objetivos tornou-se prática corrente. Assim, os apresentamos o seguinte propósito das décima e décima primeira aulas: aplicar conceitos e comandos trabalhados nas aulas anteriores para a elaborar o *PacMan*.

Os alunos precisaram criar os palcos, os *Sprites* e as programações para interligar duas fases do *PacMan*. Eles podiam utilizar contador de pontos ou outro critério para a mudança de fase do jogo. Ao percebermos que o fato deles terem que criar o jogo todo os deixaram um pouco desorientados, consideramos necessárias algumas intervenções. A fim de fazê-las, contamos com o auxílio do laboratorista. Após orientações individuais, o progresso foi perceptível.

Figura 53: PacMan – Cenário e personagens



Fonte: A autora.

Figura 54: Programação dos personagens - monstros



Fonte: A autora.

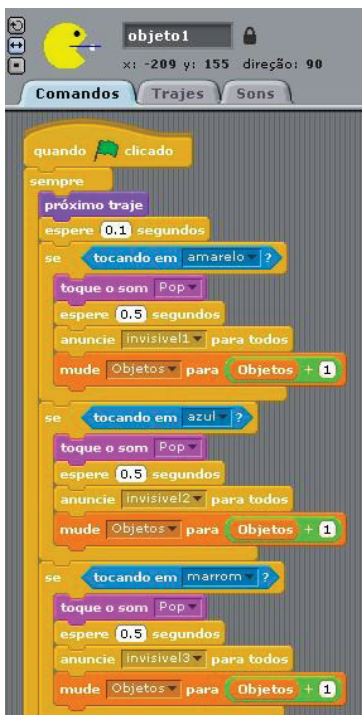
Para cada monstro deve-se utilizar um comando “quando eu ouvir – invisível1”, “quando eu ouvir – invisível2”, “quando eu ouvir – invisível3” e assim sequencialmente para cada um dos monstros.

Figura 55: Programação dos personagens – PacMan – Parte 1



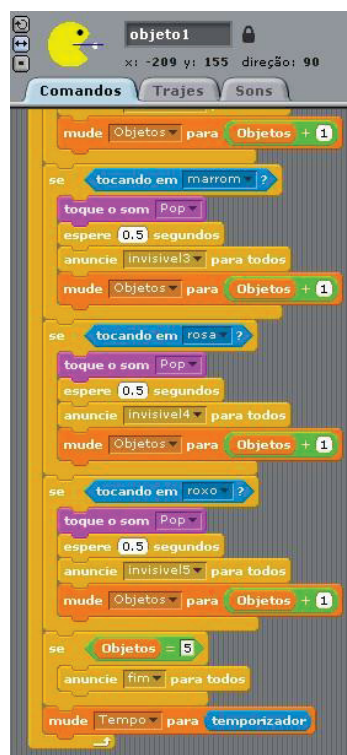
Fonte: A autora.

Figura 56: Programação dos personagens – PacMan – Parte 2



Fonte: A autora.

Figura 57: Programação dos personagens – PacMan – Parte 3



Fonte: A autora.

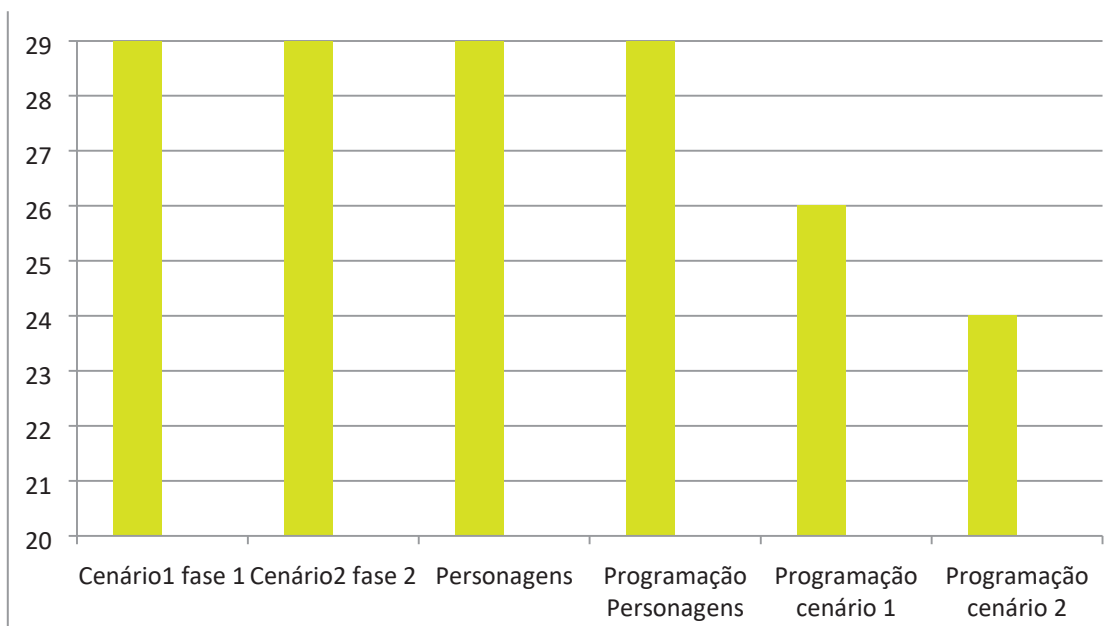
Com o desenvolvimento do jogo e os testes realizados, os alunos foram realizando as alterações necessárias para o bom andamento da atividade.

A atividade/desafio reuniu um avançado grau de complexidade devido à quantidade de *sprites*, cenários (fases) e extensa programação. Grande parte dos alunos foram capazes de finalizar a atividade em sala, porém alguns levaram para finalizar em casa. A estes, combinamos que, caso tivessem dúvidas, poderiam solicitar ajuda dos colegas, do laboratorista ou da professora.

Nesta atividade, os alunos utilizaram conteúdos matemáticos não descritos como conteúdos do 3º ano do Ensino Fundamental I, tais como: Plano cartesiano (coordenadas), valores positivos e negativos, trigonometria. Adicionalmente, também trabalharam com medidas de tempo.

No término da aula, pudemos verificar que o resultado esperado pelas turmas foi menor que nas atividades anteriores. Entretanto, o nível de dificuldade da atividade e o fato de terem que criar e programar grande parte sozinhos contribuiu para a queda do resultado.

Gráfico 15: Atividade *PacMan*



Fonte: A autora.

De acordo com o quadro de referências do pensamento computacional (BRENNAN; RESNICK, 2012), todos os conceitos, práticas e perspectivas computacionais estiveram presentes.

Figura 58: Resumo das dimensões do Pensamento Computacional presentes na atividade anterior.



Fonte: Adaptado de Brennan e Resnick (2012).

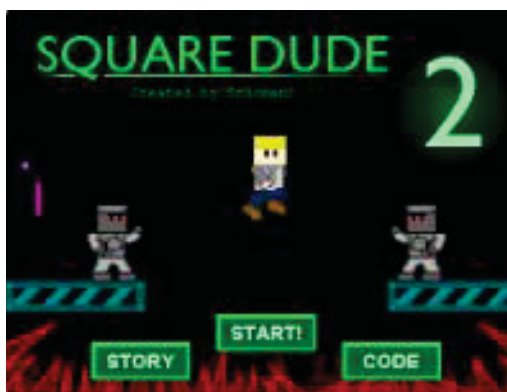
3.3.4.10 Décima segunda aula

Nesta décima segunda aula, os alunos iniciaram a elaboração do projeto. Sendo assim, apresentamos estes objetivos aos alunos: (i) apresentar aos alunos as

fases de um projeto e seus objetivos, (ii) pesquisar sobre o projeto e (iii) elaborar um rascunho do projeto.

Após apresentarmos as fases do projeto e seus objetivos, demos sequência à pesquisa do projeto e à elaboração do seu rascunho. Em seguida, demonstramos aos alunos diversos exemplos de jogos, para que pudessem ter um ponto de partida.

Figura 59: *Square Dude*



Fonte: <https://scratch.mit.edu/projects/13229729/> Acessado em: 05/04/16.

Figura 60: *Color Memory*



Fonte: <https://scratch.mit.edu/projects/154796557/> Acessado em: 05/04/16.

Após a demonstração, apresentamos um exemplo de rascunho do projeto/jogo preenchido (Apêndice I), o qual foi desenvolvido pela proponente deste estudo.

Depois da apresentação do projeto rascunho, consideramos que os alunos estavam aptos para a elaboração do próprio projeto. A função do rascunho era fazer com que eles definissem o nome e objetivo do jogo que criaram, assim com os *sprites* e os cenários.

A estratégia implementada que implementamos nesta aula funcionou parcialmente porque os alunos demonstraram algumas dificuldades no preenchimento do rascunho, mesmo após a apresentação do exemplo. Entendemos que esta dificuldade aconteceu em razão de alguns alunos terem ficados eufóricos e ansiosos.

Após verificarmos a dificuldade dos alunos em colocar no rascunho o que seria feito, solicitamos acessassem o site <https://scratch.mit.edu> e verificassem vários exemplos de jogos prontos. Após alguns minutos, juntos fomos escrevendo o rascunho, novamente. No decorrer do rascunho do projeto, verificamos que os alunos revelaram grande imaginação e vontade de desenvolver projetos interessantes e desafiadores. Sugerimos que os projetos fossem feitos em dupla, porém os alunos preferiram realizar de forma individual. Adicionalmente, eles sugeriram que os colegas apenas verificassem o mau funcionamento do projeto e opinassem sobre ele. Apesar das várias dificuldades demonstradas, os alunos conseguiram entregar o rascunho do projeto no final.

3.3.4.11 Décima terceira a vigésima aula

Logo após uma reflexão da aula anterior, constatamos que existia uma enorme variedade de projetos e com diferentes níveis de complexidade. Para a criação dos projetos, planejamos utilizar dez aulas, das quais oito seriam destinadas ao desenvolvimento individual, uma para testes dos projetos prontos e discussão deles com a turma e a última para a finalização dos projetos.

Ao final das oito aulas de desenvolvimento, os alunos podiam nos enviar os seus projetos por e-mail para que nós os acompanhassem, desse *feedback* e apoiasse na solução de problemas. Ao olhar retrospectivamente, entendemos que essa proximidade foi importante dadas as especificidades de cada projeto. Solicitamos que os alunos, a cada aula, nos entregassem o que tinham feito a fim de mostrar seus avanços com a inserção de novos elementos à programação. Assim, mantivemos sempre conosco uma cópia dos projetos ao longo das aulas, o que foi uma prevenção contra a eventual perda de arquivos, que poderia acontecer e prejudicaria a atividade proposta. Fizemos a seguinte indicação de atividades para serem adotadas no decorrer das próximas aulas:

Aula 1: Criar todos as personagens e os palcos do projeto, ou seja, da apresentação, das fases e da finalização. Começar a utilizar as imagens de *Sprites* e palcos já disponíveis no banco de dados do *Scratch*. Depois, substituir essas imagens por outras que, se quisessem, poderiam ser desenhos feitos por eles, transformados em arquivo de imagem depois de escaneados. Depois da criação dos personagens e cenários, iniciar a programação dos cenários, utilizando os comandos ocultar e exibir.

Aula 2: inserir as personagens e os cenários da primeira fase do projeto e criar a programação da fase 1.

Aula 3: Continuar a programação da fase 1 e realizar os testes.

Aula 4: Iniciar a criação da fase 2 do projeto com programação dos cenários e das personagens.

Aula 5: Continuar a programação da fase 2 e realizar os testes.

Aula 6: Iniciar a fase 3 do projeto.

Aula 7: Finalizar a fase 3 e realizar os testes.

Aula 8: Fazer os últimos ajustes na programação do projeto para deixá-lo preparado para a aula seguinte, quando uns testariam os projetos dos outros.

Analisando as propostas de projetos dos estudantes, percebemos como elas eram diferentes umas das outras. Vimos que, para que conseguíssemos orientar a todos como eu gostaríamos, precisaríamos nos preparar para conhecer toda essa diversidade. Decidimos, então, preparar materiais personalizados para os diferentes tipos de projetos. A criação destes materiais parte da prática computacional. Ela é definida por Brennan e Resnick (2012) de reutilização e reformulação. Assim, criamos trechos de códigos (disponíveis na pasta dos alunos) semelhantes aos jogos que eles iriam desenvolver. Estes fragmentos eram colocados na pasta nos minutos iniciais das aulas e visavam a apoiar os alunos e fazer com que interpretassem e adaptassem o código aos objetivos por eles pretendidos.

Como alguns dos projetos eram desafiadores, não conseguimos seguir o cronograma das oito aulas como inicialmente planejamos. Assim, primamos por tornar o andamento dos projetos respeitando os ritmos de trabalho dos estudantes e a curta duração da aula (50 minutos). Para complementar as atividades das aulas, sugerimos aos alunos que pudessem, que fizessem parte do projeto em casa. Ao longo do processo, os apoiamos, continuamente, por e-mail com o intuito de esclarecer dúvidas, lançar novos desafios nos projetos em execução e potencializar seus desempenhos.

As aulas foram animadas e dinâmicas. Consideramos que as estratégias desenvolvidas durante as foram fundamentais para o sucesso do dos trabalhos de vários alunos.

Os projetos, como dito anteriormente, foram os mais diversos possíveis. Dessa forma, apresentamos, a seguir, uma síntese dos nomes e conteúdos dos projetos

elaborados. Há um total de 39 jogos. Observamos ao leitor que houve aluno que elaborou mais de um jogo, pois eram somente 29 alunos na turma.

1. Operações – são 15 jogos: Jogo das tabuadas, Bolhas das operações, Corrida das operações, O cesto das operações, Jogo da memória, Par ou ímpar, Labirinto das adições, Círculo numérico, Descobrimo a figura, Triângulo numérico, O caminho dos números, Cruzadas dos sinais, Triângulos mágicos, Estrela mágica, Cruzada das operações
2. Reta numérica – são 4 jogos: Organize a reta numérica, Termômetro maluco, Combate dos números, Caminho dos números.
3. Problemas – são 4 jogos: Quiz, racha cuca, Cruzada matemática, Desafios.
4. Medidas de tempo – são 2 jogos: Qual o tempo, Que horas?.
5. Medidas de peso – são 2 jogos: Qual o mais pesado?, Acerte o peso.
6. Medidas de comprimento – são 2 jogos: Caça tesouro, Relacione as medidas.
7. Figuras geométricas – são 10 jogos: Palavras cruzadas das figuras planas, objeto e figura espacial, Complete com o simétrico, O empilhamento, Cara a cara dos poliedros, Jogo da memória das formas geométricas, Identificando as figuras do desenho, Caça palavras, Contando triângulos, Contando quadriláteros.

Ressaltamos que todos os personagens, cenários e programações dos jogos contaram com a ajuda e orientação dos pais e responsáveis que colaboraram de forma ativa na execução de todo o projeto.

3.3.4.12 Vigésima primeira e vigésima segunda aulas

Na vigésima primeira aula, os estudantes puderam conhecer, testar e dar sugestões nos projetos de seus colegas. Entendemos que foi um importante momento de discussão coletiva e de troca de ideias. Como resultado desta prática, os estudantes tiveram a oportunidade de receber um outro olhar sobre seus projetos. Assim, eles anotaram as críticas e sugestões e as levaram em consideração para encaminhamentos futuros.

Destinamos a vigésima segunda aula à finalização dos projetos. Este foi o momento em que os alunos puderam complementar, ajustar e corrigir a programação de seus projetos. Apesar de termos preparado e compartilhado os materiais complementares com soluções para alguns problemas, os alunos solicitaram bastante nossa ajuda à medida que faziam suas atividades. No final da aula, eles nos enviaram

a versão final de seus projetos. Algumas dessas produções ainda precisavam de ajustes, pois continham falhas de programação.

3.3.4.13 Vigésima terceira e vigésima quarta aulas

Foram sorteados 10 projetos para serem apresentados à turma nestas duas aulas. Dispomos como critérios de sorteio o tema e a quantidade de projeto de cada tema. Esclarecemos a eles que se o mesmo aluno tivesse dois projetos sorteados, seria preciso sortear outro projeto. Desse modo, o maior número de alunos poderia apresentar. Listamos, abaixo, projetos sorteados para a apresentação foram:

1. Operações: Bolhas das operações, Cruzadas dos sinais, Triângulos mágicos.
2. Reta numérica: Termômetro maluco.
3. Problemas: Racha cuca.
4. Medidas de tempo: Que horas?.
5. Medidas de peso: Qual o mais pesado?.
6. Medidas de comprimento: Caça tesouro.
7. Figuras geométricas: Objeto e figura espacial, Contando quadriláteros.

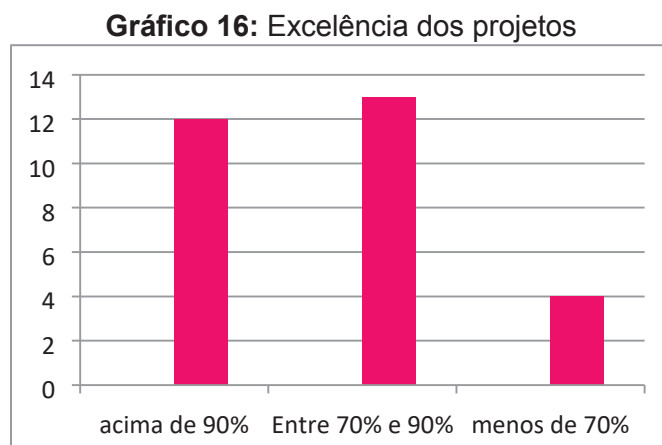
Os projetos foram apresentados utilizando o computador da professora e o projetor da sala. Os alunos explicaram os objetivos de seus projetos, fizeram uma demonstração do seu funcionamento, mencionaram as dificuldades sentidas no seu desenvolvimento e descreveram outras situações relacionadas ao desenvolvimento como, por exemplo, que elementos gostariam de ter adicionado ou que gostariam de ter feito diferente. A estrutura de apresentação foi desenvolvida por nós (Apêndice J) e visava a guiar os alunos neste momento. Nesse sentido, as apresentações seriam mais dinâmicas com duração máxima de oito minutos. Os demais projetos e suas apresentações podiam ser acessados na pasta “AlunoScratch”, criada para os projetos.

A estratégia que adotamos foi um momento produtivo de compartilhamento de ideias e de valorização das produções feitas pelos alunos. Eles apresentaram e assistiram às apresentações dos projetos dos colegas com entusiasmo, ao passo que também faziam indagações e comentários. Todos os alunos desenvolveram suas atividades de forma considerada por nós satisfatória. Ademais, entregaram os seus projetos no período estabelecido e fizeram suas apresentações conforme combinado com a turma. Parabenizamos a todos pelos projetos e pelos objetivos alcançados.

3.3.4.14 Vigésima quinta aula

Reservamos esta aula para que os alunos publicassem os seus projetos no site do *Scratch* (<http://scratch.mit.edu/>).

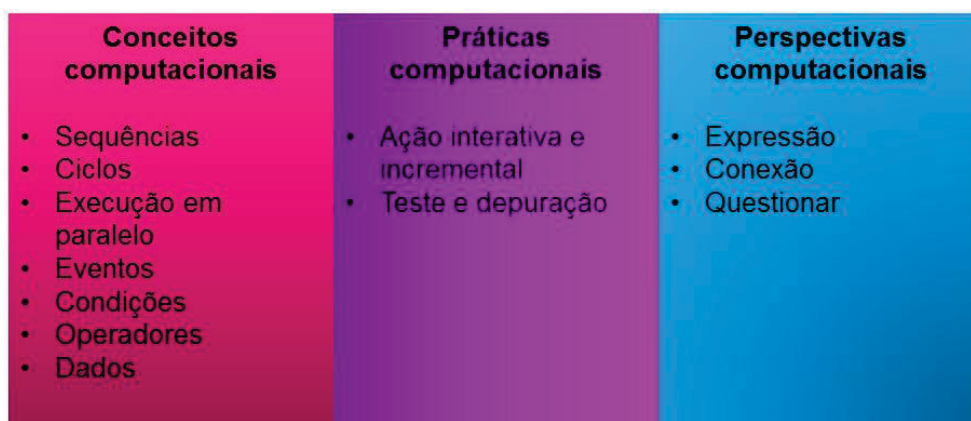
Com a posterior análise e avaliação dos projetos, confirmamos que, de fato, o resultado final foi muito satisfatório.



Fonte: A autora.

Novamente, segundo Brennan e Resnick (2012), conforme disposto em seu quadro de referências, os alunos utilizaram todos os conceitos, práticas e perspectivas computacionais.

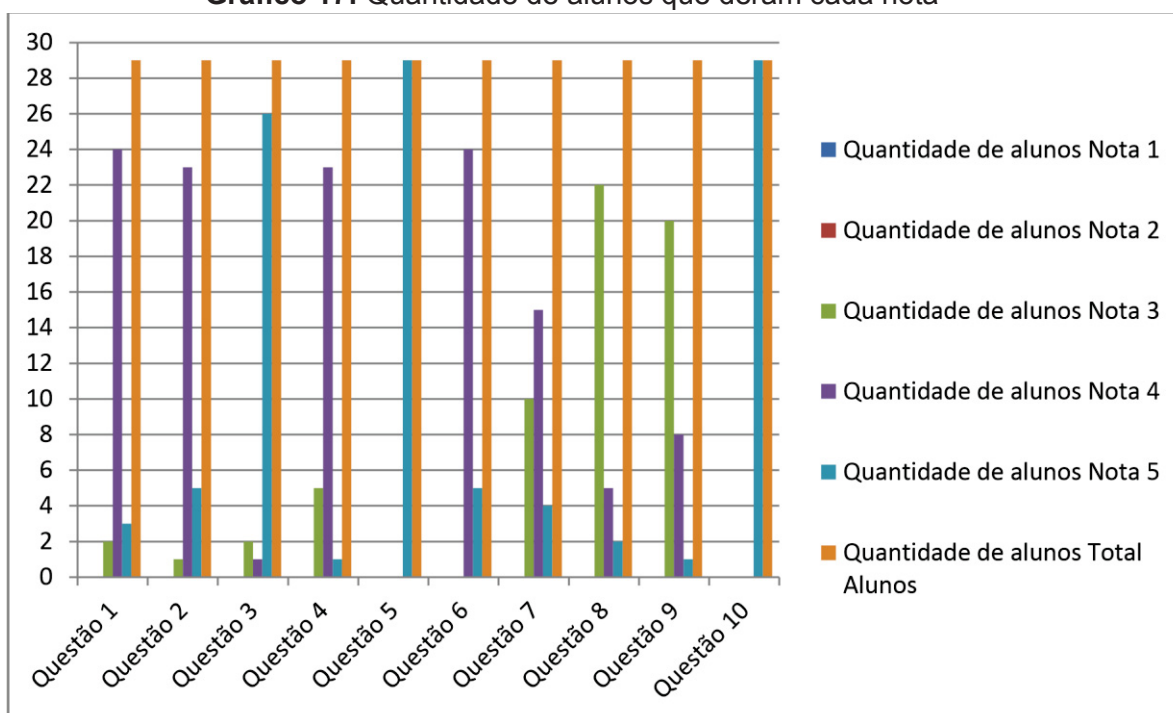
Figura 61: Resumo das dimensões do Pensamento Computacional presentes na atividade anterior.



Fonte: Adaptado de Brennan e Resnick (2012).

Nesta última aula, aplicamos o questionário constituído por dez questões (Apêndice K), no qual tínhamos como objetivo conhecer o grau de satisfação dos alunos com relação à utilização da ferramenta *Scratch*. O questionário dispunha de grau de variação entre 1 e 5, em que 1 é totalmente insatisfeito e 5 é mais que satisfeito. O questionário SUS - *System Usability Scale* foi traduzido e adaptado de Brooke (1996).

Gráfico 17: Quantidade de alunos que deram cada nota



Fonte: A autora.

Os resultados obtidos revelam um valor aproximado de 83 pontos na escala entre 0 e 100 pontos.

Tabela 1: notas e média

	Quantidade de alunos					
	Nota 1	Nota 2	Nota 3	Nota 4	Nota 5	Total Alunos
Questão 1	0	0	2	24	3	29
Questão 2	0	0	1	23	5	29
Questão 3	0	0	2	1	26	29
Questão 4	0	0	5	23	1	29
Questão 5	0	0	0	0	29	29
Questão 6	0	0	0	24	5	29
Questão 7	0	0	10	15	4	29
Questão 8	0	0	22	5	2	29
Questão 9	0	0	20	8	1	29
Questão 10	0	0	0	0	29	29
	Nota 1	Nota 2	Nota 3	Nota 4	Nota 5	
Soma por nota	0	0	62	123	105	
Total por nota	0	0	372	984	1050	
Total Geral						2406
Média geral						82,96551724

Fonte: A autora.

Assim, para além de ter sido possível atingir os objetivos propostos, a utilização desta estratégia provocou satisfação aos alunos.

4 RESULTADOS DA PESQUISA

É momento agora de avaliarmos todo o trabalho desenvolvido à luz dos objetivos de pesquisa e da literatura profissional da área.

4.1 APRENDIZADOS E DESAFIOS

A realização de uma pesquisa de mestrado envolve desafios que levam a aprendizados. O maior desafio enfrentado foi a mudança completa de foco, já que, a princípio, a pesquisa seria realizada com professores. Contudo, a mudança dos objetivos da pesquisa levou à realização de um trabalho bastante motivador que foi a realização do curso com os estudantes de 3º ao 9º ano do Ensino Fundamental. Apesar de ter que lidar com a desistência dos professores do curso e com a dificuldade de conciliar a agenda dos estudantes para realização do curso, esses aspectos nos ajudaram a conhecer uma parte da realidade das pesquisas voltadas à formação de professores e à realização de atividades práticas nas escolas.

O grande aprendizado foi perceber que nem sempre as coisas acontecem da forma como esperamos, e que devemos ter sabedoria para ouvir e refletir sobre os ocorridos. Um aprendizado pessoal, que também é apontado em muitas pesquisas, decorreu da disposição dos professores em buscar novos conhecimentos, pois é perceptível o conflito entre a busca de novos conhecimentos, a necessidade desta busca e a disposição em obtê-los.

4.2 CONSIDERAÇÕES E ANSEIOS

O estudo apresentado nesta dissertação teve por objetivo demonstrar as potencialidades do ambiente de programação *Scratch* para desenvolvimento do pensamento computacional de alunos do 3º ano do Ensino Fundamental I. Concluída a pesquisa, percebemos que ela possibilitou a identificação do impacto da: (i) estratégia na construção do pensamento computacional e (ii) da estratégia no desenvolvimento da competência de resolução de problemas. Ademais, compreendemos e a avaliação das competências dos alunos no uso do *software Scratch*.

Compreendemos ter sido possível verificar que, ao longo das aulas com os alunos, foram promovidas ações baseadas fundamentalmente na abordagem dos conteúdos necessários para o desenvolvimento do trabalho de projeto, o jogo. Os conteúdos foram divididos nas aulas iniciais e foram utilizados nas atividades, elaboração e desenvolvimento do projeto final. Nas atividades e projeto final, além de estar intrínseco o desenvolvimento das competências relacionadas com a ferramenta *Scratch*, encontram-se presentes variadas competências ligadas ao pensamento computacional. À medida que os alunos resolviam as atividades, eles também utilizavam diferentes níveis de abstração como, por exemplo, desmontar e solucionar, ou seja, o pensamento abstrato e computacional. (PHILLIPS, 2009; WING, 2006).

A título de exemplo, a atividade da quinta aula, em que era necessário verificar se o número era maior ou menor a cinquenta, poderia ser resolvido de forma rápida através de dois comandos *Se*. No entanto, a resolução mais eficaz seria através da utilização do comando *se...senão*. No decorrer desse processo, esteve inerente a formulação e a exclusão de hipótese caracterizando o pensamento lógico (PHILLIPS, 2009; WING, 2006). Durante a resolução de um problema, o *Scratch* possibilitou testar a validade ou não da solução.

Conforme as dimensões que caracterizam o pensamento computacional, definidas por Brennan e Resnick (2012), em todas as atividades estiveram presentes, de forma fracionada, os elementos das três dimensões, a saber: a) conceitos computacionais, b) práticas computacionais e c) perspectivas computacionais.

Tabela 2: Resumo dos conceitos computacionais presentes nas diversas atividades

Conceitos computacionais	Execução					
	Sequências	Ciclos	em paralelo	Eventos	Condições	Operadores Dados
Aula 1						
Aula 2	X		X			
Aula 3	X		X			
Aula 4	X		X			
Aula 5	X	X		X	X	X
Aula 6	X	X		X	X	X

Aula 7	X	X		X	X	X	
Aula 8	X	X	X	X	X	X	X
Aula 9	X	X	X	X	X	X	X
Aula 10	X	X	X	X	X	X	X
Aula 11	X	X	X	X	X	X	X
Aula 25	X	X	X	X	X	X	X

Fonte: A autora.

Para resolver um problema e chegar a uma solução, os alunos precisaram identificar as diversas etapas da atividade, ou seja, as sequências. Em contrapartida, em alguns pontos específicos das atividades, eles precisaram executar a mesma sequência diversas vezes, o que chamamos de ciclos. Em outros momentos, eles necessitaram fazer duas ou mais ações paralelamente, o que identificamos como execução em paralelo. Na última, chamamos de evento o ato de um acontecimento provocar outro.

Estiveram presentes nas atividades as funções de condições, isto é, a tomada de decisões valendo-se dos respectivos sinais aritméticos e de comparação.

Os alunos desenvolveram e testaram para verificar o funcionamento. Em outras palavras, ação iterática e incremental, a qual possibilitou-os a hipótese de correção dos erros, isto é, teste de depuração.

Ao passo que aumentava o grau de complexidade das atividades, os alunos tinham a possibilidade de se favorecerem de conhecimentos e habilidades já utilizados anteriormente ou descobertos com colegas, ou seja, reutilização e reformulação.

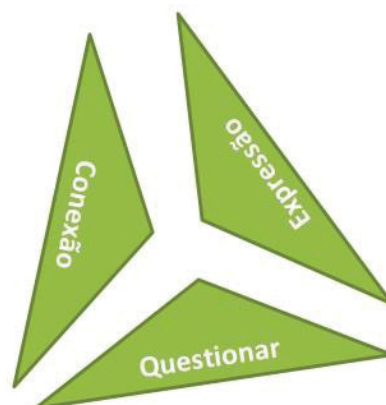
Figura 63: Práticas computacionais presentes nas atividades



Fonte: A autora.

As perspectivas computacionais estiveram de forma colaterais ao longo de todo o processo. No desenvolvimento das atividades, apesar de receberem orientações, os alunos criavam algo novo como, por exemplo, uma expressão, pois incluíam aos seus projetos elementos livremente escolhidos por eles. Da mesma forma, no decorrer das atividades, eles trabalharam em conjunto, ou seja, em conexão. Nas ocasiões em que solicitavam ajuda, era para perguntar sobre algum aspecto tecnológico.

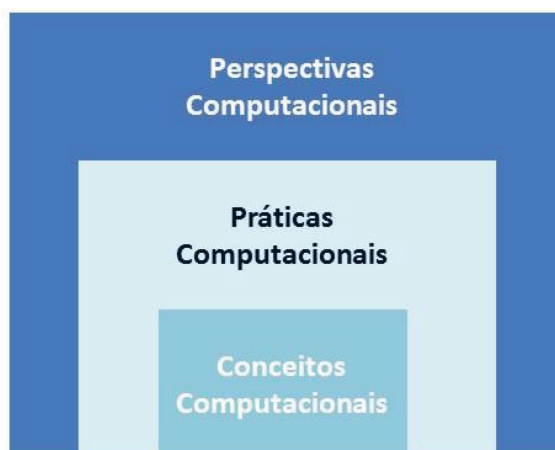
Figura 64: Perspectivas computacionais



Fonte: A autora.

No decorrer das aulas, os alunos puderam praticar conceitos, práticas e perspectivas computacionais. Os conceitos de forma subentendida davam origem a aplicação de práticas e conexão as perspectivas.

Figura 65: Resumo das dimensões do Pensamento Computacional



Fonte: A autora.

Em concordância com Brennan e Resnick (2012), a verificação do desenvolvimento do pensamento computacional deve ser realizada por meio de análise dos projetos baseando-se nas três dimensões por eles definidas. Tais práticas computacionais foram também observadas durante as aulas.

No que concerne ao segundo objetivo da investigação, identificar o impacto da estratégia no desenvolvimento da competência de resolução de problemas, todas as atividades eram abertas a sugestões e requeriam dos alunos uma participação dinâmica para auferir uma solução (ECHEVERRÍA; POZO, 1998). As atividades estavam fundadas por diversos passos, não sendo possível solucionar de maneira ininterrupta (ECHEVERRÍA; POZO, 1998). Para o desenvolvimento do projeto e para a resolução das atividades foram necessários seguir as quatro etapas (PÓLYA, 2003). As fases são: compreender o problema, elaborar um plano, executá-lo e verificar os resultados.

Os problemas das atividades foram estabelecidos quanto a estrutura, complexidade, dinamismo específico (JONASSEN, 2004). Em cada aula havia uma quantidade restrita de conceitos (estrutura e complexidade). Com relação ao dinamismo, os alunos procuravam e adaptavam soluções conforme sua compreensão ia se formando e/ou alterando. Quanto ao domínio específico, este era realizado de forma peculiar por cada aluno.

O desenvolvimento da pesquisa e a realização das atividades das aulas fizeram com que os alunos alcançassem competências e conhecimentos agremiados ao trabalho em grupo como comunicação, relação interpessoal, cooperação e respeito mútuo (LEITE; ESTEVES, 2005). Este resultado de pesquisa evidencia que os alunos podiam auxiliar seus colegas e deles requerer ajuda. Relacionadas a estas

competências e conhecimentos propriamente direcionados para o pensamento computacional, estiveram intrínsecas as competências referentes ao uso da ferramenta *Scratch*.

Ainda que as turmas se diferenciasssem por suas características, nos foi possível reconhecer que as estratégias utilizadas foram fomentadoras ao desenvolvimento e construção do pensamento computacional. Outros ganhos dizem respeito ao desenvolvimento da competência de resolução de problemas e de competências relativas ao uso do *software Scratch*.

Entendemos que o auge do projeto foi a apresentação dos projetos dos alunos a seus familiares que puderam experimentar os jogos desenvolvidos por seus filhos e colegas.

Os resultados apontam que as crianças aprenderam diversos conceitos durante o minicurso, alguns são: sequência, evento, laços, condicionais, variáveis, procedimentos, variáveis. Além de conteúdos, os jovens puderam conhecer e exercitar conceitos e práticas relacionados à resolução de problemas ligados ao desenvolvimento do pensamento computacional.

Considerando os resultados obtidos com o minicurso, avaliamos que os objetivos inicialmente traçados foram alcançados. Com o uso do *Scratch*, conseguimos explorar os conceitos computacionais apontados por Brennan e Resnick (2012) na disseminação do Pensamento computacional. Além disso, conseguimos explorar na prática conceitos da resolução de problemas, que são importantes à formação de uma geração mais aberta ao compartilhamento de ideias. Por fim, os conteúdos puderam ser avaliados continuamente por meio de projetos, o que nos possibilitou acompanhar a aprendizagem dos estudantes durante seu processo de formação.

No que concerne projetos futuros e outras práticas de ensino, entendemos que este minicurso, voltado para o ensino e avaliação do pensamento computacional entrelaçado a resolução de problemas utilizando o *Scratch*, poderá ser adaptado a outros contextos.

Considerando os potenciais e as limitações do presente estudo, propõe-se como trabalhos futuros a realização de investigações que acompanhem os alunos por um tempo maior, acompanhando o desenvolvimento do seu pensamento computacional com atividades mais regulares e com maior frequência, idealmente no currículo regular.

REFERÊNCIAS

- ALVAREZ, L. **Ensino de programação é aposta de colégios em todo o mundo**. 2014. Disponível em: <http://revistaeducacao.uol.com.br/textos/211/aposta-nofuturoo-ensino-de-programacao-tem-se-espalhado-como-330266-1.asp> Acesso em: 01 jun. 2016.
- ANDRADE, M.; SILVA, C.; OLIVEIRA, T. **Desenvolvendo games e aprendendo matemática utilizando o Scratch**. In: SBGAMES, XII, São Paulo. 2013. **Anais** 2013. p.260-263.
- AURELIANO, V. C. O.; TEDESCO, P.C. A. R. Ensino-aprendizagem de programação para iniciantes: uma revisão sistemática da literatura focada no SBIE e WIE. In: **Simpósio Brasileiro de Informática na Educação**, 23., 2012, Rio de Janeiro. **Anais...**, p. 1–10.
- BARCELOS, T. S.; SILVEIRA, I. F. Pensamento computacional e educação matemática: relações para o ensino de computação na Educação Básica. **Imago Research Group**. 2012. p.1-10.
- Batista, S. C. F.; Baptista, C. B. F. Scratch e matemática: desenvolvimento de um objeto de aprendizagem. In: ENCONTRO DE EDUCAÇÃO MATEMÁTICA, I, 2013, Fluminense, **Anais...** Fluminense: IFF, 2013, p.1-8.
- BRESSAN, M. L. Q., AMARAL, M. A. Avaliando a Contribuição do Scratch para a Aprendizagem pela Solução de Problemas e o Desenvolvimento do Pensamento Criativo, **Intersaberes**, São Paulo, v. 10, n. 21, p. 509-526. 2015.
- BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. In: AMERICAN EDUCATIONAL RESEARCH ASSOCIATION ANNUAL MEETING, 2017, San Antonio, **Proceedings...** San Antonio: AERA, 2017, p.1-25.
- CABRAL, R. V. O ensino de matemática e a informática: uso do *Scratch* como ferramenta para o ensino e aprendizagem da geometria. Sarandi: FACNORTE, 2015.
- CHAI, C.g S.; KOH, J. H. L.; TSAI, C. A Review of Technological Pedagogical Content Knowledge. **Educational Technology & Society**, Taipei, v.16, n.2, p.31–51, 2013.
- COLL, C., et al. **O Construtivismo na sala de aula**. Porto: Edições ASA, 2001.
- D'AMBRÓSIO, U. **Algumas reflexões sobre a resolução de problemas**. Disponível em . Acesso em: 25 jul. 2010.
- DANTE, L. R. **Formulação e resolução de problemas de matemática: teoria e prática**. São Paulo: Ática, 2009.
- DENNING, P. J. Is computer science? **Communications of the ACM**, New York, v.48, n.4, p. 27–31, 2005.
- DEWEY, J. **Democracy and Education**. New York: Macmillan. 1916.

ECHEVERRÍA, M.; POZO, J. Aprender a resolver problemas e resolver problemas para aprender. In: POZO, J. (Org.). **A solução de problemas: aprender a resolver, resolver para aprender**. Porto Alegre: Artmed. 1998.

FARINELLI, F. **Conceitos básicos de programação orientada a objetos**. 2007. Disponível em: <http://sistemas.riopomba.ifsudestemg.edu.br/dcc/materiais/1662272077_POO.pdf> Acesso em: 14 dez. 2016.

GOETTEN, V. J. WINCK, D. V. **AspectJ: Programação Orientada a Aspectos com Java**. São Paulo: Novatec Editora, 2006.

GOMES, R. C. S, GHEDIN, E. O Desenvolvimento cognitivo na visão de Jean Piaget In: GHEDIN, E. (Org.). **Teorias Psicopedagógicas do Ensino Aprendizagem**. Boa Vista: UERR Editora, 2012.

GUDWIN, R. R. **Linguagens de Programação: notas de aula para a disciplina EA877**. 1997. Disponível em: <<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea877/lingpro.pdf>> Acesso em: 13 dez. 2016.

ISELL, C. L.; et al. (Re)defining computing curricula by (re)defining computing. **SIGCSE Bulletin**, Nova Iorque, v.41, n.4, p. 195–207, 2010.

JONASSEN, D. H. **Learning to solve problems: an instructional design guide**. São Francisco: Pfeiffer. 2004.

KAFI, Y. B. BURKE, Q. Computer Programming Goes Back to School. **Education Week**, California, v.95, n.1, p.61-65, 2013.

LEITE, L.; ESTEVES, E. Ensino orientado para a aprendizagem baseada na resolução de problemas na Licenciatura em Ensino de Física e Química. In: SILVA, B.; ALMEIDA, L. (Org.). **Actas do Congresso Galaico-Português de Psico Pedagogia**. Braga: Universidade do Minho. 2005. p. 1751-1768.

DALLA VECCHIA, R. **A modelagem matemática e a realidade do mundo cibernético**. 2012. 275f. Tese (Doutorado em Educação Matemática) – Universidade Estadual Paulista Júlio de Mesquita Filho

MIT. **Computação Criativa - uma introdução ao pensamento computacional baseada no conceito de design**. (EduScratch, Trans.) 2011.

MEDINA, M.; FERTIG, C. **Algoritmo e Programação: Teoria e Prática**. São Paulo: Novatec, 2005.

NUNES, D. **Ciência da Computação na Educação Básica**. Disponível em 15 out. 2013: <http://www.jornaldaciencia.org.br/Detailhe.php?id=79207>. Acesso em: 9 set. 2011.

ONUICHIC, L. de La R. Ensino-aprendizagem de matemática através da resolução de problemas. In: BICUDO, M. A. V. (Org.). **Pesquisa em educação matemática: concepções e perspectivas**. São Paulo: UNESP, 1999. p.199-218.

_____. **Uma história da resolução de problemas no Brasil e no mundo**. Disponível em: . Acesso em: 25 JUL. 2010.

_____.; ZUFFI, E. M. O ensino-aprendizagem de matemática através da Resolução de Problemas e os processos cognitivos superiores. **Revista Iberoamericana de Matemática**, San Cristobal de La Laguna, v. 11, n. 11, p. 79- 97, 2007.

PAGE-JONES, M., FECCHIO, M. M. **O que todo Programador deveria saber sobre Projeto Orientado a Objeto**. Sao Paulo: Makron Books, 1997.

PAPERT, S. **The Children Machine**. New York: BasicBooks, 1993.

_____. **A Máquina das Crianças: Repensando a Escola na Era da Informática**. Porto Alegre, RS: Artes Médicas, 1993.

_____. **A família em rede**. Lisboa: Relógio D` Água, 1998.

_____. **Logo: Computadores e Educação**. São Paulo: Brasiliense, 1985.

PEREIRA, L. **Escolas Defendem Ensino de Programação a Crianças e Adolescentes**. Olhar Digital, 06 Fev. 2013. Disponível em: . Acesso em: 15 de Junho de 2016.

Phillips, P. (2009). http://www.csta.acm.org/ProfessionalDevelopment/sub/CSIT09Presentations/Phillips_Computational.pdf (Acessível em 17 de Junho de 2013)

PINTO, M. **Práticas educativas numa sociedade global**. Porto: ASA Editores, 2002.

PÓLYA, G. A. **A arte de resolver problemas**, 1978. Tradução Heitor Lisboa de Araújo. São Paulo: Interciência, 2003.

PONTE, J. P. **O computador na educação Matemática** (Cadernos de Educação Matemática, Nº 2). Lisboa: APM, 1991.

POZO, J. I. **A solução de problemas: aprender a resolver, resolver para aprender**. Porto Alegre: Artmed, 1998. 173p.

RESNICK, M. Point of View - Reviving Papert's Dream. **Educational Technology**, Londres, v.52, p.42-46. 2012.

_____, KAFAI, Y.; MAEDA, J. **A Networked, Media-Rich Programming Environment**, 2003. Disponível em: <http://web.media.mit.edu/~mres/papers/scratchproposal.pdf> Acesso em: 23 nov. 2012.

_____, et al. Scratch: Programming for All. **Communication of ACM**, Cambridge, v.52, n.11, p. 60-67. 2009.

RICHARDSON, R. J. **Pesquisa Social: Métodos e Técnicas**. São Paulo: Atlas, 2011.

SÁPIRAS, F. S.; VECCHIA, R. D.; MALTEMPI, M. V. Utilização do Scratch em sala de aula. **Educação Matemática Pesquisa**, São Paulo, v. 17, p. 973–988, 2015.

ESCOLA VICENTINA SAGRADO CORAÇÃO DE JESUS. **Site da escola**. Disponível em: <http://www.escolasagrado.com.br/> Acesso em: 16 mai. 2018.

SILVA, E. L.; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. 4. ed. Florianópolis: UFSC, 2005. 138 p.

THE CSTA STANDARDS TASK FORCE. **CSTA K-12 Computer Science Standards**, 2011. New York: ACM Computer Science Teachers Association. Disponível em: <<http://csta.acm.org/Curriculum/sub/K12Standards.html>>. Acesso em: 23/04/2017.

TRIVIÑOS, A. N. S. **Introdução à pesquisa em ciências sociais**. São Paulo: Atlas, 1987.

USA NATIONAL RESEARCH COUNCIL. **Report of a Workshop of Pedagogical Aspects of Computational Thinking**. Washington, D.C.: The National Academies Press. 2011. Disponível em: <<http://www.nap.edu/catalog/13170/report-of-a-workshop-on-the-pedagogical-aspects-of-computational-thinking>>. Acesso em: 29/10/2016.

VALENTE, J. A. **Integração do Pensamento Computacional no currículo da Educação Básica**: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. **Curriculum**, São Paulo, v.14, n.3, 2016, p.864897.

Valente, J. A. “**Pesquisa, comunicação e aprendizagem com o computador**”. Série “Pedagogia de Projetos e Integração de Mídias” - Programa Salto para o Futuro, Setembro, 2003.

_____. Por Quê o Computador na Educação? In: _____. **Computadores e Conhecimento**: repensando a educação. Campinas: Gráfica da UNICAMP, 1993. p. 24-44.

WANGENHEIM, C. G.; NUNES, V. R.; SANTOS, G. D. Ensino de Computação com SCRATCH no Ensino Fundamental: um Estudo de Caso. **Brasileira de Informática na Educação**, São Paulo, v. 22, n. 3, p. 124-146.

Wing, J. M. Five deep questions in computing. **Commun. ACM**, Nova Iorque, v.51, p.58–60, 2007.

_____. Computational thinking. **Communications of the ACM**, Nova Iorque, v. 49, n. 3, p. 33-35, 2006.

ZAPATA-ROS, M. Pensamiento computacional: Una nueva alfabetización digital. **Educación a Distancia**, Murcia, v. 46, n. 4, p. 1-47, 2015.

APÊNDICE A

FICHA DE INSCRIÇÃO PARA O CURSO

“APRENDENDO E ENSINANDO A PROGRAMAR COM O SCRATCH”

Nome: _____

Disciplinas que ministra: _____

Séries/anos que trabalha: _____

Interesses: _____

Obs.: O curso é totalmente gratuito e incluso material didático.

APÊNDICE B

TERMO DE AUTORIZAÇÃO DE USO DE IMAGEM, VOZ E DADOS DIGITAIS E RESPECTIVA CESSÃO DE DIREITOS

Eu, abaixo identificado, na melhor forma de direito, autorizo, de forma gratuita e sem qualquer ônus, ao pesquisador e à Universidade Federal do Paraná (UFPR), a utilização de meu nome, minha imagem e som de voz, relacionados ao material descrito neste termo, na dissertação de Mestrado em Educação em Ciências e em Matemática, nos Catálogos das Bibliotecas desta Instituição, e em todos os meios de divulgação possíveis, quer sejam na mídia impressa (livros, catálogos, revista, jornal, entre outros), televisiva(propagandas para televisão aberta e/ou fechada, vídeos, filmes, entre outros), radiofônica (programas de rádio/podcasts), escrita e falada, Internet, Banco de dados informatizados, Multimídia, “home video”, DVD, entre outros, e nos meios de comunicação interna, como jornal e periódicos em geral, na forma de impresso, voz e imagem, sem ressarcimento dos direitos autorais e de acordo com a Lei no 9.610/98.

A presente autorização e cessão são outorgadas livres e espontaneamente, em caráter gratuito e universal, não incorrendo a autorizada em qualquer custo ou ônus, seja a que título for, sendo que estas são firmadas em caráter irrevogável, irretratável, e por prazo indeterminado, obrigando, inclusive, eventuais herdeiros e sucessores outorgantes, produzindo seus efeitos não só no Brasil, mas em qualquer lugar situado fora das fronteiras nacionais.

Identificação:

Nome do Cedente: _____ RG: _____
 _____ CPF: _____ Telefone: (____) _____ e-mail: _____

Pesquisador(a): Sidnéia Valero Egido RG: 8.942.724-0 CPF: 039.982.739-05







 Local e Data

 Assinatura do cedente

APÊNDICE C

Aprendendo a programar com o

VENHA APRENDER A CRIAR SEUS GAMES, PROGRAMANDO E DESENVOLVENDO SEU RACIOCÍNIO.

-  o Scratch é uma linguagem de programação fácil de se aprender;
-  pode ser usado para criar histórias animadas, jogos e outros programas interativos;
-  permite misturar diferentes tipos de mídias, como imagens, sons e outros programas;
-  lembra o brinquedo Lego;
-  auxilia no desenvolvimento de raciocínio;
-  foi criado em 2003 pelo Media Lab do MIT (Massachusetts Institute of Technology).

PROGRAMAÇÃO

- ✗ Duração: 25h
- ✗ Local: Colégio Sagrado
- ✗ Totalmente gratuito
- ✗ Aulas em contraturno conforme horários abaixo:

Horários:

Dia	Horário	Turmas
6ª feira	14h45mín às 15h20mín	3ª e 4ª anos
6ª feira	15h20mín às 16h05mín	5ª e 6ª anos
6ª feira	16h05mín às 16h45mín	7ª e 8ª anos
2ª feira	16h45mín às 17h25mín	9ª anos

Parcerias:



PPGECM
Programa de Pós-Graduação
em Educação em Ciências e
em Matemática









APÊNDICE D

Aprendendo a programar com o



PROJETO VOLTADO PARA O DESENVOLVIMENTO DA CRIATIVIDADE E RACIOCÍNIO ATRAVÉS DA ELABORAÇÃO DE GAMES, FAZENDO USO DA LÓGICA DE PROGRAMAÇÃO

-  o Scratch é uma linguagem de programação fácil de se aprender;
-  pode ser usado para criar histórias animadas, jogos e outros programas interativos;
-  permite misturar diferentes tipos de mídias, como imagens, sons e outros programas;
-  lembra o brinquedo Lego;
-  auxilia no desenvolvimento de raciocínio;
-  foi criado em 2003 pelo Media Lab do MIT (Massachusetts Institute of Technology).

PROGRAMAÇÃO

- ✗ Duração: 25h
- ✗ Início: 10 de maio de 2016
- ✗ Local: Escola Vicentina Sagrado Coração de Jesus
- ✗ Totalmente gratuito
- ✗ Alunos do 3º ao 5º ano do período da tarde irão participar deste projeto, sendo que suas aulas acontecerão durante as aulas de informática.

Parcerias:



PPGECM
Programa de Pós-Graduação
em Educação em Ciências e
em Matemática



APÊNDICE E

TERMO DE AUTORIZAÇÃO DE USO DE IMAGEM, VOZ E DADOS DIGITAIS E RESPECTIVA CESSÃO DE DIREITOS

Eu, _____, responsável pelo
(a) menor _____, na melhor forma de direito, autorizo, de forma gratuita e sem qualquer ônus, ao pesquisador e à Universidade Federal do Paraná (UFPR), a utilização de meu nome, minha imagem e som de voz, relacionados ao material descrito neste termo, na dissertação de Mestrado em Educação em Ciências e em Matemática, nos Catálogos das Bibliotecas desta Instituição, e em todos os meios de divulgação possíveis, quer sejam na mídia impressa (livros, catálogos, revista, jornal, entre outros), televisiva(propagandas para televisão aberta e/ou fechada, vídeos, filmes, entre outros), radiofônica (programas de rádio/podcasts), escrita e falada, Internet, Banco de dados informatizados, Multimídia, “home video”, DVD, entre outros, e nos meios de comunicação interna, como jornal e periódicos em geral, na forma de impresso, voz e imagem, sem ressarcimento dos direitos autorais e de acordo com a Lei no 9.610/98. A presente autorização e cessão são outorgadas livres e espontaneamente, em caráter gratuito e universal, não incorrendo a autorizada em qualquer custo ou ônus, seja a que título for, sendo que estas são firmadas em caráter irrevogável, irretratável, e por prazo indeterminado, obrigando, inclusive, eventuais herdeiros e sucessores outorgantes, produzindo seus efeitos não só no Brasil, mas em qualquer lugar situado fora das fronteiras nacionais.

Identificação do Responsável:

Nome Completo: _____ RG: _____

_____ CPF: _____ Telefone: () _____ e-

mail: _____

Identificação do Menor:

Nome Completo: _____

RG: _____ RN: _____

Pesquisador(a): Sidnéia Valero Egido RG: 8.942.724-0 CPF: 039.982.739-05

Local e Data

Assinatura do cedente

APÊNDICE F**QUESTIONÁRIO INTRODUTÓRIO AOS ALUNOS****USO DO COMPUTADOR NO SEU COTIDIANO**

- 1. Para que atividade você mais utiliza o computador em casa?**
 1. Jogar
 2. Ouvir musica
 3. Pesquisar na internet
 4. Ver vídeos
 5. Fazer da escola

- 2. Que tipo de atividade mais é realizada nas aulas de informática na escola?**
 1. Jogar
 2. Pesquisar
 3. Tarefas
 4. Leituras

- 3. Você gostaria de criar seus jogos escolares?**
 1. Sim
 2. Não
 3. Talvez

- 4. Gostaria de aprender novos programas de computador?**
 1. Sim
 2. Não

- 5. Você conhece o alguma linguagem de programação?**
 1. Sim
 2. Não

- 6. Já ouviu falar no Scratch?**
 1. Sim
 2. Não

APÊNDICE G Plano de aula

Identificação: Escola Vicentina Sagrado Coração de Jesus

Professora: Sidnéia Valero Egido

Duração da Aula: 50 min

Data: ___/___/2016

Tema: Introdução ao Software livre *Scratch*

Objetivos:

Geral: Conhecer os comandos básicos do software livre Scratch **Específicos:**

- Mostrar exemplos da utilização do software livre Scratch.
- Realizar atividades introdutórias sobre o software livre Scratch.
- Propor a utilização do software livre Scratch nas elaborações de atividades.

Conteúdo:

Comando Básicos do software livre Scratch: Área de trabalho do *Scratch* e Abas: Movimento, aparência e controle.

Metodologia:

Inicialmente será feita uma exposição oral, com uso da mídia data show e depois os alunos serão orientados a começaram a fazer as atividades nos computadores.

APÊNDICE H Grades de avaliações das aulas

Aula 02

Quantidade de alunos (29 – total) Concluiu parcialmente dos (70 % a 99% dos objetivos) Concluiu parcialmente dos (100% dos objetivos) Concluiu (abixo de 70 % dos objetivos)

Aula 03

Quantidade de alunos (29 – total) Cenário Programação Cenário Personagem Programação Personagem

Aula 04

Quantidade de alunos (29 – total) Cenário Personagens Programação Personagem 1 Programação Personagem 2

Aula 05, 06 e 07

Quantidade de alunos (29 – total) Cenário Personagens Programação Compreenssão da Personagem 1 Programação


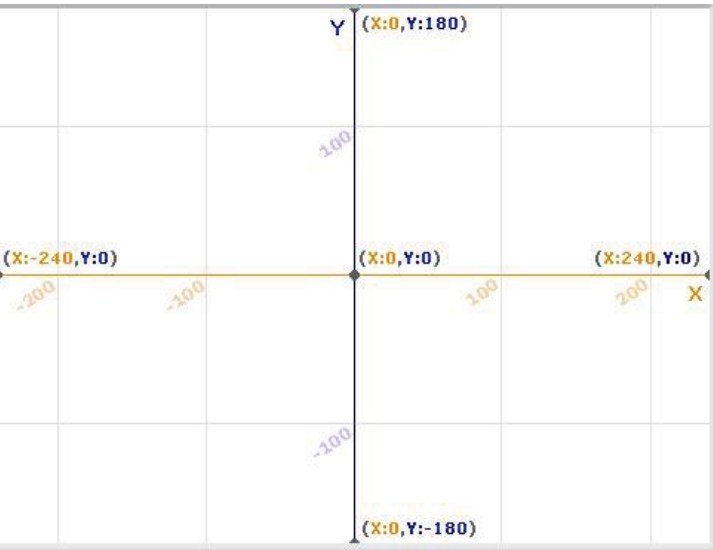
Aula 08 e 09

Quantidade de alunos (29 – total) Cenários Programação cenário Personagens Programação Personagem 1 Programação Personagem 2 Programação Personagem 2

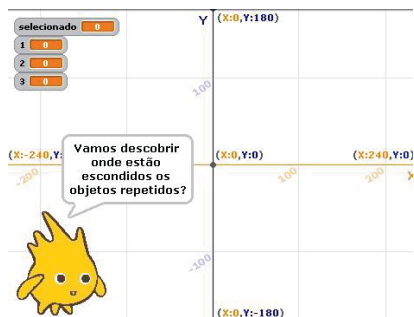
Aula 10 e 11

Quantidade de alunos (29 – total) Cenário1 fase 1 Cenário2 fase 2 Personagens Programação cenário 1 Programação cenário 2 Programação de

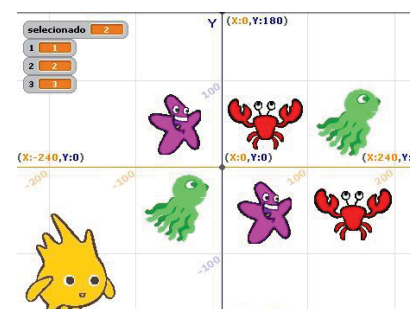
APÊNDICE I Modelo de Projeto

Aluno: _____ Turma _____
1. Nome do jogo: Jogo da memória
2. Objetivo do jogo: Encontrar os pares idênticos com o menor número de chances.
3. Sprites:

4. Palcos:

5. Telas do jogo:
Fase 1
Apresentação
<div style="background-color: #f4a460; padding: 20px; text-align: center;"> <h1 style="color: red; margin: 0;">Jogo da Memória</h1> </div>

Início do jogo




Término da fase 1



Fase 2

Apresentação

Nesta fase do jogo o seu tempo será cronometrado. Ganha que encontrar os pares em menos tempo.



APÊNDICE J

Estrutura de apresentação do projeto/jogo *Nome do projeto:*

Desenvolvedor:

Objetivos:

Funcionamento:

Dificuldades:

O que mais gostariam de acrescentar:

Outras observações:

APÊNDICE K**Questionário final**

As notas devem variar de 1 a 5, sendo:

1 – para insatisfeito

2 – para pouco satisfeito

3 – satisfeito

4 – muito satisfeito

5 – mais que satisfeito

1. Penso que gostaria de utilizar o Scratch frequentemente.

Nota: _____

2. Achei o Scratch desnecessariamente difícil.

Nota: _____

3. Acho que o Scratch foi fácil de usar.

Nota: _____

4. Penso que precisaria da ajuda de alguém para utilizar o Scratch.

Nota: _____

5. Achei as ferramentas do Scratch organizadas.

Nota: _____

6. Achei o Scratch confuso.

Nota: _____

7. Penso que a maioria das pessoas seria capaz de aprender a utilizar o Scratch facilmente.

Nota: _____

8. Achei o Scratch muito complicado.

Nota: _____

9. Senti-me confiante a utilizar o Scratch.

Nota: _____

10. Precisei de aprender muitas coisas antes de poder utilizar o Scratch.

Nota: _____