



UNIVERSIDADE ESTADUAL PAULISTA  
“JÚLIO DE MESQUITA FILHO”

FACULDADE DE CIÊNCIAS - CAMPUS DE BAURU  
PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO PARA A CIÊNCIA

**DISSERTAÇÃO DE MESTRADO**

Programação no ensino de matemática utilizando Processing 2: Um estudo das relações formalizadas por alunos do ensino fundamental com baixo rendimento em matemática

**EDUARDO CARDOSO DE SOUZA**

**ORIENTADOR**  
**PROF. DR. WILSON MASSASHIRO YONEZAWA**

**BAURU**  
**2016**

**EDUARDO CARDOSO DE SOUZA**

Programação no ensino de matemática utilizando Processing 2: Um estudo das relações formalizadas por alunos do ensino fundamental com baixo rendimento em matemática

Dissertação apresentada à Faculdade de Ciências da Universidade Estadual Paulista “Júlio de Mesquita Filho” – UNESP, Campus de Bauru, Programa de Pós-Graduação em Educação para a Ciência, como um dos requisitos para a obtenção do título de Mestre em Educação para a Ciência (Área de Concentração: Ensino de Matemática), sob a orientação do Prof. Dr. Wilson Massashiro Yonezawa.

**Bauru  
2016**

Souza, Eduardo Cardoso.

Programação no ensino de matemática utilizando Processing 2: um estudo das relações formalizadas por alunos do ensino fundamental com baixo rendimento em matemática / Eduardo Cardoso de Souza, 2016  
188 f. : il.

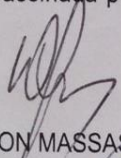
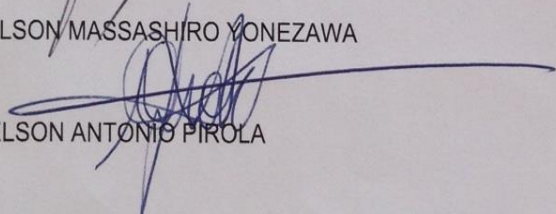
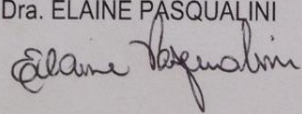
Orientador: Wilson Massashiro Yonezawa

Dissertação (Mestrado)-Universidade Estadual Paulista. Faculdade de Ciências, Bauru, 2016

1. Ensino de Programação. 2. Resolução de Problemas. 3. Pensamento Matemático. 4. Aprendizagem situada I. Universidade Estadual Paulista. Faculdade de Ciências. II. Título.

**ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE Mestrado DE EDUARDO CARDOSO DE SOUZA, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO PARA A CIÊNCIA, DA FACULDADE DE CIÊNCIAS.**

Aos 25 dias do mês de fevereiro do ano de 2016, às 10:00 horas, no(a) Anfiteatro da Pós-graduação da FC, reuniu-se a Comissão Examinadora da Defesa Pública, composta pelos seguintes membros: Prof. Dr. WILSON MASSASHIRO YONEZAWA do(a) Departamento de Computação / Faculdade de Ciências - UNESP/Bauru, Prof. Dr. NELSON ANTONIO PIROLA do(a) Departamento de Educação / Faculdade de Ciências - UNESP/Bauru, Profa. Dra. ELAINE PASQUALINI do(a) FATEC / Faculdade de Tecnologia/FATEC, sob a presidência do primeiro, a fim de proceder a arguição pública da DISSERTAÇÃO DE Mestrado de EDUARDO CARDOSO DE SOUZA, intitulada **Programação no ensino de matemática utilizando Processing 2: Um estudo das relações formalizadas por alunos do ensino fundamental com baixo rendimento em matemática.** Após a exposição, o discente foi arguido oralmente pelos membros da Comissão Examinadora, tendo recebido o conceito final: APROVADO. Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelos membros da Comissão Examinadora.

  
Prof. Dr. WILSON MASSASHIRO YONEZAWA  
Prof. Dr. NELSON ANTONIO PIROLA  
Profa. Dra. ELAINE PASQUALINI

*“Everybody in this country should learn how to program a computer... because it teaches you how to think” (Steve Job).*

*“Todos deveriam aprender a programar um computador... porque isto te ensina a pensar” (Steve Jobs).*

## **AGRADECIMENTOS**

Em primeiro lugar, a Deus, motivo maior de minha existência e força auxiliadora em minhas conquistas.

Ao Prof. Dr. Wilson Massashiro Yonezawa, que desde o tempo do Curso de Especialização em Tecnologia da Informação (CETI-Unesp/Bauru) vem contribuindo com minha trajetória profissional. Mergulhar numa prática desconhecida nem sempre é uma tarefa simples e muitas vezes nem fácil, por isso quero agradecer-lo imensamente pela orientação dedicada, por ter sempre aberto as portas do departamento de computação a mim, pelas cobranças necessárias, e pelo constante estímulo em todas as fases deste trabalho, pois sem estes elementos estaria certamente fadado a não acontecer.

À Secretaria Municipal de Educação de Santa Cruz do Rio Pardo pela oportunidade de realização desse estudo em uma das escolas de sua rede. Em especial, gostaria de agradecer a direção da unidade escolar que prontamente me autorizou a utilizar o Laboratório de Informática Educacional para realização desta pesquisa. Muito obrigado a todos os educadores e profissionais que me abriram as portas durante o trabalho de coleta de dados.

Aos amigos do convívio acadêmico e pessoal pela força, apoio, amizade, discussões pedagógicas e presença. Em especial as amigas Anália Maria Dias de Gois e Rosemeiry de Castro Prado que durante essa jornada compartilharam os perigos da estrada durante todas as idas e vindas. Agradecido por tornarem meu trajeto mais curto com seus risos.

À secretaria de Pós-Graduação da Faculdade de Ciência pela sempre tão solícita atenção em atender as demandas enquanto aluno e em especial a Denise Felipe por facilitar todos esses encaminhamentos.

À minha família e amigos, pela compreensão em minhas ausências e por abrirem caminhos ao meu desenvolvimento e crescimento profissional.

À Prof<sup>ª</sup>. Dra. Elaine Pasqualine, ao Prof. Dr. Nelson Antônio Pirola, à Prof<sup>ª</sup>. Dra. Nilcéia Aparecida Maciel Pinheiro pelas valiosas críticas e sugestões feitas no momento em que participaram das bancas do exame geral de qualificação.

## RESUMO

SOUZA, Eduardo C. **Programação no ensino de matemática utilizando Processing 2: Um estudo das relações formalizadas por alunos do ensino fundamental com baixo rendimento em matemática.** 2015. 189 f. Dissertação (Mestrado) – Faculdade de Ciências, Universidade Estadual Júlio de Mesquita Filho, Bauru, São Paulo, 2015.

O baixo desempenho em matemática obtido pelos alunos brasileiros nas avaliações externas revela que o ensino da matemática é um grande desafio tanto para as nações marcadas pela desigualdade social, quanto para as nações mais desenvolvidas. Em busca da superação dos baixos índices de rendimento dos estudantes em matemática, este trabalho de cunho qualitativo, investigou por meio da aprendizagem situada numa comunidade de prática de programadores as formas pelas quais os alunos com baixo desempenho em matemática se relacionam com a mesma durante e após participarem de oficinas de programação. A pesquisa contou com oito oficinas de programação utilizando a ferramenta Processing 2. Foi possível delinear seis categorias de análise que, conforme os objetivos da pesquisa, sinalizam arranjos da aprendizagem situada da matemática numa comunidade de prática de programadores - não linearidade, ênfase no saber fazer, informal, construção do conhecimento a partir da necessidade, interatividade e engajamento e tentou-se buscar subsídios para responder perguntas como: Quais são as vantagens de aulas de matemática utilizando ferramentas de programação? Quais são as dificuldades ao desenvolver aulas de matemática utilizando ambiente de programação? Que tipos de mudança o uso de ferramentas de programação provocam na dinâmica das aulas de matemática? Para a realização da pesquisa foi escolhida uma escola da rede municipal de ensino fundamental do município de Santa Cruz do Rio Pardo – SP, com seis alunos, na faixa etária de 12 anos que apresentavam baixo rendimento na disciplina. A programação de computadores como instrumento didático-pedagógico, no emprego da ferramenta Processing 2 aponta avanços na aprendizagem dos alunos, no tocante a promoção de um trabalho mais cooperativo, que coloca o erro como elemento natural no processo de aprendizagem, e torna a aprendizagem mais interativa, contribuindo para um feedback imediato, e avança sobretudo na apropriação de conceitos matemáticos adjacentes da atividade de programação. Foi possível evidenciar que no transcorrer das oficinas os alunos passaram a perceber a necessidade e importância da matemática enquanto constroem programas que representam seus anseios e desejos. Assim o conhecimento parte da necessidade pontual e, com isso, explorar uma aprendizagem mais ativa. No desencadeamento das oficinas os estudantes deixam de ser meramente receptores de informações, e se engajam de maneira ativa nas práticas da comunidade, o que contribui para uma aprendizagem ativa. Um dos indicativos de sucesso para a nova alfabetização do século XXI está na interatividade e no tempo rápido de respostas conforme se evidenciou neste trabalho. Enquanto os alunos trabalhavam no desenvolvimento dos projetos, criando as sequências de comandos, eles estavam a aprender sobre o processo de construção, do processo de como formalizar uma ideia e transformá-la num projeto completo e funcional. Assim, estavam a “programar para aprender”. À medida que se avança nas oficinas, os alunos aumentam sua atitude reflexiva, de modo que a ferramenta passou a ser apenas um suporte. Neste processo encontravam seus erros, procuravam corrigi-los, testavam e aprendiam os conceitos envolvidos na solução dos problemas.

**Palavras-chave:** Ensino de Programação. Resolução de Problemas. Pensamento Matemático. Aprendizagem Situada.

## ABSTRACT

SOUZA, Eduardo. C. **Programming in teaching math using Processing 2: A study of formalized relations for elementary students with low performance in mathematics.** 2015. 189 f. Dissertation (master's degree) - Faculty of Science, University Júlio de Mesquita Filho, Bauru, São Paulo, 2015.

The poor performance in mathematics obtained by Brazilian students in the external evaluation shows that mathematics teaching is a major challenge for both nations marked by social inequality, and for the most developed nations. In search of overcoming low levels of student performance in mathematics, the qualitative nature of work, investigated through learning located in a developer community of practice the ways in which students with low math performance relate to the same during and after participating in scheduling workshops. The research included eight workshops programming using Processing tool 2. It was possible to outline six categories of analysis that, as the research objectives, signal arrangements situated learning of mathematics in a developer community of practice - nonlinearity, emphasis on know-how, informal, knowledge building from the need, interactivity and engagement and we tried to get subsidies to answer questions such as: What are the advantages of math classes using programming tools? What are the difficulties in developing math classes using programming environment? What kinds of changes using programming tools cause the dynamics of math classes? For the research was chosen a school elementary school municipal system of the city of Santa Cruz do Rio Pardo - SP, with six students, aged 12 who had low performance in the discipline. The programming computers as didactic and pedagogical tool, the use of Processing 2 tool shows progress in student learning, regarding the promotion of a more cooperative work, which puts the error as a natural element in the learning process, and makes learning more interactive, contributing to an immediate feedback, and advancing above all in the ownership of adjacent mathematical concepts of programming activity. It became clear that in the course of the workshops students have come to realize the need and importance of mathematics as they build programs that represent their wishes and desires. Thus the knowledge of the specific need and, therefore, explore a more active learning. In triggering the workshops students are no longer merely information receptors, and to engage actively in community practices, which contributes to an active learning. One of the success indicative for the new literacy of the twenty-first century is on interactivity and quick response time as was evident in this work. While students worked in project development, creating the command sequences, they were to learn about the construction process, as the process of formalizing an idea and turn it into a complete and functional design. So were "programmed to learn." As we advance in the workshops, students increase their reflexive attitude, so that the tool just happened to be a support. In this process they found his mistakes, sought to correct them, were testing and learned the concepts involved in solving problems.

**Key-words:** Programming Education. Problem Solving. Mathematical Thinking. Situated Learning.

## LISTA DE ILUSTRAÇÕES

Figura 1- Níveis de proficiência em matemática. ....	16
Figura 2 - Comparativo de desempenho das edições do SARESP 2010 e 2014 - 9º Ano do Ensino Fundamental – matemática. ....	18
Figura 3 - Ambiente de programação Logo: Kturtle. ....	30
Figura 4 - Interface da primeira fase do jogo. ....	31
Figura 5 - Tabuleiro Code Monkey Island. ....	33
Figura 6 - Tabuleiro Robot Turtles. ....	33
Figura 7 - Interface kodable na web. ....	34
Figura 8 - Interface Robozzle. ....	35
Figura 9 - Interface do Cargo-Bot.....	36
Figura 10 - Ambiente Scratch (v.1.4). ....	39
Figura 11 – Estratégia de um voo senoidal. ....	40
Figura 12 - Voo Senoidal, $f(x)= 100 \cdot \text{sen } \theta$ ....	40
Figura 13 - Exemplo de aplicação em matemática. ....	46
Figura 14 - Setor circular com área $3/4$ da circunferência. ....	47
Figura 15 - Ambiente integrado de desenvolvimento Processing 2. ....	48
Figura 16 - Barra de Ferramentas Processing 2.....	49
Figura 17 - Tela de informações sobre o Processing. ....	49
Figura 18 - Referências Processing 2.....	50
Figura 19 - Sistema de coordenadas do Processing 2. ....	51
Figura 20 - Página no facebook Processing. ....	52
Figura 21 - Contribuição facultativa para download. ....	52
Figura 22 - Área de download Processing. ....	53
Figura 23 - Pasta descompactada.....	53
Figura 24 - Esquema da Aprendizagem Situada.....	59
Figura 25 - Aportes da teoria social da aprendizagem. ....	61
Figura 26 - Refinamento das intersecções de tradições teóricas. ....	62
Figura 27- Premissas da aprendizagem social de Wenger. ....	64
Figura 28 - Níveis de participação na comunidade de prática.....	67
Figura 29 - Ciclo aplicado em programação. ....	76
Figura 30 - Características do pensamento computacional. ....	77
Figura 31- Área de trabalho das versões LE 3.0 e 5.0, respectivamente. ....	79
Figura 32 - Gráficos sobre o perfil dos estudantes. ....	84
Figura 33 - Caminho metodológico percorrido .....	94
Figura 34 - Sistema de coordenada do Processing 2.....	106
Figura 35 - Código e execução do desafio inicial.....	109
Figura 36 - Segmentos de retas paralelos aluna KA. ....	109
Figura 37 - Programa do aluno MB. ....	111
Figura 38 - Desafio envolvendo formas elementares e cores.....	113
Figura 39 - Estratégia empregada pela aluna TS. ....	114
Figura 40 - Execução da atividade com arcos. ....	118
Figura 41 - Jogador Pacman. ....	119
Figura 42 - Código Pacman aluna NB. ....	120
Figura 43- Execução do programa do aluno MB.....	120
Figura 44 - Desafio envolvendo rotação e translação. ....	121
Figura 45 - Programa do aluno MB. ....	122
Figura 46- Alunos antes de iniciar a oficina 3.....	124

Figura 47 - Atividade sobre o conceito de variáveis.....	125
Figura 48 - Construção da circunferência utilizando variáveis. ....	128
Figura 49 - Erro no programa do aluno VH. ....	130
Figura 50 - Programa do aluno CA.....	132
Figura 51 - Programas dos alunos CA e MB.....	132
Figura 52 – Programa do aluno VH.....	134
Figura 53 - Estratégia utilizada pela aluna KA no desafio.....	136
Figura 54 - Execução do programa da aluna KA. ....	137
Figura 55 - Estratégia apresentada pelo aluno CA. ....	138
Figura 56 - Código da aluna TS.....	141
Figura 57 - Programa do aluno MB. ....	142
Figura 58 - Demonstração da execução do programa.....	142
Figura 59- Código do desafio apresentado pelo aluno VH. ....	144
Figura 60 – Código da aluna TS. ....	145
Figura 61 - Execução do programa da aluna <b>TS</b> . ....	146
Figura 62 - Código do programa do aluno CA. ....	148
Figura 63 - Execução do programa do aluno CA. ....	148
Figura 64 - Programa demonstração.....	150
Figura 65 - Utilização do operador NOT. ....	151
Figura 66 - Programa da aluna TS. ....	152
Figura 67 - Programa apresentado pela aluna NB. ....	153
Figura 68- Movimentação via teclado aluno KA. ....	155
Figura 69 - Problemática do círculo que não retornava ao centro.....	156
Figura 70 - Criação do retângulo proposta pelo aluno <b>CA</b> . ....	157
Figura 71 - Solução apresentada pela aluna KA.....	158
Figura 72 - Problemática dos limites da janela.....	159
Figura 73 - Programa do aluno CA, completo e funcional. ....	160
Figura 74 - Programa do aluno VH.....	163
Figura 75 - Solução apresentada pelo aluno para resolver o problema da velocidade. ....	164
Figura 76 - Execução do programa do aluno VH. ....	165
Figura 77- Dimensões observadas. ....	169

## LISTA DE QUADROS

Quadro 1 – Escala de proficiência em matemática no Pisa.....	15
Quadro 2 - Desempenho brasileiro por eixo, com média e distribuição percentual por nível de proficiência. ....	17
Quadro 3 - Análise do game segundo as características apontadas por Jane McGonical. ....	32
Quadro 4 - Características resumidas dos programas. ....	54
Quadro 5 - Alternativas para aprender programação.....	55
Quadro 6 - Elementos importantes da comunidade de prática. ....	56
Quadro 7- Elementos importantes da CoP no domínio do conhecimento. ....	63
Quadro 8 - Sujeitos da pesquisa.....	81
Quadro 9 - Datas e horários das oficinas.....	81
Quadro 10 - Etapas da pesquisa.....	87
Quadro 11 - O gosto pela matemática e suas facilidades/dificuldades .....	96
Quadro 12 - Motivação/desmotivação em estudar matemática.....	97
Quadro 13 - Comparação Motivação/desmotivação em estudar matemática .....	97
Quadro 14 - Experiências dos alunos em apreender e ensinar matemática. ....	99
Quadro 15 - Disciplina retirada da grade curricular pelos alunos. ....	100
Quadro 16 - Importância da matemática para a vida dos estudantes.....	101
Quadro 17 - Motivação dos alunos em relação ao uso de tecnologias. ....	102
Quadro 18 - O que aprendem enquanto jogam.....	103
Quadro 19 - Diferencial em uma aula de matemática para aumentar a motivação. ....	104
Quadro 20 - Episódio sobre números binários. ....	105
Quadro 21 - Atividade proposta no papel milimetrado.....	106
Quadro 22 - Episódio sobre as diagonais. ....	108
Quadro 23 - Processo de descoberta das formas básicas. ....	110
Quadro 24- Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 1. ....	112
Quadro 25 – Codificação envolvendo arcos.....	117
Quadro 26 - Episódio sobre pacman. ....	119
Quadro 27- Episódio dos problemas na construção do arco. ....	119
Quadro 28- Peculiaridade da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 2. ....	123
Quadro 29 - Interações na execução do desafio. ....	125
Quadro 30 - Dúvidas envolvendo variáveis .....	127
Quadro 31 - Pesquisa dos alunos na Internet.....	129
Quadro 32 - Questionamentos sobre condicional. ....	131
Quadro 33 - Discussão sobre o comando rect. ....	133
Quadro 34- Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 3. ....	135
Quadro 35 - Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 4.....	139
Quadro 36 - Impresso sobre as variáveis do sistema. ....	140
Quadro 37 - Dúvidas iniciais dos alunos sobre variáveis do sistema. ....	140
Quadro 38 - Compreendendo a função random.....	143
Quadro 39 - Dúvidas apresentadas no desafio das elipses. ....	144
Quadro 40- Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 5. ....	147
Quadro 41- Tabela verdade AND e OR. ....	149

Quadro 42 - Elaboração do desafio que envolve áreas específicas do plano. ....	150
Quadro 43 - Exemplo do cotidiano envolvendo operadores lógicos. ....	150
Quadro 44- Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 6. ....	154
Quadro 45 – Discussões sobre a criação da plataforma móvel inferior.....	156
Quadro 46 - Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 7.....	161
Quadro 47 - Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 8.....	165
Quadro 48 - Percepções dos alunos na aprendizagem situada. ....	170
Quadro 49 - Percepções da professora de matemática da turma. ....	171

## LISTA DE ABREVIATURAS E SIGLAS

AO	Objeto de Aprendizagem
API	<i>Application Programming Interface</i>
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CLP	<i>Controlador Lógico Programável</i>
COP	Comunidade de Prática
CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
JDK	<i>Java Development Kit</i>
LE	Linux Educacional
LIE	Laboratório de Informática Educacional
LP	Lógica de Programação
MEC	Ministério da Educação
MIT	<i>Massachusetts Institute of Technology</i>
MOOC	<i>Massive Open Online Course</i>
OA	Objeto de aprendizagem
OE	Objetivos Específicos
PHP	<i>Hypertext Preprocessor</i>
PISA	Programa Internacional de Avaliação de Estudantes
INEP	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
PNLD	Programa Nacional do Livro Didático
PPL	Participação Periférica Legítima
PPP	Projeto Político Pedagógico
RAD	<i>Rapid Application Development</i>
SARESP	Sistema de Avaliação de Rendimento Escolar de São Paulo
SO	Sistema Operacional
UE	Unidade Escolar
ZPD	Zona de Desenvolvimento Proximal

## SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO .....	13
1.1 Objetivos.....	26
1.2 Estrutura do Trabalho.....	26
CAPÍTULO 2 – REVISÃO DA LITERATURA .....	27
2.1 Programas que levam as crianças para o código .....	27
2.1.1 Logo .....	28
2.1.2 Light-bot.....	30
2.1.3 Code Monkey Island .....	32
2.1.4 Robot Turtles.....	33
2.1.5 Kodable .....	33
2.1.6 Robozzle.....	34
2.1.7 Cargo-Bot .....	35
2.1.8 Scratch .....	36
2.1.9 Code Club.....	41
2.1.10 Codecademy.....	42
2.1.11 Code.org.....	43
2.1.12 Processing 2.....	44
2.1.13 Quadro comparativo dos programas.....	53
2.2 Aprendizagem Situada.....	55
2.2.1 Aportes teóricos da aprendizagem social .....	59
2.2.2 A aprendizagem como participação social em Comunidade de Prática (CoP).....	63
2.2.3 Participação Periférica Legítima (PPL) .....	65
2.2.4 Algumas dimensões do conceito.....	67
2.2.5 Membros PPL numa comunidade de prática de programadores .....	70
2.2.6 Trabalhos correlatos .....	72
2.3 O papel do erro em programação.....	73
2.4 Programação de computadores como instrumento de ensino e o pensamento computacional .....	75
CAPÍTULO 3 – METODOLOGIA DA PESQUISA .....	78
3.1 O contexto .....	78
3.2 Os participantes .....	80
3.3 Procedimentos .....	85
3.4 Objetivos das oficinas de programação.....	88

3.4.1 Oficina 1 .....	89
3.4.2 Oficina 2 .....	89
3.4.3 Oficina 3 .....	89
3.4.4 Oficina 4 .....	89
3.4.5 Oficina 5 .....	90
3.4.6 Oficina 6 .....	90
3.4.7 Oficina 7 .....	90
3.4.8 Oficina 8 .....	91
3.4.9 Grupo focal final .....	91
3.5 Categorias de análise .....	91
I – Não linearidade .....	92
II – Ênfase no “Saber Fazer” .....	92
III – Informal.....	92
IV – Construção do conhecimento parte da necessidade .....	92
V – Interatividade.....	93
VI – Engajamento .....	93
4 - APRESENTAÇÃO DOS DADOS E DISCUSSÃO .....	95
4.1 Grupo focal inicial.....	95
4.2 - Oficinas .....	105
4.2.1 Oficina 1 – Introdução ao Processing 2.....	105
4.2.2 Oficina 2 – Exploração das formas geométricas e suas propriedades .....	113
4.2.3 Oficina 3 - Variáveis .....	124
4.2.4 Oficina 4 – Movimentação .....	135
4.2.5 Oficina 5 - Variáveis do sistema .....	139
4.2.6 Oficina 6 - Operadores lógicos .....	147
4.2.7 Oficina 7 – Utilização dos comandos Keypressed e KeyCode .....	155
7.2.8 Oficina 8 – Desafio da bolinha.....	161
4.3 Grupo focal final.....	169
CONCLUSÃO.....	173
REFERÊNCIAS .....	176
APÊNDICES .....	183

## CAPÍTULO 1 – INTRODUÇÃO

Atualmente, enquanto professor de Matemática do Ensino Fundamental e Médio, o autor deste trabalho tem vivenciado, no cotidiano escolar, um ensino da Matemática muitas vezes fragmentado e descontextualizado, desprovido de ligação entre a disciplina escolar e o mundo real, e que ainda despreza o uso de tecnologias. Com o advento da sociedade da informação e do conhecimento torna-se praticamente impossível falar de um processo de ensino e aprendizagem que não considere as influências das tecnologias, sendo sua utilização importante no processo educacional, contribuindo para formação integral do aluno.

Ainda, enquanto docente, observo que a concepção de ensinar matemática para solução de problemas, na qual primeiramente ensina-se matemática e posteriormente trabalha os problemas relativos àquele conteúdo, ainda hoje é muito utilizada pelos professores. Em contrapartida, ensinar matemática via solução de problemas, passa a ser visto como uma metodologia. O ponto inicial é uma situação-problema. Essa concepção ganhou força no Brasil com a publicação dos Parâmetros Curriculares Nacionais em 1998. Neste contexto:

A situação-problema é o ponto de partida da atividade matemática e não a definição. No processo de ensino e aprendizagem, conceitos, ideias e métodos matemáticos devem ser abordados mediante a exploração de problemas, ou seja, de situações em que os alunos precisem desenvolver algum tipo de estratégia para resolvê-las (BRASIL, 2001, p. 43).

Neste sentido, o ensino de matemática deve ser desenvolvido a partir de uma perspectiva diferenciada, ou seja, que tenha significado para o aluno. Trata-se de uma disciplina escolar em que grande parte dos conceitos estudados em sala de aula pode ser desenvolvida a partir da participação ativa, com foco nos processos interativos entre os alunos e o professor, apoiados por tecnologia em detrimento dos processos que privilegiam a memorização e a repetição.

Entusiasmado com uso de novas tecnologias para diversos fins, especialmente aquelas que envolvem o processo de ensino e aprendizagem, este pesquisador, com formação na área computacional, a partir das preocupações observadas foi buscar métodos de ensino que têm como suporte um instrumento didático-pedagógico que é a programação de computadores articulada com a concepção de ensinar matemática via solução de problemas, tendo em vista, o baixo rendimento apresentado pelos alunos deste nível, detectados em determinados indicadores, tanto nacionais quanto internacionais.

O desempenho em matemática de alunos brasileiros no Programa Internacional de Avaliação de Alunos (PISA, em inglês), em 2012, foi decepcionante. O Brasil ocupou a 58ª

posição no ranking dentre 65 países participantes, assegurando 391 pontos, abaixo da média, de 494. A pontuação brasileira só foi melhor que a de Argentina, Tunísia, Jordânia, Colômbia, Catar, Indonésia e Peru (INEP, 2014).

A avaliação do PISA tem como objetivo gerar indicadores que contribuam para a discussão da qualidade da educação e conseqüentemente, respaldar políticas públicas para a melhoria da educação básica, e no Brasil é coordenado pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP).

A avaliação é aplicada a estudantes na faixa etária dos 15 anos, idade na qual presume-se a finalização da educação obrigatória na maioria dos países. Realizada a cada três anos, abrange três áreas do conhecimento: Leitura, Matemática e Ciências. A cada edição do programa projeta-se maior ênfase em uma das áreas. As últimas edições com foco em matemática ocorreram em 2003 e 2012.

De acordo com o nível de complexidade de cada item que compõe a avaliação é possível estabelecer níveis de proficiência da área de conhecimento. Em matemática, os níveis do Pisa são apresentados resumidamente no Quadro 1.

Quadro 1 – Escala de proficiência em matemática no Pisa.

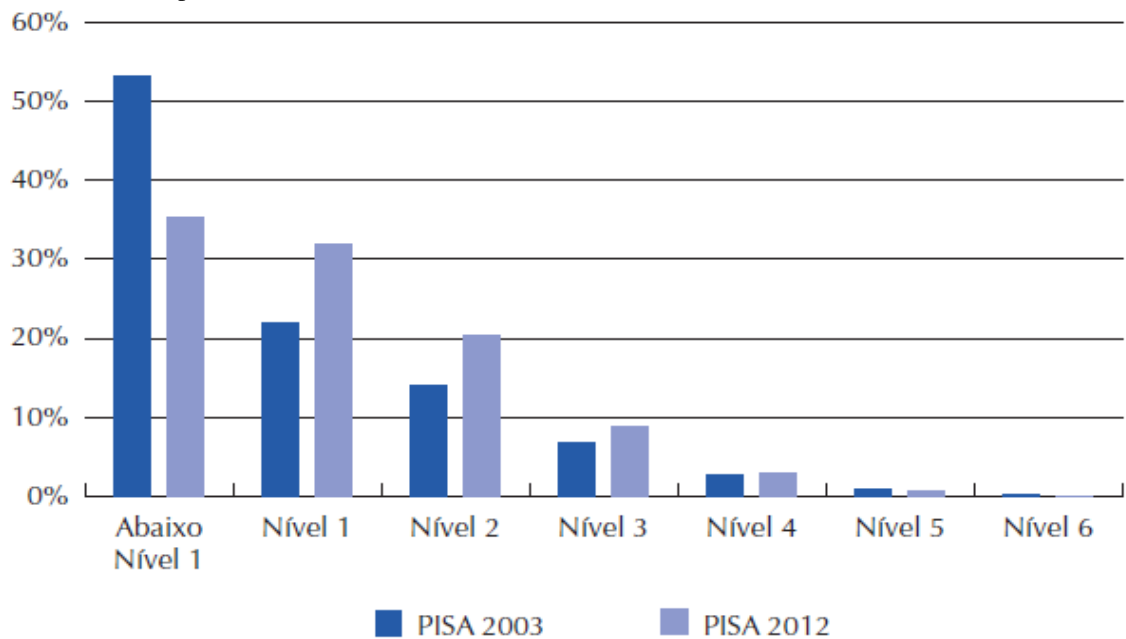
<b>Nível</b>	<b>Pontos (Limite inferior)</b>	<b>Características</b>
6	669,3	No mais alto nível, os alunos são capazes de conceituar, generalizar e utilizar informações com base em suas investigações. Conseguem realizar conexões entre diferentes fontes de informação e representações. Com avançado pensamento matemático e raciocínio, formulam e comunicam precisamente suas ações e reflexões relacionadas com constatações, interpretações e argumentos, bem como são capazes de adequá-las às situações originais.
5	607,0	Neste nível, os alunos são capazes de desenvolver modelos para situações complexas, o que possibilita selecionar, comparar e avaliar estratégias apropriadas na resolução de problemas, aptos a refletirem sobre suas ações e comunicarem suas interpretações e raciocínio.
4	544,74	Neste nível, os alunos podem trabalhar de forma eficaz com modelos explícitos para situações concretas complexas, que podem exigir formulação de hipóteses. Eles são capazes de selecionar e integrar diferentes representações empregando as habilidades desenvolvidas com flexibilidade.
3	482,4	Neste nível, os alunos são capazes de executar procedimentos, incluindo aqueles que exigem decisões sequenciais. São capazes de selecionar e aplicar estratégias simples de resolução de problemas e conseguem interpretar e utilizar representações baseadas em diferentes fontes de informação.
2	420,1	Neste nível, os alunos são capazes de extraírem informações relevantes de uma única fonte e usar um único modo de representação. Assim, conseguem empregar algoritmos, fórmulas, procedimentos ou convenções de nível de entrada.
1	357,8	Neste nível, os alunos são capazes de responder a perguntas claramente definidas envolvendo contextos conhecidos. Conseguem identificar informações e realizar procedimentos de rotina de acordo com instruções diretas em situações explícitas.
Abaixo de 1		Não especificado.

Fonte: INEP (2014).

Com foco em matemática, quando são confrontados os níveis de proficiência entre as edições de 2003 e 2012, observa-se uma redução significativa no número de estudantes situados abaixo do Nível 1, bem como um acréscimo no número daqueles situados no Nível 2. O relatório aponta uma preocupação com o fato que não houve crescimento no número de estudantes nos níveis mais altos de proficiência (Níveis 5 e 6), o que pode indicar que o país não está se preparando para formar adequadamente estudantes para funções mais complexas, que demandam conhecimento da matemática, como as áreas da engenharia e áreas afins.

O gráfico (Figura 1) mostra a comparação entre as edições de 2003 e 2012 no que se refere aos níveis de proficiência em matemática.

Figura 1- Níveis de proficiência em matemática.



Fonte: INEP (2014).

Observa-se que há no Brasil um déficit de estudantes que contemplam habilidades e competências de avançado nível de pensamento e raciocínio matemático. Estudantes nesse estágio são capazes de conceituar, generalizar, estabelecer ligações entre diferentes fontes de informação e representações, e transitar com flexibilidade entre elas. De acordo com o relatório INEP (2014, p. 20), os alunos dos mais elevados níveis são capazes de “associar sua percepção e compreensão a um domínio de operações e relações matemáticas simbólicas e formais, de modo a desenvolver novas abordagens e estratégias para enfrentar novas situações”.

Outro ponto que merece destaque, é que os resultados nacionais mostram que os estudantes brasileiros têm melhor desempenho na área de indeterminação e dados, apresentando uma diferença de cerca de 30 pontos quando comparados com a área de mudanças e relações, na qual registrou o pior desempenho. Isso significa em termos matemáticos, que os estudantes brasileiros possuem dificuldades de estabelecerem relações por meio de funções e equações adequadas.

O conteúdo Espaço e Forma também aparece entre os eixos em que os estudantes brasileiros apresentam mais dificuldades (Quadro 2), o que permite apontar problemas no ensino e aprendizagem da geometria enquanto fundamento do referido eixo, com mais de 40% dos alunos abaixo do Nível 1.

Quadro 2 - Desempenho brasileiro por eixo, com média e distribuição percentual por nível de proficiência.

Conteúdo	Média	EP	Abaixo Nível 1	Nível 1	Nível 2	Nível 3	Nível 4	Nível 5	Nível 6
Indeterminação e dados	402,1	2,0	26,5	35,1	25,5	10,0	2,5	0,3	0,0
Quantidade	392,9	2,5	36,5	27,0	20,2	10,5	4,3	1,3	0,2
Espaço e forma	380,7	2,0	40,3	30,6	18,8	7,3	2,4	0,6	0,1
Mudanças e relações	371,5	2,7	46,3	24,0	16,5	8,4	3,3	1,1	0,3

Fonte: INEP (2014).

Com base nos dados apresentados é possível verificar que as maiores dificuldades dos alunos brasileiros estão relacionadas com as subáreas da matemática Mudanças e Relações e Espaço e forma.

No eixo Mudança e Relações requer-se que os estudantes compreendam os tipos fundamentais de mudança e reconheçam quando tais mudanças ocorrem, de forma a utilizar modelos matemáticos para descrevê-las. Em termos matemáticos, isto significa modelar essas relações por meio de funções, bem como criar, interpretar e transitar entre as diversas representações gráficas e simbólicas (INEP, 2014).

No eixo Espaço e Forma, a geometria é considerada um fundamento primordial. De acordo com Van Hiele (1986), a geometria é um ramo da matemática que estuda as formas, plana e espacial, bem como suas propriedades, ou ainda, ramo da matemática que estuda a extensão e as propriedades das figuras (plana) e dos sólidos (espacial). Assim, é fundamental que os alunos reconheçam as formas geométricas presentes no mundo real, para formalização e construção de conceitos dentro da geometria. Os estudantes devem ser capazes de compreender uma variedade de fenômenos do mundo físico e visual, como: padrões, representações e propriedades dos objetos, posição e orientação, interação dinâmica com formas reais, entre outras habilidades (INEP, 2014).

Ainda, sobre o eixo Espaço e Forma, Van Hiele (1986) destaca em sua teoria que a aprendizagem em geometria acontece por níveis graduais de pensamento, em que a visualização e o reconhecimento pertencem ao nível elementar, e preliminar para o entendimento da formalização do conceito geométrico. A ênfase na visualização e no reconhecimento geométrico leva-nos a trabalhar essa habilidade nos espaços escolares atentos a expressiva importância.

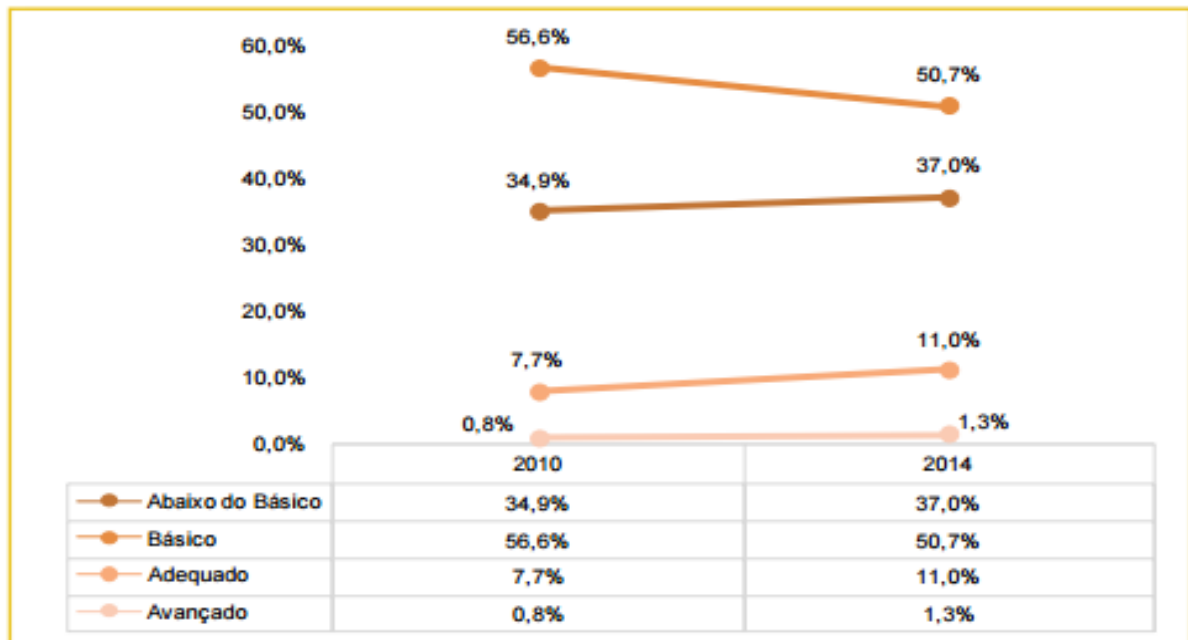
A partir desses resultados preocupantes, que colocam o ensino da matemática escolar no Brasil num patamar inferior, emerge o seguinte questionamento: o que fazer para mudar esse cenário? Uma resposta é evidente: o Brasil precisa encontrar formas de trabalhar com esses alunos de baixo rendimento e promover um ensino comprometido, atento às necessidades deles e ao nível de proficiência de cada um e que saibam superar situações como: baixo desempenho escolar, desmotivação, repetência, evasão e baixa renda.

O baixo resultado obtido pelos brasileiros revela que o ensino da matemática é um grande desafio tanto para as nações marcadas pela desigualdade social, quanto para as nações mais desenvolvidas.

Esses indicadores do PISA não são diferentes daqueles que outras avaliações em larga escala têm apontado, como por exemplo, o SARESP no Estado de São Paulo.

Um comparativo dos estudantes dos 9<sup>os</sup> anos nas edições do SARESP 2010 e 2014 aponta que o desempenho em matemática apresentou melhora em 2014, mas está longe do nível considerado “adequado” para o encerramento de um ciclo, e, conseqüentemente, para o ingresso dos estudantes no Ensino Médio (SÃO PAULO, 2015).

Figura 2 - Comparativo de desempenho das edições do SARESP 2010 e 2014 - 9º Ano do Ensino Fundamental – matemática.



Fonte: SÃO PAULO (2015).

Observa-se (Figura 2) que o percentual de alunos no nível Abaixo do Básico é maior em 2014, porém há um indicativo de trabalho que envolve recuperar os estudantes que apresentam baixa proficiência, fornecendo-lhes subsídios para que cada vez mais estudantes atinjam a proficiência Adequada ou Avançada (SÃO PAULO, 2015).

Estes resultados são preocupantes, uma vez que para muitos dos estudantes, ainda há conceitos elementares mal compreendidos ou assimilados que se arrastam para além do Ensino Fundamental, como por exemplo: resolver problemas modelados por uma função do 1º grau ou identificar e resolver sistemas lineares, entre outros (SÃO PAULO, 2015).

Cortella (2014) aponta sobre os novos paradigmas da educação e enfatiza que hoje as mudanças são cada vez mais velozes, a memória torna-se cada vez mais fugaz e a história muito rápida. Assim, faz-se necessário um novo olhar, um novo jeito de pensar, e

especialmente de construir a educação. Os tempos mudaram, assim também os alunos, as ferramentas, e as habilidades necessárias (conhecimento).

Segundo o Relatório Nacional Pisa 2012, o domínio da matemática aparece como habilidade essencial para enfrentar os desafios futuros, assim:

A matemática é um elemento fundamental na preparação dos jovens para a vida moderna, permitindo que enfrentem desafios em sua vida profissional, social e científica. Espera-se que os jovens desenvolvam capacidade de raciocínio matemático, utilizem ferramentas e conceitos matemáticos; que sejam capazes de descrever, explicar e prever fenômenos (INEP, 2014, p.18).

O relatório mostra a necessidade de utilização da matemática em uma situação contextualizada e, para que isso seja possível, é importante que a experiência em sala de aula seja suficientemente rica, ofertando um pluralismo metodológico.

Em seu trabalho Laburu e Carvalho (2001) destacam algumas estratégias que podem contribuir para a uma proposta pluralista em sala de aula. A recomendação de selecionar variadas estratégias, muito provavelmente deva estimular mais os estudantes e, como consequência, promover com maior certeza a competência do aprendiz. Os autores atentam que o maior perigo em uma proposta pedagógica é assumi-la como um único pensamento gerador das melhores estratégias de ensino e aprendizagem comum a todos os aprendizes.

Considerando esses aspectos, o letramento em matemática no PISA 2012 é definido como:

A capacidade do indivíduo de formular, aplicar e interpretar a matemática em diferentes contextos, o que inclui o raciocínio matemático e a aplicação de conceitos, procedimentos, ferramentas e fatos matemáticos para descrever, explicar e prever fenômenos. Além disso, o letramento em matemática ajuda os indivíduos a reconhecer a importância da matemática no mundo, e agir de maneira consciente ao ponderar e tomar decisões necessárias a todos os cidadãos construtivos, engajados e reflexivos (INEP, 2012, p.18).

Os estudantes com letramento em matemática ao se depararem com um problema inserido num contexto, são capazes de transitar para um problema matemático, susceptível de uma solução matemática, utilizando-se de conceitos, ferramentas e procedimentos matemáticos para obter resultados matemáticos. Logo, são capazes de interpretar esses resultados e estimá-los. Os processos de formular, empregar e interpretar são centrais no ciclo de modelagem matemática, e são habilidades dos indivíduos com letramento em matemática.

O termo letramento é cada vez mais empregado no Brasil. Mas em que difere da alfabetização? Segundo Soares (2004) os dois conceitos apresentam diferenças fundamentais, pois estão relacionados com concepções distintas.

O termo letramento aparece relacionado à compreensão da leitura e escrita como práticas sociais, que privilegia a visão de língua que se usa durante a comunicação. O termo alfabetização relaciona-se à concepção de escrita como sistema ordenado pelas regras gramaticais, ou mesmo de escrita como código, que é preciso decifrar (SOARES, 2004).

Apesar das diferenças apresentadas, Soares (2014) sustenta-se na indissociabilidade entre alfabetização e letramento. Não basta apenas alfabetizar, isto é, ensinar os aspectos da língua como código, também é imprescindível trabalhar a língua em seus usos sociais. Assim, a escola deve trabalhar com os dois processos simultaneamente.

Alguns pesquisadores, como Prensky (2005, 2008, 2010), Resnick (2012, 2013), Rushkoff (2012), sobre o futuro da educação discutem como seria a alfabetização no século XXI e questionam sobre quais são as principais habilidades que os indivíduos devem possuir para serem considerados alfabetizados.

Alguns autores propuseram outros tipos de alfabetizações. Kress e Leeuwen (2006) abordam a alfabetização visual que inclui competências relacionadas ao suporte multimídia da linguagem que, segundo Martino (2005), refere-se ao armazenamento, veiculação, percepção e engloba estruturas de código visual, auditivo e tátil. Outro, como Squire (2008) vai além, e aponta que a alfabetização emergente está na interatividade e nos games.

Resnick (2013) enxerga a codificação (programação de computadores) como a nova alfabetização desse século, na qualidade de extensão da escrita. A programação não está substituindo a escrita, mas é bastante entrelaçada com ela e direciona para uma pedagogia mais abrangente. A capacidade de codificação permite escrever novos tipos de coisas - histórias interativas, jogos, animações e simulações. E, como acontece com a escrita tradicional, há fortes razões para que todos possam aprender o código.

Para Prensky (2008) a fluência em línguas faladas continuará a ser importante, assim como a interatividade e outros dispositivos oriundos de jogos multimídia serão ferramentas indispensáveis para a comunicação do pensamento no século XXI. No entanto, para o autor a verdadeira alfabetização do novo século se encontra fora de todos estes domínios. Para ele o que distinguirá uma pessoa alfabetizada é a alfabetização em programação, a capacidade de fazer a tecnologia digital. Essa habilidade é conhecida como interação homem-máquina ou simplesmente programação.

Atualmente, as pessoas com habilidades altamente desenvolvidas na área de programação são vistas como nerds<sup>1</sup>. Para Prensky (2008) assim como as máquinas se tornaram componentes importantes na comunicação, trabalho, educação e nos lares, a capacidade de programá-las, ensinando-as a fazer o que se quer, tornará uma habilidade cada vez mais valiosa e essencial.

Deste modo, a programação torna-se uma habilidade fundamental, atributo de classes sociais e intelectuais superiores, assim como a leitura e a escrita fez no passado. Alguns séculos atrás, a leitura e a escrita foram restritas a uma pequena classe de especialista, os escribas (PRENSKY, 2008).

Com isso, emergem alguns questionamentos: Será que indivíduos com formação adequada realmente devem contemplar habilidades de programação? Se em algum momento precisarem de programação não é possível comprá-la?

Para o autor, na idade média ou no antigo Egito um indivíduo, para comunicar seus pensamentos no papel, não poderia fazê-lo sozinho, sendo necessário o auxílio de um escriba, alguém que conhecia o código escrito. Na outra extremidade, precisava de alguém para ler ou decodificá-lo, a menos que o indivíduo dominasse as competências escritoras e leitoras.

De acordo com Prensky (2008), tudo caminha para que neste século, não exista necessidade de uma tribo escriba de programadores, uma vez que o conjunto de habilidades que um indivíduo necessita para ser considerado alfabetizado incluirá, tão logo, domínio de programação.

Atualmente telefones móveis, carros, geladeiras já exigem certo conhecimento de programação. Em breve a programação será a forma como interagimos com todos os objetos, e um componente importante da forma como interagimos uns com os outros (PRENSKY, 2008).

Há uma hierarquia de níveis de linguagem de programação. As linguagens de programação geralmente são divididas em três categorias. Uma linguagem de programação de baixo nível representa diretamente as instruções em código de máquina (representação binária) e é projetada de modo a ser compreendida pela máquina em lugar de ser entendida pelas pessoas.

---

<sup>1</sup> Nerd é uma expressão utilizada desde o final da década de 1950 no Massachusetts Institute of Technology (MIT), atribuída àqueles indivíduos que trabalhavam no laboratório de tecnologia, que passavam noites em claro nas suas pesquisas. Na década de 1960 difundiu-se a sua conotação pejorativa, aplicado a pessoas com inteligência geralmente acima da média, com alguma dificuldade em se relacionar socialmente. PEREIRA, Camila. A Redenção dos Nerds. **Revista Veja**, Editora Abril, Edição 2050, ano 41, n. 9, 05 mar. 2008. Acessado em novembro de 2015: <http://www.fndc.org.br/clipping/a-redencao-dos-nerds-233145/>.

As linguagens de baixo nível não apresentam sentenças em idioma utilizado por seres humanos, portanto, elas são mais difíceis de aprender. Todavia, as linguagens de baixo nível permitem uma execução mais rápida, como o código Assembler.

Por outro lado, uma linguagem de alto nível é elaborada para ser mais facilmente compreendida pelas pessoas. As palavras-chaves e funções de uma linguagem de alto nível são, em geral, muito semelhantes a uma sentença em língua inglesa e são usualmente mais fáceis de aprender, ler e compreender, como a linguagem Java. No entanto, as linguagens de alto nível necessitam ser compiladas e convertidas em um programa equivalente em linguagem binária de máquina (MONTEIRO, 2007).

As futuras gerações estarão satisfeitas por muito pouco tempo na qualidade de consumidores de tecnologia, programando em nível de configuração (Menu de opções). Assim, crianças do século XXI são direcionadas a se tornarem programadores em algum grau. Para preparar as crianças para a vida neste século, temos que ajudá-las a potencializar suas ferramentas, estendendo suas habilidades de programação (PRENSKY, 2005).

Atualmente crianças aprendem linguagens de programação como o Scratch (descrito no item 2.1.8) e posteriormente migram para linguagens mais poderosas. A autoaprendizagem de programação pode ser definida como o processo do aluno autoadquirir conhecimento e o desenvolvimento da lógica de programação é vista como a “viga mestra” para tais ramificações. A autoaprendizagem em programação possibilita a essas crianças se expressarem na língua de seus próprios tempos.

Segundo Rushkoff (2012) a alfabetização em programação é requisito para a participação em um mundo digital. Para ele, quando se adquire a linguagem, não apenas se aprende a ouvir, mas também como falar. Quando se apropria da habilidade leitora, não apenas se aprende a ler, mas também a forma de escrever. Porém, quando se aprende a usar computadores não se aprende como programa-los. Assim, quando não alfabetizados a codificar têm que aceitar os softwares com as limitações e agendas desenvolvidas por seus criadores, pois não temos habilidades para customizar tais softwares.

Todos possuem ideias e necessidades passíveis de soluções em programação. Assim, ser fluente digitalmente vai muito além de saber fazer uso de ferramentas tecnológicas na condição de consumidor. Espera-se saber como construir coisas significativas por meio dessas ferramentas (RESNICK, 2012).

Recentemente, as escolas ao redor do mundo têm se preocupado em realizar o letramento de suas crianças e adolescentes em uma “nova alfabetização”. Crianças do ensino fundamental estão aprendendo fundamentos da lógica de programação por meio de jogos,

plataformas online e ferramentas visuais, melhorando continuamente suas habilidades à medida que avançam por meio das séries. A ferramenta Scratch desenvolvida no MIT é um bom exemplo de apoio (RESNICK, 2009).

O ambiente gráfico de programação utiliza a mesma metáfora do brinquedo LEGO. O Scratch ajuda os jovens a aprender a pensar de maneira criativa, refletir de maneira sistemática e trabalhar de forma colaborativa, habilidades essenciais para a vida no século XXI.

Utilizando-se da lógica de programação visual, por meio de blocos lógicos que se encaixam, a ferramenta proporciona uma lógica de programação mais simples, lúdica e intuitiva, diferenciando das linguagens tradicionais. A ferramenta é fornecida gratuitamente e destina-se a crianças a partir dos oito anos e foi desenvolvida pela equipe Lifelong Kindergarten do MIT Media Lab e coordenada por Mitchel Resnick (EDUSCRATCH, 2015).

A programação está se tornando mais fácil a cada dia. Os *premade*<sup>2</sup> de objetos de programação, disponíveis na Internet, e na sua maioria livre, podem simplesmente serem conectados para executar determinadas tarefas. Segundo Prensky (2008) esses bancos de dados de *premade* aumentam consideravelmente as capacidades dos alunos, possibilitando ampliar suas capacidades de programação e de resolução de problemas, e encurtar o tempo para fazer as coisas. Em certo sentido, esses pedaços de código são como um alfabeto de programação.

Com pedaços de código cada vez mais acessíveis na web, expandem as oportunidades envolvendo programação. Muitos jovens, na busca de atender as suas necessidades e desenvolver seus próprios jogos, aventuram-se nas linguagens de programação mais sofisticadas como Java e C++. Assim surge a nova tribo escriba de programadores e torna-se a elite intelectual do século XXI.

À medida que o século avança, aqueles que não contemplam habilidades de programação poderão ter grandes dificuldades na vida cotidiana. Para Prensky (2008) os jovens cidadãos do futuro não terão respeito aos adultos que não conseguirem programar seu próprio smartphone, um computador, ou qualquer outra coisa. As crianças de hoje já enxergam seus pais e professores como analfabetos.

Se a programação é de fato a alfabetização chave deste século, como nós, enquanto educadores, faremos alunos alfabetizados? Essa questão não será enfoque da análise que se

---

<sup>2</sup> Premade de objetos de programação refere-se a pedaços de código construídos antecipadamente que podem ser conectados aos programas para executar tarefas específicas. Acessado em setembro de 2015: <http://www.collinsdictionary.com/dictionary/english/premade>.

pretende realizar neste trabalho. Muitos dos professores, até mesmo os mestres em matemática e ciências, não possuem as habilidades necessárias, mesmo as mais elementares em programação. A maioria das ferramentas e até mesmo o conceito de programação foram desenvolvidos muito tempo depois que esses professores nasceram ou foram escolarizados.

Assim, a alfabetização pertencerá àqueles que podem dominar uma variedade de poderosas e expressivas interações homem-máquina.

Atento às exigências na nova alfabetização do século XXI, no início de segundo semestre de 2014, escolas públicas da Inglaterra começaram a inserir seus alunos mais jovens no mundo da programação.

O governo do Reino Unido realizou uma reestruturação na maneira que se ensina computação para crianças no país, acrescentando aulas obrigatórias de programação. Convencidos de que o currículo estava em dissonância com as exigências requeridas para este século, o governo pretende formar não apenas consumidores de tecnologia, mas desenvolvedores. Sendo o primeiro país do G20<sup>3</sup> a colocar a ciência da computação no centro do seu currículo, crianças de cinco anos vão jogar games que ajudam a desenvolver o pensamento algoritmo e computacional sem a complexidade exigida em uma linguagem de programação convencional. No momento em que o aluno atingir a idade de 14, os professores irão orientá-los sobre como usar duas ou mais linguagens de programação profissional, uma vez que possuirão o pensamento computacional bastante desenvolvido (BLOOMBERG, 2014).

Da nova política pública educacional Inglesa, pode-se observar que entre a idade de cinco a quatorze anos, em que o estudante é orientado a utilizar linguagem de programação, transcorreu-se aproximadamente 10 anos. Segundo Prensky (2010) estudiosos, como Howard Gardner e Malcolm Gladwell, apontam em suas pesquisas que leva cerca de 10.000 horas, alguns dizem 10 anos para tornar-se realmente especialista em alguma coisa. Assim, ao término desse período, alunos ingleses contemplarão habilidades e competências requeridas em programação.

Como muitos países desenvolvidos, o Reino Unido enfrenta uma crise pela falta de profissionais da área de tecnologia. O país projeta uma carência de 249 mil trabalhadores para empregos qualificados tecnologicamente até 2020, conforme dados da pesquisa empírica preparada pela comissão europeia (BLOOMBERG, 2014).

---

<sup>3</sup> G20 é o grupo constituído pelos 20 países de economia mais desenvolvidos no mundo, para debater questões relacionadas a assuntos financeiros internacionais.

Para especialistas, aprender o código permite as crianças construir suas próprias coisas, ser criativo, e futuramente conseguir um emprego em uma área na qual haverá um déficit de profissionais. Assim, a fluência em novas tecnologias, abre um leque de aprendizagem:

A capacidade de escrever programas de computador é uma parte importante da alfabetização na sociedade de hoje. Quando as pessoas aprendem a programar, elas aprendem estratégias importantes para a resolução de problemas, projetos de design e comunicação de ideias (EDUSCRATCH, 2015).

Enquanto as crianças aprendem a programar, o mais importante é que estão a programar para aprender. Para exemplificar, Resnick (2012) faz uma analogia com a leitura e escrita. Para o autor, quando se aprende a ler e a escrever criam-se oportunidades para aprender outras coisas. Quando se aprende a ler, pode-se então ler para aprender. Assim, a mesma coisa acontece para a programação. Quando se aprende por meio da programação e programa-se para aprender, está a aprender num contexto significativo, e esta é a melhor forma de aprender (RESNICK, 2012).

Não sabemos se um aluno vai se transformar em um profissional de programação, porém independentemente do que ele fizer em sua vida profissional, ele poderá fazer uso dessas competências que adquiriu. Resnick (2012) destaca que mais uma vez é necessário fazer uma analogia a um idioma. Quando se aprende a ler e a escrever, não há intenção de se tornar escritores profissionais. Na realidade, poucos se tornarão escritores profissionais, mas é útil para todos que aprendam a ler e a escrever. Assim, o mesmo acontece com a programação. Quando essa geração crescer não serão programadores profissionais, mas terão a capacidade de pensar de forma criativa, com pensamento sistêmico para resolução de problemas, com habilidade de trabalhar em colaboração. Todas essas competências são desenvolvidas enquanto se aprende a programar.

Considerando as dificuldades dos alunos brasileiros frente à aprendizagem matemática, os resultados ruins nas avaliações externas e a tendência da nova alfabetização, este trabalho norteia-se pelo seguinte questão problema: como o conhecimento de programação melhora e/ou potencializa o conhecimento em matemática dos alunos do ensino fundamental que possuem baixo rendimento na disciplina.

## 1.1 Objetivos

O objetivo deste trabalho é investigar as relações formalizadas por alunos do ensino fundamental, com baixo rendimento em matemática, por meio da aprendizagem situada numa comunidade prática de programadores, apoiados por linguagem de programação visual.

Utilizando-se de fundamentos da aprendizagem situada, participação periférica legítima, comunidades de prática pretende-se atingir os seguintes Objetivos Específicos (OE):

OE1- apontar como os alunos percebem a necessidade e/ou importância da matemática enquanto constroem “coisas”, e como a programação interfere em sua motivação na aprendizagem matemática.

OE2- identificar, à medida que os alunos aprendem programação, quais são os conceitos matemáticos implícitos nessa construção que os mesmos se apropriaram, as relações estabelecidas, e as facilidades e/ou dificuldades que demonstraram durante esse processo.

OE3- explorar a aprendizagem ativa em matemática, com feedback rápido e isento da noção de pré-requisito em programação.

## 1.2 Estrutura do Trabalho

Este estudo está organizado para responder a questão norteadora apresentada, e contempla as seguintes seções: Capítulo 2. *Revisão da literatura*, no qual se realiza uma apresentação de programas que levam as crianças para o código, que ajuda a desenvolver o pensamento computacional, produzindo um quadro comparativo entre essas ferramentas, que justifica a escolha da ferramenta empregada nesta pesquisa. Ainda nesta seção será apresentada a teoria da Aprendizagem Situada, seus principais pressupostos, delimitações e dimensões do conceito. Capítulo 3. *Metodologia de Pesquisa* mostra as escolhas metodológicas e os caminhos percorridos para obtenção dos objetivos. O Capítulo 4 *Apresentação dos dados e discussões* mostra o desenrolar de cada uma das oito oficinas de programação, evidenciando as relações formalizadas pelos estudantes em uma comunidade de prática de iniciação em programação de computadores, sobretudo no tocante a apropriação de conceitos matemáticos adjacentes à atividade de programação. Ao final, a *conclusão* descreve se o alcance dos objetivos traçados foram atendidos, as contribuições para a educação matemática, os questionamentos que ainda permanecem e sugestões para continuidade deste trabalho.

## **CAPÍTULO 2 – REVISÃO DA LITERATURA**

Descreve-se neste capítulo doze ferramentas que ensinam programação de computadores seguida de um quadro comparativo. Discorre sobre a aprendizagem situada numa comunidade de prática de programadores, o papel do erro em programação e aspectos da programação como instrumento de ensino, bem como resgata alguns trabalhos correlatos.

### **2.1 Programas que levam as crianças para o código**

Em todo o mundo, crianças têm participado de atividades que ensinam programação. O conhecimento básico de conceitos de programação melhora a resolução de problemas, entre outras habilidades. O Presidente da Microsoft, Bill Gates disse recentemente que aprender a escrever programas estende a mente e ajuda a pensar melhor, cria maneiras de pensar sobre as coisas e é útil em todos os domínios (CODE, 2015).

Há uma variedade de plataformas que dão suporte para que crianças criem seus próprios jogos e animações utilizando métodos de codificação simplificada. Antes de se aventurar nessas plataformas, uma abundância de jogos ensina as noções básicas de programação de forma divertida e acessível.

A programação de computadores, até pouco tempo, era destinada a um grupo de especialistas que dominavam competências para realizá-las. Como se observa na justificativa deste estudo, especialistas defendem que a programação é uma atividade acessível a todos que passam a enxergar a ciência da computação como uma arte liberal e que deve ser ensinada desde muito cedo às crianças.

Segundo Alves (2011), o conceito de artes liberais refere-se ao conjunto de estudos que provê conhecimentos, métodos e habilidades intelectuais gerais aos seus estudantes, ao invés de focar em especializações. Ao analisar o contexto histórico do ensino de programação de computadores, é possível observar que, desde as experiências com a linguagem Logo de Papert até os dias de hoje, existe um forte argumento a favor da inclusão dessa disciplina nas escolas de ensino regular. Trabalhos publicados por Valente (1993, 1997, 1999), já assumiam essa perspectiva.

As atividades de programação auxiliam o aprendiz a alcançar a fase de compreensão de conceitos. Ele pode refletir sobre os resultados de suas ações e ideias, e essa reflexão é o mecanismo pelo qual o aprendiz se torna consciente de seu conhecimento e, assim, pode transformar seus esquemas mentais em operações mais complexas (VALENTE, 1999).

As crianças, hoje, nascem imersas em um mundo digital, mas, ao contrário do que se possa imaginar, elas não conhecem o funcionamento desse mundo, apenas utilizam suas ferramentas passivamente, como consumidores de tecnologia. Aprender uma linguagem de programação nos dias atuais é tão importante quanto aprender a ler e escrever, assim qualquer pessoa que saiba criar um programa de computador será capaz de criar a realidade em que o restante estará inserido (RUSHKOFF, 2012).

Essas afirmações reforçam a importância da aprendizagem das metodologias e técnicas utilizadas para a programação de computadores não só para quem deseja se profissionalizar na área, mas por todas as pessoas.

Como a programação se tornou uma habilidade extremamente útil, diversas iniciativas públicas e privadas legitimam o seu ensino nas escolas regulares. Para os especialistas aprender programação no século XXI é tão importante quanto ler e escrever foram no século XX, se configurando como a nova alfabetização desse século. Assim, não precisa ser um gênio para programar, assim como não precisa ser gênio para ler e escrever.

Em um vídeo recente, o presidente dos Estados Unidos, clamou para a juventude de seu país: “Não compre um novo game, desenvolva um. Não se limite a fazer download de um novo aplicativo, ajude a projetá-lo. Não apenas use seu celular, programe-o” (OBAMA, 2013).

A seguir serão apresentadas algumas ferramentas, jogos em diversas plataformas (digital/não-digital) que ensinam programação e/ou conceitos de programação de computadores para crianças de forma divertida, sem o rigor de uma linguagem de programação profissional.

### **2.1.1 Logo**

Logo é uma linguagem de programação que foi desenvolvida por Seymour Papert do Instituto de Tecnologia de Massachusetts (MIT) em 1968. Papert (1986) propôs uma metodologia que chamou de “construcionismo”, articulando conceitos computacionais com a teoria Piagetiana. Posteriormente, sua abordagem foi aplicada em outras plataformas além do Logo. Nessa abordagem o aprendiz constrói, por intermédio do computador, o seu próprio conhecimento (PAPERT, 1986).

Os princípios construcionistas proposto por Papert (1980) relacionam ideias de Dewey, Piaget e Vigotsky e são aplicados na construção de ferramentas educacionais que auxiliam as crianças na aprendizagem de conceitos de programação, e na exploração de seus próprios micromundos. Segundo Papert (2007), o micromundo constitui num universo

simbólico formalizado pelas crianças, em que a imaginação e criatividade são exploradas sem restrição.

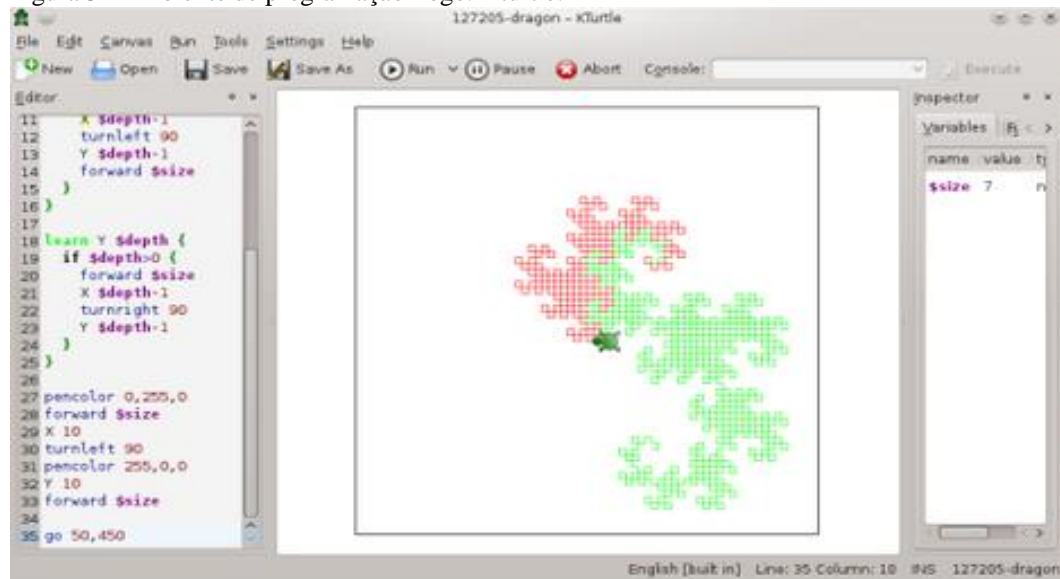
Um dos ambientes de programação Logo é o kturtle. Utiliza-se de uma interface gráfica contendo uma tartaruga que se movimentará a partir de comandos estruturados do usuário. Os comandos básicos para movimentar a tartaruga são palavras chaves seguido do seu atributo. Como exemplificação, o comando para *parafrente* juntamente com seu atributo tamanho do passo ou o comando *paradireita* seguido do ângulo do giro.

O Kturtle é um ambiente de programação educacional KDE Desktop (Figura 3) e tem o objetivo de tornar a programação mais fácil, sendo muito utilizado para ensinar as crianças as noções básicas de programação.

A ferramenta permite traduzir a linguagem de programação (os comandos, a documentação e as mensagens de erro) para a língua nativa do programador. Assim, traduzindo a linguagem de programação para a linguagem nativa do programador torna-se o aprendizado mais simples. Outras características para ajudar a atingir esse objetivo são: destaque de sintaxe intuitivo, mensagens de erro simples, função de ajuda integrada, execução passo a passo, entre outras.

O Kturtle permite a utilização de procedimentos. Os procedimentos são funções como aquelas encontradas em outras linguagens de programação. Se há uma rotina que é recorrente podemos escrever um procedimento. Por exemplo, um procedimento chamado quadrado. Cada vez que o usuário precisar desenhar um quadrado, ele realizará o chamando do procedimento já salvo, no qual economizará linhas de código e tempo, uma vez que todos os comandos linha a linha da construção do quadrado já estão salvos no procedimento. Alguns sistemas operacionais, como o S.O. Linux Educacional contempla um ambiente de programação e traz o Kturtle integrado.

Figura 3 - Ambiente de programação Logo: Kturtle.



Fonte: KDEdu (2015).

A linguagem Logo ainda hoje é utilizada em escolas do mundo todo, não apenas como uma ferramenta de aprendizagem para iniciantes em programação, mas principalmente para auxiliar o professor no processo de aprendizagem de outros conceitos, em diversas áreas do conhecimento. O uso da linguagem Logo no ensino de matemática tem sido discutido em várias pesquisas, entre elas, os trabalhos realizados por Matos (1991) e Valente (1996), como um ambiente que possibilita atividades de investigação geométrica e apropriação de conceitos matemáticos.

### 2.1.2 Light-bot

O Light-bot é um jogo de programação desenvolvido por Danny Yaroslavski, um jovem estudante canadense. O game ensina conceitos como planejamento, teste, depuração, procedimentos e loops e aponta que escolas têm utilizado o Light-bot para ajudar a ensinar conceitos básicos de programação, e ele ainda tem um lugar em alguns currículos universitários, em aulas de introdução à programação (TECHCRUNCH, 2013).

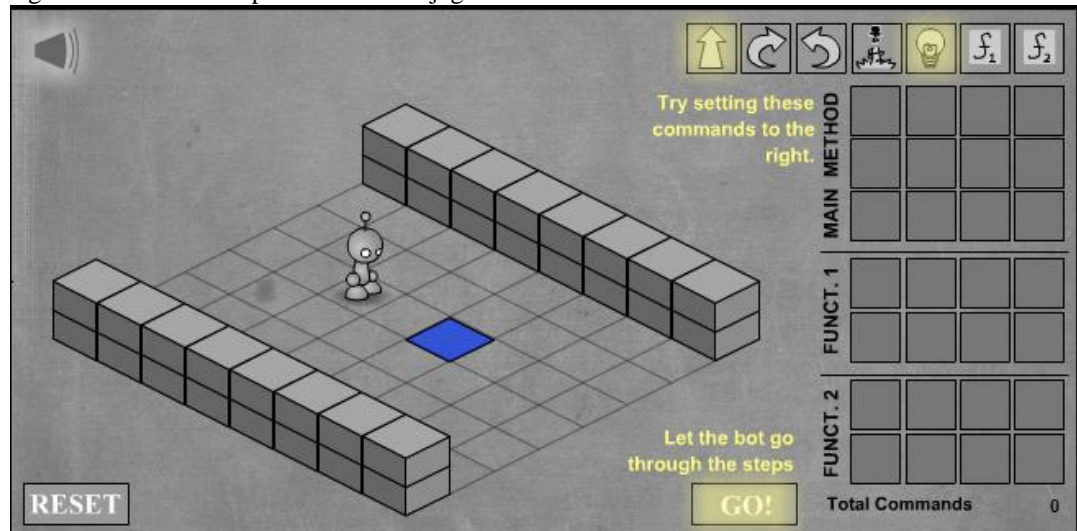
É possível jogá-lo gratuitamente na plataforma kongregate<sup>4</sup> (Figura 4). Há também versões completas de Light-bot, para diferentes grupos etários, à venda no iTunes e Google Play.

Pais (2002) salienta que os softwares computacionais com finalidade educacional devem proporcionar uma interface de fácil interação com os alunos. O Light-bot atende este quesito, apresentando uma plataforma simples. Seu objetivo consiste em sequenciar

<sup>4</sup> Kongregate é um site de que busca criar uma comunidade de jogadores e desenvolvedores de jogos online desenvolvidos em flash. Criado em 2006, o site possui milhares de usuários espalhados pelo mundo.

instruções que levem o ator principal do jogo (robozinho) a acender todas as posições do tabuleiro que estão definidas na cor azul. Para atingir tal objetivo, o número de instruções é limitado, exigindo que o usuário faça uso de funções (procedimentos), a fim de executar tarefas que se repetem.

Figura 4 - Interface da primeira fase do jogo.



Fonte: Autor (2015).

Em seus estudos Csikszentmihalyi (2009), destaca que um jogo para prender a atenção do usuário deverá equilibrar o grau de dificuldade (desafios propostos) com as habilidades adquiridas pelo usuário ao longo do jogo. O autor chama esse canal de *flow* (fluxo). Se o jogo exige muita habilidade do usuário quando ele ainda não a detém, o mesmo sente-se estressado. Na contramão, se o usuário já adquiriu habilidades e o jogo não apresenta um grau de dificuldade maior, o mesmo sente-se entediado (*bored*). Essa característica deverá ser observada em jogos educacionais, para que o game siga este canal desafiador ao aluno, para que ele adquira novas habilidades e não se sinta desestimulado.

Jogar um game não é uma tarefa fácil e requer desenvolvimento de muitas habilidades. Partindo desse princípio, cabe o seguinte questionamento: Por que o aluno passa horas à frente de um game e não pensa em abortar o seu envolvimento, sentido-se desafiado a todo instante? A resposta certamente nos leva a entender que um bom game, está nesse canal de fluxo, estruturado em dois eixos: habilidade versus dificuldade.

O ensino da matemática deveria se pautar nesses mesmos princípios apresentados do game designer, de tal forma que o aluno siga por um canal de construção do conhecimento, equilibrando desafios e habilidades.

Ao analisar o Light-bot percebe-se que o game consegue manter este equilíbrio entre desafios e habilidades. O jogo inicia-se orientado a execução das tarefas. Conforme o jogador

se apropriada de novas habilidades e competência ele aumenta o grau de dificuldades nos desafios, apresentando rotinas que requer uso das funções (procedimento).

Apresenta-se uma análise (Quadro 3) do game Light-bot sob as características que um bom jogo deve apresentar, apontadas por McGonical (2012).

Quadro 3 - Análise do game segundo as características apontadas por Jane McGonical.

<b>REGRAS</b>	Sequenciar instruções que levem o ator principal do jogo (robozinho) a atingir o objetivo. Para atingir tal objetivo, o número de instruções é limitada, exigindo que o usuário faça uso de funções (procedimentos), a fim de executar tarefas que se repetem (rotinas).
<b>METAS</b>	O jogo tem como meta levar o ator principal do jogo (robozinho) a acender todas as posições do tabuleiro que estão definidas na cor azul.
<b>FEEDBACK</b>	O game apresenta um feedback imediato. O usuário consegue ver a execução sequenciada dos comandos e observar os possíveis erros, garantindo ao jogador repensar novas estratégias para contemplar seus objetivos.
<b>PARTICIPAÇÃO VOLUNTÁRIA</b>	O game permite participação voluntária, permitindo ao usuário entrar ou sair dessa atividade a qualquer tempo. O usuário (jogador) pode abortar para posteriormente, num outro momento, se aventurar em novos desafios.

Fonte: Autor (2014).

Atualmente contamos com muitos recursos digitais que possibilitam a iniciação das crianças de forma fácil e divertida no mundo da programação. Há uma infinidade de jogos digitais como o Light-bot apresentado anteriormente. Muitos desses games permitem que as crianças escolham um tema de seu interesse para que a experiência seja mais agradável a elas e levam em consideração diferentes idades e nível de conhecimento. Mas ainda contamos com variados jogos não digitais que também proporcionam a iniciação das crianças na codificação.

A seguir serão apresentados dois jogos de tabuleiro que ajudam as crianças a se apropriarem de conceitos de programação.

### ***2.1.3 Code Monkey Island***

O Code Monkey Island é um jogo de tabuleiro (Figura 5) que, de forma divertida, introduz as crianças no mundo da programação. Aborda conceitos como lógica booleana e para seu criador, Raj Sidhu, o jogo proporciona uma experiência na qual toda a família pode brincar e aprender juntos conceitos de programação (CODEMONKEYISLAND, 2015).

No jogo, as crianças podem aprender conceitos lógicos por meio de macacos coloridos em vez de linhas de código. A entrada de dados é realizada por meio de cartões e numa corrida movimentada com seus macacos ao redor da ilha, analisam-se os melhores movimentos para a próxima jogada.

Figura 5 - Tabuleiro Code Monkey Island.



Fonte: Página oficial do Code Monkey Island<sup>5</sup> (2015).

### 2.1.4 Robot Turtles

Outro jogo de tabuleiro que ensina os fundamentos da programação (Figura 6) usando cartões de instrução para comandar tartarugas em torno de um labirinto. As crianças usam instruções sequenciadas, e depuração quando cometem erro, e levam apenas segundos para aprender o jogo de acordo com o criador Dan Shapiro (ROBOTTURTLES, 2015).

Figura 6 - Tabuleiro Robot Turtles.



Fonte: Página oficial do Robot Turtles<sup>6</sup> (2015).

### 2.1.5 Kodable

É um jogo para crianças de 5 a 8 anos de idade. O jogo é indicado para crianças não alfabetizadas, por possuir instruções não escritas e inicialmente auto instrutivas. Assim que as crianças aprendem a arrastar e soltar em um tablet, em torno de quatro ou cinco anos, eles estão prontos para jogar. O game pode ser jogado no próprio navegador Internet (Figura 7), a partir de um cadastro prévio. O jogo também está disponível na plataforma iOS, com a finalidade de ajudar a ensinar crianças os conceitos como condições, loops, funções e

<sup>5</sup> Disponível em: <http://codemonkeyplanet.com/>; Acesso em ago. 2015.

<sup>6</sup> Disponível em: <http://www.robotturtles.com/>; Acesso em ago. 2015.

depuração, ensinando as crianças a pensar como programadores para resolver problemas. O game evita o uso de textos, atraindo assim, crianças que ainda não dominam a habilidade leitora (KODABLE, 2015).

Figura 7 - Interface kodable na web.



Fonte: Página oficial do Kodable<sup>7</sup> (2015).

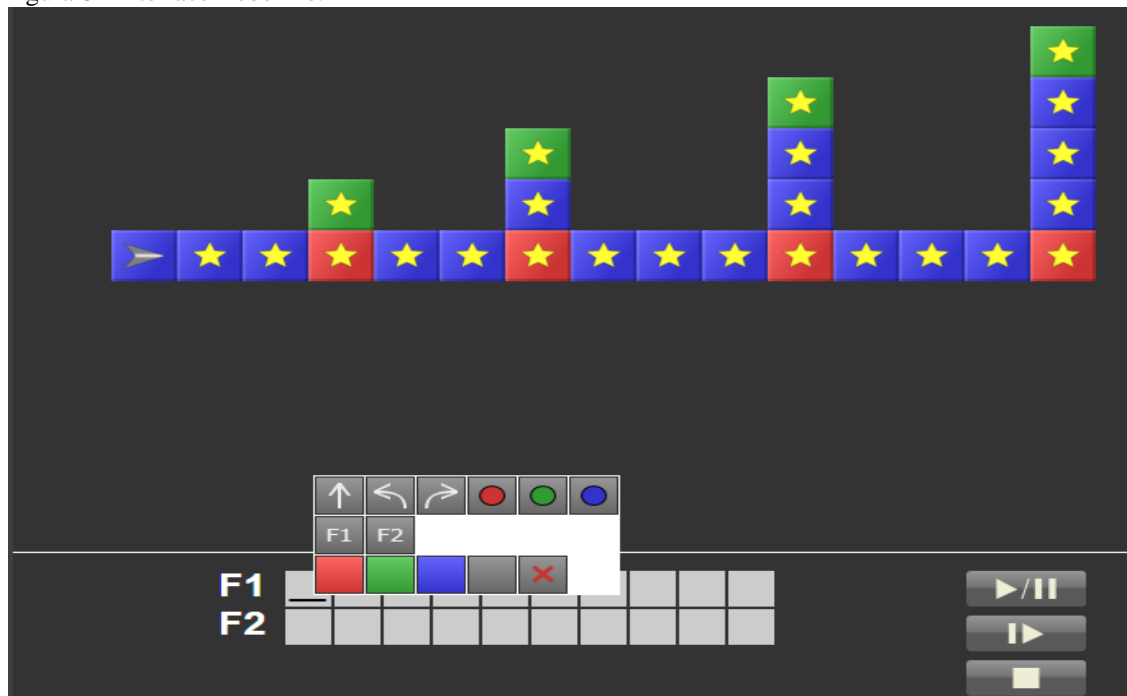
### **2.1.6 Robozzle**

Disponível em diversas plataformas como: navegadores web, iOS, Android, Windows Phone, o Robozzle permite guiar um robô por meio de uma série de labirintos usando comandos limitados. Os níveis de dificuldades variam de acordo com a idade. É livre para jogar em navegadores, e conta com o apoio da comunidade que permite aos jogadores desenvolver novos desafios e comentar sobre novos níveis.

O objetivo em cada desafio é coletar todas as estrelas disponíveis no labirinto (Figura 8). Para isso, o jogador fará uso dos grupos de caixas, que são chamadas de "função", que poderá ser usada de forma recursiva para atendimento da meta do game. O jogo desenvolve o pensamento criativo e habilidades de programação sem exigir conhecimentos técnicos previamente (ROBOZZLE, 2015).

<sup>7</sup> Disponível em: <http://game.kodable.com;>. Acesso em ago. 2015.

Figura 8 - Interface Robozzle.



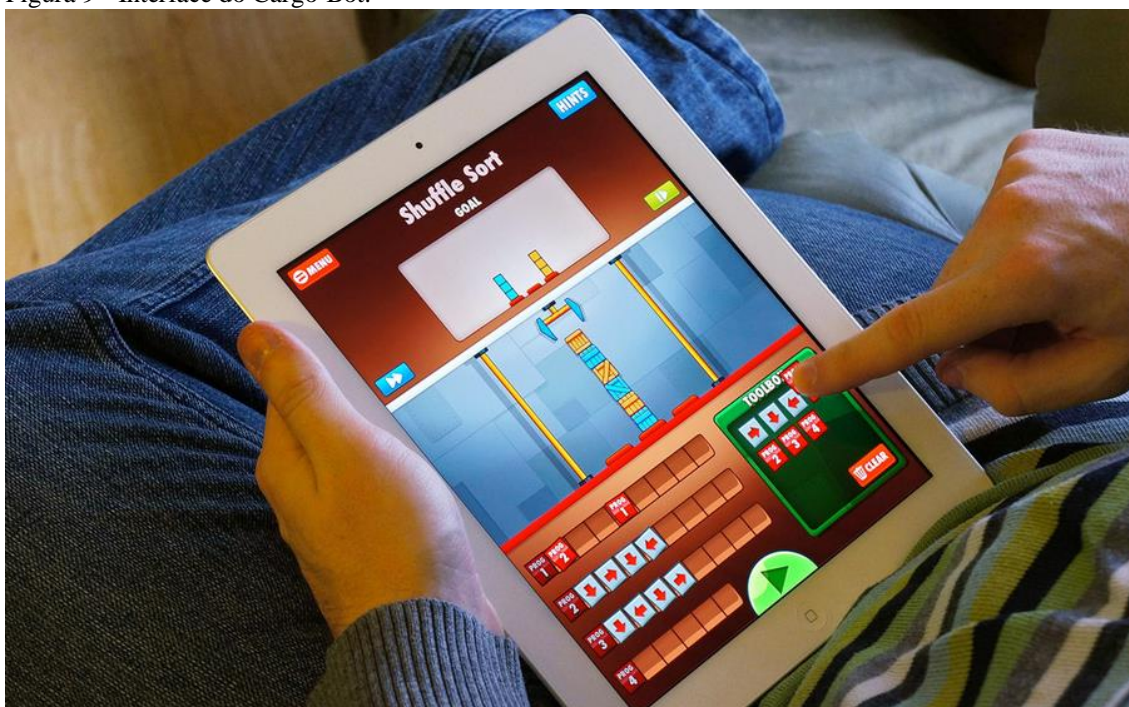
Fonte: Página oficial do Robozzle<sup>8</sup> (2015).

### 2.1.7 Cargo-Bot

O Cargo-Bot é o primeiro jogo desenvolvido utilizando Codea, um aplicativo iPad para criar rapidamente jogos e simulações. É acessível para crianças em seus níveis mais fáceis, mas oferece desafios à medida que progride e adquire habilidades. Os jogadores usam um guindaste para mudar caixas em torno de uma fábrica (Figura 9). Ao fazer isso, eles usam codificação, conceitos como loops e procedimentos e realiza-se depuração. O Cargo-Bot é um quebra-cabeças projetado para ensinar as crianças sobre a programação. O conceito é simples - controlar um braço robótico para mover caixas para um local designado. A lógica passo-a-passo ensina as crianças a enfrentar um problema maior, quebrando-o em etapas menores. O usuário ainda pode gravar suas soluções e compartilhá-las no YouTube (CARGOBOT, 2015).

<sup>8</sup> Disponível em: <http://www.robozzle.com/js/play.aspx?puzzle=7251;> Acesso em ago. 2015.

Figura 9 - Interface do Cargo-Bot.



Fonte: Página oficial do Cargo-Bot<sup>9</sup> (2015).

### 2.1.8 Scratch

Mitchel Resnick, professor e pesquisador do programa acadêmico do MIT Media Lab, juntamente com seu grupo de pesquisa desenvolveram uma variedade de ferramentas educacionais que envolvem pessoas em novas experiências de aprendizagem, entre eles o premiado LEGO Mindstorm, que recebeu este nome após a publicação do livro de Seymour Papert no ano de 1980, intitulado “Mindstorms: Children, Computers and Powerful Ideas” no qual o autor, um dos fundadores do MIT, propõe um ambiente único de aprendizado baseado em computação que ele denominou "Microworld" e que pode influenciar a qualidade do conhecimento adquirido.

O Lego Mindstorms consiste em uma série de kits que contêm software e hardware para criar robôs programáveis. Eles incluem um bloco inteligente, uma espécie de CLP<sup>10</sup>, um conjunto de sensores e motores. Lego Mindstorms pode ser usado para construir um modelo de um sistema embarcado com peças eletromecânicas controladas por computador. Muitos tipos de sistemas embarcados da vida real podem ser modelados utilizando Mindstorms. O mais recente sistema chama-se Lego Mindstorms EV3 lançado em 2013.

<sup>9</sup> Disponível em: <http://twolivesleft.com/CargoBot/>; Acesso em ago. 2015.

<sup>10</sup> Controlador Lógico Programável (CLP) ou Controlador Programável é um computador especializado, baseado em um microprocessador que desempenha funções de controle através de softwares desenvolvidos pelo usuário.

Em seus projetos, Resnick procura associar a criatividade e a aprendizagem, assumindo que na educação infantil as crianças constroem brinquedos, ferramentas criativas que lhes permitem pensar, testar e aprender. Além disso, argumenta que no ensino fundamental e médio o caminho da aprendizagem não deve ser diferente. Resnick descreve que a ciência da computação deve desenvolver materiais para que as crianças possam crescer, desenvolver soluções criativas envolvendo as diversas áreas do conhecimento. Neste processo, o computador se torna aliado na aprendizagem criativa (RESNICK, 2006).

O conhecimento deve ser apresentado aos estudantes de forma a fazer sentido para eles, e deve fazê-lo de acordo com seus próprios interesses e necessidades e utilizar as novas tecnologias em soluções de busca para seus problemas. Aprender por meio de projetos e experiências fortalece o conceito de sociedade do conhecimento (RESNICK, 2006).

O autor apresenta uma evolução, sendo esta a "sociedade criativa", na qual as pessoas precisam continuar aprendendo toda a sua vida e dar soluções criativas para seus problemas e necessidades. Com isso, devemos proporcionar aos alunos novas oportunidades de aprendizagem, em que eles possam fazer uso da tecnologia de forma inovadora e produtiva, com experiências criativas e infinitas possibilidades de aprendizagem. Isso torna possível operar a pesquisa para o desenvolvimento de um pensamento criativo sistêmico e deliberado. Pensando nisso, o grupo de Resnick desenvolveu uma nova linguagem de programação, chamada Scratch, que torna mais fácil para as crianças a criar suas próprias histórias animadas, jogos de vídeo e arte interativa.

Utilizado em mais de 150 países e disponível em mais de 40 idiomas, o Scratch ajuda os jovens a aprender a pensar de maneira criativa, refletir de maneira sistemática e trabalhar de forma colaborativa, habilidades essenciais para a vida no século XXI (EDUSCRATCH, 2015). Utilizando-se da lógica de programação visual, por meio de blocos lógicos que se encaixam, a ferramenta proporciona uma lógica de programação mais simples, lúdica e intuitiva, diferenciada das linguagens tradicionais. A ferramenta é fornecida gratuitamente e destina-se a crianças a partir dos oito anos e foi desenvolvida pela equipe Lifelong Kindergarten do MIT Media Lab e coordenada por Mitchel Resnick (EDUSCRATCH, 2015).

A primeira versão desktop do Scratch foi apresentada em 2003. Desde 2007, os projetos podem ser compartilhados on-line com outros usuários. Na página oficial do Scratch é possível inserir e também visualizar programas criados por outras pessoas, experimentar, reutilizar e ainda adaptar imagens e scripts.

O Scratch tornou-se popular no Reino Unido por meio do Code Club, que utiliza nos cursos introdutórios a ferramenta para ensinar as bases de programação de computador,

contribuindo assim para sua difusão. O uso do Scratch justifica-se como linguagem introdutória por causa de sua relativa facilidade para construir programas, uma vez que as habilidades aprendidas na ferramenta podem ser aplicadas a outras linguagens de programação, como por exemplo, a linguagem Python<sup>11</sup>.

A ferramenta não é exclusiva para a criação de programas, uma vez que fornece uma variedade de recursos de programação visual para criar histórias animadas. A ferramenta possibilita aos professores criar aulas conceituais, visuais e trabalhar em laboratório de ciências, por meio de jogos e animações que estimulam a mente e a interação com o aluno.

Há uma área de palco onde são apresentados os resultados. Existem várias maneiras de criar um ator e palcos. Em primeiro lugar, os usuários podem desenhar seus próprios atores manualmente ou podem escolher um *script* da biblioteca padrão ou até mesmo uma foto usando uma câmera, ou clip art. Com um ator selecionado, blocos de comandos podem ser aplicados a ele, arrastando-os para a área de comandos. Todos os blocos disponíveis estão listados e classificados como: movimento, aparência, som, caneta, variáveis, controle, sensores, operadores e mais blocos. Além do guia Script, há duas guias adicionais, uma guia Costumes e a guia Sons. Na guia Costumes, o usuário pode selecionar uma nova roupa para o ator e na guia Sons, podem inserir sons e músicas para um ator.

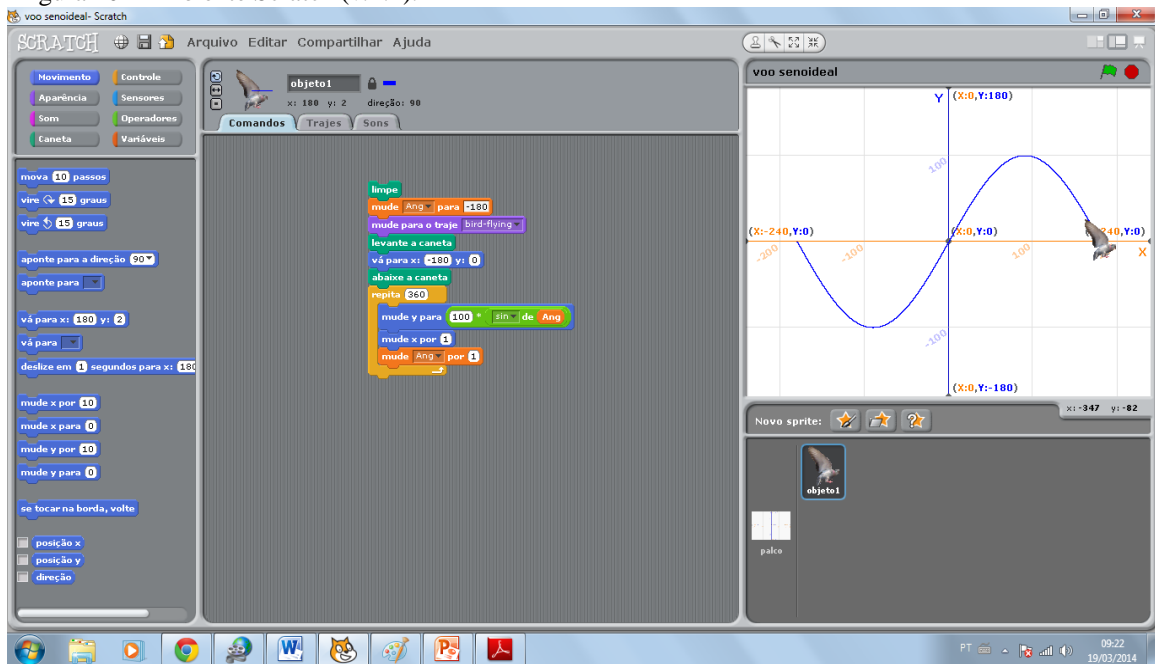
Com o avanço tecnológico, ferramentas como o Scratch contribuem na construção de softwares educativos, como objetos de aprendizagem (OA). De acordo com Wiley (2000, p. 4) um “OA refere-se a qualquer recurso digital que possa ser reutilizado em diferentes ambientes de aprendizagem, com o objetivo de apoiar a aprendizagem”. Exemplos de recursos digitais reutilizáveis pequenos, incluem entre outros, animações que podem ser facilmente elaboradas utilizando o Scratch como ferramenta.

A seguir (Figura 10), apresenta-se uma visão geral do Scratch no desenvolvimento de um OA que apoia o estudo de um movimento senoidal.

---

<sup>11</sup> Python é uma linguagem de programação de computadores de propósito geral, classificado entre as oito melhores linguagens de programação mais populares do mundo. Pode ser usada para criar muitas coisas, incluindo aplicações web e aplicações de desktop.

Figura 10 - Ambiente Scratch (v.1.4).



Fonte: Autor (2015).

O slogan da comunidade on-line do Scratch *Imagine, Program, Share* indica que a partilha e os aspectos sociais da criatividade são partes importantes da filosofia do Scratch. Todos os projetos no site são compartilhados e podem ser executados em um navegador web usando o Flash Player, que não está disponível para dispositivos iOS.

A tecnologia e as práticas pedagógicas podem exercer influências sobre a aprendizagem. Segundo o construcionismo<sup>12</sup> apresentado por Seymour Papert (1986), a construção do conhecimento baseia-se em uma ação concreta e intencional podendo esta fazer uso do computador como recurso a sua construção. Deste modo, quando um aluno utiliza os recursos computacionais, ele cria a possibilidade de contato com novas ferramentas a favor da aprendizagem.

Recursos tecnológicos como o Scratch possibilitam uma interação entre sujeito-objeto de modo a repensar estratégias empregadas na resolução de problemas e a aprendizagem gerada por tais interações pode permitir a reconstrução do conhecimento. Assim, o uso de novas ferramentas tecnológicas a favor da aprendizagem como o Scratch favorece uma prática docente baseada em projetos, em que diversas áreas do conhecimento podem ser integradas na resolução de diferentes problemas, de forma cooperativa, mediada pelo professor. O Scratch poderá servir de suporte para o docente acompanhar todo o processo, o raciocínio lógico da criança e analisar o que ela fez e apontar eventuais erros. O aluno pode criar diversos

<sup>12</sup> O construcionismo é uma teoria proposta por Seymour Papert, em que a construção do conhecimento é baseada na realização de uma ação concreta. O construcionismo implica numa interação aluno-objeto, mediada por uma linguagem de programação.

algoritmos<sup>13</sup> e utilizar de conceitos matemáticos para solucionar situações problemas, além de retomar muitas outras habilidades e conceitos envolvidos na construção.

Ao encontro das possibilidades mencionadas e para melhor contextualizá-las, um exemplo criado foi traçar um voo senoidal (Figura 11).

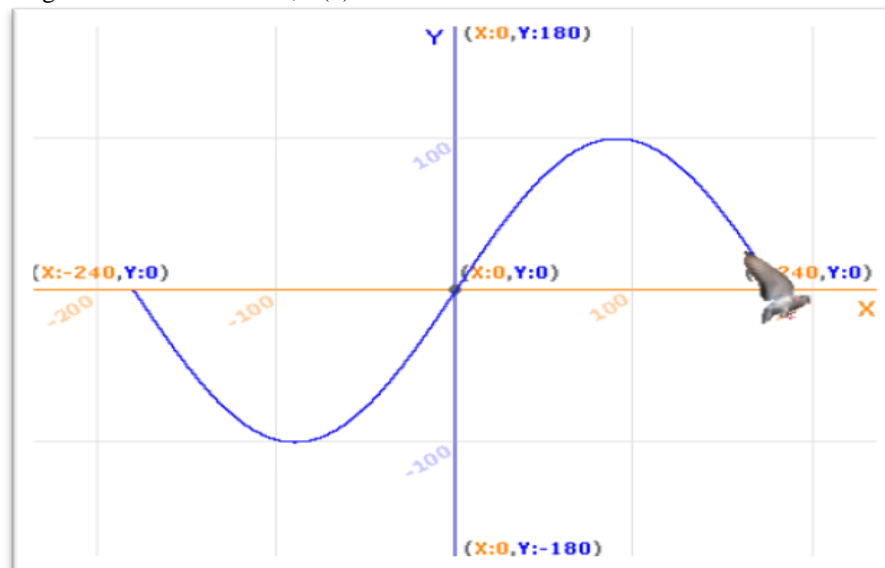
Figura 11 – Estratégia de um voo senoidal.



Fonte: Autor (2015).

Como resultado da execução dos blocos de comandos apresentados (Figura 11) será realizado um voo senoidal por um pássaro variando  $2\pi$  radiano, ou seja  $360^\circ$  (Figura 12).

Figura 12 - Voo Senoidal,  $f(x) = 100 \cdot \sin \theta$



Fonte: Autor (2015).

Dentro deste contexto será possível o aluno visualizar e analisar diversos conceitos, dentre eles: o comportamento da função, domínio da função  $D(f) = \mathbb{R}$  e a imagem da função  $Im(f) = [-100; 100]$ .

<sup>13</sup> Algoritmo é uma sequência lógica, que utiliza da lógica matemática para alcançar um objetivo específico.

O uso do Scratch no âmbito educacional pode desenvolver o pensamento lógico, pois trabalha com construção de algoritmos (lógica de programação), a qual envolve conceitos matemáticos e potencializa o pensamento criativo em crianças do ensino fundamental, que consegue criar e fazer uso de um repertório de estratégias para atingir objetivos complexos (MARTINS, 2012).

O recente aumento de interesse em aprender programação é refletido em sites como codecademy.com e code.org, que tem centrado especialmente em empregos e oportunidades de carreira, mas há razões mais profundas e amplas para aprender programação.

Nesse processo de aprender o código, as pessoas aprendem muitas outras coisas. Além de aprender ideias matemáticas e computacionais aprendem-se estratégias para a resolução de problemas, elaboração de projetos e comunicação de ideias. Estas habilidades são úteis não apenas para cientistas da computação, mas para todos, independentemente da idade, origem, interesses ou ocupação (RESNICK, 2013).

Assim, o que é necessário é uma mudança de mentalidade, de modo que as pessoas comecem a ver a codificação não só como um caminho para bons empregos, mas como uma nova forma de expressão e um novo contexto para a aprendizagem.

### ***2.1.9 Code Club***

O Code Club é uma iniciativa voluntária no Reino Unido. Criada pela professora britânica Clare Sutcliffe e pela desenvolvedora Linda Sandvik, em 2012, destinado a ensinar crianças com idades entre 9 a 11 como codificar. As autoras lançaram o projeto com o plano de chegar a um quarto das 21.000 escolas da Grã-Bretanha até 2015. Para isso, contam com o apoio da Google, e hoje já são 1.668 clubes no Reino Unido e 142 em outros países. O Code Club é uma rede mundial de atividades extracurriculares gratuitas, gerenciada por voluntários, e tem como principal objetivo ensinar programação de computadores a crianças (CODECLUB BRASIL, 2015).

A filosofia do Code Club é diversão, criatividade e aprendizagem pela descoberta. É importante que as crianças aproveitem o tempo que elas passam no Code Club e que não se sintam como se estivessem em mais uma aula da escola. As crianças devem entender que são responsáveis pelo que o computador faz, e podem (e devem) fazer com que a máquina faça o que desejam (CODECLUB BRASIL, 2015).

Para ensinar programação como atividade extracurricular nas escolas, voluntários utilizam projetos criados pelo Code Club que contemplam jogos, animações e páginas da internet. Qualquer pessoa pode se tornar um voluntário, criar um clube próximo de sua casa e

ministrar as aulas de programação, uma hora por semana. Os cursos 1 e 2, disponibilizados pelo Code Club, utilizam o Scratch como ferramenta para ensinar as bases da programação de computadores. O curso 3 é uma introdução ao desenvolvimento web e utiliza HTML<sup>14</sup> e CSS (Cascading Style Sheets) que é uma linguagem que possibilita ao desenvolvedor criar uma ligação para uma página que contém os estilos. O curso 4 ensina a linguagem de programação Python.

No Brasil, o Code Club nasceu da iniciativa de Heverton Herman, que em 2013 se interessou pelo projeto quando assistiu ao vídeo *O que as escolas não ensinam* (<https://www.youtube.com/watch?v=iDLpJK8qfFk>), que num primeiro momento focou em traduzir alguns vídeos e posteriormente entrou em contato com as criadoras do projeto na Inglaterra e organizou o site brasileiro com o slogan Programação é coisa de criança (CODECLUB BRASIL, 2015).

### **2.1.10 Codecademy**

Criado pelos americanos Zach Sims e Ryan Rubinski, o Codecademy é um site que oferece aulas gratuitas de programação de computadores por meio de uma plataforma interativa. “Programação é parte das linguagens necessárias no século XXI. Nós achamos que programar ajuda a pessoa a ter diferentes pontos de vista e alcançar seus sonhos” (CODECADEMY, 2015).

Ele usa o conceito de Mooc Mechanic, um curso virtual projetado para não depender do professor. O termo Massive Open Course Online (MOOC) em Português (Curso Online Aberto e Massivo) é um tipo de curso aberto oferecido por meio de ambientes virtuais de aprendizagem, ferramentas Web 2.0 e redes sociais destinadas a fornecer para um grande número de estudantes a oportunidade de expandir seus conhecimentos num processo de co-produção. O termo Mechanic é usado porque, em vez de um professor ou universidade organizar a classe, um planejador de e-mail irá realizá-lo.

Os MOOCs são geralmente livres e não necessitam de pré-requisitos, mas também não oferecem certificados de participação. Eles são oferecidos a um grande número de estudantes e possuem vasta quantidade de material. Talvez, no futuro, possa ser validado por uma universidade por meio de uma avaliação em sala de aula.

---

<sup>14</sup> HyperText Markup Language (HTML) é uma linguagem de marcação utilizada na construção de páginas na web. Os documentos HTML podem ser interpretados por navegadores.

Para participar de um MOOC não é obrigado pagar taxa, embora possa haver custos de manutenção, mas, em princípio, são totalmente gratuitos e abertos a qualquer tipo de público e muitos cursos tradicionais dependem de um número de participantes e alguns professores para ser iniciado. A proposta de Mooc, ser massivo, sugere que o curso é projetado para suportar qualquer número de participantes. De acordo com a Wiley (2012), uma característica que pode ser associada a Mooc está relacionada com o conceito de comunidade de prática, que será discutido na fundamentação teórica deste trabalho.

Para Figueiredo (2012) os MOOCs podem ter características tais como: um percurso exploratório, ciclos de tentativa e erro-reflexão e definem uma semente para romper com a forma clássica de ensino e aprendizagem, explorando novas práticas em educação e adicionando o potencial de inteligência notícias Web 2.0. Assim, o conhecimento está sendo co-produzido por todos e é projetado com ênfase no contexto, não no conteúdo. Sob essas características, o Codecademy foi pensado e cada indivíduo que entra tem o seu próprio perfil. Para motivar os usuários a participar, o site fornece feedback por exercícios contemplados e, além disso, o site permite a qualquer pessoa criar e publicar um novo curso utilizando a ferramenta Course Builder, e fornece um fórum no qual os programadores iniciantes e avançados podem interagir e ajudar entre si.

### ***2.1.11 Code.org***

O Code.org foi criado pelo iraniano Hadi Partovi, formado em Ciências da Computação pela Universidade de Harvard e investidor de empresas como Facebook e Dropbox (CODE, 2015).

Em 2011 ele teve a ideia de criar um vídeo para estimular jovens a aprender programação, usando depoimentos de personalidades do mundo da tecnologia. A decisão de dedicar-se integralmente ao projeto veio depois de saber que o vídeo sobre o projeto havia recebido mais de 11 milhões de visualizações na Internet. Assim, em dezembro de 2013, o Code.org lançou a iniciativa “Hour of Code 2013”, em que qualquer pessoa poderia dedicar 1 (uma) hora para aprender a programação de computadores.

Na plataforma o aluno pode aprender os primeiros passos na programação de computadores. No tutorial para iniciantes, o aluno tem o primeiro contato com a programação de computadores por meio da linguagem Blockly<sup>15</sup>, que possibilita arrastar e soltar blocos para escrever os códigos. Além de um tutorial para iniciantes, é possível encontrar outros

---

<sup>15</sup> Blockly é uma linguagem de programação visual baseada em blocos apresentada pela Google. Todo o código do Google Blockly é gratuito e open source.

tutoriais para aprendizagem de linguagens, como JavaScript e até uma introdução à robótica (CODE, 2015).

Na página oficial do Code há um depoimento de uma estudante do 2º ano, onde ela diz “Você pode fazer o que quiser com a programação”, isto é, a programação possibilita aprender novas coisas, novos ensinamentos, descobrir algo novo, criando possibilidades de aprendizagem. Sob esta ótica entendemos que a programação encaminha como uma arte liberal.

### **2.1.12 Processing 2**

O Processing 2 é uma linguagem de programação visual e não é o primeiro a seguir este paradigma. Como exemplo, o Logo apresenta esse contexto visual, quando um programador escreve instruções para dirigir uma tartaruga ao redor da tela, produzindo formas e desenhos. Apesar da simplicidade e inovação, o Logo tem limitações. O Processing é um descendente direto do Logo e nasceu em 2001 no grupo de pesquisa do Instituto de Tecnologia de Massachusetts Media Lab pela iniciativa dos estudantes de pós-graduação Casey Reas e Benjamin Fry.

Processing é uma linguagem de programação de código aberto para as pessoas que querem programar imagens, animação e som. Ele é usado por estudantes, artistas, designers, arquitetos, pesquisadores e entusiastas para aprendizagem, criação de protótipos e produção. Ele é criado para ensinar fundamentos da programação de computadores dentro de um contexto visual e para servir como um caderno de desenho de software e ferramenta de produção profissional. O processamento é desenvolvido para artistas e designers como uma alternativa para ferramentas de software proprietário no mesmo domínio (PROCESSING, 2015).

Processing 2 é livre e gratuito e é construído baseado em Java<sup>16</sup>, orientado a objetos<sup>17</sup> e se constitui numa linguagem funcional, sem algumas das limitações do Logo. O Processing 2 permite que a comunidade de desenvolvedores, professores, artistas se reúnam para compartilhar trabalhos, contribuir com ideias e expandir as características do ambiente.

Não existe uma linguagem de programação que resolva todos os problemas e nenhuma linguagem é perfeita. O Processing 2 vem com muitas de limitações, no entanto, é um excelente lugar para começar (e ficar). Ele é usado para aplicações web, projetos de arte em museus, entre as mais variadas aplicações.

---

<sup>16</sup> Java é uma linguagem de programação interpretada orientada a objetos, desenvolvida na década de 1990 pela empresa Sun. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é executado por uma máquina virtual.

<sup>17</sup> A orientação a objetos é um modelo de análise, projeto e programação de sistemas de *software* baseado na composição e interação entre diversas unidades de software chamadas de objetos.

Aprender a programar é um processo de aprender a pensar, conforme enfatizado neste trabalho. Enxerga-se o Processing 2 como uma nova ferramenta para os artistas (alunos) e tecnólogos usarem em seu trabalho, como meio e como um contexto.

O Processing 2 está situado no local de convergência entre as artes e a tecnologia, facilitando a busca e alfabetização de ambos. O Processing introduz uma linguagem das artes visuais por meio de um vocabulário mais familiarizado da ciência da computação. Conceitos de ciência da computação são centrais, tais como, variáveis, funções e condicionais são ensinados conjuntamente com noções básicas visuais como imagem, forma e cor (PROCESSING, 2015).

A chave para a nossa filosofia é a fusão entre os dois: vemos software como um meio, ou algo que conecta duas coisas. Nós vemos-la como um meio para fazer, e vemos o nosso papel como o fornecimento de um novo meio para os artistas e usuários através de uma variedade de campos. Em uma sociedade que exige uma alfabetização e sensibilidade para a mídia de todos os tipos, nós nos vemos fazendo isto possível para um amplo público (PROCESSING, 2015).

Assim, o Processing 2 atende a uma variedade de comunidades em diferentes contextos, desde a sala de aula do ensino fundamental até a universidade, usado por todos, desde o profissional amador ao especializado.

O Processing 2 vêm sendo usado como ferramenta educacional nas salas de aula do mundo todo. Nos últimos anos, ele começou a se tornar uma presença na escola nas salas de aula do ensino fundamental e médio, sendo particularmente eficaz quando usado para ensinar ciência, matemática, e mais geralmente, ciência da computação (PROCESSING, 2015).

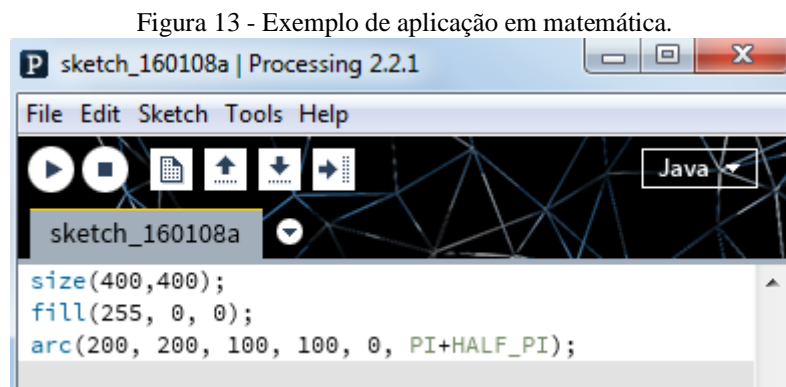
O uso crescente do Processing na sala de aula ao longo dos últimos anos aponta a repensar a forma como os alunos são ensinados a programar. Em vez de aprender uma habilidade profissional específica (linguagem de programação), o ensino do Processing segue a partir de uma filosofia que defende uma forma mais generalizada e estratégica de pensar. Como a tecnologia muda rapidamente, o objetivo de qualquer educação em software tende a ensinar os fundamentos da programação. Com o conhecimento generalista, um estudante é capaz de programar em qualquer outra linguagem. Além disso, ele usa um quadro visual, de modo a ser mais acessível para as pessoas que naturalmente não pensam como matemáticos ou cientistas da computação. Isso possibilitou ao Processing 2 tornar-se parte da caixa de ferramentas de um artista de uma forma nunca antes vista.

No âmbito da educação superior, Processing 2 alavancou como uma ferramenta de sala de aula em disciplinas variadas como a ciência da computação, inglês, arte, arquitetura, design, e as humanidades digitais (REAS; FRY, 2007).

Um exemplo bem simples para que se possa entender o uso do Processing 2 no ensino de matemática é na construção de setores circulares, a ferramenta torna esse processo fácil.

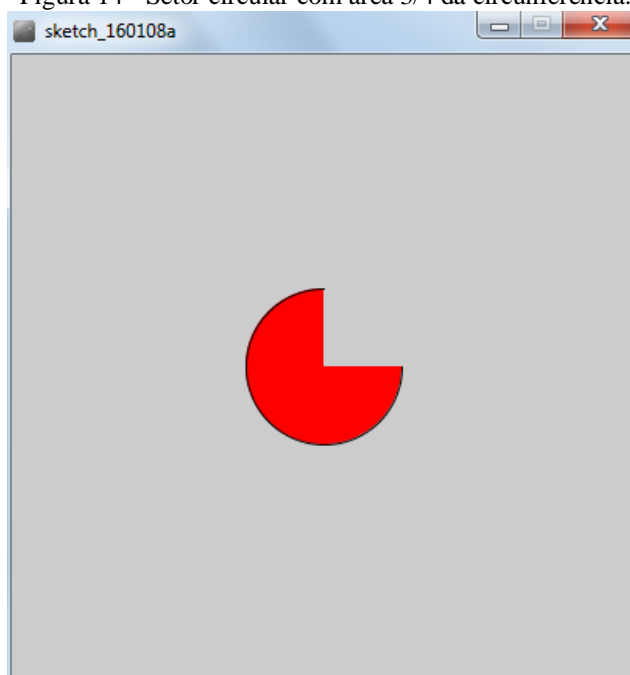
O comando `arc()` é responsável por essa construção e possui a seguinte sintaxe: `arc (a, b, c, d, início, fim)`. Os parâmetros necessários a serem fornecidos nessa construção - a: coordenada x do centro do setor circular; y: a coordenada y do centro do setor circular; c: a largura do raio do setor circular; d: a altura do raio do setor circular; início: ângulo para iniciar o arco em radiano; fim: o ângulo para terminar o arco em radiano.

As constantes `PI` e `TWO_PI` são comumente empregadas e equivalem a  $180^\circ$  e  $360^\circ$ , respectivamente. `PI`, `QUARTER_PI`, `HALF_PI`, e `TWO_PI` correspondem a  $180^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $360^\circ$ , respectivamente. Como exemplo, será construído um setor circular com área três quartos da circunferência. Uma das possibilidades de solução é o código apresentado na Figura 13.



Fonte: Autor (2015).

No código apresentado na Figura 13, o comando `size` define o tamanho da janela de apresentação. O comando `Fill` é responsável pelo preenchimento do setor circular, no caso do exemplo a cor vermelha é utilizada, e o comando `arc` é responsável pela construção do arco. Para conseguir um arco de  $270^\circ$  como estratégia foi empregada a soma das constantes `PI` e `HALF_PI`. O resultado da execução do programa é apresentado na Figura 14.

Figura 14 - Setor circular com área  $3/4$  da circunferência.

Fonte: Autor (2015).

O exemplo apresentado foi uma pequena demonstração em relação às inúmeras potencialidades de criação que a ferramenta Processing 2 proporciona, que podem ser utilizadas em aulas de matemática.

A utilização da ferramenta dentro das artes visuais e campos de design tem crescido continuamente desde que o software foi lançado. Aparecendo em um momento crítico, quando os artistas começaram a introduzir novas mídias e tecnologias em suas práticas, Processing 2 tem permitido a milhares de profissionais criativos mudar o cenário das artes no início do século XXI. O uso da ferramenta Processing 2 é diversificado e tem contribuído em projetos de destaques em exposições e museus como o Museum of Modern Art (MoMA) em Nova Iorque e o Victoria e Albert Museum, em Londres (PROCESSING, 2015).

O software tem ganhado destaque no meio artístico devido a sua capacidade de produzir formas dinâmicas, processar gestos, simular sistemas naturais, e integrar outros meios de comunicação incluindo som, imagem e texto. Usado para criar projeções em espetáculos de dança e música, criar instalações interativas em galerias, a ferramenta permite uma aproximação de design e programação, e possibilita o surgimento de uma geração de artistas visuais que consideram a programação parte essencial da sua prática criativa (REAS; FRY, 2007). No site <http://www.designculture.com.br/processing> encontram-se exemplos de aplicações do uso do Processing no âmbito das artes visuais e design.

O Ambiente Integrado de Desenvolvimento ou IDE, do inglês *Integrated Development Environment*, é um programa de computador que reúne características e ferramentas de apoio

ao desenvolvimento de *software* com o objetivo de agilizar este processo de construção. Geralmente os IDEs facilitam a técnica de RAD (Rapid Application Development, ou em português Desenvolvimento Rápido de Aplicativos), que visa à maior produtividade dos desenvolvedores.

O ambiente integrado de desenvolvimento do Processing 2 faz com que seja fácil escrever programas, num editor de texto. A capacidade do Processing 2 pode ser estendida com bibliotecas e ferramentas adicionais. Há centenas de bibliotecas criadas pela comunidade Processing 2 que podem ser adicionadas aos programas para permitir novas funcionalidades.

O Processing 2 tem diferentes modos de programação, o que permite sua utilização em diferentes plataformas e programação em diferentes linguagens. Os modos de programação padrão são Java e Experimental. Outros modos de programação podem ser adicionados como o modo JavaScript e Android. O Ambiente de Desenvolvimento do Processing 2 consiste em um editor de texto simples para escrever código, uma área de mensagens, um console de texto, guias para gerenciamento de arquivos, uma barra de ferramentas com botões para as ações mais comuns, e uma série de menus. As opções de menus mudam de acordo com o modo. A Figura 15 mostra o modo Java.



Fonte: Autor (2015).

Os programas escritos usando Processing são chamados de sketches. Estes esboços são escritos no editor de texto. A área de mensagem emite um feedback ao salvar e ao exportar e

também exibe erros. A saída de console exibe texto por esboços de transformação, incluindo mensagens de erro completas e saída de texto a partir de rascunhos com o uso das funções `print()` e `println()`, conforme apresentado na Figura 15.

Os botões na barra de ferramentas são respectivamente (Figura 16): executar o programa, parar programas, criar um novo, abrir, salvar e exportar.

Figura 16 - Barra de Ferramentas Processing 2.

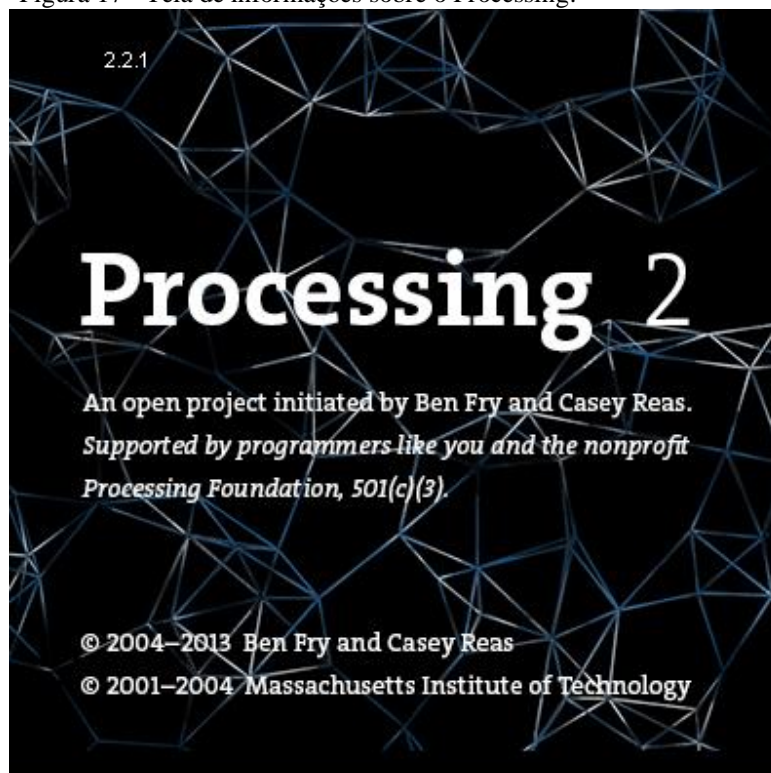


Fonte: Autor (2015).

Os comandos adicionais são encontrados dentro dos cinco menus: Arquivo, Editar, Esboço, Ferramentas e Ajuda. Os menus são sensíveis ao contexto, o que significa que, apenas os itens relevantes para o trabalho que está sendo realizado, estão disponíveis. Dentre os menus é interessante destacar algumas opções do menu Ajuda.

Na opção Sobre o Processing (Figura 17) é possível encontrar informações sobre o software, como sua versão e os nomes dos criadores.

Figura 17 - Tela de informações sobre o Processing.



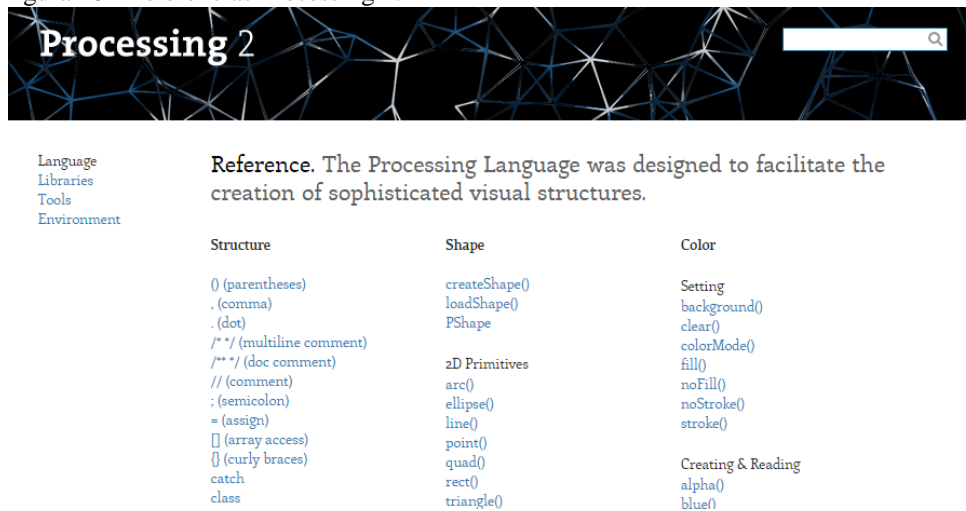
Fonte: Autor (2015).

As funções possuem parâmetros que são intuitivos, e isto é considerado como potencialidade da linguagem Processing para um novato. No entanto, a única maneira de saber com certeza a sintaxe do comando será recorrendo à referência online.

A referência pode ser encontrada online no site <http://www.processing.org> sob o link referência. Podemos navegar por todas as funções disponíveis por categoria, ou em ordem alfabética.

A página de referência (Figura 18) traz uma documentação completa para cada função, incluindo: nome, exemplos de código, descrição do que a função realiza, sintaxe de como escrever a função e parâmetros. A opção Referências do menu Ajuda abre as referências no navegador padrão.

Figura 18 - Referências Processing 2.



Fonte: Autor (2015).

As referências se constituem como recurso acoplado ao Processing 2 para sanar dúvidas de sintaxe dos comandos, conceitos, além de contar com alguns exemplos práticos que auxiliam na compreensão.

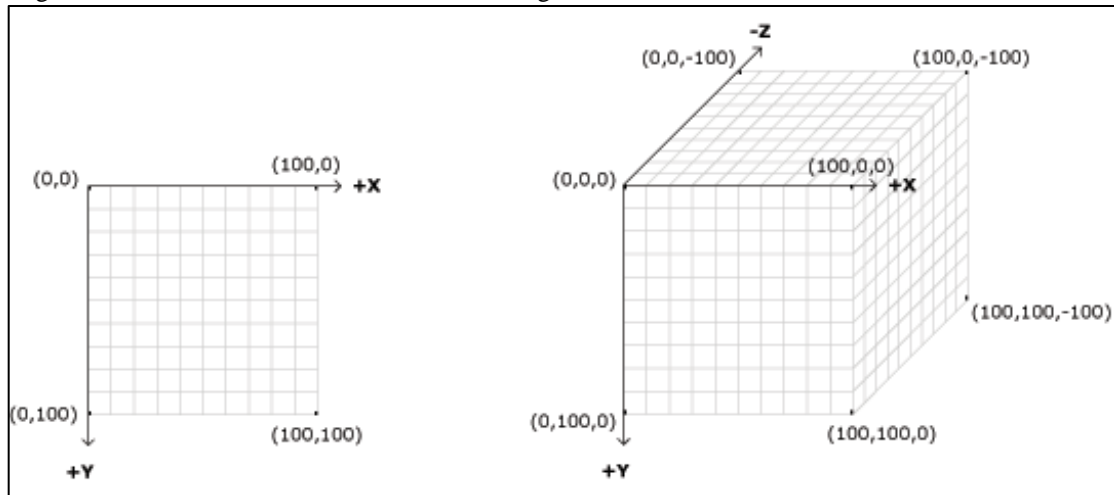
Os programas escritos em Processing 2 são informalmente conhecidos como esboços. A pasta na qual se armazenam os esboços é chamada de Sketchbook<sup>18</sup>. Cada esboço de processamento consiste de uma pasta (com o mesmo nome que o seu esboço) e um arquivo com a extensão (.pde). Por exemplo, se o desenho é nomeado "Sketch\_150822a", a pasta para o esboço será chamada de "Sketch\_150822a" e o arquivo principal será chamado de "Sketch\_123.pde". A extensão do arquivo (.pde) é uma designação para o ambiente de desenvolvimento Processing.

A linguagem Processing foi projetada para gerar e modificar imagens, assim, o trabalho na ferramenta é realizado sob um sistema de coordenadas cartesianas com origem no canto superior esquerdo. O eixo x tem seus valores crescendo na horizontal da origem para a direita e o eixo y cresce na vertical da origem até a parte inferior da janela (Figura 19). É

<sup>18</sup> Um sketchbook é um caderno para esboços frequentemente utilizado por artistas para desenhar ou pintar como parte do processo criativo.

possível alterar a origem do eixo utilizando uma função específica, dependendo da necessidade do programador em suas construções.

Figura 19 - Sistema de coordenadas do Processing 2.



Fonte: Processing (2015).

Há diferentes modos de programação incorporados ou que podem ser adicionados ao Processing, o que possibilita criar aplicações em diferentes linguagens e para diferentes plataformas. Os modos de programação padrão encontrados são Java e Experimental. Outros modos de programação, tais como JavaScript e Android, são adicionais. Atualmente, ainda em fase de desenvolvimento, há suporte para programação no modo Python.

O modo *Java* possibilita escrever programas desde os mais simples aos complexos. É utilizado por iniciantes e pode ser expandido para desenvolvimento de software Java profissional. Os programas escritos neste modo podem ser exportados para executar em Sistemas Operacionais Linux, Mac OS X e Windows (PROCESSING, 2015).

O modo *experimental* é um protótipo de uma versão futura do Development Environment Processing. Ele inclui recursos para verificar os erros no código enquanto ele está escrito, para acompanhar variáveis (como seu conteúdo é alterado), para depurar um programa com pontos de quebra (PROCESSING, 2015).

Os esboços escritos no modo *JavaScript* podem ser exportados para executar dentro dos navegadores web usando HTML5<sup>19</sup>, incluindo áudio e vídeo. Já os esboços escritos no modo Android podem ser exportados para serem executados em smartphones e tablets. Este modo é documentado na página wiki em <https://github.com/processing/processing-android/wiki>. O principal objetivo deste projeto é tornar possível a criação de apps para

<sup>19</sup> HTML5 é uma evolução do padrão que define o HTML. O termo representa dois diferentes conceitos: uma nova versão da linguagem HTML, com novos elementos, atributos, e comportamentos e um conjunto maior de tecnologias que permite o desenvolvimento de aplicações e web sites mais diversos e poderosos. < <https://pt.wikipedia.org/wiki/HTML5>>. Acesso em ago.2015.

Android usando a API (Interface de Programação de Aplicativos) do Processing 2. A página orienta ainda que é uma versão beta, incompleta e contém bugs (PROCESSING, 2015).

O Processing 2 mantém um fórum oficial para sanar dúvidas da comunidade sobre o funcionamento do sistema Processing disponível em <http://forum.processing.org/two/>. A comunidade também mantém uma página oficial no facebook (Figura 20) e uma conta oficial no twitter com 12,1 mil seguidores.

Figura 20 - Página no facebook Processing.



Fonte: Autor (2015).

O software pode ser baixado gratuitamente na área de download do website [processing.org](http://processing.org). Antes da realização do download solicita-se uma doação para financiar a Fundação Processing, uma vez que o software constitui-se *open source*, desenvolvido por uma organização sem fins lucrativos. Assim, caso queira contribuir o usuário poderá fazê-lo (Figura 21).

Figura 21 - Contribuição facultativa para download.

Processing is open source, free software. All donations fund the [Processing Foundation](#), a nonprofit organization devoted to advancing the role of programming within the visual arts through developing Processing.

No Donation
  \$10
  \$25
  \$50
  \$100
  \$

[Donate & Download](#)

Fonte: Autor (2015).

No website são fornecidas versões para Sistemas Operacionais Linux, Mac OSX e Windows (Figura 22). A ferramenta não necessita de uma instalação específica, bastando descompactar os arquivos em uma pasta elegida para guardar os arquivos do software.

Figura 22 - Área de download Processing.

**Download Processing.** Processing is available for Linux, Mac OS X, and Windows. Select your choice to download the software below.



3.0.1 (23 October 2015)

**Windows** 64-bit  
**Windows** 32-bit

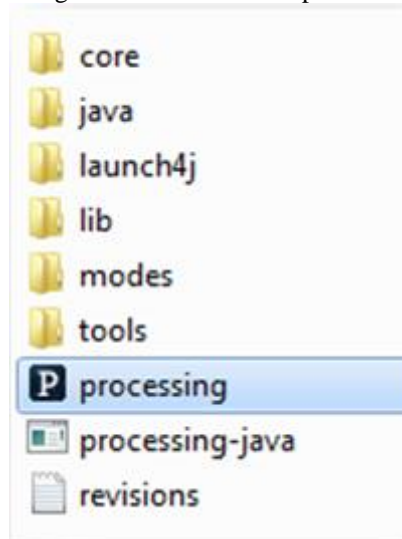
**Linux** 64-bit  
**Linux** 32-bit  
**Linux** ARMv6hf

**Mac OS X**

Fonte: Autor (2015).

Ao descompactar o arquivo na pasta elegida encontraremos seguinte estrutura (Figura 23).

Figura 23 - Pasta descompactada.



Fonte: Autor (2015).

O Processing 2 pode ser iniciado facilmente com dois cliques no arquivo executável, como destacado na Figura 23. As outras pastas do pacote contemplam bibliotecas e arquivos essenciais para a compilação dos programas desenvolvidos na ferramenta. Cabe salientar que a instalação ocorreu em todas as máquinas do laboratório, disponibilizando um atalho na área de trabalho, para facilitar utilizações futuras.

### 2.1.13 Quadro comparativo dos programas

Um quadro geral foi elaborado para resumir as propriedades/características de cada programa descrito nesta seção.

Quadro 4 - Características resumidas dos programas.

Ferramentas	Características	O que desenvolve?
<b>Kturtle</b>	Ferramenta gratuita, em que o usuário programa a tartaruga, usando os comandos do TurtleScript para executar uma determinada tarefa.	A ferramenta tem o objetivo de tornar a programação mais fácil, sendo utilizado para ensinar às crianças as noções básicas de matemática, geometria e programação.
<b>Ligth-bot</b>	É um puzzle clássico, gratuito e divertido, que tem como objetivo acender as luzes apresentadas no tabuleiro.	É um jogo para estimular o pensamento lógico, que requer pensamento analítico. Algumas ideias básicas de programação, como sequências e rotinas são empregadas, contribuindo para o desenvolvimento do pensamento recursivo.
<b>Code Monkey Island</b>	Jogo de tabuleiro que ensina às crianças a lógica da programação de computadores por meio da dinâmica lúdica de macacos.	Apresenta uma solução low-tech (baixa-tecnologia) para um problema de alta tecnologia, no qual o jogador deve dominar conceitos de lógica booleana e instruções condicionais.
<b>Robot Turtles</b>	Jogo que tem como objetivo levar uma tartaruga pelo tabuleiro evitando os obstáculos e outros jogadores para conseguir alcançar uma jóia.	O jogo ensina os fundamentos da programação para as crianças de forma implícita. Utiliza-se de instruções sequenciadas e depuração quando cometem erros. Assim, ao invés de utilizar um dado, os jogadores utilizar cartas com instruções básicas de programação que determinarão o movimento da tartaruga.
<b>Kodable</b>	Obtido gratuitamente para Ipad. A premissa é conduzir “bolinhas alienígenas” sobre determinados trajetos para coletar as moedas.	Ensina conceitos de sequência, condições e loops apenas arrastando e soltando os comandos para programar as ações de um personagem chamado “Smeeborg”.
<b>Robozzle</b>	Disponível gratuitamente, o puzzle conta com o apoio da comunidade que permite aos jogadores desenvolver novos desafios.	Desenvolve o pensamento recursivo na chamada a funções. O jogo desenvolve o pensamento criativo e habilidades de programação sem exigir conhecimentos técnicos previamente.
<b>Cargo-bot</b>	Programado inteiramente no iPad, pode ser adquirido gratuitamente com 36 desafios inteligentes na App Store.	É um puzzle onde se ensina um robô a mover caixas. Conceitos como loops, procedimentos e depuração são empregados. A lógica passo-a-passo ensina as crianças a enfrentar um problema maior, quebrando-o em etapas menores.
<b>Scratch</b>	Ferramenta gratuita que possibilita construir histórias interativas, jogos e animações.	O uso do Scratch desenvolve o pensamento lógico, pois trabalha com construção de algoritmos (lógica de programação), nos quais envolve conceitos matemáticos e potencializa o pensamento criativo em crianças do ensino fundamental conforme investigado por Martins (2012).
<b>Code Club</b>	Plataforma online destinada a ensinar as crianças com idades entre 9 a 11 como codificar.	Os cursos iniciais disponibilizados pelo Code Club utilizam o Scratch como ferramenta para ensinar as bases da programação de computadores.
<b>Codeacademy</b>	Um site que oferece aulas gratuitas de programação de computadores por meio de uma plataforma interativa.	Plataforma que ensina, de maneira interativa, diversas linguagens de programação e marcação. À medida que realizam as lições, pontos e medalhas como são recebidas como forma de incentivo.
<b>Code.org</b>	Organização sem fins lucrativos cujo objetivo é divulgar e ensinar programação a pessoas de todas as idades gratuitamente.	No tutorial para iniciantes, o aluno tem o primeiro contato com a programação de computadores por meio da linguagem Blockly, que possibilita arrastar e soltar blocos para escrever os códigos.
<b>Processing 2</b>	Ferramenta livre e gratuita, projetada para gerar e modificar imagens. Baseada em Java, originou-se no MIT.	Nos últimos anos, começou a se tornar uma presença na escola e as salas de aula do ensino fundamental e médio, sendo particularmente eficaz quando usado para ensinar ciência, matemática, e mais geralmente, ciência da computação.

Fonte: Autor (2015).

Para organizar cada uma das doze ferramentas apresentadas no Quadro 4, foram criadas as seguintes categorias, que permitem a classificação das ferramentas de acordo com interfaces, plataforma e finalidade. A seguir, uma breve descrição de cada uma das categorias elaboradas por este pesquisador e apresentadas no Quadro 5.

- Ferramentas de programação que empregam CUI (command-line user interface), em que linhas de código são usadas para interagir com programas de computador.

- Ferramentas de programação que utilizam interface GUI (Graphical User Interface) que permitem a interação de elementos gráficos como ícones e outros indicadores visuais, em contraste à interface de linha de comando.

- Puzzles, um tipo de game que apresenta desafios e para atender ao objetivo os jogadores fazem uso de fundamentos de programação como exemplo, laços de repetição, que podem desenvolver o pensamento recursivo.

- Jogos de tabuleiro os quais ensinam fundamentos do código por meio de uma plataforma não digital.

- Plataformas online que ensinam programação, e utilizam de ferramentas diversas, como Scratch, Blockly, entre outras.

Quadro 5 - Alternativas para aprender programação

<b>Ferramentas GUI (programação de arrastar e soltar)</b>	<b>Ferramentas CUI (programação que usa um editor de texto)</b>	<b>Plataformas não digitais (Jogos de tabuleiro)</b>	<b>Plataformas online que ensinam programação</b>	<b>Puzzles que ensinam fundamentos de programação</b>
Scratch	KTurtle (Logo)	Code Monkey Island	Code Club	Ligth-Bot
	Processing 2	Robot Turtles	Codecademy	Kodable
			Code.org	Robozzle
				Cargo-Bot

Fonte: Autor (2015).

Este trabalho de pesquisa utiliza a ferramenta Processing 2. Tal escolha se justifica pelo fácil aprendizado que o Processing 2.0 proporciona, ampla disponibilidade de tutoriais, exemplos e comunidades, o que torna o aprendizado rápido. Com poucas linhas de código conseguem-se obter resultados visuais. Outro fator considerado foi a fácil instalação, uma vez que basta fazer o download da IDE compilada para a plataforma desejada. O Processing apresenta uma IDE simplificada e amigável, o que torna a interação tranquila.

## 2.2 Aprendizagem Situada

O conceito de aprendizagem situada, os seus pressupostos e as principais questões associadas a esta teoria serão aqui apresentados. Os conceitos relacionados com a aprendizagem como um processo social são delimitados: aprendizagem situada, comunidade de prática e participação periférica legítima. O objetivo nesta seção não é consumir todas as possibilidades em torno da origem desse conceito, mas tentar incluir algumas das principais

ideias e as conexões necessárias para uma compreensão abrangente do assunto. O texto também se desenrola sobre os aspectos relacionados com algumas das dimensões do conceito.

A ideia de que a aprendizagem envolve um processo de participação numa comunidade de prática tem ganhado espaço significativo nos últimos anos. O termo comunidade de prática foi apresentado por Jean Lave, antropóloga, e Etienne Wenger, cientista da área computacional no livro “Situated learning - Legitimate peripheral participation” (1991).

A antropóloga social, Jean Lave da Universidade da Califórnia, em Berkeley, ao observar a atividade cotidiana, argumentou que "a aprendizagem é onipresente em toda atividade, embora muitas vezes não reconhecida como tal" (Lave, 1993, p.5). Assim, o argumento apresentado por Lave (1993) é que as comunidades de prática estão por toda parte: trabalho, escola, casa, ou em nossos interesses sociais e de lazer. Para Wenger (1998) comunidades de prática são formadas por pessoas que se envolvem em um processo de aprendizagem coletiva que compartilham uma paixão por algo que fazem e aprendem como fazê-lo melhor.

Em alguns grupos, alguns membros fazem parte do núcleo, em outros, estão nas partes mais periféricas. No entanto, os membros são reunidos por meio de atividades comuns e aprendem por meio do empenho mútuo nas atividades. Assim, uma comunidade de prática é diferente de uma comunidade de interesse ou de uma comunidade geográfica, pois contempla uma prática compartilhada (WENGER, 1998).

De acordo com Wenger (1998) são três elementos importantes (Quadro 6) para distinguir uma comunidade de prática de outros grupos.

Quadro 6 - Elementos importantes da comunidade de prática.

DOMÍNIO	COMUNIDADE	PRÁTICA
Comunidade de prática possui uma identidade definida por compartilhar um domínio de interesse, na qual implica num compromisso com o domínio. Assim, uma comunidade de prática é mais do que uma rede de conexões entre pessoas.	Ao prosseguir o seu interesse em seu domínio, os membros se envolvem em conjunto atividades e discussões, compartilham informações, constroem relações e interações que permitem aprender uns com os outros.	Os membros de uma comunidade de prática desenvolvem um repertório partilhado de recursos: experiências, ferramentas, formas de abordar problemas por meio de uma prática compartilhada.

Fonte: Wenger (1998).

Considerando os três elementos apresentados (Quadro 6), uma comunidade de prática compreende muito mais do que o conhecimento técnico ou especializado associado com a realização de alguma tarefa. Os membros estão envolvidos em um conjunto de relacionamentos ao longo do tempo, e a organização em torno de uma área específica de conhecimento proporciona aos membros uma identidade. (LAVE; WENGER, 1991, p. 98).

Todas as teorias de aprendizagem se baseiam em pressupostos fundamentais sobre a pessoa, o mundo e as suas relações. O foco principal desta teoria é na *aprendizagem como participação social*. A participação aqui não se refere apenas a eventos locais de engajamento em certas atividades com determinadas pessoas, mas, a um processo mais abrangente de participantes ativos nas práticas de comunidades sociais e na construção de *identidades* em relação a essas comunidades.

A perspectiva proporciona meios alternativos de aprendizagem, que segundo Lave e Wenger (1991), em vez de definir a aprendizagem como a aquisição de conhecimento proposicional, que segue uma proposição lógica, formal e fundamentada, argumentam que a aprendizagem “situa” em certas formas de coparticipação social.

Assim, ao invés de perguntar quais os tipos de processos cognitivos e estruturas conceituais estão envolvidos na aprendizagem, a pergunta oportuna seria quais tipos de compromissos sociais fornecem o contexto adequado para que a aprendizagem ocorra.

Lave e Wenger (1991) argumentam que o aprendizado não deve ser visto como uma simples transmissão de conhecimento abstrato e descontextualizado de um indivíduo para outro, mas um processo social pelo qual o conhecimento é co-construído. Sugere-se que a aprendizagem esteja situada em um contexto específico e incorporada dentro de um ambiente social e físico particular.

Os autores ressaltam que,

Repensar a aprendizagem a partir da perspectiva proporcionada pela Participação Periférica legítima passaria a ser um exercício fecundo. Tal estudo também levanta questões sobre a organização social das próprias escolas em comunidades de prática com variadas formas de associação (WENGER, 1991, p. 41).

A aprendizagem situada não é uma forma de educação, muito menos uma pedagogia estratégica, no entanto, essas ideias tiveram repercussões em muitos campos de estudo e, vieram a ser aplicadas a situações e ambientes diferentes, resultando em publicações recentes em educação matemática, e colocando a aprendizagem situada no pensamento sobre educação e pesquisa educacional.

Muitos dos exemplos da literatura da aprendizagem situada envolvem matemática, e tem particularmente influenciado pesquisadores na educação matemática. Particularmente importante foi a aprendizagem situada sobre a incompatibilidade das situações escolares típicas e situações do "mundo real", como a do local de trabalho, onde é preciso implantar conhecimento matemático. Maior ênfase deve ser dada à relação entre o que é aprendido em sala de aula e o que é necessário do lado de fora da sala de aula, e isso tem sido uma contribuição valiosa do movimento aprendizagem situada (ANDERSON; REDER; SIMON, 1996, p. 2).

Assim, aprendizagem situada passa a ser olhada como uma abordagem de ensino e segue o trabalho de Dewey, Vygotsky, e outros que afirmam que os alunos estão mais inclinados a aprender participando ativamente da experiência de aprendizagem.

Esta maneira de abordar a aprendizagem é algo mais do que simplesmente "aprender fazendo" ou aprendizagem experiencial. Para Wenger (1998) conceito de contextualização envolve pessoas para serem participantes plenos no mundo e na geração de significado.

Aprendizagem situada depende de duas reivindicações: não faz sentido falar de conhecimento que está descontextualizado, abstrato ou geral e os novos conhecimentos e aprendizagem são devidamente concebidos como sendo localizadas em comunidades de prática (TENNANT, 1997, p. 77).

A despreocupação com um modelo ou currículo pré-definido promove um relacionamento mais favorável à aprendizagem do que em relações de hierarquia explícita. O ensino informal está relacionado ao processo de socialização do homem. Os limites entre o ensino formal, informal e não formal são permeáveis, uma vez que estamos aprendendo constantemente por diferentes vias e agentes. Na aprendizagem situada, a atenção está voltada para o processo da prática social e a estrutura pedagógica é vista como uma fonte secundária (LAVE; WENGER, 1991).

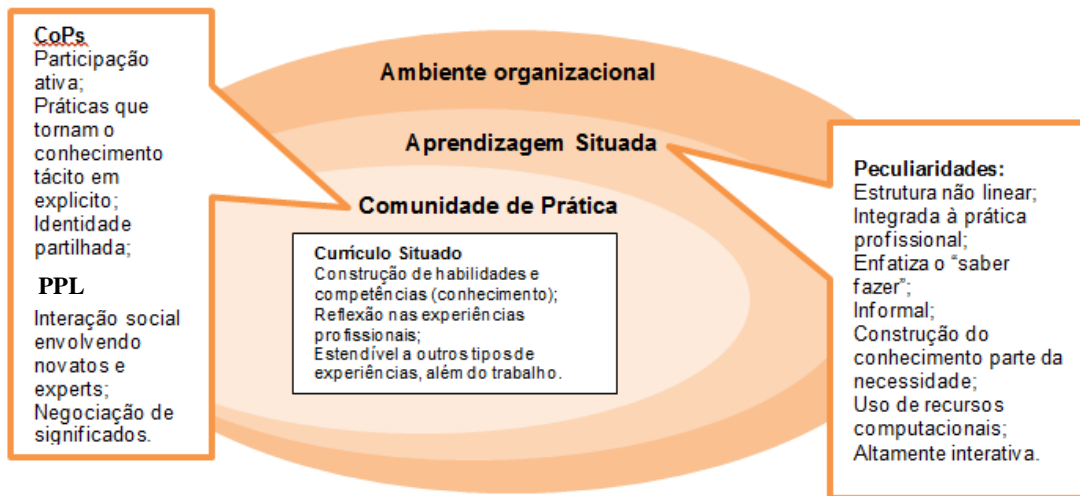
O aprendizado situado é essencialmente uma questão de criar significado para as atividades reais da vida diária onde a aprendizagem ocorre em relação ao ambiente de ensino (STEIN, 1998, p. 2).

Para reproduzir ambientes reais de trabalho, as salas de aula se transformam em laboratórios. Estudantes estão ativamente envolvidos na abordagem dos reais problemas do mundo (OTEC, 2007). Assim, constroem o conhecimento a partir de experiências que trazem para a situação de aprendizagem e o sucesso da aprendizagem situada baseia-se na interação social.

Muitas vezes o ensino ocorre a partir do abstrato, com experiências fora do contexto, num momento em que o conhecimento não é necessário, de forma passiva, linear, desassociado do fazer e com uso de poucos recursos computacionais. A aprendizagem situada, por outro lado, sugere que a aprendizagem ocorra por meio das relações entre as pessoas com a aprendizagem contextual autêntica, informal, e muitas vezes não intencional (OTEC, 2007).

Carvalho e Steil (2012, p. 8) apresentam um esquema que mostra a identificação das características e resultados da Aprendizagem Situada no ambiente organizacional (Figura 24).

Figura 24 - Esquema da Aprendizagem Situada.



Fonte: Baseado em Carvalho e Steil (2012, p. 8).

Como implicações da aprendizagem situada ocorrem: aperfeiçoamento profissional, soluções eficientes para as situações-problemas, conhecimentos e geração de novas ideias. Ambientes de aprendizagem situada colocam os alunos em situações de aprendizagem autênticas onde eles estão ativamente imersos em uma atividade ao usar a resolução de problemas (habilidades de pensamento crítico). Estas oportunidades devem envolver uma comunidade social que reproduz situações do mundo real. A experiência de aprendizagem situada deve incentivar os alunos a explorar seu conhecimento prévio e desafiar os outros em sua comunidade (STEIN, 1998).

### 2.2.1 Aportes teóricos da aprendizagem social

Desde a publicação de "Aprendizagem situada: participação periférica legítima" Lave e Wenger (1991), comunidades de prática têm sido o foco de atenção, em primeiro lugar como uma teoria da aprendizagem e mais tarde como parte do campo da gestão do conhecimento.

Para entender como a aprendizagem ocorre fora da sala de aula, os autores estudaram como os novatos em grupos informais (comunidades de prática) tornam-se membros efetivos desses grupos. Usou-se pela primeira vez o termo comunidades de prática para descrever a aprendizagem por meio da prática e participação, de onde surgiu o nome de aprendizagem situada.

A estrutura da comunidade é constituída ao longo do tempo por meio de um processo de Participação Periférica Legítima (PPL). *Legitimação e participação* definem as

características dos integrantes de uma comunidade enquanto *periférico e participação* estão preocupados com a localização e identidade no mundo social (LAVE; WENGER, 1991).

Em sua primeira obra, Lave e Wenger (1991) observaram a aprendizagem situada dentro de uma comunidade de prática entre parteiras, alfaiates, cortadores de carne, entre outros. Essa pesquisa descobriu que, quando os recém-chegados se juntam a um grupo ou comunidade estabelecida, eles gastam algum tempo inicialmente para observar como o grupo funciona e como eles podem participar. Os autores descrevem este processo de socialização como a Participação Periférica Legítima. O termo "comunidade de prática" refere-se ao grupo que compartilha um interesse comum e um desejo de aprender e contribuir com a comunidade, com sua variedade de experiências.

De forma geral, a maior parte da formação desses praticantes se dá na relação direta com seu orientador na execução do trabalho. Com o objetivo de ampliar a dimensão da relação orientador/aprendiz que incluísse a participação e a transformação da identidade no seio das comunidades de prática, em 1998, Wenger publicou o livro *Communities of practice: learning, meaning and identity*.

A sustentação da teoria social da aprendizagem proposta por Wenger envolve diversas áreas do conhecimento. A aprendizagem social seria então compreendida na intersecção entre campos teóricos distintos (Figura 25).

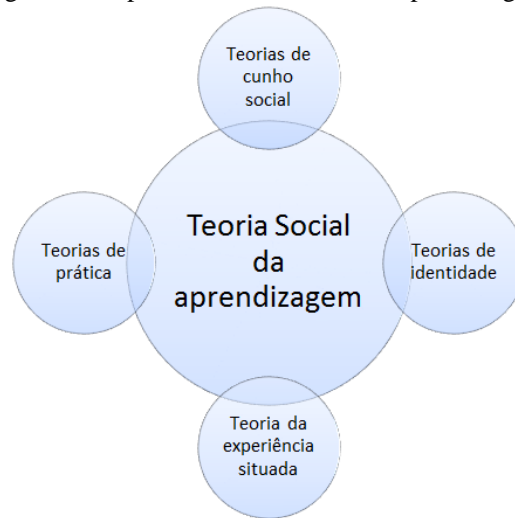
Wenger (1998) explica como ele localiza a sua teoria social da aprendizagem, na intersecção de dois eixos principais. Na representação (Figura 25), o eixo vertical trata das teorias da estrutura social e teorias de experiência situada e o eixo horizontal abrange a teorias de prática e as teorias de identidade.

No eixo vertical, as teorias da estrutura social fornecem insights sobre uma variedade de formações sociais e como as práticas decorrentes dessas formações sociais afetam nossas vidas num nível quase não visto. No outro extremo, as teorias de experiência situada fornecem explicações sobre como as regras são alteradas, como as normas mudam e como os indivíduos resistem às estruturas dominantes e escrevem suas próprias histórias (WENGER, 1998, p.14).

No eixo horizontal, as teorias de prática fornecem insights sobre as atividades sociais e culturais cotidianas, como o trabalho, escola, entre outros. Estas teorias, portanto, fornecem alguns insights sobre o que se faz em ambientes sociais, como se veste, se expressa e até mesmo maneiras pelas quais essas práticas são incorporadas na vida social, contextos históricos e nas relações de classe. Na outra extremidade desse mesmo eixo, encontram-se as teorias da identidade, as quais fornecem um quadro para compreender a formação e

construção da pessoa. Naturalmente, as questões de classe, raça e gênero, entre outras, são abordadas e reconhecidas como tendo influência sobre identidade (WENGER, 1998, p.16).

Figura 25 - Aportes da teoria social da aprendizagem.



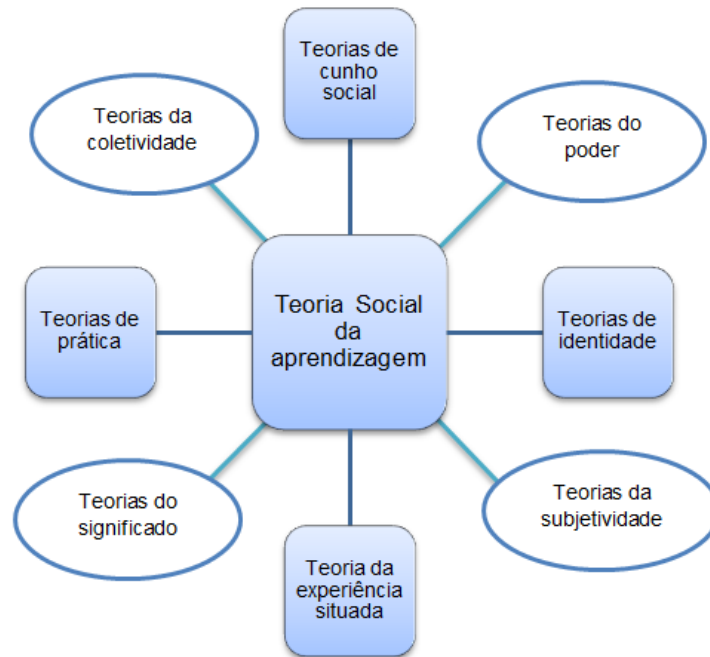
Fonte: Reprodução adaptada (WENGER, 1998, p.12).

Assim, para Wenger (1998) a aprendizagem como participação social seria central, entre os eixos, e ocorre pelo envolvimento em interações e ações, sendo este envolvimento incessantemente imerso na cultura e no contexto histórico.

O autor refina ainda mais a imagem, adicionando eixos diagonais (Figura 26). A teoria da identidade é colocada entre o poder e a subjetividade. Assim, o poder (estruturas socioeconômicas e políticas) e subjetividade (do agente) são dialeticamente envolvidos na (re) construção da identidade, e nessa relação a identidade é também influenciada.

A teoria da coletividade contempla diversas configurações sociais, desde as locais, como as famílias, comunidades, ou grupos até configurações mais amplas como: estados, classes sociais, associações, movimentos sociais, entre outros. Na outra extremidade do eixo diagonal, encontramos a teoria da subjetividade, que contempla a natureza da individualidade como uma prática de organização e como essa prática direciona para um engajamento no mundo social (WENGER, 1998, p.14).

Figura 26 - Refinamento das intersecções de tradições teóricas.



Fonte: Reprodução adaptada (WENGER, 1998, p.21).

Nessa perspectiva,

Aprender, não é nem totalmente subjetiva, nem totalmente englobado na interação social, e não é constituída separadamente do mundo social (com as suas próprias estruturas e significados). Isto recomenda uma visão descentralizada do significado da aprendizagem, em que a aprendizagem é reconhecida como um bem social, constituída na experiência, através da legítima participação periférica na prática social em curso; o processo de mudança de experiente habilidade é subsumido em processos de mudança de identidade através de participação em uma comunidade de praticantes (LAVE; WENGER, 1991, p.64).

No outro eixo diagonal, apresenta-se a teoria do poder, no qual o desafio consiste em encontrar conceituações de poder que evitem perspectivas conflituosas, como o poder enquanto dominação, opressão ou violência, bem como modelos simplesmente consensuais. No extremo, encontram-se as teorias do significado que tentam explicar como as pessoas produzem seus próprios significados. Assim, as comunidades de prática se desenvolvem e se estabelecem sob as relações políticas e econômicas associadas às instituições, que necessariamente implicam algum tipo de poder social (WENGER, 1998).

A internalização é ainda fundamental para um trabalho sobre aprendizagem preocupado com a sua natureza social. O autor esclarece que estamos conscientes de que o conceito de Vygotsky sobre Zona de Desenvolvimento Proximal (ZPD) recebeu amplamente diferentes interpretações, segundo o qual o conceito de internalização desempenha funções diferentes.

Sob esta ótica,

Engestrom (1987 apud LAVE e WENGER, 1991 p. 49) define a Zona de Desenvolvimento Proximal como a distância entre as ações cotidianas dos indivíduos e a forma historicamente nova da atividade social que pode ser gerada coletivamente como uma solução para um problema incorporado em ações cotidianas.

Interpretações sociais do conceito da Zona de Desenvolvimento Proximal mostra o interesse em estender o estudo da aprendizagem para além do contexto pedagógico, incluindo a estrutura do mundo social. Assim, a ênfase volta-se para as questões de transformação sociocultural com as mudanças das relações entre os novatos e veteranos no contexto da prática partilhada.

### **2.2.2 A aprendizagem como participação social em Comunidade de Prática (CoP)**

O foco principal do trabalho mais recente de Wenger está em entender como a participação social do indivíduo torna-o ativo nas práticas de comunidades sociais, e na construção de identidade por meio dessas comunidades (WENGER, 2000).

As características estruturais de uma comunidade de prática são novamente redefinidas para um domínio de conhecimento, uma noção de comunidade e uma prática (Quadro 7).

Quadro 7- Elementos importantes da CoP no domínio do conhecimento.

DOMÍNIO	COMUNIDADE	PRÁTICA
Um domínio do conhecimento cria um terreno comum, inspira os membros a participar, orienta a sua aprendizagem e dá sentido às suas ações.	A noção de uma comunidade cria o tecido social para a aprendizagem. Uma comunidade forte promove interações e incentiva uma vontade de partilhar ideias.	A prática é o foco específico em torno do qual a comunidade se desenvolve, as ações e mantém o seu núcleo de conhecimento.

Fonte: Wenger (2000).

Num grupo de pessoas interessadas em compartilhar informações e discutir sobre um determinado tema que lhes interessa, os membros não são necessariamente especialistas ou técnicos sobre o tema em torno do qual a Comunidade de interesse (CoI) se formou. O objetivo do CoI é fornecer um lugar onde as pessoas que compartilham um interesse comum possam trocar informações, fazer perguntas e expressar suas opiniões sobre o assunto. A afiliação em um CoI não depende de prática, necessita apenas de interesse no assunto. Diferentemente, uma CoP é um grupo de pessoas que são praticantes ativos. O propósito de uma CoP, é fornecer uma maneira para os participantes compartilhar dicas e melhores práticas, fazer perguntas a seus colegas, e prestar apoio para o outro. Assim, os membros das comunidades de prática são pensados para serem fontes mais eficientes e eficazes de informações e experiências (WENGER, 2000).

Estudos mostram que os trabalhadores gastam um terço de seu tempo à procura de informações e são cinco vezes mais propensos a recorrer a um colega de trabalho, em vez de uma fonte explícita de informação como um livro, manual, ou banco de dados (DAVENPORT; PRUSAK, 2000).

Os membros da comunidade possuem conhecimento tácito, que pode ser difícil de armazenar e recuperar. Assim, além do conjunto de práticas explícitas construídas historicamente, as comunidades possuem um repertório tácito (implícito) que se configura de regras não ditas, percepções, sinais e premissas subliminares. Com isso, um indivíduo pode compartilhar a melhor maneira de lidar com uma situação baseado em suas experiências e reduzir a curva de aprendizagem.

O tipo de informação que é partilhada e apropriada em uma CoP é ilimitada e com isso, as comunidades de prática ajudam os indivíduos na lacuna entre o que saber e o como saber. Como membros de comunidades de prática, indivíduos relatam aumento da comunicação com as pessoas, menos dependência de proximidade geográfica e a geração de novos conhecimentos (DUGUID, 2005).

A aprendizagem era considerada um processo individual, separado do restante das atividades. Assim, ensinava-se em salas de aulas isoladas e livre da participação do mundo exterior. Nessa abordagem o aprendiz está atento ao docente ou aos exercícios propostos e o processo avaliativo ocorre por meio de testes que apresentam o conhecimento fora de um contexto. Segundo Wenger (1998) a aprendizagem deriva de quatro premissas básicas conforme Figura 27.

Figura 27- Premissas da aprendizagem social de Wenger.



Fonte: Wenger (1998).

De acordo com Wenger (1998) somos seres sociais e este aspecto se configura como central na aprendizagem em relação ao ator, mundo, atividade, significado, cognição, aprendizagem e conhecimento. Para o autor, o conhecimento é uma questão de competência

no tocante às iniciativas que são significativas, sendo mediado socialmente, historicamente contextualizado e sempre aberto. Assim, o conhecer se transforma numa questão de participação na busca de determinadas iniciativas que proporcionem um engajamento ativo no mundo. Assim, o significado advém da habilidade de vivenciar o mundo.

Neste sentido,

[...] qualquer poder de abstração é completamente situado na vida das pessoas e na cultura que faz com que seja possível. O que é chamado de conhecimento geral não é privilegiado em relação a outros tipos de conhecimento. Ele pode ser adquirido apenas em circunstâncias específicas. A generalidade de qualquer forma de conhecimento sempre reside no poder de renegociar o significado do passado na construção do significado de circunstâncias presentes (WENGER, 1991, p. 34).

A teoria sobre a Participação Periférica Legítima não se destina à abstração, mas a uma tentativa de explorar as suas relações concretas. Pensar sobre um conceito como Participação Periférica Legítima é argumentar que o seu significado teórico deriva da riqueza das suas interligações na história, por meio do tempo e entre culturas (LAVE; WENGER, 1991). Assim, a abstração nesse sentido decorre da desconexão de uma prática cultural particular.

### ***2.2.3 Participação Periférica Legítima (PPL)***

Lave e Wenger (1991), ao invés de olhar para a aprendizagem como a aquisição de certas formas de conhecimento tentam colocá-la nas relações sociais em situações de coparticipação. Não se pergunta quais tipos de processos cognitivos e estruturas conceituais estão envolvidas, mas sim, quais tipos de compromissos sociais fornecem um contexto adequado para aprendizagem (WENGER, 1997).

As pessoas ingressam em uma comunidade como membros periféricos. À medida que os membros se tornam mais competentes, tornam-se mais envolvidos nos principais processos da comunidade particular. Assim, aprendizes novatos participam inevitavelmente em comunidades de praticantes e os conhecimentos e habilidades exigidos dos novatos movem-se em direção à participação plena nas práticas sociais de uma determinada comunidade (LAVE; WENGER, 1991, p.37).

Nesse sentido,

Parece tudo muito natural para decompô-la em um conjunto de três pares contrastantes: legítimos contra ilegítimo, periférica contra central, participação contra não-participação. Mas é nossa intenção que o conceito passe a ser tomado como um todo. Cada um dos seus aspectos é indispensável para a definição dos outros e não pode ser considerada isoladamente (WENGER, 1991, p. 35).

Assim, orienta-se para que o conceito seja compreendido numa totalidade e não seja fragmentado em legítima *versus* ilegítima, periférico *versus* central ou participação *versus* não participação. De acordo com Lave e Wenger (1991) a participação parcial dos recém-chegados não significa "desconectada" da prática de interesse e também é um conceito dinâmico. Neste sentido, corresponde a uma forma de ganhar acesso a fontes para a compreensão por meio de um envolvimento crescente.

Os alunos inevitavelmente participam de comunidades de praticantes e o domínio do conhecimento e habilidades requer que os novatos encaminhem para a plena participação nas práticas socioculturais de uma comunidade. O significado da aprendizagem está configurado por meio do processo de se tornar um participante completo em uma prática sociocultural. Este processo social inclui a aprendizagem de habilidades experientes (LAVE; WENGER, 1991, P. 29).

Deste modo a interação social que ocorre em comunidades de prática entre especialistas e iniciantes é crucial para a teoria da aprendizagem situada. Os novatos começam aprender observando os membros da comunidade e, em seguida, mover-se lentamente a partir da periferia da comunidade para participar plenamente como membros.

Wenger (1998) identifica três principais níveis de participação (Figura 28).

1) O grupo central (menor), com membros participantes intensamente na vida da comunidade, que geralmente promove discussões e projetos e normalmente assume papéis de liderança na condução do grupo.

2) O grupo ativo que participa regularmente, mas não ao nível dos líderes.

3) O grupo periférico que, enquanto participantes passivos na comunidade, ainda aprendem com seu nível de envolvimento. O autor observa que o terceiro grupo representa tipicamente a maior parte da comunidade.

Figura 28 - Níveis de participação na comunidade de prática.



Fonte: Adaptado de Wenger (1998).

Alguns membros ocupam um lugar central na comunidade e isto está relacionado com uma maior compreensão da prática, o que não implica necessariamente relação hierárquica. Para os membros periféricos se tornarem insiders<sup>20</sup> necessitam engajar nos processos internos, integrar o aspecto subjetivo da comunidade, assimilar sua língua, conhecer suas histórias e proceder como membros da comunidade (WENGER, 1997).

Deste modo, para o aluno, ser membro periférico é altamente produtivo, uma vez que o aproxima de profissionais engajados, e ao observar suas práticas, atentar a suas elucidações será formalizado um envolvimento efetivo. Enquanto participante nas práticas de uma comunidade, cada membro define sua identidade, sua maneira de agir, pensar e se relacionar como membro.

#### 2.2.4 Algumas dimensões do conceito

Lave e Wenger (1991) desenvolveram uma compreensão da natureza da aprendizagem no seio das comunidades de prática, e como o conhecimento gerado permite que os educadores pensem um pouco diferente sobre os grupos com os quais estão envolvidos. A noção de comunidade de prática e a conceituação mais ampla de aprendizagem situada fornecem indicações importantes para a prática.

<sup>20</sup> Insider é uma expressão em inglês para uma pessoa que tem acesso a informações privilegiadas nas empresas.

### 2.2.4.1 Identidade como pertencente

Wenger (1998) identifica três modos distintos de pertencer a uma comunidade de prática quando se considera a formação da identidade e da aprendizagem. Estes são: engajamento, imaginação e alinhamento.

**Engajamento** - Esta maneira de pertencer é evidenciada pela participação ativa nos processos mútuos de negociação de significado incluindo formação de trajetórias, histórias compartilhadas de aprendizagem, relações, interações e práticas.

Há uma essência de mutualidade no engajamento, em que as pessoas interagem com os outros, dentro e fora da comunidade. Este tipo de envolvimento dá origem a comunidades de prática que se fortalecem ao longo do tempo. O engajamento tem um carácter limitado no tempo e no espaço, porque há limites para o que podemos fazer. No entanto, o envolvimento pode ser estreito. Seu poder para sustentar identidade como praticante em uma comunidade pode levar uma pessoa a desenvolver uma "visão de túnel"<sup>21</sup>. Assim a passagem de fronteira e limite de trabalho poderiam ser minimizados, ou mesmo excluídos.

**Imaginação** - No contexto de pertencer, a imaginação não é uma fuga da realidade, mas a capacidade do modo de se ver em outros contextos ou mundos imaginários. Possivelmente este modo de pertencer prevê uma trajetória de diversas maneiras, incluindo: a criação de imagens de possibilidades, a criação de imagens do mundo, a criação de imagens de nós mesmos, o reconhecimento de imagens do passado e do presente, o reconhecimento da própria experiência em padrões mais amplos, o equilíbrio de um ato delicado de identidade, jogando com a participação e a não participação, no interior e no exterior, real e possível.

**Alinhamento** - Como forma de pertencimento, o alinhamento é um escopo amplo, envolvendo uma pessoa em: discursos, estilos, conformidade, complexidade. Tornar-se parte de algo grande, ao contrário de imaginação e engajamento, encaixando-se em estruturas mais amplas. No entanto, o alinhamento pode envolver o poder de forma destrutiva, em que partidários simplesmente estão em conformidade com as exigências feitas sobre elas permitindo identidades de ser oprimido. Como exemplo, um novo currículo, com um tipo diferente de alinhamento, os professores podem simplesmente ensinar de uma forma baseada em resultados cumprindo exigências feitas sobre eles, suprimindo sua própria identidade docente.

---

<sup>21</sup> Em teoria da identidade, o termo "passagem de fronteira" refere-se à passagem das fronteiras ou limites entre comunidades de prática.

#### **2.2.4.2 Identidade - Trajetórias dentro de Participação**

Dentro desses modos de pertencer são incorporadas formas de participação e não participação. A participação pode ser entendida como as maneiras pelas quais nos envolvemos nas práticas da comunidade.

Uma identidade coerente, portanto, é uma "mistura de estar dentro e estar fora" (WENGER, 1998). Um grau de interação entre essas formas (participação e não-participação) define a trajetória, sendo este o trajeto que um participante segue à entrada numa comunidade de prática.

Diferentes graus de interação implicam diferentes experiências, diferentes trajetórias e diferentes formas de participação/não-participação nas práticas sociais da comunidade. Os recém-chegados em qualquer comunidade são periféricos para a comunidade e suas práticas em matéria de entrada, mas, na medida em que embarcam em uma trajetória, tornam-se mais tarde "veteranos", na medida que adquirem mais habilidades e competências. A trajetória é muito diferente daquela de marginalidade na qual a não participação é dominante e a plena não é alcançada, o que pode gerar uma situação de desqualificação (desempowering position) como abordam os autores (LAVE; WENGER, 1991).

Com o envolvimento nas atividades periféricas, novatos se familiarizam com as tarefas, vocabulário e princípios de organização da comunidade. Gradualmente, com o passar do tempo, os recém-chegados tornam-se veteranos e surgem situações que permitem aos participantes avaliar quão bem estão a contribuir com seus esforços, assim, a participação periférica legítima fornece um meio de auto avaliação (LAVE; WENGER, 1991).

#### **2.2.4.3 Repertório compartilhado**

Um repertório compartilhado é um recurso comum que cria, mantém e reforça uma comunidade de prática. O engajamento mútuo em uma comunidade ao longo do tempo pode dar origem a um repertório compartilhado que envolve normas linguísticas e práticas sociais (WENGER, 1998, p. 83).

Os estilos linguísticos incluem as formas de pronúncia de palavras, gírias, jargões compartilhados e terminologia especializada. O campo de aplicação das práticas sociais engloba maneiras comuns de fazer as coisas, pensamentos, discursos, entendimentos e memórias compartilhadas.

O repertório compartilhado é um recurso fundamental para que os membros de uma comunidade se envolvam conjuntamente na negociação de significado e, por sua vez facilita

um maior envolvimento mútuo na prática. Assim, um produto de uma comunidade de prática ao longo do tempo, inclui rotinas, palavras, ferramentas, formas de fazer as coisas, histórias, gestos, símbolos, gêneros, ações ou conceitos que a comunidade produz ou adapta ao processo de existência, e que se tornam parte da prática.

#### **2.2.4.4 A presença social**

Uma comunidade de prática envolve a criação de presença social. Define a presença social como o grau de importância de uma pessoa numa interação e a sua importância numa relação interpessoal (TU, 2002, p. 38). De acordo com o autor, acredita-se que a presença social afeta a probabilidade de um indivíduo participar de uma CoP. A gestão de uma comunidade de prática muitas vezes enfrenta muitas barreiras que inibem as pessoas de se envolver em troca de conhecimentos. Alguns fatores permitem melhor engajamento numa comunidade de prática, como a motivação e colaboração.

A motivação para compartilhar o conhecimento é fundamental para o sucesso em uma comunidade de prática. Estudos mostram que os membros são motivados ao se tornarem participantes ativos em uma CoP quando veem o conhecimento como significativo para o bem da comunidade (ARDICHVILLI; WENTLING, 2003).

A colaboração é essencial para que as comunidades de prática prosperem. Fatores podem indicar um maior nível de colaboração no intercâmbio de conhecimento em uma rede de negócios, em que participantes mais experientes tendem a promover uma cultura mais colaborativa, como por exemplo, o maior nível de escolaridade (SVEIBY; SIMON, 2002).

#### **2.2.5 *Membros PPL numa comunidade de prática de programadores***

A aprendizagem começa com as pessoas tentando resolver problemas. Quando a aprendizagem é baseada num problema, as pessoas exploram situações da vida real para encontrar respostas. Quando os alunos estão nessas situações da vida real são direcionados a aprender. A abordagem de aprendizagem baseada em problemas conduz os estudantes para um maior nível de pensamento (HUNG, 2002).

O aprendizado situado está envolvido com tecnologia. A interação entre máquina e usuário se constitui em mais uma possibilidade de fazer da aprendizagem situada bem sucedida, e proporciona aos alunos uma oportunidade de se ter outro meio para aprendizagem (HALVERSON, 2009, p. 83).

De acordo com Gee (2008), pesquisadores de diversas áreas têm levantado a possibilidade da aprendizagem por meio de tecnologias digitais (jogos digitais) por facilitar a aprendizagem situada no contexto da atividade e experiência. Desta forma, a compreensão dos jogos de vídeo game modernos como ambientes de aprendizagem está se tornando cada vez mais importante. O autor argumenta que a natureza convincente de participação no jogo digital é, em parte, devido aos princípios subjacentes de aprendizagem sociais, cognitivas e de desenvolvimento em torno dos quais jogos de sucesso são construídos, pois, para o autor, bons jogos comerciais são forçados a incorporar bons princípios de aprendizagem, uma vez que se nenhum jogador conseguisse aprender os games, ninguém os compraria. Jogos fáceis, que requer domínio de poucas habilidades não são atrativos (GEE, 2008).

Assim, os estudantes que aprendem situados são capazes de usar com precisão as habilidades apropriadas por meio da experiência e com essas experiências eles são capazes de ser produtivos em suas vidas.

Para Halverson (2009) quase todas as habilidades relacionadas ao trabalho podem ser ensinadas. A resolução de problemas e a geração de novas ideias podem ser melhor apoiadas em um ambiente de aprendizagem social onde todas as partes interessadas experimentam os efeitos positivos da aprendizagem contínua.

A aprendizagem situada oferece uma oportunidade de trabalhar com outras pessoas e considerar a melhor forma de aplicar novos conceitos relacionados com o contexto específico de sua prática. É perceptível que se os alunos estivessem em um ambiente educacional onde aprendizagem situada fosse implementada, eles estariam preparados para o seu futuro. Assim, quando colocados em situações onde se aprende fazendo, provavelmente serão bem sucedidos. Há metas fortes para o aprender situado em que os alunos, como participantes periféricos podem desenvolver uma visão do todo. Um currículo de aprendizagem situada se desdobra em oportunidades de participação na prática e não é especificado como um conjunto de ordens para uma boa prática (LAVE; WENGER, 1991).

Assim, o conhecimento do aluno é definido pelo processo social de interagir e fazer. Ele permite aos alunos se apropriarem do conhecimento naturalmente como resultado de um comportamento social. O aplicativo de computador atua como um guia enquanto os alunos aprendem naturalmente a partir de sua própria experiência. Com isso, a aprendizagem situada acelera o processo e a capacidade de aprendizagem do estudante.

Lave e Wenger (1991) ao invés de aprender por replicar as performances dos outros, ou por adquirir conhecimentos transmitidos em instrução, sugerem que o lugar do

conhecimento está dentro de uma comunidade da prática, as questões de aprendizagem devem ser abordadas dentro dos ciclos de desenvolvimento daquela comunidade.

Um ambiente de aprendizagem situado prevê treinamento em momentos críticos, e os andaimes de apoio, em que o professor fornece as habilidades e estratégias aos alunos incapazes de completar a tarefa são importantes. Aos poucos, o suporte (o andaime) é removido até que o aluno seja capaz de avançar sozinho (YOUNG, 1993).

Assim, compreender a teoria da aprendizagem situada em comunidades de prática formalizadas por Lave e Wenger (1991) e Wenger (1998), permitiu olhar para o ambiente educacional dentro dos Laboratórios de Informática Educacional (LIEs) voltado à formação de comunidades de prática de programadores, enquanto membros periféricos legítimos, possibilitando a aprendizagem situada.

Como resultado do desenvolvimento de suas práticas que são compartilhadas mutuamente, entender como a identidade é compartilhada no grupo de programação, permite analisar como elas se organizam para (re)pensar as atividades educativas no LIEs, e ainda, ao analisar as relações que se estabelecem, espera-se identificar as inúmeras habilidades e quais habilidades são reificadas no processo de se construir uma identidade.

Nossa sociedade pressupõe que muitos profissionais tenham seu local apropriado de trabalho. O bom desempenho de todo profissional depende também dos ambientes e dos instrumentos disponíveis. Para aqueles que possuem uma visão atualizada de educação matemática, o LIE se configura como ambiente favorável à criação de situações pedagógicas desafiadoras, por meio da formação de comunidades de prática de programadores.

Esperamos que as discussões a serem levantadas na análise desta pesquisa promovam uma reflexão acerca da emaranhada rede de relações que se estabelecem numa comunidade de prática constituída por meio das oficinas de programação dentro dos LIEs e da maneira como os educadores e alunos constroem suas práticas e aprendem situados nesses espaços.

### ***2.2.6 Trabalhos correlatos***

Recentes pesquisas têm procurado compreender a partir da teoria da aprendizagem situada cunhada por Lave e Wenger (1991) a aprendizagem situada da matemática em diferentes comunidades de prática, sob uma abordagem que envolve contexto, participação, engajamento, identidade e aprendizagem em CoPs. Os trabalhos de Fernandes (2004) e Batista (2014) são exemplos encontrados na literatura. Fernandes (2004) procura identificar e caracterizar a atividade matemática dos alunos em práticas não socialmente definidas como Matemática (Práticas de Serralharia) e perceber como essa atividade pode ser ligada ao

currículo da Matemática escolar e ao seu desenvolvimento. O autor evidencia diferenças significativas nos dois contextos de aprendizagem e as conclusões do estudo permitem caracterizar as diferentes matemáticas dos alunos, bem como discutir a origem do conhecimento matemático evidenciado em práticas profissionais e em práticas escolares (FERNANDES, 2004).

O trabalho de Batista (2014) investiga que aprendizagens são produzidas no interior de uma comunidade de aprendizes de matemática e como a identidade dos participantes se transforma, apropriando-se dos saberes e práticas da comunidade. As análises permitiram constatar que os alunos em uma comunidade de aprendizes de matemática, podem compartilhar saberes e práticas que trazem de seu mundo social, integrando-as com os saberes escolares, mobilizando processos de negociação de significados e transformação de identidades ao longo das aulas de matemática (BATISTA, 2014).

Assim, este trabalho de pesquisa vem somar aos trabalhos apresentados por Fernandes (2004) e Batista (2014) e trará contribuições no campo da aprendizagem situada da matemática, uma vez que aborda questões relacionadas à aprendizagem de alunos com baixo rendimento em matemática em uma comunidade de prática de programadores, tendo como suporte um instrumento didático-pedagógico que é a programação de computadores articulada com a resolução de problemas que, após levantamento do estado da arte, parece não se encontrar na literatura em educação matemática com tanta frequência.

Buscou-se em diferentes plataformas, entre elas a da CAPES, trabalhos relacionados com o uso do Processing no ensino e aprendizagem da matemática e parece não se encontrarem estudos nesse campo com regularidade. Assim, a presente pesquisa aborda questões que, do meu ponto de vista, ainda necessitam ser aprofundadas na área do desenvolvimento do raciocínio e aprendizagem matemática.

### **2.3 O papel do erro em programação**

Na atividade de depuração<sup>22</sup>, o aprendiz analisará como o programa se comporta, e nesse processo reflete sobre o que fez e realiza o “pensar sobre o pensar”. Se algo está errado com seu raciocínio, o aprendiz pensará sobre o que poderia estar errado e tentará com base em erros observados, encontrar soluções corretas para problemas, alterando o programa.

Esse processo permite que o aluno deixe de pensar no correto e no errado e se volte na busca de uma solução aceitável. O erro passa a ser então um revisor de ideias e não mais um

---

<sup>22</sup> Entende-se como depuração o processo de testar e encontrar erros na construção do software.

motivo de punição, intimidação e frustração. A forma como o aluno encara a ocorrência de erros, procurando uma melhor compreensão das estratégias e dos conceitos envolvidos na solução adotada, identifica seu estilo de pensar sobre si mesmo e de relacionar-se com o mundo (ALMEIDA, 2000).

Os erros são parte importante da nossa experiência e se não os conhecemos podemos vir a repeti-los. Assim, o erro representa um papel construtivo na aquisição do conhecimento. Errar é um direito do aluno, mas compete ao mediador (professor/computador) fazer com que os alunos/usuários reflitam sobre erros realizados (MATUI, 1995).

Valente (1997) reflete sobre mudanças profundas no sistema educacional, entre elas o papel do erro:

[...] como a alteração do papel atribuído ao erro (não mais para ser punido, mas para ser depurado), a não segregação das disciplinas, a promoção da autonomia do professor e dos alunos e a flexibilização de um sistema rígido, centralizado e controlador. Enfim, transformar a escola que nós conhecemos (VALENTE, 1997, p.4).

De acordo com Valente (1999) as características disponíveis no processo de programação como o da linguagem Logo, alteram o papel do erro e o processo de achar e corrigir o erro constitui uma oportunidade única para o aluno aprender sobre um determinado conceito envolvido na solução e/ou elaboração de estratégias.

Valente (1997) salienta que geralmente o erro é tratado como um indicativo de ignorância ou não aprendizagem, mas em programação o erro faz parte do processo. É necessário notar o erro do ponto de vista de um evento rico, que também fornece aos alunos e professores a oportunidade de compartilhar ideias em conjunto e coordenar suas ações, a fim de identificar o erro e tomar medidas para mudar isso. Assim, em programação o erro é tratado como uma tentativa de acertar.

Com isso, o erro é colocado em um lugar de destaque e utilizado como importante mediador da aprendizagem. Ao desenvolver um programa usando uma linguagem de programação visual, o aluno tem a oportunidade de pensar sobre o que ele está fazendo e criar possibilidades de resolução de problemas. Neste contexto, o erro deve ser visto como um importante aliado no processo ensino-aprendizagem.

## 2.4 Programação de computadores como instrumento de ensino e o pensamento computacional

O emprego da programação como instrumento didático-pedagógico favorece a aprendizagem ativa, uma vez que a interação estabelecida entre as ações dos estudantes e o feedback do programa favorece a dinâmica. Nesta abordagem, o aluno descreve os passos necessários para atingir o objetivo, sendo direcionado a transformar seus conhecimentos em procedimentos (ALMEIDA, 2000).

Com o objetivo de possibilitar o uso pedagógico do computador em uma aproximação construcionista<sup>23</sup> em detrimento da instrucionista<sup>24</sup>, Papert (1986), desenvolve a linguagem de programação “Logo”, que possibilita a criação de novas situações de aprendizagem que levam o aprendiz a refletir sobre o processo de construção.

Assim que um programa é executado pelo computador, podem ocorrer duas situações:

- O resultado atendeu as expectativas, portanto a atividade é concluída.
- O resultado não correspondeu ao esperado, e exige rever o processo de estruturação das estratégias lógicas/matemáticas empregadas na construção do algoritmo.

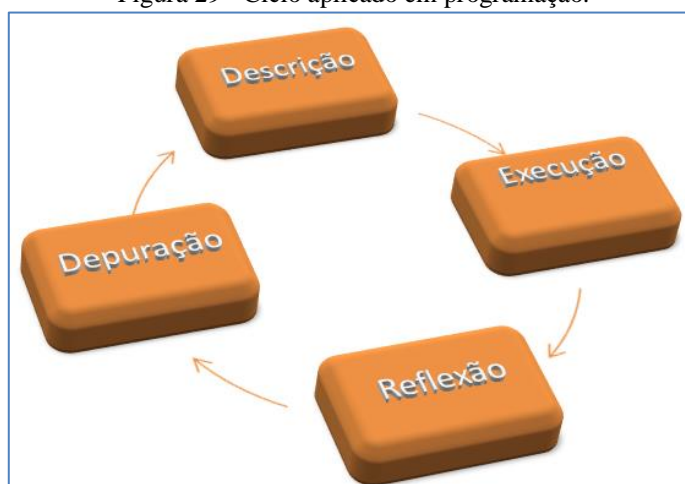
A execução do programa promove um *feedback* imediato, levando o estudante a promover reflexões acerca do problema. Tenta-se compreender as estratégias empregadas, os conceitos envolvidos, os erros e as possíveis formas de corrigi-los. Assim, o aluno passa a depurar seu programa, e inserir nele novas estratégias. O ciclo volta a se repetir até a obtenção de um resultado satisfatório, que atenda o propósito do desafio e/ou situação-problema proposta (Figura 29).

---

<sup>23</sup> Na abordagem construcionista o computador proporciona situações de aprendizagem que levam o aluno a refletir sobre tudo o que está sendo representado (ALMEIDA, 2000, p. 32).

<sup>24</sup> Na abordagem instrucionista o programa de ensino é o mesmo, a diferença é que o computador passa somente transmitir informações e não deixa explícito o pensamento do aluno (ALMEIDA, 2000, p. 26).

Figura 29 - Ciclo aplicado em programação.



Fonte: Autor (2015).

Segundo Almeida (2000) o ciclo de descrição-execução-reflexão-depuração, primeiramente empregado nas atividades de programação na linguagem “Logo”, possibilita a transferência da abordagem para outros ambientes, externos ao ambiente computacional. Assim, o “ciclo descrição-execução-reflexão-depuração é uma característica da atividade computacional, transferível e aplicável em outras situações” (Almeida, 2000, p. 47).

O ensino dos princípios computacionais proporciona o desenvolvimento do pensamento computacional, que segundo Wing (2006) é uma habilidade fundamental para todos, não apenas para cientistas da computação e pode ser observada nos processos de leitura, escrita e matemática como parte integrante da habilidade analítica dos estudantes.

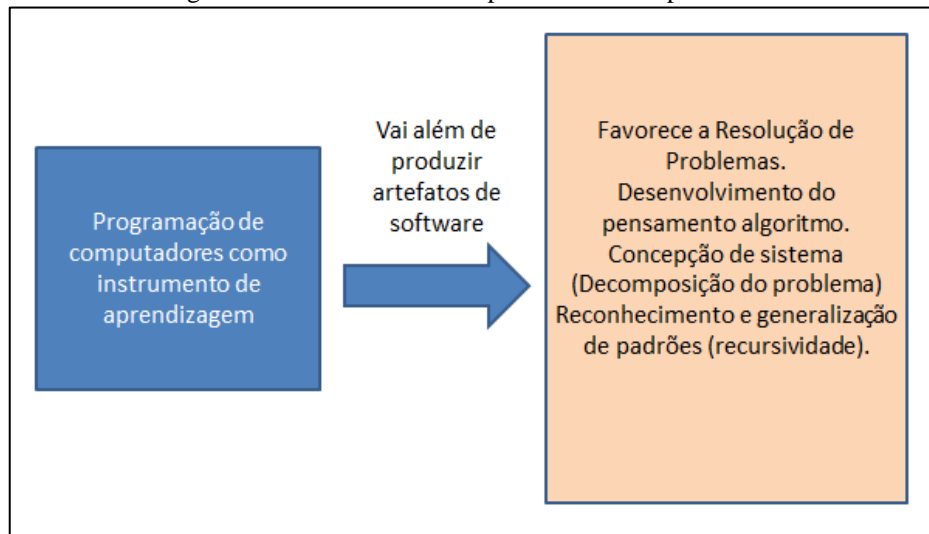
Pensamento computacional envolve a resolução de problemas, a concepção de sistemas, a compreensão do comportamento humano, construídos sobre os conceitos fundamentais da ciência da computação. Pensamento computacional inclui uma variedade de ferramentas mentais que refletem a magnitude do campo da ciência da computação (WING, 2006).

O pensamento computacional sistematiza os passos da solução de problemas, favorece a abstração e decomposição de uma tarefa complexa em que os alunos conseguem fragmentar os problemas da vida em partes funcionais para compreender o todo (WING, 2008).

O pensamento computacional é uma habilidade analítica que se constrói durante o desenvolvimento cognitivo da criança para encontrar os passos necessários para resolver os problemas (algoritmos). Este tipo de pensamento vai além do uso mecânico do computador, e utiliza os princípios básicos da computação para resolução de problemas, expandindo a capacidade do pensamento humano (WING, 2006).

Para sistematizar algumas características do pensamento computacional apresentadas por Wing (2006, 2008) foi elaborado um diagrama (Figura 30).

Figura 30 - Características do pensamento computacional.



Fonte: Autor (2015).

A proposta de Wing (2006) de resolver um problema aplicando o pensamento computacional significa reduzir grandes problemas em menores e mais simples de se resolver. Isso exige a capacidade de pensar de forma abstrata e em múltiplos níveis, e não apenas se restringe à aplicação de técnicas de programação. O pensamento computacional conjuga matemática e engenharia e representa a maneira que as pessoas pensam, gera ideias e não simplesmente artefatos de software e assim pode ser útil a todas as pessoas (WING, 2006).

Os trabalhos de Setti (2009) investigam o ensino e a aprendizagem de introdução à programação de computadores, focados na elaboração de algoritmos computacionais. A autora observa que alunos com conhecimentos matemáticos apropriados, encontram dificuldades na passagem do raciocínio matemático para o correspondente computacional. Na realização dessa passagem, a autora destaca que é necessário utilizar os conhecimentos matemáticos apropriados com um novo formato, devido ao processo de discretização necessário para transformar o raciocínio matemático no correspondente computacional e identifica aspectos de estruturas de repetição como necessárias para realizar o processo de particionar o todo em partes com menor complexidade.

Com isso, justifica-se a utilização da programação de computadores como instrumento de ensino que contribui para tornar o pensamento computacional difundido em nossas salas de aula, uma vez que pode desenvolver e aprimorar outras habilidades importantes conforme apresentadas.

## CAPÍTULO 3 – METODOLOGIA DA PESQUISA

Apresenta-se neste capítulo a metodologia empregada na pesquisa e as categorias de análise que foram estruturadas para constituí-la. O trabalho apresenta características de uma abordagem qualitativa, na qual se investigou as falas dos alunos e suas interações enquanto aprendiam programação de computador utilizando a ferramenta Processing 2.

### 3.1 O contexto

Para a realização da pesquisa foi escolhida uma escola da rede municipal de ensino fundamental (EMEF) do município de Santa Cruz do Rio Pardo – SP. A instituição se configura a maior escola da rede com cerca de mil alunos. Localizada num bairro residencial a 6 minutos do centro da cidade, a escola conta com um laboratório de informática educacional conquistado pela adesão ao Programa Nacional de Tecnologia Educacional (ProInfo) que levou à escola solução multiterminal com 8 CPUs, 16 terminais de acesso e 1 servidor multimídia. Na ocasião, a escola conquistou dois laboratórios, sendo constituída de uma infraestrutura única com 32 terminais de acesso.

O Sistema Operacional (SO) instalado nas máquinas segue diretrizes do governo federal, nas quais o Ministério da Educação (MEC) incentiva a utilização de softwares livres e produz conteúdos específicos, voltados para o uso didático-pedagógico, associados à distribuição Linux-Educacional (LE). A escola conta com a versão 3.0. e algumas máquinas com a versão atualizada do LE 5.0 segue a tendência das interfaces de dispositivos móveis e dos SOs mais recentes. A seguir um panorama geral da área de trabalho do LE 3.0 e LE 5.0 (Figura 31), em que se nota a Edubar<sup>25</sup> simplificada e um papel de parede mais simples, com a imagem de um símbolo nacional, o Sabiá-Laranjeira.

---

<sup>25</sup> Edubar é uma barra localizada na área de trabalho que permite ao usuário acesso rápido aos conteúdos educacionais e aos portais do MEC.

Figura 31- Área de trabalho das versões LE 3.0 e 5.0, respectivamente.



Fonte: Autor (2015).

Em sua versão mais atual a busca e a ativação dos aplicativos pelo usuário são realizadas de forma simplificada, aproveitando recursos de estruturação e de busca do Gnome<sup>26</sup> presentes nos dispositivos móveis.

A rede não conta com aula de informática na grande curricular, bem como extracurricular, sendo o Laboratório de Informática Educacional utilizado como ambiente de apoio pedagógico, compartilhado com mais diferentes disciplinas do currículo.

A rede municipal de Santa Cruz do Rio Pardo incorporou no ano de 2007 um sistema apostilado de ensino. A adoção do sistema apostilado tem se expandido na educação básica, em substituição aos tradicionais livros didáticos fornecidos pelo Programa Nacional do Livro Didático (PNLD). Nesse programa a equipe escolar escolhe coleções de livros didáticos que melhor atendem ao Projeto Político Pedagógico (PPP) da escola, que tem como principal objetivo subsidiar o trabalho pedagógico dos docentes.

A tendência de apostilamento dos sistemas de ensino é notável e se alastrou pelas prefeituras da região de Santa Cruz do Rio Pardo. Impulsionada por grupos privados que visam a esse nicho de mercado, em atendimento aos gestores públicos, busca a padronização dos ritmos de aprendizagem e dos conteúdos curriculares. Assim, nessa perspectiva padronizar e homogeneizar constitui umas das funções atribuídas aos sistemas de ensino apostilados, desconsiderando-se a diversidade e as especificidades de cada unidade escolar (ADRIÃO et al., 2009). Sob essa lógica a prefeitura firmou parceria com o Sistema Anglo de Ensino, pertencente ao grupo Abril, que há oito anos fornece suporte pedagógico à rede municipal.

O principal motivo da escolha da escola foi o trânsito facilitado com a direção da unidade escolar que prontamente permitiu a realização da pesquisa, sendo solícita em todos os

<sup>26</sup> Gnome é um projeto de software livre. Com ênfase especial à usabilidade, acessibilidade e internacionalização, a comunidade de desenvolvimento do GNOME conta com voluntários e o pode ser utilizado por vários sistemas, principalmente por sistemas Linux.

momentos, cedendo por duas semanas o laboratório de informática no período matutino para a realização da coleta de dados. Outro ponto de destaque refere-se à composição do quadro docente da disciplina de matemática que conta com profissionais dispostos a fazer uso de ferramentas tecnológicas inovadoras, com a finalidade de diversificar a metodologia empregada, proporcionando aos estudantes novas formas de aprendizagem.

Em seu Projeto Político Pedagógico (PPP) a unidade escolar enfatiza que o eixo da tecnologia deve permear o currículo e suas disciplinas por meio de uma abordagem interdisciplinar. Assim a proposta reconhece a tecnologia como eixo integrador de todas as áreas do conhecimento, e ainda considera que o impacto do progressivo acesso às tecnologias digitais vem transformando, em ritmo acelerado, as metodologias de ensino e o conceito de aprendizagem no ambiente escolar e fora dele, configurando num fator estimulador para realização deste trabalho de pesquisa na referida unidade.

Inicialmente foi estabelecido contato com a Secretaria Municipal de Educação para autorização do desenvolvimento desta pesquisa. Assim, após autorização, o pesquisador iniciou as atividades de organização e planejamento do Laboratório de Informática Educacional com a instalação da ferramenta Processing 2. Neste ponto é interessante ressaltar que não houve nenhum problema decorrente da instalação da ferramenta Processing 2 no SO Linux-Educacional 3.0, disponível no LIE da Unidade Escolar.

### **3.2 Os participantes**

Para seleção dos sujeitos da pesquisa, foi solicitado para que a professora de matemática das turmas do 7<sup>o</sup> ano do período vespertino, apontasse seis alunos que apresentassem baixo rendimento na disciplina. Assim, a seleção ficou a critério da professora e foram indicados seis alunos representados nesta pesquisa conforme demonstrado no quadro 8.

E por que alunos com baixo desempenho? Tomando por base uma curva de distribuição normal, o grande desafio é trazer os alunos que se encontram nas extremidades dessa curva para a região do desvio padrão da média. Ensinar um aluno bom é na maioria das vezes fácil e o grande desafio consiste em fazer com que o aluno do nível abaixo do básico, ou básico se torne um aluno adequado, que demonstre domínio pleno de habilidades e competências desejáveis para o ano/série escolar em que se encontra (SÃO PAULO, 2015).

Quadro 8 - Sujeitos da pesquisa.

<b>Estudante (iniciais do nome)</b>	<b>Sexo</b>	<b>Idade</b>
<b>CA</b>	Masculino	12 anos
<b>KA</b>	Feminino	12 anos
<b>MB</b>	Masculino	12 anos
<b>NB</b>	Feminino	12 anos
<b>TS</b>	Feminino	12 anos
<b>VH</b>	Masculino	13 anos

Fonte: Autor (2015).

Após os apontamentos realizados pela professora da disciplina de matemática, o pesquisador encaminhou aos pais/responsáveis dos alunos selecionados um Termo de Consentimento Livre e Esclarecido (Apêndice A) com o objetivo de informar aos responsáveis o objetivo da pesquisa e obter seu consentimento. O termo foi entregue aos alunos que o levaram para casa em duas vias, que foram assinadas.

As oficinas transcorreram no período da manhã, conforme planejado e especificado no Quadro 9, com duração prevista de três horas cada uma, das 8h às 11h.

Quadro 9 - Datas e horários das oficinas.

<b>Oficinas</b>	<b>Data</b>
1ª oficina	23/02/2015
2ª oficina	24/02/2015
3ª oficina	25/02/2015
4ª oficina	27/02/2015
5ª oficina	02/03/2015
6ª oficina	03/03/2015
7ª oficina	04/03/2015
8ª oficina	06/03/2015

Fonte: Autor (2015).

Com finalidade única de traçar um panorama do grupo selecionado, sem a preocupação com o rigor de um método de análise, elaborou-se um questionário semiestruturado (Apêndice B), que contemplava algumas questões abertas a fim de evidenciar variáveis não presentes nas oficinas que posteriormente pudessem ser confrontadas, tais como: desempenho em matemática, acesso à tecnologia no âmbito familiar, familiaridade com a língua inglesa, contato com linguagem de programação entre outros.

O questionário foi aplicado sem a presença do pesquisador, com a colaboração do monitor de informática da unidade escolar, e foi respondido de forma individualizada e sem comunicação com os demais estudantes.

Os dados foram tabulados, conforme **Apêndice C** e foi possível levantar algumas questões inicialmente importantes, como indicadas a seguir:

**I-** O grupo apresenta-se homogêneo em relação à idade/série. Apenas o aluno **VH** apresenta um atraso de um ano para a idade requerida na série.

**II-** Somente os alunos **CA** e **TS** participam de uma constituição familiar tradicional convivendo com pai e mãe.

**III-** Todos os integrantes informaram ter computador com acesso à Internet em casa, possibilitando assim, acesso doméstico ao computador e as suas ferramentas. Mas, foi possível identificar que o uso do computador em casa é compartilhado com outros familiares, o que pode ocasionar uma concorrência para sua utilização. O tempo médio de utilização do computador pelos participantes é de 3h/dia.

**IV-** Em relação ao aproveitamento em matemática, todos os alunos atribuíram o qualificador BOM para representarem seus conhecimentos em matemática, e a média na disciplina, os mesmos quantificaram em 5. A escola tem como critério mínimo para aprovação em disciplinas a média 5, sendo assim, observamos que o grupo se encontra no limite entre o básico e abaixo do básico. A aluna **NB** não quantificou sua média na disciplina.

**V-** Na indicação do gosto por matemática, todos apresentaram uma resposta positiva. Mesmo o grupo apresentando um rendimento baixo na disciplina, os alunos não indicam aversão à matemática.

**VI-** Nenhum integrante do grupo teve contato anterior com alguma linguagem de programação (LP).

**VII-** Em relação ao uso do LIE da Unidade Escolar indicaram que os professores de matemática nunca os levaram para realizar atividades no laboratório, e os docentes de outras áreas quando os levam, foram unânimes em relatar que a atividade principal é a de “Escrever textos”. Observa-se que os docentes da disciplina de matemática não utilizam de metodologias diferenciadas, mediadas por computador, pois a frequência no laboratório é nula. Quanto aos professores de outras áreas, promovem a utilização do computador como recurso de complementação e limitado apenas a processos mecanizados que não favorecem a construção do conhecimento, opondo as ideias de Kenski (2007) em que o autor enfatiza que é preciso organizar novas experiências pedagógicas, quando novas tecnologias possam ser utilizadas em processos colaborativos de aprendizagem, que garantam a participação de todos os envolvidos no processo.

**VIII-** O grupo demonstrou não gostar do estudo da língua estrangeira moderna: Inglês. Nesse questionamento é importante destacar que os alunos contam com a disciplina de Inglês desde o primeiro ano do ensino fundamental na Unidade Escolar. As linguagens de programação possuem sintaxe em língua inglesa, assim, o conhecimento nesse idioma atua

como facilitador para a construção de tais sintaxes, mas está longe de ser pré-requisito para a aprendizagem de programação.

**IX-** Os estudantes têm vínculo sólido com a escola, uma vez que estudam desde o primeiro ano do ensino fundamental na UE, adaptados ao sistema de ensino, trabalho dos docentes e regras da instituição. Somente a aluna **TS** iniciou seus estudos na escola nas séries finais do ensino fundamental.

**X-** Sobre a questão aberta “de que forma gostam de aprender matemática”, percebe-se uma diversidade nas respostas apresentadas. Assim, entende-se que os alunos gostam de aprender matemática de diferentes formas, inclusive apoiados pelo uso do computador (**Q13 - Apêndice C**).

**XI-** Os alunos, de alguma forma, acham importante ter conhecimento em alguma linguagem de programação.

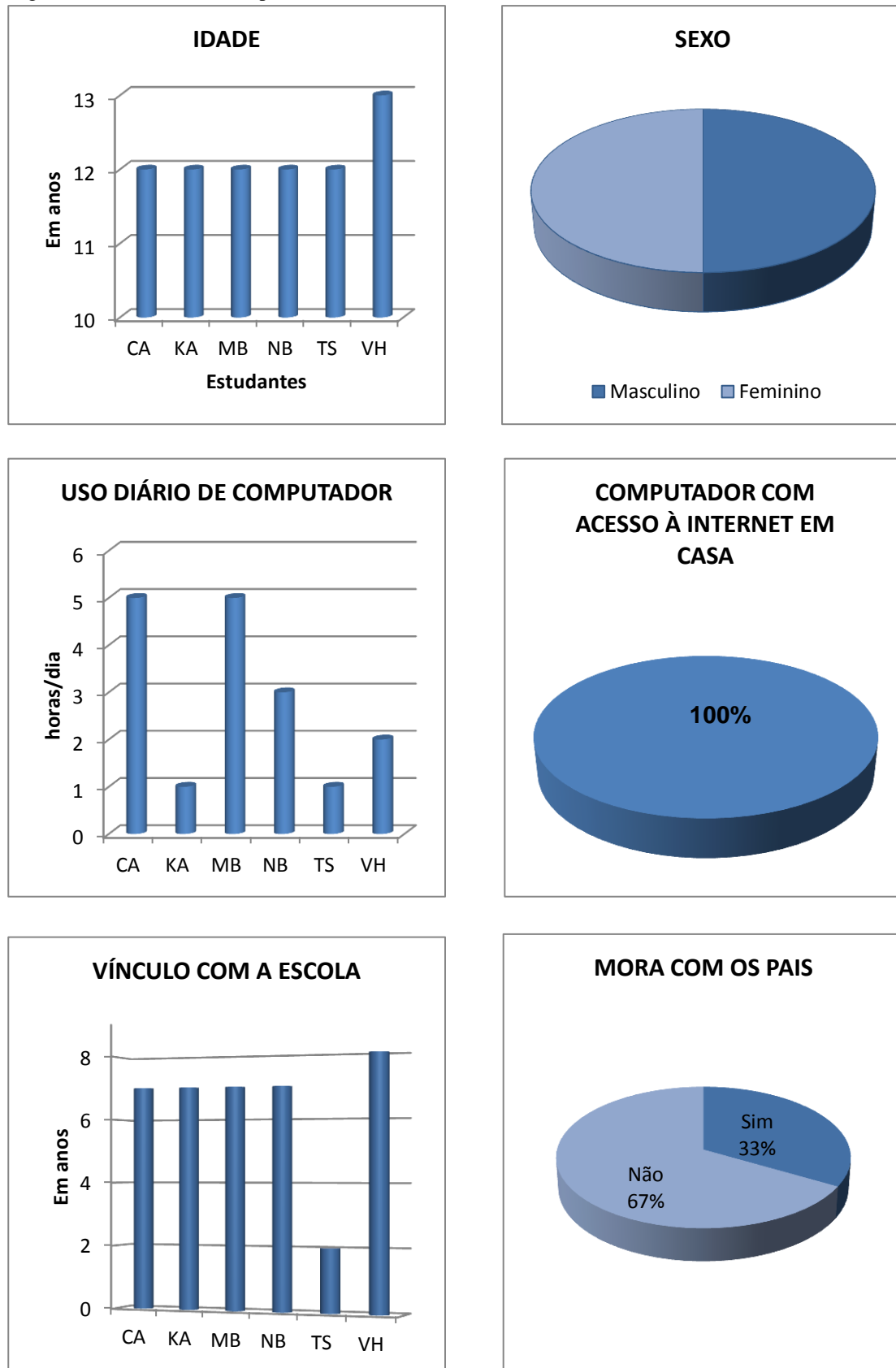
**XII-** Sobre a questão que envolvia o domínio de novas tecnologias pelos docentes, as respostas projetavam para uma perspectiva negativa. Apenas o aluno **MB** disse entender, pois essas tecnologias não fizeram parte da geração de seus professores. Tais afirmações vêm ao encontro da fala Prensky (2008) em que jovens alunos desse século enxergam professores alheios à tecnologia como “analfabetos”. Respostas como: “incapacitado”, “muito burro” não se referem à falta de habilidades do professor com tecnologia, mas externam o inconformismo, onde não se entende o porquê do professor permanecer na zona de conforto, de não tentar o novo, o diferente.

Segundo Almeida (2000a, p. 135) os cursos de licenciaturas destinados à formação de professores deixam a educação tecnológica em segundo plano, e em seus currículos dão ênfase aos conhecimentos pedagógicos e conteúdos, desprezando o conhecimento tecnológico. A autora salienta a necessidade urgente de ser criado na formação inicial dos professores um espaço para a apropriação dos conhecimentos necessários à utilização crítica dos computadores na prática pedagógica.

Nas vozes dos alunos percebe-se que eles estão mergulhados no mundo digital, se interessam, e clamam por um professor que tente o novo, assim como eles, sem medo de errar. Os cursos de formação inicial de professor devem repensar seus currículos, com atenção especial aos conhecimentos tecnológicos, para formarem profissionais que atendam às exigências dessa geração digital.

Para facilitar uma leitura geral do grupo selecionado, gráficos acerca do perfil dos estudantes foram elaborados (Figura 32).

Figura 32 - Gráficos sobre o perfil dos estudantes.



Fonte: Autor (2015).

### 3.3 Procedimentos

Depois de aplicado o questionário com o objetivo de traçar o perfil do grupo selecionado, empregou-se o método de grupo focal para levantamento dos dados como técnica de análise exploratória, na etapa inicial desse estudo.

Segundo Gatti (2012) o grupo focal como técnica exploratória possibilita a realização de um estudo preliminar do principal objetivo do trabalho que será realizado, de modo que a pesquisa subsequente possa ser concebida com maior compreensão, delineamento e refinamento. Em sua obra Gatti (2012), justifica o emprego da técnica de grupo focal para levantamento de dados, pois favorece a dinâmica interacional, procurando entre os participantes, consensos, dissensos, rupturas, descontinuidades e silêncios. Destaca-se que os recortes podem ser pontos de partidas das análises, sendo as sequências de falas importantes, pois dão respaldos às inferências do pesquisador.

No grupo focal, privilegia-se a seleção dos participantes segundo algum critério, desde que eles possuam alguma característica comum que os qualifica para a discussão da questão (GATTI, 2012). A característica comum do grupo selecionado é o baixo desempenho na disciplina de matemática.

Para registro das interações promovidas no grupo focal, optamos pelo registro em áudio e vídeo, juntamente com registros escritos realizados por este pesquisador com objetivo de auxiliar as análises posteriores. Os trabalhos no grupo focal foram conduzidos pelo próprio pesquisador que atuou como moderador.

No processo de obtenção dos dados, as falas dos estudantes foram registradas, os materiais escritos de apoio durante as oficinas foram recolhidos, os programas desenvolvidos pelos alunos foram salvos, e as anotações do pesquisador registravam momentos importantes das oficinas que não puderam ser capturados pela câmera.

Em nenhum momento os alunos foram informados sobre os temas da discussão do grupo focal para que não viessem com ideias pré-formadas e a todo instante foi incentivado um clima aberto às discussões, proporcionando confiança aos estudantes para expressarem suas opiniões com sinceridade.

Após a realização de cada oficina foi realizado um backup das gravações para que em caso de problemas no cartão de memória, os dados pudessem estar disponíveis em outra mídia para recuperação.

Na condução do grupo focal foi respeitado o princípio da não diretividade e o facilitador (moderador) atentou para que o grupo desenvolvesse a comunicação sem

ingerências indevidas do mesmo, realizando encaminhamentos que facilitassem as trocas, procurando sempre manter os objetivos do trabalho em grupo (GATTI, 2012, p. 35).

A realização do grupo focal teve como objetivo o levantamento das relações que os estudantes estabelecem inicialmente com a aprendizagem, o ensino da matemática e suas relações referentes ao uso de tecnologias computacionais.

As transcrições das falas dos estudantes e do moderador foram realizadas pelo próprio pesquisador fiéis ao áudio gerado, contribuindo para maior rigor e qualidade das transcrições para análise.

Após as transcrições, foram selecionados episódios, que são caracterizados como sequências coerentes de sentenças de um discurso e definidas em termos de algum tipo de unidade temática (VAN DIJK, 2002).

Assim, foram destacados alguns episódios com o objetivo de revelar respostas aos questionamentos desta pesquisa, enquanto os alunos aprendiam programação e resolviam os desafios propostos.

Das oito oficinas de Processing 2 realizadas, revelaram-se muitos episódios, dos quais, aqueles que mais chamaram a atenção desse pesquisador, compõem a análise de dados.

Após as etapas de coleta dos dados, foram realizadas as compilações à luz dos referências deste trabalho e sob as categorias de análise, que serão descritas posteriormente.

As transcrições foram fiéis às falas dos estudantes durante os grupos focais e no transcorrer das oficinas. Algumas das falas que não puderam ser compreendidas em áudio foram interpretadas com ajuda das anotações feitas pelo pesquisador durante e após cada oficina.

As etapas da pesquisa (Quadro 10) foram constituídas de oito oficinas de programação utilizando a ferramenta Processing 2, e dois grupos focal, sendo um inicial e outro promovido ao término das oficinas. Este trabalho consistiu em compreender como os estudantes se relacionavam inicialmente com a matemática, suas experiências, motivações, facilidades e dificuldades. Em seguida foi realizada uma ação, por meio de oficinas de programação, utilizando a ferramenta Processing 2. Ao fim, foi possível evidenciar se houve avanços em relação ao que os estudantes compreendiam e percebiam da matemática inicialmente, fornecendo subsídios para responder perguntas como: Quais são as vantagens de aulas de matemática utilizando ferramentas de programação? Aqueles que as utilizam em sala de aula de que forma fazem isso? Modificam o método de ensino? Existe alguma relação entre método e aluno? Quais são as dificuldades ao desenvolver aulas de matemática utilizando ambiente de programação? Que tipos de mudança o uso de ferramentas de programação podem provocar na dinâmica das aulas de matemática?

Quadro 10 - Etapas da pesquisa.

<b>Grupo focal inicial</b>	Como os estudantes enxergavam a matemática inicialmente?
<b>Oficina 1</b>	Introdução ao Processing 2 - características e noções iniciais.
<b>Oficina 2</b>	Formas primitivas e Sistema de coordenadas.
<b>Oficina 3</b>	Variáveis e condicionais.
<b>Oficina 4</b>	Função draw( ).
<b>Oficina 5</b>	Variáveis do sistema.
<b>Oficina 6</b>	Operadores lógicos.
<b>Oficina 7</b>	Movimentação (keyPressed).
<b>Oficina 8</b>	Aplicação dos conceitos apropriados anteriormente.
<b>Grupo focal final</b>	Como os estudantes passaram a enxergar a matemática após as oficinas de Programação utilizando Processing 2.

Fonte: Autor (2015).

No primeiro grupo focal verificaram-se as experiências dos alunos com a matemática, suas motivações<sup>27</sup> e desmotivações em relação à matemática e suas experiências em relação ao uso de novas tecnologias, levantando suas dificuldades e expectativas.

Utiliza-se da análise de conteúdo para identificar características da aprendizagem situada numa comunidade de prática de programadores, a assimilação de conceitos e conteúdos matemáticos, experimentação e descobertas de cada estudante evidenciados em suas falas no decorrer das oficinas, frente aos desafios propostos.

Segundo Bardin (2011) a análise de conteúdo conduz entre dois polos: a precisão da objetividade e a riqueza da subjetividade.

Nesse sentido a autora descreve:

Se um olhar imediato espontâneo já é fecundo, não poderá uma leitura atenta, aumentar a produtividade e a pertinência? Pela descoberta de conteúdos e de estruturas que confirmam o que se procura demonstrar a propósito das mensagens ou pelo esclarecimento de elementos de significações susceptíveis de conduzir a uma descrição de mecanismos de que a priori não detínhamos a compreensão (BADIN, 2011, p 29).

A análise de conteúdo consiste em uma técnica refinada, que busca um texto atrás de outro texto, um texto que não está evidente em uma primeira leitura objetiva e que precisa de uma metodologia para ser revelado e isso exige do pesquisador dedicação e um certo grau de imaginação e criatividade, sobretudo na definição das categorias de análise. A categorização são classes que reúnem um grupo de elementos sob um título genérico, promovendo a organização em razão de características comuns. Assim, o que possibilita o agrupamento é a

<sup>27</sup> Neste trabalho entende-se por motivação a definição apresentada por Heckhausen (1967) como sendo voltado para alcançar um objetivo e composto por todos os fatores que incentivam e direcionam o comportamento de um indivíduo na obtenção desse objetivo.

parte comum entre os elementos e a categorização tem como objetivo fornecer uma representação simplificada dos dados brutos (BARDIN, 2011).

Ao final, após a realização das oficinas, realizou-se outro grupo focal para identificar a apropriação tecnológica, os momentos de identificação do pensamento matemático, a apropriação dos conceitos matemáticos, as relações estabelecidas, bem como o crescimento social e atitudinal.

As oficinas de *Processing 2* foram ministradas por este pesquisador, por exigir conhecimento de lógica de programação e conhecimento da ferramenta empregada na pesquisa. Vale destacar que, o autor possuía conhecimento de programação em outras linguagens, sendo iniciado na ferramenta Processing 2 durante a disciplina de jogos digitais, ministrada neste programa de pós-graduação.

Apesar de a escola contar com professores dispostos a fazer uso de novas tecnologias, não havia no momento nenhum docente da disciplina de matemática que tivesse experiência em programação de computadores, exigindo do pesquisador a promoção das oficinas.

As oficinas de programação foram ministradas numa abordagem instrumentalista, com objetivo de apropriação técnica na ferramenta em foco, visando ao ensino de programação. Assim, à medida que os alunos aprendiam programação foi possível analisar como os estudantes ampliavam seus conhecimentos em matemática, sem dizer-lhes, em nenhum momento, que estavam aprendendo matemática.

As oficinas utilizam como base, o tutorial descrito no texto “*Getting Started with Processing*”<sup>28</sup>, devidamente adaptado para a língua portuguesa que conta com diversos exemplos de programas e comentários. O manual foi projetado para ser usado tanto como um texto de apoio para um curso de programação de nível introdutório ou para autoinstrução. Foi utilizado o modo de programação Java.

### **3.4 Objetivos das oficinas de programação**

As oficinas tinham um caráter geral técnico, na qual visava ao ensino de programação numa linguagem específica, mas à medida que os alunos aprendiam programação buscou-se analisar como os estudantes ampliavam seus conhecimentos em matemática, sem dizer-lhes, em nenhum momento, que estavam aprendendo matemática por meio da aprendizagem situada numa comunidade de prática de programadores.

---

<sup>28</sup> Disponível em: <[http://www.petronet.ir/documents/10180/2323248/Getting\\_Started\\_with\\_Procesing](http://www.petronet.ir/documents/10180/2323248/Getting_Started_with_Procesing)>. Acesso em: 08 de abr. 2015.

A seguir, uma descrição resumida dos objetivos e conteúdos de cada uma das oito oficinas planejadas. É importante ressaltar que em muitos momentos, por conta da dinâmica das oficinas houve flexibilidade em relação ao planejado inicialmente.

### ***3.4.1 Oficina 1***

Etapa inicial, logo após a realização do grupo focal, teve como objetivo apresentar a ferramenta de estudo da pesquisa aos estudantes, suas características, principais funcionalidades e uma exploração inicial do sistema de coordenadas da área de trabalho do Processing 2 (sketch). As primeiras noções em geometria são chamadas primitivas e a oficina procurou explorar três itens que se destacam por serem conhecidos intuitivamente: ponto, reta e plano cartesiano.

### ***3.4.2 Oficina 2***

O objetivo nessa segunda etapa era ampliar a apropriação técnica na ferramenta e consistiu na exploração dos comandos de formas geométricas primitivas em 2D, entre os quais destacam-se: `rect()` `ellipse()`, `line()`, juntamente com a exploração dos comandos `arc()`, `rotate()` e `translate()`, além do comando `fill()` e `noFill()` e os relacionados à temática cor.

### ***3.4.3 Oficina 3***

A terceira oficina abordou o conceito de variável e seus principais tipos (`int` e `float`). Os programas servem essencialmente para manipular dados ao serem executados. Os dados podem se apresentar em duas formas distintas: como constantes ou variáveis. Em programação, uma variável se configura como uma posição localizada na memória do computador capaz de reter e representar um valor ou expressão. Assim, é necessário que os estudantes compreendam seu conceito e emprego. No segundo momento dessa oficina foi apresentada aos alunos a estrutura condicional (decisão simples). As estruturas de decisão são essenciais dentro de um programa. Elas são capazes de alterar o processamento de dados de acordo com circunstâncias diferentes. Assim, apresentamos aos estudantes o comando (SE) que executa a ramificação.

### ***3.4.4 Oficina 4***

A quarta oficina explorou a função `draw()`. Ela é o coração do modo contínuo e indica o que deve se repetir infinitamente. Em muitos programas, a todo instante, o computador verifica o que você está fazendo com o mouse, realiza cálculos e atualizações da tela para

processar todos os gráficos. Este ciclo de cálculo e desenho acontece infinitamente e aproximadamente 30 por segundo para boa animação. Assim, o conceito é fundamental para que os alunos se apropriem do conceito e formalizem competências para irem além de esboços estáticos utilizando Processing 2.

#### **3.4.5 Oficina 5**

A intenção dessa etapa foi explorar algumas das principais variáveis do sistema. O Processing 2 possui um conjunto de variáveis reservadas, disponíveis livremente. As variáveis reservadas armazenam informações como as dimensões da janela de display ou a posição do mouse. Como exemplificação, as variáveis *width()* e *height()* sempre mantém as dimensões da janela determinada pela função *size()*. Assim, a oficina buscou aprofundar o estudo das variáveis possibilitando aos estudantes maior liberdade em suas construções, sempre com o cuidado de nomear suas próprias variáveis evitando nomes de variáveis reservadas.

#### **3.4.6 Oficina 6**

Nessa fase, buscamos demonstrar a importância da utilização de operadores lógicos dentro da construção de um programa. Os operadores lógicos de sentença servem para comparar uma combinação de outras comparações. As expressões lógicas e relacionais complexas são geralmente empregadas nas estruturas de decisão e repetição. Com isso, os estudantes tiveram a oportunidade de ampliar seus laços de decisões dentro de suas construções, implicando programas mais eficientes, com um menor número de códigos.

#### **3.4.7 Oficina 7**

A finalidade dessa etapa foi ampliar a interatividade utilizando as setas de movimentação do teclado por meio do deslocamento de objetos num espaço cartesiano.

Os teclados são normalmente usados para inserir caracteres, mas eles têm potencial para uso além de suas intenções originais. Ao escrever o seu próprio programa, você tem a liberdade de usar os dados de teclado da forma que desejar, assim esta oficina objetivou essa ampliação de utilização do teclado. O Processing 2 registra a tecla mais recentemente pressionada e se a tecla está atualmente pressionada, isso possibilita um leque de possibilidades de exploração tendo como via de entrada o teclado.

### **3.4.8 Oficina 8**

A intenção dessa última etapa foi permitir aos estudantes explorar todos os recursos e conceitos apropriados durante as oficinas, englobados num único desafio que consiste em construir uma plataforma móvel que não deixa uma bolinha cair, após terem internalizados os padrões, ideias, técnicas e práticas na ferramenta empregada neste estudo. Nessa oficina não foi apresentado nenhum comando ou conceito inédito aos alunos, tendo como intencionalidade observar a capacidade dos alunos de transitar para outro desafio, susceptível de uma solução, utilizando-se de conceitos, ferramentas e procedimentos já apropriados para obter resultados.

### **3.4.9 Grupo focal final**

Esse grupo focal ocorreu ao término da última oficina, com duração aproximada de 30 minutos, com o objetivo de verificar como os alunos passaram a enxergar a necessidade da matemática e se houve avanços no tocante ao que eles compreendiam e percebiam da matemática após as oficinas de programação.

## **3.5 Categorias de análise**

Após as etapas de coleta dos dados, foram então realizadas as compilações à luz das categorias de análise evidenciadas de acordo com os pressupostos de Lave e Wenger (1991) e Wenger (1998).

Foi possível delinear seis categorias de análise que podem, conforme os objetivos da presente pesquisa, sinalizar arranjos da aprendizagem situada da matemática numa comunidade de prática de programadores. Sendo elas:

### ***I – Não linearidade***

Um ambiente de aprendizagem situado fornece aos estudantes a oportunidade de investigar várias perspectivas (Wenger, 2008). Muitas metodologias são projetadas num formato linear, que pressupõe que o aluno tenha um início único e avança por meio de um programa para a conclusão única. Possibilitar aos alunos múltiplos papéis e oportunidades para explorar o programa a partir de várias perspectivas implica ricas situações de aprendizagem. Todas as estruturas não apresentam um único sentido, apresentam múltiplos caminhos e destinos, desencadeando em múltiplos finais.

### ***II – Ênfase no “Saber Fazer”***

De acordo com Wenger (1998) esta maneira de abordar a aprendizagem é algo mais do que simplesmente "aprender fazendo" ou aprendizagem baseada na experiência e o conceito de contextualização envolve pessoas que sejam participantes plenos no mundo e na geração de significado. Aprender a fazer e aprender a conhecer são processos indissociáveis. Em uma sociedade altamente tecnológica, é necessário ampliar o desenvolvimento das competências de gestão dos alunos, para torná-los capazes de gerenciar projetos, otimizar recursos e trabalhar em equipe. Assim, a aprendizagem situada enfatiza o saber fazer e se relaciona com o processo de participação ativa e reificação. A reificação consiste em transformar uma ideia em uma coisa, sendo uma operação mental que consiste em transformar conceitos abstratos em realidades concretas ou objetos.

### ***III – Informal***

A despreocupação com um modelo ou currículo pré-definido promove um relacionamento mais favorável à aprendizagem do que em relações de hierarquia explícita. O ensino informal está relacionado ao processo de socialização do homem. Na aprendizagem situada, a atenção está voltada para o processo da prática social e a estrutura pedagógica é vista como uma fonte secundária (LAVE; WENGER, 1991). Assim, o conhecimento do aluno é definido pelo processo social de interagir e fazer. Ele permite aos alunos se apropriarem do conhecimento naturalmente como resultado de um comportamento social.

### ***IV – Construção do conhecimento parte da necessidade***

Um ambiente de aprendizagem situado fornece um contexto autêntico que reflete a forma como o conhecimento vai ser usado na vida real, a partir de uma necessidade, que

preserva o contexto completo da situação sem fragmentação e decomposição, que convida à exploração e permite a complexidade natural do mundo real (WENGER, 2008).

### ***V – Interatividade***

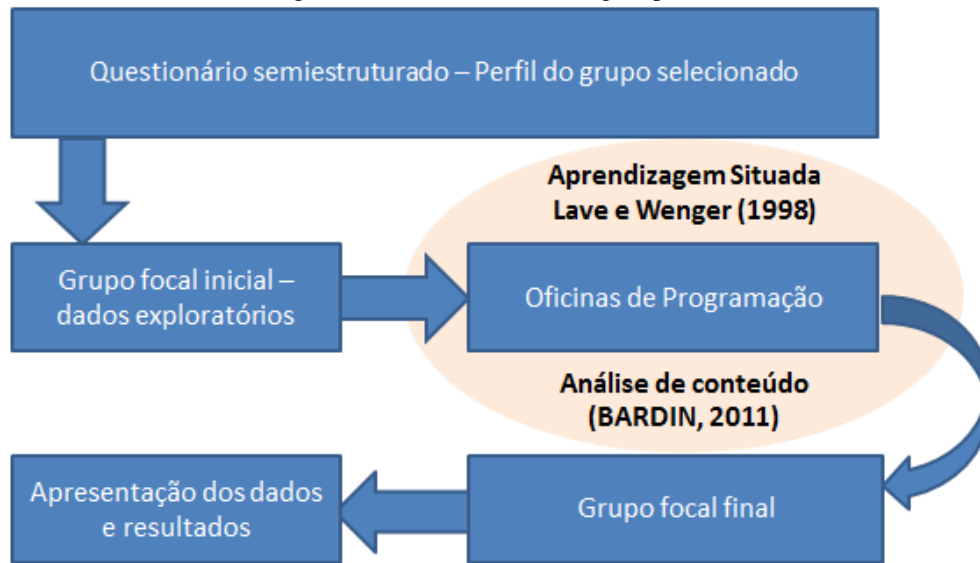
A interação entre máquina e usuário se constitui em mais uma possibilidade de fazer da aprendizagem situada melhor sucedida, bem como proporcionar aos alunos uma oportunidade de se ter outro meio para aprendizagem. A interatividade é um princípio básico de sistemas computacionais por meio do retorno ao usuário sobre as ações que foram executadas e seus resultados obtidos. Sem um *feedback*, o usuário sente-se perdido, sem a certeza de que sua solicitação foi executada com sucesso ou se ocorreu algum erro. Muitos sistemas mudaram o paradigma de interação humano-computador e se baseiam na linguagem visual. Em vez de um meio computacional abstrato, toda programação é feita graficamente. De acordo com Gee (2008), pesquisadores de diversas áreas têm levantado a possibilidade da aprendizagem por meio de tecnologias digitais interativas por facilitar a aprendizagem situada no contexto da atividade e experiência.

### ***VI – Engajamento***

O engajamento consiste nos processos internos em integrar o aspecto subjetivo da comunidade, assimilar sua língua, conhecer suas histórias e proceder como membros da comunidade (WENGER, 1997). Para o aluno, ser membro periférico é altamente produtivo, uma vez que o aproxima dos especialistas (interação social envolvendo novatos e especialistas) e ao observar suas práticas, atentar a suas elucidações será formalizado um envolvimento efetivo (WENGER, 1998). Esta maneira de pertencer é evidenciada pela participação ativa nos processos mútuos de negociação de significado, incluem: formação de trajetórias, histórias compartilhadas de aprendizagem, relações, e práticas que tornam o conhecimento tácito em explícito.

Para sintetizar o caminho metodológico percorrido neste trabalho foi construído um diagrama que fornece uma visão geral (Figura 33).

Figura 33 - Caminho metodológico percorrido



Autor (2015).

O percurso metodológico relaciona-se ao lugar a que se quer chegar. Assim, o questionário contribuiu para traçar o perfil inicial do grupo. Em seguida, o grupo focal inicial, na obtenção de dados exploratórios, respaldou a compreensão de como os estudantes se relacionavam inicialmente com a matemática, suas experiências, motivações, facilidades e dificuldades. Por meio das oficinas de programação foi possível evidenciar as relações que os alunos formalizaram, os conteúdos matemáticos assimilados, enquanto membros periféricos legítimos em uma comunidade de prática de programadores. Ao fim, no grupo focal final foi possível evidenciar avanços em relação àquilo que os estudantes compreendiam e percebiam da matemática inicialmente.

## **4 - APRESENTAÇÃO DOS DADOS E DISCUSSÃO**

Neste capítulo são apresentados os dados obtidos de acordo com o percurso metodológico realizado e sua discussão.

### **4.1 Grupo focal inicial**

Na data e horário marcados compareceram todos os alunos selecionados e os mesmos foram encaminhados ao laboratório de informática. Num primeiro momento, antes de iniciá-los na ferramenta Processing 2, os alunos foram convidados a se sentarem em círculo para realização do grupo focal.

Primeiramente explicou-se aos alunos a necessidade das atividades serem registradas em áudio/vídeo, para fins de análise posterior do pesquisador. No início alguns alunos ficaram incomodados com a câmera. Gatti (2012, p. 29) aponta que a forma de registro do trabalho deve ser explicada e é necessário obter a anuência dos participantes. Assim, depois de explicado aos estudantes a importância do registro pela câmera, com a garantia do sigilo dos mesmos, criaram-se condições favoráveis à participação de todos os integrantes, já que passado o momento inicial, a presença da câmera passou despercebida.

Foram elaboradas anteriormente algumas questões norteadoras da entrevista, que foi promovida na forma de bate papo, sem ordem estabelecida de respostas, deixando os alunos livres para expressarem o que pensavam, atentando sempre para que todos pudessem ter a oportunidade de responder ao menos uma vez cada questão e enfatizando que todas as ideias e opiniões eram interessantes e que não havia certo ou errado.

A primeira questão em foco foi sobre o gosto pela matemática e quais as facilidades e/ou dificuldades em relação a esta área do conhecimento. O quadro 11 apresenta as interações na ordem proferida.

Quadro 11 - O gosto pela matemática e suas facilidades/dificuldades

<b>CA</b>	<i>“Não sei se gosto da matemática, mas eu tenho dificuldades com operações com frações, eu não entendi o que foi ensinado no final do ano passado, como chama aquilo mesmo?”.</i>
<b>MB</b>	<i>“Eu gosto de matemática, eu tenho facilidade com contas normais, mas eu tive muita dificuldade em compreender as conversões de Bytes, fui muito mal na prova”.</i>
<b>NB</b>	<i>“Eu gosto, mas eu tenho dificuldades em muitas coisas”.</i>
<b>VH</b>	<i>“Eu gosto de matemática, a aula de matemática é legal, mas não gosto de conta complicada, prefiro as fáceis”.</i>
<b>TS</b>	<i>“Eu gosto de matemática e tenho facilidade quando eu consigo ver. A professora levou para aula o icosaedro, daí eu entendi o que é truncar, para formar a bola”.</i>
<b>KA</b>	<i>“Gosto da matemática quando a professora explica e eu entendo. Quando eu não entendo eu acho chato”.</i>

Fonte: Primária (2015).

Diante do primeiro contato percebe-se que a aluna **NB** se demonstrou bastante tímida em expor sua opinião se comparado aos demais integrantes. A maioria dos estudantes expressou positivamente gostar da matemática. O aluno **CA** não soube informar se gosta ou não da matemática, mas apresenta imediatamente suas dificuldades em relação a um conteúdo do qual não se recorda e pede ajuda aos amigos do grupo. Com a ajuda do grupo, descobre-se que o aluno disse ter dificuldade no conceito de Mínimo Múltiplo Comum (MMC).

Em relação às facilidades do aluno **MB** ao dizer “contas normais” entende-se que se referem as quatro operações elementares (adição, subtração, multiplicação e divisão). O mesmo quando diz “conversões de bytes” comete um erro conceitual, ele refere-se às conversões do sistema binário para o sistema decimal, um indicativo que o aluno teve dificuldades em operações que utilizam potenciação de base 2.

O aluno **VH** disse preferir contas fáceis, assim, relaciona seu gosto pela matemática com o grau de complexidade das operações, indicando que se o problema exigir cálculos mais elaborados e/ou envolver domínio de variadas habilidades tende a não gostar.

Observa-se na resposta da aluna **TS**, a ênfase dada no concreto. Ela relaciona seu gosto pela matemática quando ela consegue “ver”. Essa condição está relacionada à contextualização das expectativas de aprendizagens e a dinâmica entre teoria e prática. O relato da aluna refere-se a um icosaedro regular constituído por 20 triângulos equiláteros, em que este sólido pode ser truncado (remoção de partes de um sólido colocadas simetricamente no sólido) formando um icosaedro truncado que se constitui de 12 faces pentagonais regulares e 20 hexagonais regulares. A aluna faz referência à bola de futebol que costuma ser feita a partir deste sólido.

Nota-se que os estudantes, em relação às facilidades e dificuldades apresentadas referem-se a assuntos recentes, a maior parte deles abordados na série anterior.

Após a primeira temática, foi conduzido o tema “Motivação para estudar matemática”. Manifestaram-se na ordem descrita os seguintes alunos (Quadro 12).

Quadro 12 - Motivação/desmotivação em estudar matemática

<b>CA</b>	<i>“A motivação que tenho é o material anglo que é bom, a desmotivação é os alunos que falam demais e atrapalham a concentração e perde muito tempo. Quando eu fico desmotivado eu não presto atenção”.</i>
<b>KA</b>	<i>“O que me motiva a estudar matemática é pensar num futuro melhor, e quando eu for trabalhar com administração ou alguma coisa assim, que usa matemática, tá sabendo mais”.</i>
<b>TS</b>	<i>“Eu me sinto desmotivada quando preciso fazer aquele monte de exercícios da tarefa de casa, dá muita preguiça”.</i>
<b>MB</b>	<i>“Sabe, eu fico desmotivado em aprender matemática toda vez que tenho que aprender um assunto que não gosto. A professora nunca diz pra que serve aquilo. Agora o que mais me motiva é quando eu tiro nota boa na prova”.</i>
<b>VH</b>	<i>“Eu fico sempre desmotivado quando não consigo entender o que a professora explica, matemática é legal, mas às vezes é difícil. Quando é facinho e eu consigo entender, daí eu gosto”.</i>

Fonte: Primária (2015).

A estudante **NB** não se manifestou nessa temática, e quando solicitada a sua opinião a mesma fez um sinal gestual negativo e imediatamente questionou “quantas perguntas são?”. Tal atitude se configura como um possível indicativo de que a estudante não estava sentindo-se confortável em participar da discussão. Então, nesse momento o pesquisador seguindo as recomendações da Gatti (2010) para continuidade de um clima aberto às discussões enfatizou que o espaço era um momento de expressar suas opiniões e encaminhar pelos ângulos que quisessem; “aqui não tem certo ou errado”, tranquilizando os participantes e levando-os a sentir confiança para se expressarem.

Da temática pode-se evidenciar (Quadro 13) os seguintes elementos encontrados nas falas dos estudantes.

Quadro 13 - Comparação Motivação/desmotivação em estudar matemática

<b>Motivação</b>	<b>Desmotivação</b>
Bons instrumentos de apoio à aprendizagem.	Indisciplina nas aulas.
Projeção de um futuro melhor conquistado por meio da educação.	Grande quantidade de exercícios nas tarefas de casa para fixação do conceito.
Aprender aquilo que faz sentido, do interesse do aluno.	Não obter respostas convincentes do professor sobre a aplicação do conceito matemático.
Boas notas.	Dificuldade de compreensão da explicação do professor.
Compreensão da explicação do professor.	

Fonte: Primária (2015).

Percebe-se na opinião do estudante **CA** a valorização de bons instrumentos de apoio à aprendizagem, em que considera o material apostilado como sendo de qualidade e fator de

motivação. Assim, os alunos levam em consideração, como fator de motivação em suas experiências de aprendizagem a utilização de bons instrumentos e ferramentas.

Outro elemento é aprender aquilo que faz sentido para o aprendiz e de acordo com o interesse do aluno. Para Papert (2007) a melhor aprendizagem se faz quando o aprendiz assume o comando do seu próprio desenvolvimento, por meio de atividades que sejam significativas e prazerosas.

Boas notas foram outro elemento considerado como fator de motivação na temática. Assim, a importância de uma avaliação que considera o processo que cada aluno perfaz, sua evolução pessoal e coletiva se configura altamente importante. Com isso a avaliação assume uma função formativa e integradora, como instrumento a serviço da aprendizagem, que ajuda o aluno a aprender, permite-lhe perceber seus próprios avanços e dificuldades, possibilitando ao professor rever seus procedimentos e (re)planejar o trabalho. Ela acompanha permanentemente o processo, de forma abrangente, diversificada e condizente com os objetivos.

Como fator desmotivador é citada a grande quantidade de exercícios na tarefa de casa. Percebemos na fala da aluna **TS** que o excesso de exercícios propostos para casa é motivo de desmotivação. O ensino apostilado impõe ritmos de aprendizagem e propõe tarefas de casa ao final de cada módulo, muitas vezes extensas, o que representa um indicativo de desmotivação.

Sobre a dificuldade de compreensão da explicação do professor, enfatiza-se que o professor deve apresentar um pluralismo metodológico considerando o pressuposto que os alunos aprendem de diferentes formas e em ritmos diferentes. Assim, todo aluno pode aprender desde que lhe sejam oferecidas condições adequadas; isso implica considerar que não existe aluno incapaz e que variadas condições e caminhos devem ser disponibilizados para que o aluno aprenda. Segundo Almeida (2010) as crianças têm diferentes habilidades e, por isso, diversas maneiras e ritmos para aprender. Em um curto período de tempo, eles podem superar um problema familiar ou adquirir mais segurança com a ajuda de um professor e esses fatores repercutem profundamente na sua capacidade de aprender.

Na fala do aluno **MB** há indícios de problemas na formação do professor, quando ele diz: “a professora nunca mostra pra que serve aquilo”. Isso mostra um despreparo na formação do professor, configurando num indicativo para a melhoria da formação continuada do professor.

A escola é um espaço onde todos os estudantes devem ter as mesmas oportunidades de aprendizagens asseguradas. Para isso, as estratégias empregadas devem ser as mais diversificadas. Segundo Soligo (2003) o aprendiz mais rápido não é melhor do que o mais

lento e “todos os aprendizes estão sempre evoluindo, mas em diferentes ritmos e por caminhos singulares e únicos”.

Logo, foi conduzido para a próxima temática e solicitado ao grupo que socializasse as suas experiências de aprender e ensinar matemática (Quadro 14).

Quadro 14 - Experiências dos alunos em aprender e ensinar matemática.

<b>TS</b>	<i>“Eu não gosto de explicar matemática. Meu sobrinho tem 7 anos, e algumas vezes eu ajudo ele na tarefa de casa”.</i>
<b>KA</b>	<i>“Eu não tenho muita experiência em ensinar matemática. Quando um amigo pede pra eu explicar eu peço pra ele pedir explicação para o professor porque ele explica melhor. Explicar eu não gosto”.</i>
<b>NB</b>	<i>“Eu tenho medo de explicar coisa errada, porque eu não sei se está certo”.</i>
<b>MB</b>	<i>“Muitas vezes eu até sei fazer (exercício), mas eu não sei como explicar”.</i>
<b>VH</b>	<i>“Eu não ensino porque eu não gosto”.</i>
<b>CA</b>	<i>“Quando a professora pede pra sentar em dupla pra resolver os exercícios e eu sei fazer, pois tenho confiança, eu explico”.</i>

Fonte: Primária (2015).

Verifica-se nas discussões sobre as experiências de ensinar e aprender matemática uma maior ênfase nas experiências de ensinar matemática e muito pouco com as experiências relacionadas com a aprendizagem matemática.

Nota-se que os alunos possuem pouca experiência em ensinar matemática, e dizem não gostar de ensinar. Evidenciamos a presença do medo, na fala da aluna **NB** o que ocasiona falta de segurança em formalizar uma explicação. Tal postura leva-nos a observar a fala do aluno **MB** que muitas vezes o aluno sabe fazer a atividade proposta, mas ele não consegue sistematizar uma explicação, talvez porque ainda não domina plenamente as habilidades requeridas na resolução do problema. O aluno **CA** quando se sente confiante frente ao problema não hesita em fazer a experiência de explicar.

Prosseguindo, foi lançada uma situação na qual fosse dado poder aos alunos de retirar apenas uma disciplina da grade curricular, qual eles retirariam. As repostas imediatas foram tabuladas (Quadro 15) e apresentadas na ordem a seguir.

Quadro 15 - Disciplina retirada da grade curricular pelos alunos.

<b>CA</b>	<i>“Português, porque tem muito texto, e quando eu tenho que ler bastante eu fico entediado”.</i>
<b>MB</b>	<i>“Geografia, não por causa da professora, mas é que eu não me identifico com a matéria”.</i>
<b>NB</b>	<i>“Inglês é difícil. Eu tenho muita dificuldade na pronuncia e no dia a dia a gente não usa inglês”.</i>
<b>KA</b>	<i>“Inglês, porque a língua inglesa se usa pra fora do Brasil e eu não pretendo sair pra fora do Brasil”.</i>
<b>TS</b>	<i>“Eu retiraria a disciplina de Inglês porque temos que aprender bem o Português que é a nossa língua”.</i>
<b>VH</b>	<i>“Eu não retiraria nenhuma, porque acho que todas são importantes. Mas se eu tivesse poder eu retiraria o Inglês, a apostila é muito difícil”.</i>

Fonte: Primária (2015).

Nesta fase, constata-se uma interação maior entre os participantes. Quando o aluno **MB** disse que retiraria Geografia do currículo, o aluno **CA** imediatamente disse: “mas se você quiser ser um arqueólogo precisará desses conhecimentos”.

Quando o aluno **CA** disse que retiraria a disciplina de Português a aluna **TS** indagou “vai retirar a mais importante?”.

É notável no grupo a aversão pela disciplina de Inglês, em que a grande maioria não consegue visualizar a importância do estudo e domínio de uma língua estrangeira moderna, como o Inglês em sua formação integral.

Os estudantes têm aulas de inglês desde o primeiro ano na escola, e tal aversão se configura como um indicativo de que algo está errado no tocante ao ensino e aprendizagem da língua inglesa. Pela análise do discurso podemos entender que os alunos apresentam uma visão limitada.

Na busca de entender melhor o contexto, procura-se após a transcrição das falas dos alunos, a professora de Inglês **WS** que atua há 10 anos na unidade escolar trabalhando com as séries iniciais do ensino fundamental, para emitir sua contribuição no tocante à compreensão sobre as opiniões dos alunos. A professora **WS** relata que os alunos que possuem mais dificuldades na alfabetização em língua portuguesa, têm grandes dificuldades na língua inglesa. A apostila do 1º ao 5º ano trabalha de uma forma bastante lúdica, porém a professora destaca que o material passa a ter um nível de complexidade maior a partir do 4º ano quando se inicia o trabalho com leitura de textos e com a pronúncia. Assim, neste momento o aluno que não está alfabetizado em Língua Portuguesa passa a ter dificuldades na aprendizagem da língua e constrói essa aversão à língua inglesa.

As professoras de inglês das séries finais do Ensino Fundamental **EO** e **DB** também contribuíram para a análise. Em suas colocações apontam que a apostila é vista como

complexa por conter textos e vocabulários sofisticados. A metodologia da apostila de inglês se reduz a texto, interpretação e gramática, salientando que falta algo diferenciado para atrair os alunos. Ainda, destacam que não há tempo para se trabalhar atividades complementares como, por exemplo, música, por conta do sistema apostilado que impõe um ritmo de trabalho, e se não for mantido, ou dispensado com atividades complementares, não conseguem contemplar os conteúdos programados no material. Assim, fica evidenciada na temática a aversão dos alunos a língua inglesa.

Em nenhum momento os estudantes citaram retirar a disciplina de matemática da composição da grade, confirmando assim o gosto e predileção pela matemática.

Logo foi apresentada a próxima temática que permitia aos alunos discutir sobre a importância da matemática em suas vidas (Quadro 16).

Quadro 16 - Importância da matemática para a vida dos estudantes.

<b>KA</b>	<i>“Eu considero matemática uma matéria muito importante por causa que futuramente no trabalho vou precisar fazer uso. Matemática e língua portuguesa é muito importante”.</i>
<b>TS</b>	<i>“Eu acho importante porque quando formos comprar algo, não sermos enganados no troco”.</i>
<b>NB</b>	<i>“Eu também acho importante porque... (risos).”</i>
<b>MB</b>	<i>“A matemática é importante na minha vida, eu pretendo fazer faculdade que mexe com cálculo, então vou precisar dela. Mas, a gente precisa dela em várias outras situações, até no futebol tem matemática, calcula a velocidade da bola”.</i>
<b>VH</b>	<i>“Eu consigo ver a importância dela (matemática) na minha vida. Quando a professora, ano passado levou a gente para conhecer a matemática do sítio, eu consegui conversar com a minha avó lá no sítio sobre alqueires, ela ficou admirada porque eu sabia”.</i>
<b>CA</b>	<i>“Matemática é importante porque se você for, tipo assim, um banqueiro, você vai ser expulso do trabalho, porque banqueiro é uma das coisas que mais usa matemática”.</i>

Fonte: Primária (2015).

Nessa discussão, cada estudante de uma forma particular apresenta sua visão positiva em relação à importância da matemática em suas vidas. Alguns estudantes projetam a importância da matemática para seu uso futuro, preocupados com o trabalho e a formação acadêmica.

Nesse ponto ainda, percebe-se que a aluna **NB** muito tímida, com baixa intensidade tonal, apresenta dificuldades de concluir sua posição frente ao tema proposto.

Chama atenção os relatos dos alunos **MB** e **VH** nos quais se consegue visualizar a importância da matemática no dia a dia. Os alunos citam a matemática presente no futebol e a matemática presente no sítio. O interessante é que o aluno **VH** consegue discutir com sua avó medidas agrárias como alqueire paulista. Isso implica o domínio de uma matemática que geralmente está longe das realidades urbanas. Percebe-se que o aluno consegue diferenciar

outros tipos de alqueires. Quando questionado se sabia quanto valia um alqueire, o mesmo rapidamente “depende se é alqueire paulista, mineiro ou baiano. O paulista vale 24.200 metros quadrados”.

Por fim, encaminha-se para a última temática prevista como norteadora, na qual se desejava descobrir as motivações dos alunos em relação ao uso das tecnologias e informática (Quadro 17).

Quadro 17 - Motivação dos alunos em relação ao uso de tecnologias.

<b>KA</b>	<i>“Eu gosto muito da tecnologia e da informática. Eu gosto de mexer muito no meu celular e eu acho que minha maior motivação é ficar mais atualizada”.</i>
<b>TS</b>	<i>“Eu gosto de usar, principalmente ficar na internet, eu não me canso de mexer, minha mãe precisa pedir pra eu desligar o computador”.</i>
<b>NB</b>	<i>“Eu uso a internet para fazer pesquisa da escola. Eu gosto de ficar na internet e eu entro no facebook e tem muita coisa legal”.</i>
<b>MB</b>	<i>“Eu pretendo fazer uma faculdade que vai mexer com esse ramo (tecnologia). Eu não me canso de mexer, na minha casa é o lugar que eu mais gosto de ficar”.</i>
<b>VH</b>	<i>“Tudo de tecnologia motiva mais porque a gente não vê o tempo passar quando tá usando. Eu mesmo fico baixando muita coisa, ontem eu baixei no meu Andróide o Perguntados”.</i>
<b>CA</b>	<i>“Eu gosto da tecnologia, principalmente dos jogos, eu fico lá jogando, ouvindo músicas e vendo vídeo no youtube”.</i>

Fonte: Primária (2015).

Nesta temática evidencia-se a motivação dos alunos em relação à tecnologia, em que cada estudante demonstra uma motivação particular para fazer uso. Os estudantes falam com mais facilidade sobre a temática, sinalizando uma motivação intrínseca, externada nas expressões “coisa legal”, “baixei no meu Andróide”, “não me canso de mexer”.

O estudante **VH** demonstra ter conhecimentos sobre o tipo do sistema operacional do seu smartphone, e aponta que instalou o aplicativo “Perguntados”. Tal aplicativo é um jogo de perguntas e respostas sobre conhecimentos gerais e aborda diversas áreas, como ciências, esportes, entre outras.

O último aluno a manifestar sua opinião nesse eixo foi o aluno **CA** o qual manifesta sua motivação intrínseca por games virtuais.

Identifica-se que a resposta à pergunta, como os alunos gastam seu tempo, está relacionada a tecnologias e Internet, entre elas games digitais e vídeos online. Assim, traça-se o perfil de um grupo que pertence a uma geração com muita habilidade com tecnologia, mas mesmo assim tem notas baixas em matemática.

Nesse momento, o pesquisador (P) lançou uma nova questão não prevista inicialmente, para fomentar a discussão sobre o posicionamento do aluno.

**P:** “Interessante, o **CA** disse que tem motivação em fazer uso de tecnologia, principalmente com jogos. Eu gostaria de ouvir o que vocês aprendem enquanto jogam”. Alguns alunos manifestaram, veja no quadro 18.

Quadro 18 - O que aprendem enquanto jogam.

<b>CA</b>	<i>“Eu aprendo a melhorar a técnica do jogo e muito sobre conhecimentos gerais”.</i>
<b>MB</b>	<i>“Dependendo do jogo você aprende um pouco de inglês”.</i>
<b>CA</b>	<i>“No Minecraft eu aprendo mais matemática nas montagens dos blocos. Você não conhece esse jogo?”.</i>
<b>TS</b>	<i>“Aqui no computador tem também um monte de jogo que a gente aprende um monte de coisas de matemática. No Tuxmath que a gente joga em rede é bem legal”.</i>

Fonte: Primária (2015).

Apenas quatro alunos se manifestaram e em seus discursos apresentam a credibilidade de que o game os leva a aprender mais. Ainda podemos evidenciar a empolgação dos alunos na discussão.

Considerando o aprimoramento das técnicas do jogo, do inglês, chama atenção a fala dos alunos **CA** e **TS** que dizem que aprendem mais matemática jogando Minecraft e Tuxmath. Tais afirmações reforçam a visão dos alunos de que por meio dos jogos são levados as mais diversas experiências de aprendizagens, conforme apontadas por Gee (2008). No mundo do Minecraft tudo são cubos e no Tuxmath os jogadores tentam proteger suas cidades dos cometas realizando operações aritméticas.

A estudante **TS** utiliza a expressão jogar em rede. Isso representa domínio por parte da aluna em relação ao modo multijogador e ao conceito implícito de rede de computadores. Assim, deduz-se que os estudantes possuem ricas experiências com jogos, e domínio de alguns termos técnicos da área de informática.

E por fim, retorna-se ao foco inicial perguntando qual o diferencial em uma aula de matemática para que eles fiquem mais motivados (Quadro 19), focados na aprendizagem da matemática.

Quadro 19 - Diferencial em uma aula de matemática para aumentar a motivação.

<b>CA</b>	<i>“Uma mesada...(risos). Mas, a escola é chata e as aulas no geral não tem nada relacionado com que a gente faz na vida lá fora”.</i>
<b>MB</b>	<i>“O professor de matemática poderia dar aulas mais interessantes trazendo a tecnologia pra sala, tipo um telão de computador no lugar da lousa”.</i>
<b>KA</b>	<i>“Poderia ter acesso à Internet dentro da sala e nas aulas de matemática a gente poder usar, como complementação da matéria depois que a gente terminasse o bloco da apostila”.</i>
<b>MB</b>	<i>“Eu acho interessante usar a internet nas aulas, mas se o professor deixar usar internet nas aulas, o aluno não vai prestar atenção no professor, vai ficar no facebook e no whatsapp. Por isso que é proibido celular aqui na escola, mas o ano passado eu descobri a senha do wi-fi (risos)”.</i>
<b>VH</b>	<i>“A aula ficaria mais legal se o professor não ficasse só na apostila e fizesse igual à professora de ciências que leva a gente para o laboratório para fazer os experimentos”.</i>

Fonte: Primária (2015).

Quando lançado o questionamento, o aluno **CA** prontamente ironizou e disse que se fosse pago uma mesada ele ficaria mais motivado e logo abriu o sorriso, descontraindo o grupo. Apesar da descontração promovida, isso indica um tipo de motivação extrínseca, quando há algum tipo de recompensa em suas atividades. Mas o ponto interessante de seu argumento é quando ele diz que a escola é chata e não tem nada de relacionado com a vida fora dos muros da escola. Para Willingthan (2011), o aprendizado tem que ser uma experiência envolvente e significativa, na qual os alunos realmente compreendam as perguntas para poder respondê-las. Assim, a importância do que é ensinado deve ser muito bem explicada aos alunos.

Acredita-se que ao falar que a escola “é chata” o aluno clama para que se inverta o tradicional e a escola inove suas práticas. Levanta-se o questionamento de como melhorar a escola, formando alunos bem sucedidos e preparados para trabalhar e crescer no mundo depois da escola.

Nas últimas falas se evidencia a preocupação do aluno **MB** em permitir o uso da internet na sala de aula. Acreditamos que em uma instituição chamada escola, onde a maior parte da informação se encontra fora dela na grande rede, proibir o acesso à internet, implica limitar a aprendizagem e reduzir seus horizontes. A escola precisa encontrar meios de promover acesso a todos os seus estudantes e criar mecanismos que incentivem sua utilização nas aulas.

Na fala do estudante **VH** fica evidenciada a preocupação com a diversificação nas formas de aprendizagem, demonstrando o descontentamento do aluno em realizar atividades somente na apostila e a comparação com a professora de ciências que promove aulas mais dinâmicas e que despertam o interesse dos alunos.

Nota-se que os estudantes relacionam o diferencial nas aulas de matemática com o uso de tecnologias para aguçar suas motivações.

Assim, transcorridos aproximadamente 50 minutos desde o início do bate papo, os estudantes foram convidados a assumir de forma aleatória um computador no laboratório para o início das oficinas de Processing 2.

## 4.2 - Oficinas

### 4.2.1 Oficina 1 – Introdução ao Processing 2

A oficina iniciou-se apresentando a ferramenta e o que configura uma linguagem de programação visual. O conceito de programação visual é bastante antigo e baseia-se no uso de interfaces gráficas para interação com o usuário.

A primeira característica da linguagem apresentada ao grupo é que a mesma é *case sensitive*. O Processing 2 distingue letras maiúsculas e minúsculas no reconhecimento de palavras-chave e nomes de variáveis, ou seja, para o programa, “XYZ” não é o mesmo que “xyz”.

Quando o pesquisador, disse que por trás daquela bonita área de trabalho que eles estavam visualizando havia muitas linhas de código e surgiram os seguintes questionamentos, (Quadro 20) vejamos o episódio:

Quadro 20 - Episódio sobre números binários.

<b>MB</b>	<i>Não seriam números binários?</i>
<b>P</b>	<i>A programação é feita realizada em alto nível, e será necessário um compilador para converter este código em linguagem de máquina.</i>
<b>MB</b>	<i>Então, nunca se programa em 0 e 1? Porque a professora disse no ano passado que o sistema binário é a linguagem do computador.</i>
<b>P</b>	<i>Sim. Mas, sempre programamos em uma sintaxe de linguagem próxima da nossa compreensão para facilitar. Seria muito difícil de programar em binário, praticamente impossível. Existe outro programa que se encarrega de fazer essa “tradução” em binário.</i>

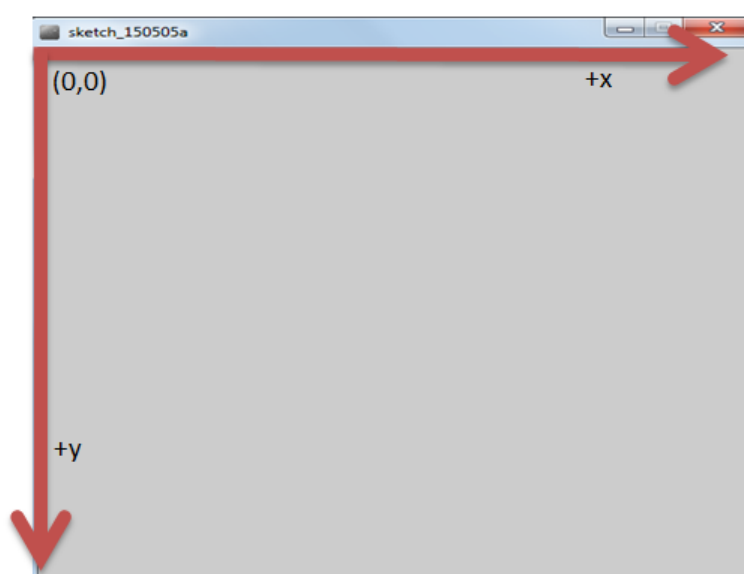
Fonte: Primária (2015).

Nota-se que a pergunta do aluno **MB** foi coerente com a situação inicial e o mesmo conseguiu estabelecer relação com situação de aprendizagem explorada na aula de matemática no ano anterior, quando a professora abordou provavelmente o sistema binário de numeração, no qual tinha os objetivos de identificar as características de um sistema de numeração na base 2, transformar os registros de base 2 em base 10 e vice-versa, efetuar operações de adição no sistema binário e conhecer as medidas de memória do computador. Os demais

integrantes não questionaram nesse momento, mas ouviram atentamente a interação entre o pesquisador e o aluno.

Após a apresentação da interface da ferramenta e suas principais características iniciamos a exploração do sistema de coordenada do Processing 2 (Figura 34). O sistema de coordenadas da área de trabalho do Processing 2 (sketch) está ancorado no canto superior esquerdo com o eixo x que se prolonga horizontalmente da esquerda para a direita, e o eixo y que se prolonga verticalmente de cima para baixo. Nas oficinas não foi abordado o eixo z, para fins 3D.

Figura 34 - Sistema de coordenada do Processing 2.



Fonte: Autor (2015).

Para ajudar na compreensão do sistema de coordenadas da ferramenta, o pesquisador levou em papel impresso, uma malha quadriculada para propor uma representação similar ao sistema de coordenadas do Processing, na qual os estudantes pudessem colocar alguns pontos que foram sugeridos pelo pesquisador. Nesta etapa realizou-se a atividade juntamente com os alunos, que consistia em desenhar na malha quadriculada as formas primitivas especificadas pelo código (Quadro 21).

Quadro 21 - Atividade proposta no papel milimetrado.

Usando a malha quadriculada em branco, desenhe as formas primitivas especificadas pelo código: `line(0,0,10,7); point(0,3); point(0,5); rect(6,1,5,4); ellipse(4,8,5,5);`

Fonte: Autor (2015).

Inicia-se a oficina trabalhando com os comandos linha a linha. Nesse primeiro momento os alunos conheciam a sintaxe do comando `size()`, `point()` e `line()`. Assim, foi proposto o primeiro desafio diretamente na ferramenta, que na visão desse pesquisador era

muito simples, mas que ao ser lançado podemos vivenciar experiências profundas de aprendizagem.

O desafio consistiu em criar uma janela (400x400 pixel) e colocar um ponto no centro da região. O aluno **CA** foi o primeiro a terminar com êxito o desafio proposto e imediatamente disse: *“Professor, terminei. O ponto parece uma sujeira no monitor”*. A aluna **NB** apresentava dificuldades para identificar a coordenada do ponto central. O aluno **MB** disse para a aluna **NB**, *“lembra da batalha naval que fizemos o ano passado? É igualzinho!”*

O aluno **MB** faz referência ao jogo tradicional, bem antigo que usa um sistema de coordenadas. Atualmente, existem versões eletrônicas desse jogo. Para jogar Batalha Naval, utiliza-se de uma malha quadriculada, onde cada quadrado é identificado por um sistema de coordenadas. Esta atividade está proposta na apostila do Sistema Anglo utilizada pelos alunos no 6º ano – Módulo 12. Assim, notamos que o aluno conseguiu estabelecer relações entre os conhecimentos matemáticos apropriados com o sistema de coordenada do Processing 2.

O programa do aluno **VH** não rodava e o mesmo não conseguiu descobrir o erro. A aluna **TS** conseguiu identificá-lo e notou que faltava o ponto e vírgula ao final da linha do comando point.

O aluno **VH** disse: *“a tela do computador agora é o papel milimetrado, como ficou bem mais legal colocar os pontos”*. Quando o aluno se referiu dessa forma, acreditamos que o grupo conseguiu perceber que o sistema de coordenadas do Processing 2 não é nada mais do que um pedaço de papel milimetrado e compreender que cada pixel da tela é uma coordenada, que determina a localização de um ponto no espaço.

A seguir, foi proposta uma continuidade do desafio inicial, na qual os estudantes tinham que traçar um segmento do canto superior direito da janela ao canto inferior esquerdo da janela, considerando a mesma área da janela (400x400 pixels).

O aluno **MB** em menos de um minuto, novamente disse: *“pronto, acabei!”* Nesse momento o pesquisador verificou a coerência da estratégia empregada e valorizou o trabalho do aluno parabenizando-o pelo procedimento assertivo.

Os demais alunos concluíram o desafio, porém com um pequeno erro. Como mediador do processo, solicitei para que os alunos analisassem as coordenadas apresentadas nos atributos do comando  $\text{line}(x_1, y_1, x_2, y_2)$ . Durante todas as oficinas tivemos a preocupação de não adiantar respostas, possibilitando aos alunos se tornarem ativos e protagonistas do processo de descoberta e aprendizagem.

A aluna **KA** percebeu o erro: *“professor, não é 400, o último ponto é 399 porque começa em zero”*. Nesse momento notou-se o compartilhamento de experiências no grupo.

O aluno **CA** tinha dificuldades de organização com o código, disposto numa sequência linear, sendo realizada a intervenção do pesquisador. Não havia nenhum erro, apenas uma questão de organização do código. A interferência, nesse momento, levou o aluno a uma melhora da organização do código de seu programa. Quando se programa, melhora a organização dos estudantes, pois os mesmos desenvolvem habilidades no tocante a estruturação de blocos de comando, isto implica um aprimoramento das habilidades de organização, uma vez que programar exige organização, o que permite facilitar a análise do desencadeamento lógico.

Para completar o desafio inicial, a proposta consistia em traçar outra linha que unisse o canto superior esquerdo com o canto inferior direito. Observe um episódio interessante registrado durante a execução do desafio inicial (Quadro 22).

Quadro 22 - Episódio sobre as diagonais.

<b>CA</b>	<i>“MB, precisa fazer o ponto do meio?”</i>
<b>MB</b>	<i>“Não, você vai fazer só as duas linhas cruzando”</i>
<b>CA</b>	<i>“Posso começar por qualquer uma então?”</i>
<b>KA</b>	<i>“Tanto faz, você coloca o ponto que começa e o ponto que termina”</i>
<b>NB</b>	<i>“Me ajuda, o risco tá vindo aqui no meio”</i>
<b>KA</b>	<i>“Você deve ter colocado a coordenada errada, são as duas diagonais”</i>

Fonte: Primária (2015).

Na ocasião, a aluna **NB** apresentava dificuldades em entender o sistema de coordenada da área de trabalho do Processing 2. Ela recebeu feedback da execução do programa, e reconheceu que seus objetivos não foram atingidos, mas não conseguia enxergar seu erro. A aluna **KA** demonstrava domínio e confiança no que desenvolvia, e imediatamente sabia que o erro se encontrava na coordenada final (399,399). Em uma de suas falas a aluna **KA** relata: *“... esse sistema de coordenadas a gente usa em mapas pra gente localizar os lugares, lá em casa tem um desse, só que lá tem as letras e os números”*.

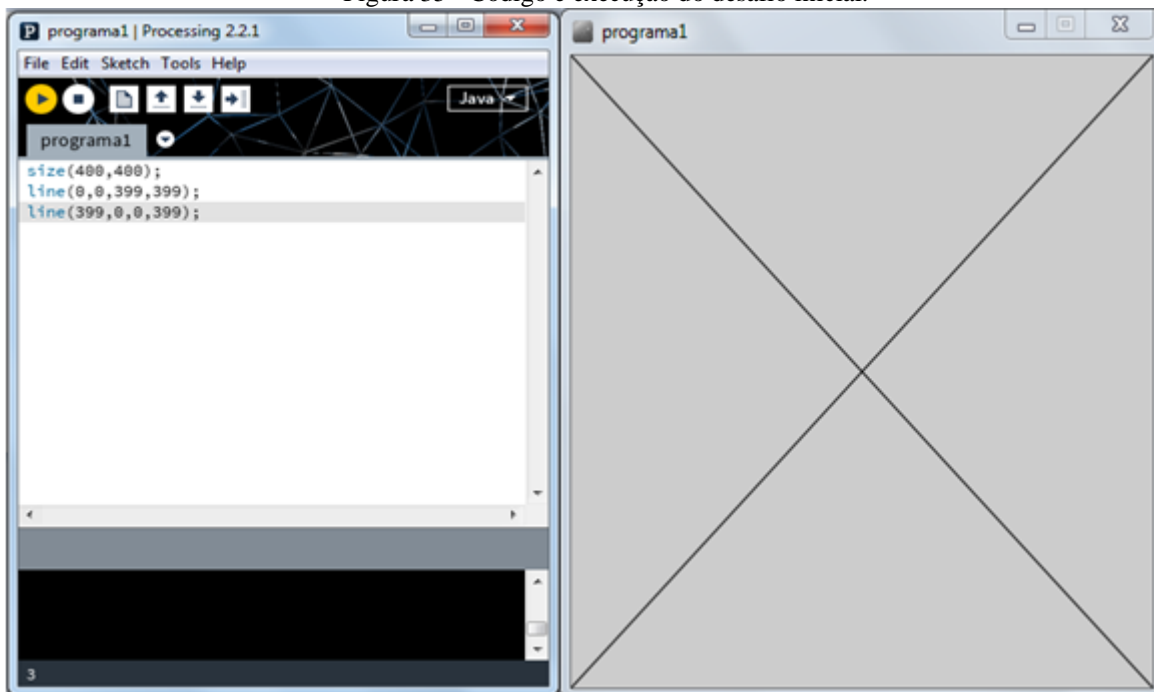
O aluno **CA** acreditava que era necessário criar o ponto de intersecção dos dois segmentos, mas foi orientado pelos amigos do grupo **MB e KA**, que a criação não era necessária. A aluna **KA** reconhece que são as duas diagonais do quadrado, sendo que em nenhum momento foi utilizado o termo “diagonal” pelo pesquisador.

Enquanto aguardávamos a aluna **NB** finalizar seu programa, os alunos **CA e MB** estavam explorando o comando `line()` em outras coordenadas, e trocavam experiências entre eles. Não foi estabelecido um tempo para o término do desafio, pois os estudantes estavam se habituando à dinâmica das oficinas na ferramenta e foi priorizado inicialmente os princípios

fundamentais da programação em Processing 2, que consiste no domínio de utilização do sistema de coordenadas.

Como produto final do desafio, o aluno **CA** apresentou o seguinte código (Figura 35).

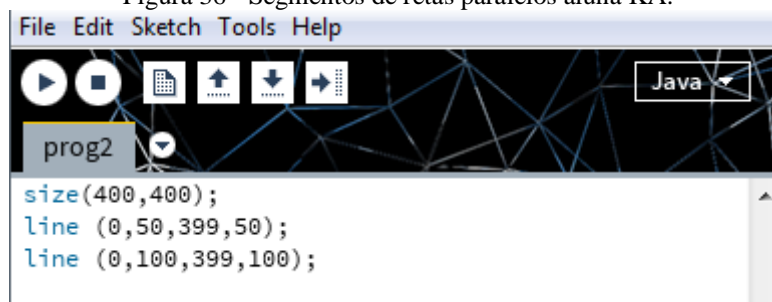
Figura 35 - Código e execução do desafio inicial.



Fonte: Primária (2015).

No segundo momento da oficina foi proposto que os alunos criassem duas linhas que não se encontrassem. A aluna **KA** apresentou a seguinte estratégia (Figura 36).

Figura 36 - Segmentos de retas paralelos aluna KA.



Fonte: Autor (2005).

Verifica-se que a aluna fez retas paralelas horizontais. O aluno **MB** apresentou estratégia diferente da aluna **CA** utilizando retas paralelas verticais.

O aluno **VH** chegou numa inferência interessante: *“Professor, olha só que quando a gente coloca os x com mesmos valores as linhas não se encontram e ficam na horizontal. Agora se fizer como o MB fez, o valor de y ficou igual, por isso que as linhas ficaram na vertical”*.

Nesta oficina foi possível explorar ainda os comandos triangle ( $x_1, y_1, x_2, y_2, x_3, y_3$ ), quad ( $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$ ), rect ( $x, y, largura, altura$ ) e ellipse ( $x, y, largura, altura$ ). Deixou-se um tempo livre para que o grupo pudesse realizar a experimentação dos comandos apresentados.

Evidencia a seguir (Quadro 23) algumas falas dos alunos durante o episódio de descoberta.

Quadro 23 - Processo de descoberta das formas básicas.

<b>TS</b>	<i>“Nossa, isso aqui é muito diferente, estou gostando”.</i>
<b>NB</b>	<i>“Meu retângulo ficou pequeno demais, vou ter que aumentar!”.</i>
<b>NB</b>	<i>“Como que fica o comando da circunferência, primeiro coloco o centro?”.</i>
<b>MB</b>	<i>“Eu fiz um de cada, mas olha aqui vou arrumar porque o triângulo ficou sobre o meu retângulo”.</i>
<b>VH</b>	<i>“Quando não dá erro dá até uma emoção”.</i>

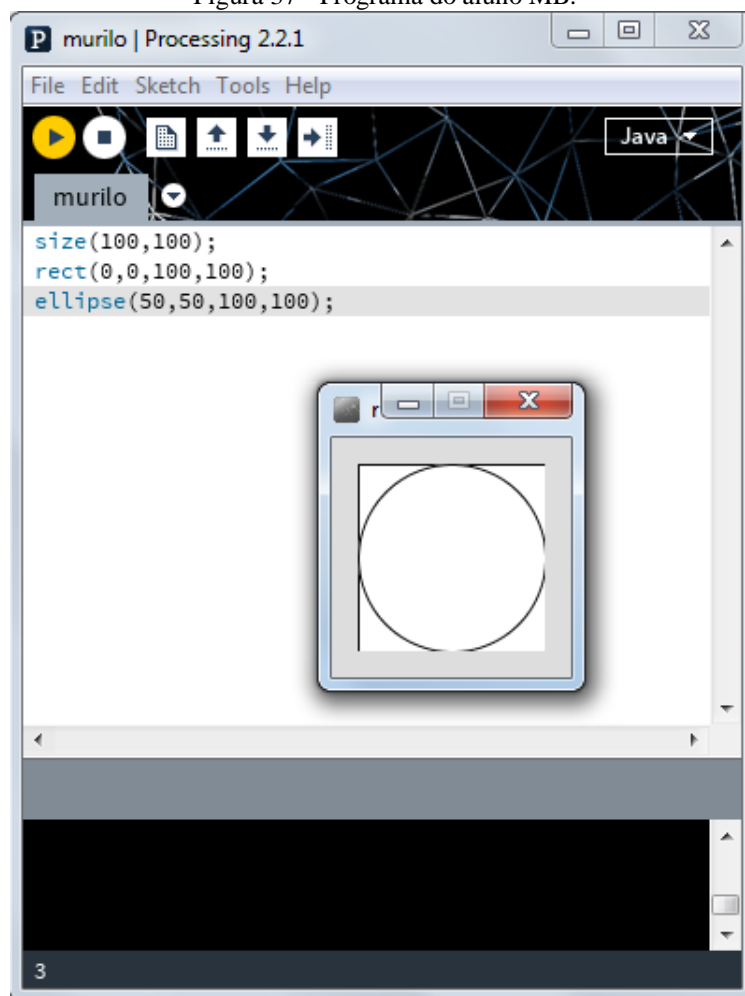
Fonte: Primária (2015).

A partir dos discursos no processo de descoberta constata-se que para os alunos o trabalho na ferramenta era algo prazeroso, como evidenciado na fala da aluna **TS**. A aluna **NB** percebeu a necessidade de melhorar o resultado de seu programa a partir do feedback visual que o Processing apresenta. Foi evidenciado um trabalho mais colaborativo entre os integrantes. O aluno **MB** teve grande clareza ao sanar as dúvidas da aluna **NB** sobre os parâmetros do comando ellipse(). O aluno **VH** demonstrou compreensão de seus insucessos e destaca que é emocionante quando executa seu programa e o mesmo sai como planejado.

Na última atividade desse encontro, foi proposto que os estudantes criassem um quadrado de mesma dimensão da janela definida e colocassem em seu interior uma circunferência, sendo esta a maior possível.

Na resolução desse desafio, a construção do quadrado foi muito tranquila, porém surgiram muitos questionamentos entre o grupo relativos à construção da circunferência. Um primeiro questionamento estava relacionado aos parâmetros das dimensões, e chegaram a uma inferência por tentativa e erro, na qual descobriram que se tratava não do raio da circunferência, mas sim do seu diâmetro. O aluno **MB** foi o primeiro a apresentar uma solução para o desafio (Figura 37). Este desafio desenvolveu pedidos de ajuda para o pesquisador. Eles resolviam seus problemas num trabalho colaborativo e com o auxílio do feedback visual da ferramenta. Em nenhum momento o pesquisador adiantou algum tipo de resposta ao grupo.

Figura 37 - Programa do aluno MB.



Fonte: Primária (2015).

Na oficina foi possível explorar de forma situada o desenvolvimento e/ou aprimoramento da habilidade de localização de pontos num sistema de coordenadas em um plano, identificar a diferença entre uma reta e um segmento de reta, depreender os conceitos de retas paralelas, retas concorrentes e retas perpendiculares.

Ao final da primeira oficina, compreende-se que foi executada com êxito por ter cumprido seus objetivos traçados inicialmente. De acordo com as categorias de análise propostas apresenta-se no Quadro 24 a classificação das falas evidenciadas.

Quadro 24- Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 1.

<b>Peculiaridade</b>	<b>Fala dos alunos</b>	<b>Como foi observado</b>
Não linearidade	<p><b>CA</b> “Posso começar por qualquer uma então?”</p> <p><b>KA</b> “Tanto faz, você coloca o ponto que começa e o ponto que termina”.</p>	Possibilitou aos alunos múltiplas oportunidades para explorar o programa a partir de várias perspectivas, o que implicou ricas situações de aprendizagem. Todas as estruturas não apresentam um único sentido, apresentam múltiplos caminhos e destinos, desencadeando em múltiplos finais.
Ênfase no “saber fazer”	<p><b>KA</b> “...esse sistema de coordenadas a gente usa em mapas pra gente localizar os lugares, lá em casa tem um desse, só que lá tem as letras e os números”.</p> <p><b>MB</b> “lembra da batalha naval que fizemos o ano passado é igualzinho!”.</p>	Quando os alunos conseguem estabelecer relações com algo prático, o interesse é despertado.
Informal	<p><b>VH</b> “Quando não dá erro dá até uma emoção”.</p>	Quando o aluno erra, ele sempre pode voltar para adequar ou mudar a estratégia. O erro é algo muito natural, assim ocorre constantemente nas relações sociais. Os estudantes são assim encorajados a correr riscos, a explorar, a tentar coisas novas. Na verdade, fracassar frente a um desafio proposto foi tido pelos alunos como uma coisa boa. A escola tradicional costuma oferecer muito menos espaço para o risco, exploração e insucesso. Constatamos que a ferramenta colaborou nesse sentido.
Interatividade	<p><b>NB</b> “Meu retângulo ficou pequeno demais, vou ter que aumentar!”.</p> <p><b>MB</b> “Eu fiz um de cada, mas olha aqui vou arrumar porque o triângulo ficou sobre o meu retângulo”.</p> <p><b>TS</b> “Nossa, isso aqui é muito diferente, estou gostando!”.</p>	No uso da apostila os estudantes se viam passivos e com o Processing ocorre uma resposta aos alunos que mantêm um canal de interação. Para que aconteça algo é necessário que o aluno entre numa relação dialógica, e isso acontece muito facilmente com a ferramenta empregada, por meio do feedback. As palavras e os atos são colocados no contexto de uma relação interativa entre o estudante e o mundo.

Fonte: Autor (2015).

Nessa primeira etapa de compilação de análise, também foi possível identificar que o assunto programação não assustou os alunos, sendo introduzido tranquilamente, uma vez que o grupo pertence à geração digital. Foi interessante ver o grupo trocando ideias, empenhando-

se, querendo criar algo diferente do que tinha visto. Era surpreendente, uma vez que todos estavam entretidos e não havia discussões fora da temática, como ocorrem rotineiramente em sala de aula tradicional.

#### **4.2.2 Oficina 2 – Exploração das formas geométricas e suas propriedades**

Os estudantes aprenderam na oficina anterior instruir o computador para desenhar uma forma com local e tamanho específicos. No entanto, um elemento fundamental faltava: a cor. Os estudantes a todo instante perguntavam como inserir cor nos objetos. Era notória a inquietude do grupo em descobrir como colorir seus objetos construídos.

Discute-se que no mundo digital, a cor é definida antes do objeto com um intervalo de números e explica-se o caso mais simples: preto e branco ou tons de cinza.

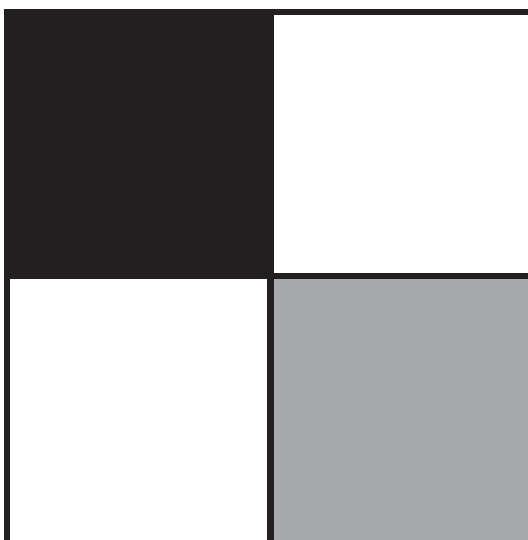
**P:** “...temos o seguinte: 0 significa preto, 255 significa branco. No meio, qualquer outro número, um tom de cinza que vai do preto ao branco”

**VH:** Professor, tem que ser antes de cada coisa que a gente fizer?

**MB:** “É como pedir pra alguém fazer um desenho com canetinha, antes tem que avisar a cor que queremos”.

Em seguida, um novo desafio foi proposto, no qual tinham que apresentar algo parecido com o que foi representado no quadro branco, disponível no laboratório. O desafio proposto consistia em construir algo semelhante ao desenho (Figura 38).

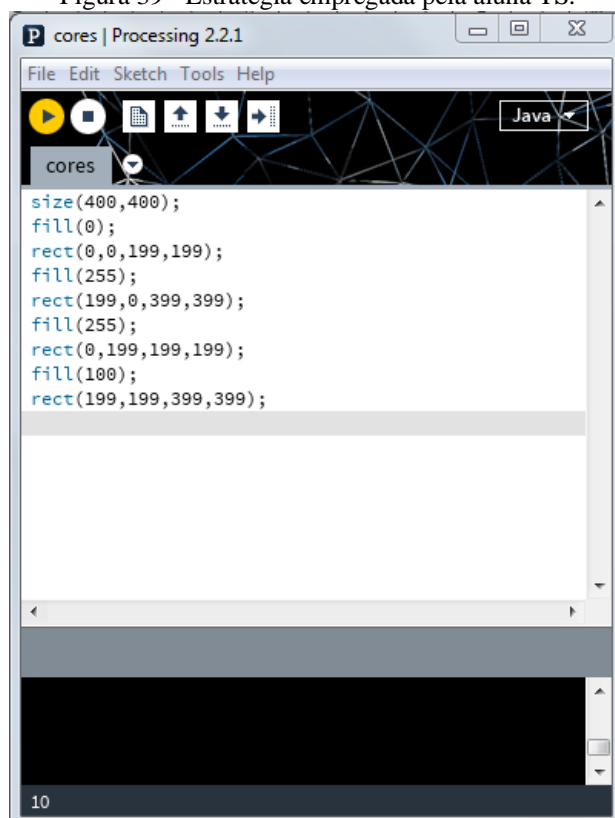
Figura 38 - Desafio envolvendo formas elementares e cores.



Fonte: Autor (2015).

Neste desafio o que chamou atenção foi a utilização de duas estratégias diferentes para atingir o mesmo objetivo. A aluna **TS** foi a primeira a apresentar o código e demonstrava uma fisionomia que esboçava sorrisos por ter conseguido alcançar o objetivo (Figura 39).

Figura 39 - Estratégia empregada pela aluna TS.



```

cores | Processing 2.2.1
File Edit Sketch Tools Help
cores
size(400,400);
fill(0);
rect(0,0,199,199);
fill(255);
rect(199,0,399,399);
fill(255);
rect(0,199,199,199);
fill(100);
rect(199,199,399,399);
10

```

Fonte: Primária (2015).

A aluna **TS** solucionou o desafio colocando quadrados justapostos e antes de cada objeto, estipulando a cor desejada para preenchimento. A aluna aproveitou a janela definida no `size()`. A aluna **KA** procedeu de forma análoga em sua estratégia.

Interessante destacar que **CA** atingiu o mesmo objetivo proposto, partindo de uma estratégia diferente. Analisando seu código, o aluno primeiramente traçou um quadrado maior, depois traçou duas linhas perpendiculares e depois sobrepôs aos quadrantes formando dois novos quadrados para conseguir colorir. O aluno **CA** conseguiu atingir os objetivos, porém seu código ficou extenso e redundante. O aluno percebeu que quando criou os quatro quadrantes não conseguiu inserir a cor no preenchimento e como solução realizou a sobreposição.

O professor elogiou o trabalho desenvolvido pelo aluno **CA**, porém destacou para o grupo que o interessante era fazer um programa com o menor número de códigos, que atendesse aos objetivos, evitando assim desperdício de memória e alcançando um melhor desempenho do programa.

Os alunos **VH** e **NB** foram os últimos a terminarem o desafio, pois apresentaram demora nas coordenadas iniciais e finais da construção dos retângulos, descobrindo seus erros por meio da execução do programa. Quando os estudantes compilam seus programas recebem *feedback* imediato durante as suas experiências, assim reconhecem seus erros e mesuram a

distância que estão da meta proposta no desafio. Assim, evidenciamos que para aprender, os estudantes precisam receber, o mais rápido possível, feedbacks claros, específicos e explicativos.

O aluno **CA**, disse durante conversa com o aluno **MB**, “...direto no computador fica mais fácil, eu não gosto de fazer no papel”. Na resolução dos desafios notou-se um desconforto sempre que era necessário fazer uso do papel milimetrado que o pesquisador trouxe para auxiliar no desenvolvimento das atividades. Inicialmente fizeram uso constante do apoio, mas gradualmente foram se desprendendo do papel milimetrado, atuando diretamente no programa. Com isso, a motivação positiva foi percebida na dinâmica da resolução dos problemas à medida que adquiriam autonomia nos processos.

Ainda os alunos insistiam em colorir, para eles o preto, branco e a escala de cinza não satisfaziam a seus anseios. A aluna **KA** exclamou: “Professor, não quero pôr cinza, quero colocar cor mesmo!”, como se as variações de cores apropriadas não fossem cores.

Foi explicado aos alunos sobre a formação das cores digitais que são construídas por meio da mistura de três cores primárias: vermelho, verde e azul (RGB). Para formação de uma tonalidade era necessária a combinação de intensidade dessas três cores. A princípio os alunos estranharam e logo questionaram:

**VH:** “...não posso escrever o nome da cor em inglês? Não funciona?”

**NB:** “...vamos ter que ficar tentando?”

Assim, foram apresentadas aos alunos duas maneiras para facilitar o trabalho com cores. O Processing 2 possui um seletor de cores para auxiliar na escolha de cores, com acesso via TOOLS (a partir da barra de menus) e também foi possibilitado por meio do editor de imagens (GIMP) do Linux Educacional. Os alunos deram preferência ao aplicativo GIMP por estarem familiarizados com sua utilização. Rapidamente conseguiram compreender a formação das cores e a utilização da ferramenta de auxílio.

Interessante destacar que, imediatamente, os alunos começaram a utilizar o comando fill(R,G,B) em suas criações e, imediatamente, desenharam uma figura geométrica e começaram a testar diversas cores, sem instrução ou comando do professor. Os estudantes conseguiram entender o processo de formação da cor digital e suas combinações e realizavam seus testes. A aluna **NB** enquanto testava o comando disse “Eu não sabia disso, como é simples”. A aluna na verdade estava querendo dizer que ela entendeu o conceito de combinação para se trabalhar com cores, e a partir desse momento aquilo que ela entendia como sendo algo complicado, foi desmistificado e passou a ser algo simples.

Percebe-se que após os alunos descobrirem o mundo das formas primitivas e das cores no padrão RGB, a exploração no Processing 2 começou a ganhar liberdade. Foi notório que, a partir desse momento, o grupo começou a fazer uso da criatividade, que é um passo importante para implementar esses conhecimentos em um cenário de programação do mundo real. Para o novato em programação, a simplicidade e a interação rápida do Processing 2 foi fundamental para os trabalhos com iniciantes, evidenciado na fala da aluna **KA**: “*Professor, programar é muito animado*”. Foram contemplados os comandos primitivos em 2D e o comando `arc()` foi explorado logo na sequência. Nesse momento os alunos começaram a tomar suas ideias e transformá-las em projetos reais de software.

Os estudantes gostavam porque tudo é muito rápido. Nas aulas tradicionais, em muitos momentos os alunos queriam que o professor adiantasse respostas e que o professor resolvesse os problemas propostos na apostila. O trabalho com a ferramenta foi diferente nesse aspecto, eles queriam ser os protagonistas e resolver seus problemas. **CA**: “*Professor, tá dando erro, mas não fala eu vou descobrir*”. Nota-se que, com essa geração de estudantes para que se mantenham engajados, o tempo de feedback tem que ser muito curto e o Processing 2 proporciona ricas interações, conforme características da aprendizagem situada apresentadas por Wenger (2000).

Foi disponibilizado um intervalo de 10 minutos aos alunos. Nenhum aluno quis sair do laboratório. Estavam a explorar os conceitos e comandos que estavam dominando de forma livre, mas queriam saber mais. O aluno **CA** estava assistindo a um vídeo no Youtube sobre Processing 2. A aluna **KA** descobriu antes de apresentá-los o guia de referências da ferramenta disponível em <<https://processing.org/reference/>>.

Os estudantes começam a explorar os comandos relacionados à cor, e o que estava programado na sequência didática foi adiado para explorar os anseios dos alunos. Percebe-se uma condução não linear, na qual o aluno direciona o que deseja aprender.

No intervalo, os alunos descobriram pesquisando a função `background()`. A aluna **TS** disse: “*olha aqui, esse comando vai trocar esse fundo cinza*”.

Nesse momento, a aluna reconheceu que o fundo padrão é cinza claro e que para definir o plano de fundo ou `background` necessitava utilizá-lo apenas uma vez. Interessante a rapidez com que todos começaram a testar o comando como trocavam informações.

Terminado o intervalo, passou-se a explorar o comando `arc()`, que está relacionado com a construção de arcos nos círculos. Primeiramente apresentamos aos alunos a compreensão básica de como os ângulos são definidos no Processing.

No resgate dos conhecimentos prévios dos alunos sobre ângulos evidenciou-se que o grupo reconhecia o grau como unidade de medida de rotações.

**P:** *“Uma rotação completa vai de zero a 360°. Um ângulo de 90° é um quarto de 360°. É bastante intuitivo para nós pensar ângulos em termos de graus. Mas gente, aqui no Processing requer ângulos especificados em radianos<sup>29</sup>”.*

Consecutivamente, o aluno **MB** indagou: *“O que é isso, radiano?”* Para os alunos um conceito totalmente inédito. Porém, o pesquisador explicou que não era necessário se preocupar com isso naquele momento, uma vez que, felizmente, o Processing 2 torna isso fácil. A função radians() converte automaticamente os valores de graus em radianos, mas ela não foi abordada nesse momento. O aluno **VH** indaga *“Então a gente não precisa saber isso não?”*.

Além disso, apresentamos que as constantes PI e TWO\_PI estão disponíveis para acesso conveniente a esses números comumente utilizados, equivalente a 180° e 360°, respectivamente, sem entrar em detalhes sobre radianos. Assim, apresentamos no quadro branco um resumo de quatro valores em radianos que são utilizados com frequência. PI, QUARTER\_PI, HALF\_PI, e TWO\_PI pode ser usado para substituir os valores de radiano para 180°, 45°, 90°, 360°, respectivamente. Os alunos **NB** e **KA** se manifestaram nesse momento.

**NB:** *“Vamos ter que decorar esses nomes aí?”.*

**KA:** *“Não é difícil de guardar, eu acho que já peguei”.*

Foi proposta uma atividade similar, encontrada no material de apoio e solicitado aos alunos que digitassem o código (Quadro 25) para que pudessem visualizar o que aconteceria.

Quadro 25 – Codificação envolvendo arcos.

```
size(480, 120);
arc(90, 60, 80, 80, 0, HALF_PI);
arc(190, 60, 80, 80, 0, PI+HALF_PI);
arc(290, 60, 80, 80, PI, TWO_PI+HALF_PI);
arc(390, 60, 80, 80, QUARTER_PI, PI+QUARTER_PI);
```

Fonte: Autor (2015).

Como resultado da execução do código temos o resultado (Figura 40).

<sup>29</sup> Um radiano (Rad) é o ângulo definido em um círculo por um arco de circunferência com o mesmo comprimento que o raio do referido círculo. Um ângulo de 180° vale PI radianos. Um ângulo de 360° vale 2 \* PI radiano, e 90° vale PI/2 radianos, e assim por diante. A fórmula para converter de graus em radianos é: Rad = 2 \* PI \* (graus / 360). PI, o que é? Na matemática a constante PI (ou π) é um número irracional definida como a razão entre o comprimento da circunferência (perímetro) pelo seu diâmetro. É aproximadamente 3,14159.

Figura 40 - Execução da atividade com arcos.



Fonte: Adaptado de Getting Started with Processing (2010).

Durante a execução a aluna **TS**: “*A função `arc()` desenha um pedaço de uma elipse*”.

Imediatamente ao ver o resultado da execução do programa a aluna relaciona a construção de arcos e/ou setores circulares com pedaço de uma elipse.

**P**: “*O primeiro e segundo parâmetros definem a coordenada do centro, o terceiro e quarto definem as dimensões da largura e altura. O quinto parâmetro define o ângulo para iniciar o arco, e o sexto define o ângulo para parar.*”.

Aparentemente, trabalhar com arcos no Processing com alunos de 12 anos parecia ser algo impossível, mas o grupo se apropriou muito rapidamente da sintaxe do comando. Observe o episódio (Quadro 26).

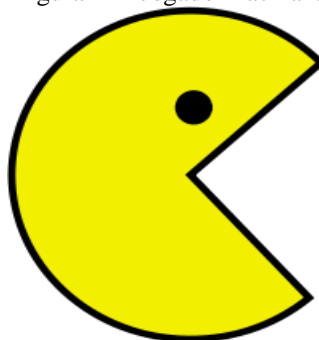
Quadro 26 - Episódio sobre pacman.

<b>MB</b>	<i>“... professor olha que legal, com isso vou conseguir fazer um pacman!”</i>
<b>P</b>	<i>“Então está lançado o desafio! Construir um jogador pacman... todo mundo sabe como é um jogador pacman?”</i>
<b>CA</b>	<i>“Quem não sabe professor o que é pacman!” (risos).</i>

Fonte: Primária (2015).

É interessante que o desafio partiu do próprio grupo. Pacman é um game da década de 1980 e tornou-se um dos jogos mais populares. A dinâmica do jogo é simples e consiste num jogador (Figura 41) representado por uma cabeça redonda com uma boca que se abre e fecha, posicionado em um labirinto simples, repleto de pastilhas e fantasmas que o perseguem. O objetivo é comer todas as pastilhas sem ser alcançado pelos fantasmas, em ritmo progressivo de dificuldade.

Figura 41 - Jogador Pacman.



Fonte: Autor (2015).

Para tal construção os alunos tinham que reconhecer que o jogador pacman é um setor circular. Vejamos outro episódio (Quadro 27).

Quadro 27- Episódio dos problemas na construção do arco.

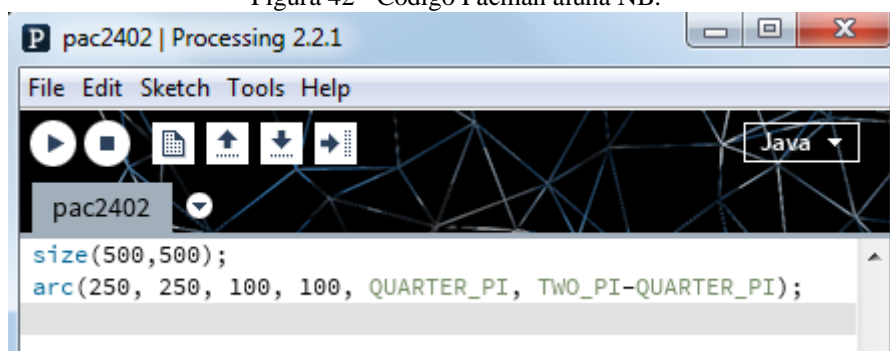
<b>TS</b>	<i>“professor, o ângulo de início e fim não está dando certo?”</i>
<b>NB</b>	<i>“o meu também”</i>
<b>P</b>	<i>“Façam os testes e descubram o que está errado”</i>
<b>CA</b>	<i>“...é diferente do que estamos acostumado”</i>
<b>P</b>	<i>“Mas como vocês estão acostumados?”</i>
<b>CA</b>	<i>Assim professor. (O aluno faz a representação gestual no sentido anti-horário)</i>
<b>P</b>	<i>Interessante CA, mas isso é fácil solucionar!”</i>

Fonte: Primária (2015).

O aluno **CA** notou que na construção do setor circular, o ângulo de início para o ângulo de término obedecia ao sentido horário. Essa foi uma dificuldade encontrada pelos alunos, pois os mesmos estavam habituados a trabalharem com a construção de arcos no

sentindo anti-horário. Superado pelo grupo tal problemática, a aluna **KA**, apresentou o seguinte código (Figura 42) e explicou seu raciocínio ao grupo.

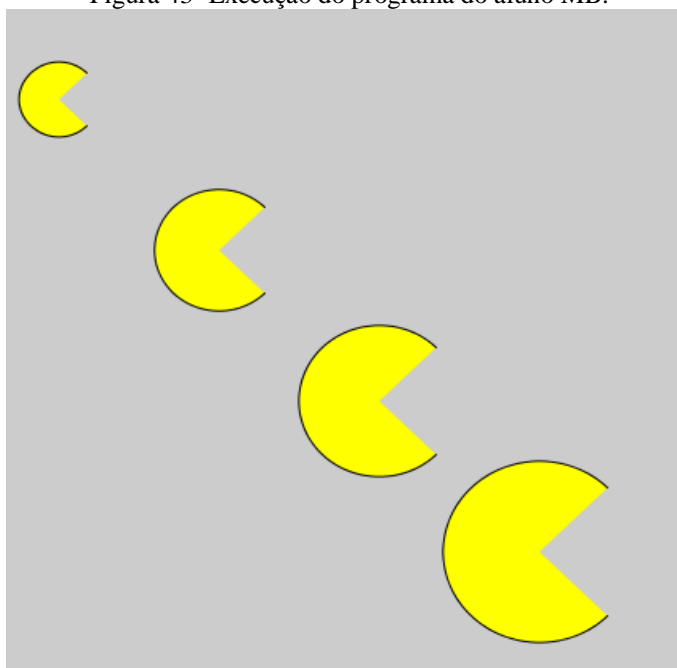
Figura 42 - Código Pacman aluna NB.



Fonte: Primária (2015).

O grupo passou a fazer vários jogadores pacman espalhados pela janela (Figura 43), alterando a coordenada do centro, de diversos tamanhos e inserindo a cor amarela. Os estudantes **KA**, **MB**, **TS** mostravam refletir sobre seus erros, enquanto os demais integrantes ainda faziam em alguns momentos uso da tentativa e erro. Mas era notável a motivação do grupo, um ajudando o outro na superação dos problemas e dificuldades. Todos conseguiram atingir o objetivo do desafio e externavam motivação para desbravar a ferramenta.

Figura 43- Execução do programa do aluno MB.



Fonte: Primária (2015).

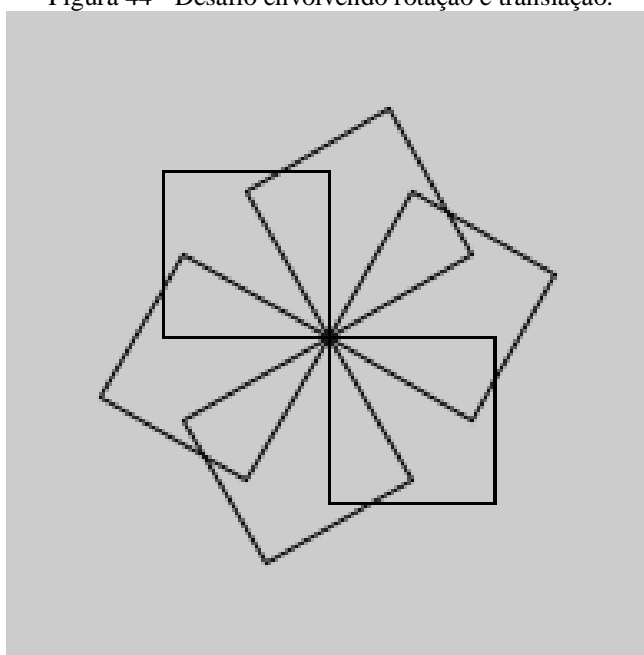
Nesse desafio foi percebido que os alunos **MB**, **CA** e **KA** conseguiram generalizar o conceito em outros objetos. Eles projetavam diferentes arcos de circunferência, sendo evidenciado o domínio do conceito, fator elencado por Lave e Wenger (1991) em sua teoria.

O aluno **CA** descobriu que conseguia visualizar a execução de seu programa em tela cheia, compartilhando a novidade com o grupo.

Após esse desafio iniciou-se uma experiência com o comando `rotate()` e `translate()`. A translação é um deslocamento do plano das coordenadas. Assim como a rotação, não deslocamos as formas em si, mas sim o plano onde foram desenhadas. Utilizamos alguns exemplos para demonstrar aos alunos sua aplicação. Observa-se que o grupo teve maior facilidade em compreender o comando `translate()` em relação ao comando `rotate()`, como evidenciado na fala da aluna **NB**: *“Eu entendi que a tela começa agora nessa posição aqui do meio que foi colocada, não é isso?”* A aluna apresentava certa insegurança, mas referia-se corretamente ao deslocamento do sistema de coordenadas à posição (100,100).

Depois de ultrapassado o tempo previsto para a realização da oficina, para o grupo se divertir com os comandos e conceitos apropriados, sugerimos um desafio para combinar rotação e translação, que consistia em fazer a seguinte representação (Figura 44), com seis quadrados sem preenchimento com centro rotacionando  $60^\circ$  em torno do ponto central da janela.

Figura 44 - Desafio envolvendo rotação e translação.



Fonte: Autor (2015).

Os alunos não conseguiram entrar num consenso. **Nenhum** estudante conseguiu descobrir a estratégia para o desafio proposto, apesar das várias tentativas e erros. Tiveram dificuldade em reconhecer que bastava fazer o primeiro quadrado com origem na posição central e depois realizar a translação do plano de coordenada para este ponto e rotacionar em torno deste novo eixo.

**TS:** “Preciso de ajuda para fazer. Alguém conseguiu?”.

**MB:** “Professor, ajuda a gente nesse aqui”.

Sob suporte (andaime), o aluno **MB**, foi o primeiro a apresentar o desafio com êxito. Vejamos o código apresentado pelo aluno (Figura 45).

Figura 45 - Programa do aluno MB.

```

size(200,200);
noFill();
rect(100,100,50,50);

translate(100,100);
rotate(radians(60));

rect(0,0,50,50);
//translate(100,100);
rotate(radians(60));
rect(0,0,50,50);

rotate(radians(60));
rect(0,0,50,50);

rotate(radians(60));
rect(0,0,50,50);

rotate(radians(60));
rect(0,0,50,50);

```

Fonte: Primária (2015).

Ao analisar seu código percebe-se a organização do aluno **MB** que para melhor entendimento fragmentou, deixando uma linha em branco. Percebe-se que o mesmo tentou realizar a translação do sistema de coordenada novamente, e quando notou seu erro, deixou como comentário. As notas explicativas incorporadas dentro do código são comuns em programação. Comentários de linha única são representados com os dois caracteres de barra (//). Os comentários são ignorados pelo compilador, utilizado pelo aluno neste caso, com o intuito de desprezar o código escrito.

Os alunos **CA** e **VH** não conseguiam perceber que a rotação era cumulativa. Estavam a fazer a rotação 60°, 120°, 180°. O sistema de coordenada já havia girado 60°, bastava girar novamente em 60°. O aluno **MB** auxiliou os alunos com o término do desafio, demonstrando

ter se apropriado bem dos conceitos. Este desafio consumiu aproximadamente 25 minutos, e gerou bastante discussão envolvendo conceitos de rotação, translação e ângulo em giros.

Foi evidenciado que os alunos fizeram uso da função radians() sem demonstrar maiores problemas, tinham consciência de que o Processing trabalha com ângulos em radianos e para isso era necessário converter em graus.

Quadro 28- Peculiaridade da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 2.

<b>Peculiaridade</b>	<b>Fala dos alunos</b>	<b>Como foi observado</b>
Ênfase no “saber fazer”	<p><b>TS:</b> “<i>Preciso de ajuda para fazer. Alguém conseguiu?</i>”</p> <p><b>MB:</b> “<i>Professor, ajuda a gente nesse aqui</i>”.</p>	Um ambiente de aprendizagem situado prevê momentos críticos, e os andaimes de apoio, onde o professor fornece as habilidades, estratégias quando os alunos têm dificuldades na construção das ideias do desafio. Aos poucos, o suporte (o andaime) é removido até que o aluno seja capaz de caminhar sozinho, conforme apontado por YOUNG (1993).
Construção do conhecimento parte da necessidade	<b>MB:</b> “ <i>... professor olha que legal, com isso vou conseguir fazer um pacman!</i> ”.	Foi muito mais interessante para o grupo, quando a opção em construir o Pacman partiu do interesse e necessidade dos estudantes naquele momento.
Interatividade	<b>KA:</b> “ <i>Professor, programar é muito animado</i> ”.	Os estudantes estavam gostando porque tudo é muito rápido. O tempo curto de feedback fez toda a diferença.
Engajamento	<b>NB:</b> “ <i>Eu não sabia disso, como é simples</i> ”.	Um ambiente de aprendizagem situado promove articulação para permitir o conhecimento tácito para ser explicitado. Lave e Wenger (1991) apontam que ser capaz de falar o vocabulário e contar as histórias de uma cultura de prática é fundamental para a aprendizagem.

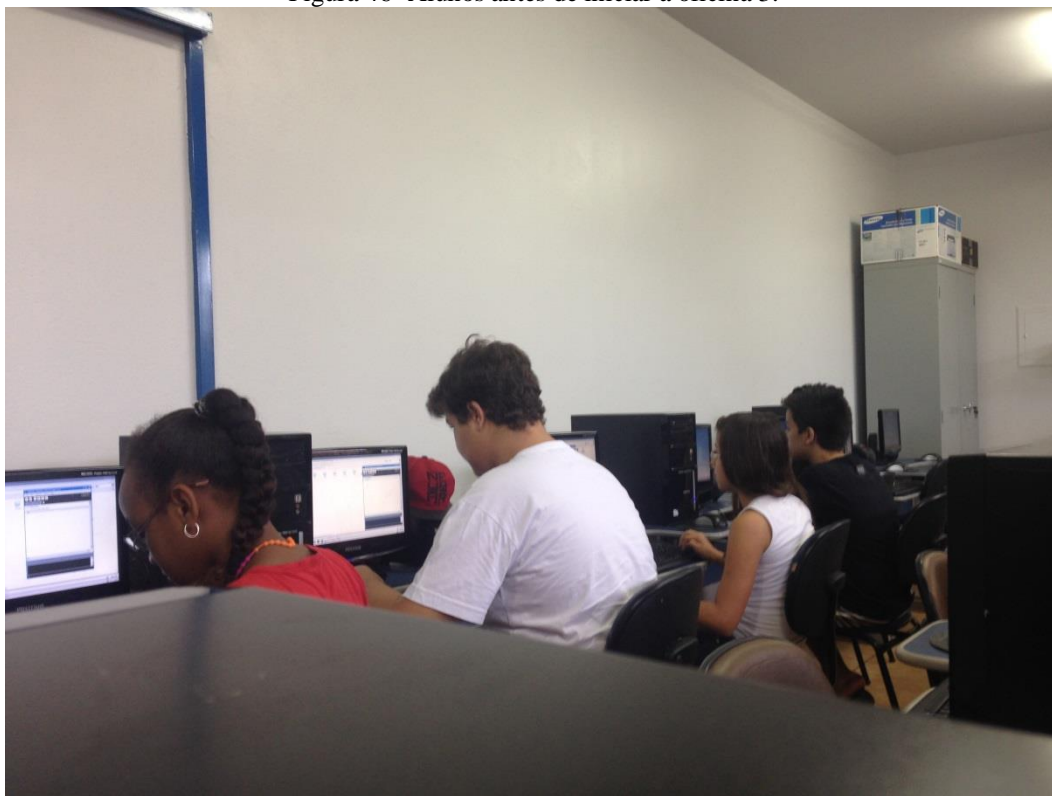
Fonte: Autor (2015).

Na oficina foi possível explorar a aprendizagem situada trabalhando habilidades matemáticas como: identificar o grau como uma das unidades de medida de ângulo, associar rotações com a ideia de ângulo, reconhecer rotação e translação como transformações geométricas entre muitas outras habilidades exploradas de forma situada numa comunidade de prática de programação.

### 4.2.3 Oficina 3 - Variáveis

Chegado à escola para ministrar a oficina, alguns alunos já estavam no LIE (Figura 46) desvendando o mundo da programação e se aventurando no Processing 2, resolvendo problemas individual e coletivamente que eles mesmos criaram. Assim, achamos interessante registrar o momento.

Figura 46- Alunos antes de iniciar a oficina 3.



Fonte: Autor (2015).

Nota-se que os estudantes estavam trabalhando com empenho em suas construções. Alguns construindo programas mais elaborados (aluno **MB**) e outros apenas reproduzindo os desafios anteriores (aluno **TS**).

O trabalho na ferramenta possibilitava um canal de fluxo, ampliando a cada oficina um leque de possibilidades. Todos os alunos, sem exceção, queriam conhecer mais. À medida que os alunos ampliavam suas habilidades, requeriam desafios com um grau de dificuldade maior, mas cada um dentro do seu nível de desenvolvimento.

O método torna interessante por possibilitar ao professor trabalhar com o mesmo desafio respeitando o nível cognitivo de cada aluno. O importante é que os alunos estavam a avançar em seus conhecimentos. Devido à heterogeneidade de alunos encontrados na sala de aula a ferramenta possibilita um trabalho respeitando as diferenças. Não houve ausência e alguns alunos do grupo já destacaram-se como líderes, como os alunos **MB**, **KA** e **TS**. A

aprendizagem situada prevê a presença social dentro da comunidade de prática, conforme apontados por Wenger (2000).

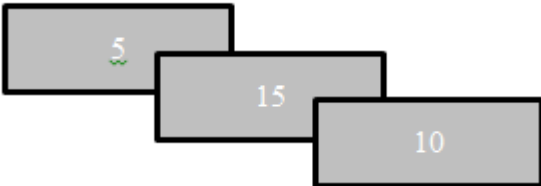
Enquanto professor nessa unidade escolar, quando se chega a uma sala de aula é sempre um grande desafio motivar os alunos para iniciar a aula e fazer com que eles abram a apostila. E isso não aconteceu no laboratório com o uso do Processing. Antes do início da oficina todos estavam com o Processing 2 aberto, vivenciando novas ou velhas experiências. A motivação está no “diferente”, no feedback rápido que a ferramenta possibilita e na liberdade de exploração que a ferramenta oferece aos estudantes.

Nessa oficina aborda-se o uso de variáveis e laço de condicionais. Em uma concepção inicial seria a oficina chave e mais complexa. Pensou-se em iniciá-la com analogias, numa tentativa de explicar o conceito de uma variável de uma forma intuitiva.

Foi entregue aos alunos, na forma impressa, uma atividade conforme modelo (Figura 47), adaptado do livro Introdução à Programação de computadores (SILVA, 1989).

**P:** “Suponhamos um armário dividido em portinhas numeradas, e que cada portinha poderá conter uma instrução ou valor numérico e consideremos uma pilha de cartões, em que cada cartão terá escrito um valor numérico que será utilizado durante a execução da tarefa”.

Figura 47 - Atividade sobre o conceito de variáveis.

			
<b>1</b> Coloque o valor zero na porta 10	<b>2</b> Coloque o valor zero na porta 11	<b>3</b> Pegue o 1º cartão da pilha e copie seu valor para a porta 12	<b>4</b> Some o valor que está na porta 10 com o que está na porta 12 e o resultado coloque na porta 10
<b>5</b> Some 1 ao valor que está na porta 11 e coloque o resultado na porta 11	<b>6</b> Se o valor que está na porta 11 for menor que 3, vá para a porta 3 e continue deste ponto. Caso contrário para próximo (7)	<b>7</b> Divida o valor que esta na porta 10 pelo valor que esta na porta 11 e coloque o resultado na porta 10.	<b>8</b> Copie o valor que está na porta 10 para apresentar ao grupo.
<b>9</b> Encerre o serviço.	<b>10</b>	<b>11</b>	<b>12</b>

Fonte: Adaptada de Silva (1989).

Durante a execução dessa atividade introdutória, realizamos alguns recortes. Vejamos as interações no Quadro 29:

Quadro 29 - Interações na execução do desafio.

<b>KA</b>	“Professor, quando eu coloco na portinha apaga o valor que tá lá?”.
-----------	---

<b>P</b>	<i>“Sim. O valor anterior não está mais armazenado”.</i>
<b>VH</b>	<i>“Professor três é menor que três?”.</i>
<b>P</b>	<i>“O que vocês acham?”.</i>
<b>CA</b>	<i>“É igual, não é maior e nem menor!”.</i>
<b>P</b>	<i>“Então ele atende ou não a condição?”.</i>
<b>CA</b>	<i>“Não atende, vai para a portinha 7. É dez! não é?”.</i>
<b>P</b>	<i>“Vocês perceberam que na portinha 11 há uma condição, quando é que vai sair e vai para o próximo?”.</i>
<b>NB</b>	<i>“Enquanto era menor que 3 ficava voltando lá no 3, tudo de novo”.</i>
<b>P</b>	<i>“Olha só que interessante, enquanto o valor na portinha 11 fosse menor do que três, ficaria nesse laço. Percebem que ele saiu desse laço quando não atendeu mais a condição. Então agora me respondam: Quais dessas instruções são de atribuição, entrada de dados, decisão, saída, e parada?”.</i>
<b>MB</b>	<i>“Parada é o 9 e decisão é o 6, agora os outros tem que ver!”.</i>
<b>MB</b>	<i>“No 1 e 2 está atribuindo o valor zero. Não entendo pra que atribuir zero. Zero é nada, então não precisava”.</i>
<b>TS</b>	<i>“Lá na portinha 7 tá fazendo uma divisão e guardando na 10. Então tá atribuindo também”.</i>
<b>CA</b>	<i>“Agora só restou a 3 que é a entrada de dados, que vem da pilha. E a saída é pra gente mostrar o resultado”.</i>

Fonte: Primária (2015).

O aluno **VH** apresentava dúvidas em relação à condição proposta, envolvendo equivalência e desigualdade numérica, que foi elucidada durante a atividade. O grupo reconheceu de forma fácil o laço condicional apresentado. O aluno **MB** achava desnecessário atribuir o valor zero.

Foi explicado que no Processing 2 as variáveis podem ser de diferentes tipos e somos obrigados a definir explicitamente o tipo de valor antes de utilizar uma determinada variável.

O aluno **CA** indagou: *“variáveis deve ter um tipo. Por quê?”.*

**P:** *É a forma como o computador sabe exatamente quanta memória deve ser alocada para armazenar dados.*

Nessa oficina são apresentadas ao grupo somente as variáveis do tipo int (número inteiro) e float (número decimal). Dois tipos de declarar uma variável e atribuir um valor foi abordado; como declarar e inicializar uma variável em duas linhas de código ou como combinar as duas linhas em uma única.

Nota-se que nas atividades dessa oficina o grupo optou pela forma mais sucinta, apesar de que a aluna **NB** preferiu fazer com duas linhas.

Observe algumas dúvidas acerca das variáveis apresentadas pelos estudantes (Quadro 30).

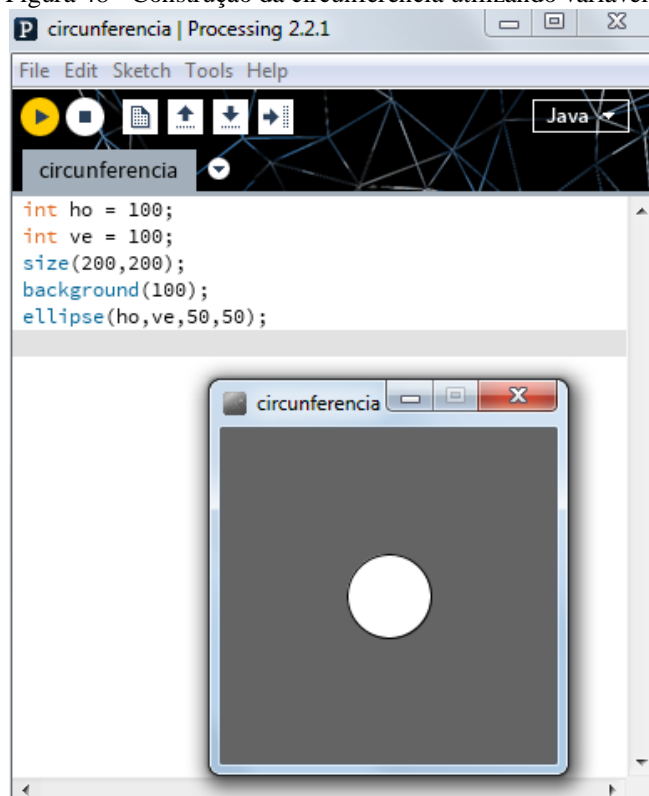
Quadro 30 - Dúvidas envolvendo variáveis

<b>NB</b>	<i>“Pode ser qualquer nome, professor?”.</i>
<b>P</b>	<i>“Evite o uso de palavras que aparecem em outras partes da linguagem Processing.”</i>
<b>VH</b>	<i>“Como assim?”</i>
<b>P</b>	<i>“Não chame uma variável de size pois é um comando da linguagem. Já existe um! E outra, use nomes que significam algo. Parece óbvio, mas é importante. Exemplo, se você estiver usando uma variável para manter o controle da pontuação de um jogo, chamá-lo de placar é mais significativo do que você chamar de número”.</i>
<b>KA</b>	<i>“Professor, mas quando eu devo usar uma variável, não consigo saber”.</i>
<b>P</b>	<i>“Pessoalmente, gostaria de dizer se um número aparece uma vez no programa, use uma variável. Sempre use variáveis!”.</i>

Fonte: Primária (2015).

Foi proposto um desafio para que se construísse um círculo de raio 50 de forma que as coordenadas centrais pudessem ser alteradas nas variáveis criadas. No desafio o aluno **CA** ficou auxiliando a aluna **NB**, discutindo suas dúvidas. Era notório o domínio e a clareza com que o aluno explicava aos amigos. A solução ao desafio veio rapidamente, com o código (Figura 48) apresentado pela a aluna **NB**.

Figura 48 - Construção da circunferência utilizando variáveis.



Fonte: Primária (2015).

Ao analisar o código da aluna questionei o motivo pelo qual chamaram as variáveis de “ho” e “ve”. A aluna alegou “*Eu coloquei horizontal e vertical abreviado*”. Percebemos uma coerência agregada ao nome da variável utilizada pela aluna **NB**, porém sugerimos ao grupo que procurassem se referir ao eixo horizontal de x e ao vertical de y.

Os alunos realizaram vários testes, alterando o valor atribuído à variável, para que o círculo se movimentasse na tela. Logo o aluno **MB** pergunta: “*Professor como faço pra movimentar o círculo com o mouse na tela?*”.

Nesse momento há um interesse em colocar movimentação. Os estudantes são levados a partir da necessidade de fazer um objeto (círculo a mover na área do sketch). Essa motivação é importantíssima em aprendizagem. O aluno é levado a pensar logicamente como realizar a movimentação desse círculo pela área do Sketch e só depois implementar em uma linguagem de programação. Assim a ferramenta passa a ser apenas um suporte.

O aluno **MB** mostra para a turma o que ele andou pesquisando na Internet (comunidades) sobre o Processing 2 (Quadro 31).

Quadro 31 - Pesquisa dos alunos na Internet.

<b>NB</b>	<i>“Pode ser qualquer nome, professor?”</i>
<b>MB</b>	<i>“Professor, eu pesquisei como que faz capturar a posição da coordenada por meio do mouse”.</i>
<b>P</b>	<i>“Legal, acredito que vamos precisar disso. Vamos por parte pra gente não se perder.”</i>
<b>VH</b>	<i>“Eu baixei o Processing!”</i>

Fonte: Primária (2015).

A aprendizagem situada conforme apresentada por Wenger e Lave (1991) prevê trajetórias de participação na comunidade. Impressionante que todos os estudantes manifestaram que tinham realizado download da ferramenta em seus computadores pessoais. Quando o aluno **MB** indagou anteriormente como se fazia para movimentar o círculo com o mouse, o mesmo já sabia que tinha que capturar as coordenadas x,y da posição mouse para poder realizar tal movimentação, apenas não sabia como fazer isso em Processing 2.

Com base no desafio anterior, propusemos um novo desafio que consistia em colocar 2 circunferências seguidas, que se encontrassem num único ponto.

Para realizar o presente desafio os alunos teriam que pensar na divisão da área do sketch de forma a solucionar o desafio, observando as coordenadas, raio, e diâmetro estipulados por eles.

A aluna **TS** quando fez sua segunda circunferência, disse: *“Professor, vimos isso essa semana na aula de matemática. A professora disse que isso aqui são duas circunferências tangentes, porque ela se encontra só nesse ponto”*. Percebemos que a aluna conseguiu estabelecer relação com aquilo que aprendeu em sala de aula e mostrava que estava dominando o conceito de tangência.

O aluno **VH** apresentava dificuldades e não conseguiu terminar o desafio proposto. A aluna **NB**, ofereceu ajuda, mas também não conseguiu localizar o erro, e logo o aluno **MB** veio ajudá-lo. Assim solicitou-se para que o aluno salvasse o programa.

Figura 49 - Erro no programa do aluno VH.



Fonte: Primária (2015).

O aluno reaproveitou seu código da primeira construção do círculo. Percebeu-se que o mesmo não atentou que manteve as mesmas variáveis (Figura 49) e com isso a ferramenta emitiu a seguinte mensagem *“duplicate variable local xcentro”*.

Nesse tipo de interação não houve em nenhum momento a rotulação que nós, enquanto professores, ouvimos em algumas ocasiões, de que esse aluno é “fraco” por não ter conseguido solucionar o desafio. Até mesmo entre eles, essa rotulação acontece e essa cultura diferenciada com o uso da ferramenta encoraja os estudantes, e possibilita estimular o sucesso uns dos outros de forma deliberada. Nas oficinas, os estudantes reconhecem que parte do seu propósito é certificar-se de que todos os seus amigos tenham sucesso na resolução do desafio.

Neste momento da oficina retorna-se ao trabalho com expressão booleana (condicionais), permitindo que o nosso trabalho tome caminhos diferentes, dependendo do valor atual armazenado na variável. Quando mostrado no quadro a expressão  $x > 5$  me dá um aperto de mão, obtiveram-se as seguintes interações (Quadro 32).

Quadro 32 - Questionamentos sobre condicional.

<b>VH</b>	<i>“Quanto vale o x. Se eu soubesse te daria um aperto de mão”.</i>
<b>MB</b>	<i>“Professor, condicional é sempre uma pergunta?”.</i>
<b>TS</b>	<i>“Professor e caso contrário, o que acontece?”</i>
<b>P</b>	<i>“A estrutura pode ser expandida com parte falsa. Quando a aluna disse “caso contrário” ela estava se referindo a parte falsa da expressão booleana, caso não atendida a condição inicial”.</i>
<b>MB</b>	<i>“Com a condicional somos nós que controlamos o programa para fazer alguma coisa quando aquilo é verdade”.</i>
<b>CA</b>	<i>“Professor como faço para comparar dois valores, tipo se a variável x é igual a y. Porque eu usei o sinal de igual e está dando erro”.</i>
<b>P</b>	<i>“Use dois sinais de iguais para indicar a comparação”.</i>

Fonte: Primária (2015).

O questionamento do aluno **MB** levou o mediador a refletir que realmente as expressões booleanas operam dentro do programa como perguntas. Em seguida foi apresentada aos alunos a sintaxe e estrutura da expressão booleana (condicional).

A aluna **NB** demonstrava não compreender muito bem estrutura, mas os alunos **CA** e **MB** estavam testando a lógica e se depararam com um problema.

Os alunos tinham a necessidade de comparar o conteúdo de duas variáveis, porém não sabiam como fazer isso em Processing 2. Nesse momento a construção do conhecimento parte da necessidade, conforme os pressupostos de Lave e Wenger (1991).

Antes de partir para outro desafio mais elaborado foi apresentada uma estrutura em pseudocódigo<sup>30</sup> e em seguida, o desafio constituía em implementá-lo em Processing 2.

**P:** *“Qual será o resultado da execução?”.*

**CA:** *“Isso é fácil, nem tem graça de fazer!”.*

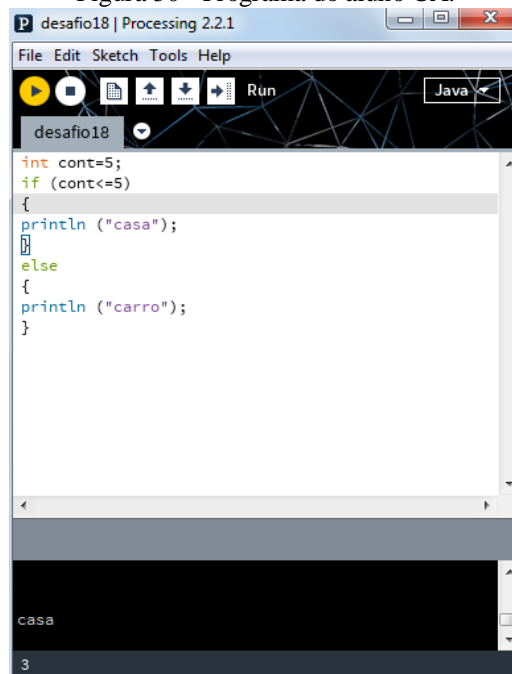
Percebe-se que quando o aluno detém muita habilidade e o desafio contempla pouca dificuldade, o aluno fica “entediado”, quando expressa: “não tem graça de fazer” conforme elencado por Csikszentmihalyi (2009). Na construção do programa a aluna **KA** pediu para não explicar em pseudocódigo, uma vez que confundiu as estruturas.

**KA:** *“Professor quando o senhor explica usando o mesmo código eu acho bem mais fácil de fazer”.* Os alunos apresentaram dificuldades na estruturação do código, principalmente no que se refere a abertura e fechamento das chaves. O trabalho com pseudocódigo apresentou-se nesse momento como elemento dificultador, o que levou este pesquisador a não utilizá-lo em demonstrações futuras.

<sup>30</sup> Pseudocódigo é uma forma genérica de escrever um algoritmo, utilizando uma linguagem nativa, que pode ser entendida facilmente, sem a necessidade de conhecer a sintaxe de uma linguagem de programação.

O aluno **CA** foi o primeiro a apresentar o código completo e funcional (Figura 50). Os alunos **CA** e **MB** posteriormente construíram juntos outro programa envolvendo condicional, com o operador de igualdade (Figura 51).

Figura 50 - Programa do aluno CA.

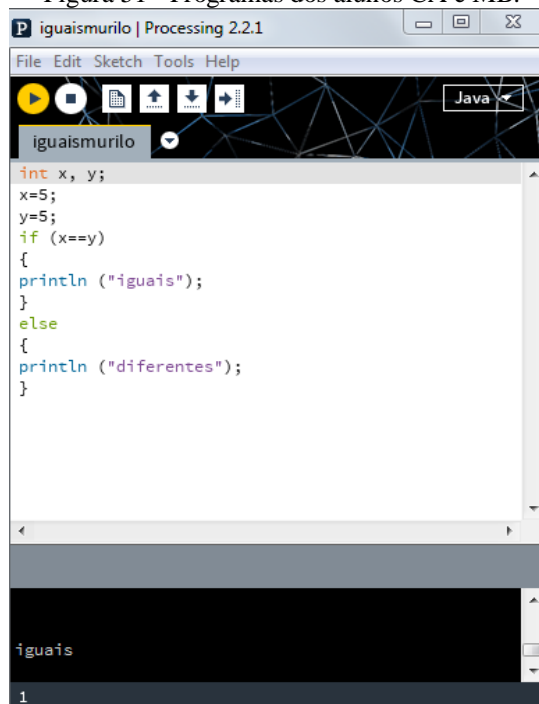


```
desafio18 | Processing 2.2.1
File Edit Sketch Tools Help
desafio18
int cont=5;
if (cont<=5)
{
println ("casa");
}
else
{
println ("carro");
}

casa
3
```

Fonte: Primária (2015).

Figura 51 - Programas dos alunos CA e MB.



```
iguaismurilo | Processing 2.2.1
File Edit Sketch Tools Help
iguaismurilo
int x, y;
x=5;
y=5;
if (x==y)
{
println ("iguais");
}
else
{
println ("diferentes");
}

iguais
1
```

Fonte: Primária (2015).

Os alunos **CA** e **MB** queriam fazer a “pergunta” diferente. Ao invés de perguntarem se os valores são iguais, eles queriam descobrir como era o operador que representa o diferente.

**CA:** “*Professor, mas se eu quiser fazer a pergunta se eles são diferentes eu posso, porque o resultado vai ser o mesmo, só vai inverter!*”. Percebemos a visão antecipada do resultado. O trabalho com o Processing 2, veio contribuir para um pensamento de estimativa de resultados e previsão.

O último desafio dessa oficina constituía em elaborar um programa que dependesse do valor de uma variável  $x$ . Se esse valor fosse maior que 100 e menor que 300, se desenharia na tela uma elipse qualquer. Caso contrário (fosse um valor maior que 100 e maior que 300) se desenharia uma linha qualquer. Caso contrário (fosse menor que 100) se desenharia um quadrado qualquer.

O aluno **VH** quando estava trabalhando na construção de um quadrado levantou um questionamento que desencadeou uma discussão interessante (Quadro 33).

Quadro 33 - Discussão sobre o comando `rect`.

<b>VH</b>	“ <i>Esse comando deveria chamar outro nome e não <code>rect</code></i> ”.
<b>MB</b>	“ <i>Deve ser porque todo quadrado é um retângulo</i> ”.
<b>P</b>	“ <i>Não só retângulo, mas também losango, paralelogramo, e trapézio</i> ”.
<b>TS</b>	“ <i>Nós aprendemos isso em matemática, ano passado, a professora ensinou!</i> ”.
<b>MB</b>	“ <i>Quadriláteros notáveis</i> ”.
<b>P</b>	“ <i>O que é um quadrilátero notável?</i> ”.

Fonte: Primária (2015).

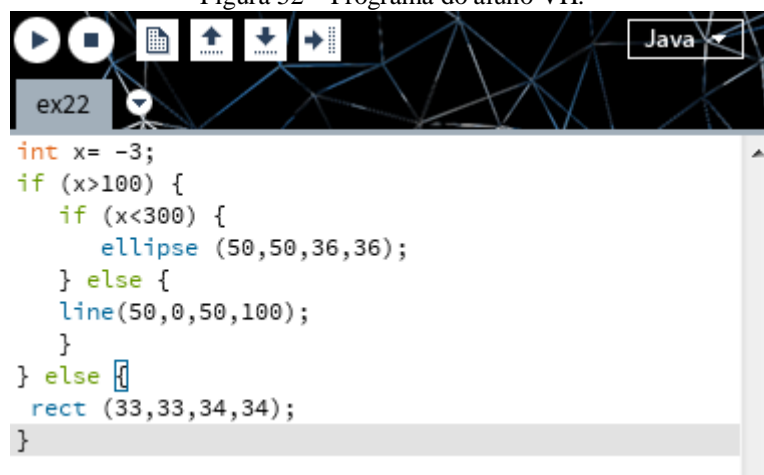
O aluno se referia ao conceito de retângulo, pois possui quatro ângulos retos, assim concluiu que quadrado também é um retângulo. A aluna **TS**, recordou que a professora do ano anterior tinha trabalhado essa habilidade, porém o aluno **MB** relacionou ao assunto quadrilátero notável. Quando questionados sobre o que era um quadrilátero notável ninguém soube explicar.

O aluno **VH** conseguiu apresentar o seguinte código (Figura 47) e fazia consecutivos testes para testar as condições. **VH:** “*Professor, olha o meu programa ficou diferente de todos. Já testei vários valores*”.

Para tal, o aluno variava o valor de  $x$ . O procedimento demonstra o domínio da habilidade de relação, em função da variável  $x$ . Quando o aluno alterava o valor de  $x$ , sabia que  $x$  poderia assumir qualquer valor inteiro, por ter declarado o tipo da variável como inteiro.

O grupo ainda não havia iniciado o estudo da álgebra em sala de aula. Este trabalho contribuiu para o entendimento da álgebra, o uso de letras representando números. A compreensão de generalizações associadas ao uso de letras (variáveis no caso dos programas) é habilidade elementar dentro do curso de matemática no Ensino Fundamental.

Figura 52 – Programa do aluno VH.



```
int x= -3;
if (x>100) {
    if (x<300) {
        ellipse (50,50,36,36);
    } else {
        line(50,0,50,100);
    }
} else {
    rect (33,33,34,34);
}
```

Fonte: Primária (2015).

No programa (Figura 52) o aluno faz teste com valores negativos, demonstrando domínio do conjunto numérico trabalhado.

Ao término da oficina a aluna **TS** perguntou se as variáveis podem ser declaradas em qualquer lugar dentro de um programa. Então, aproveitamos a oportunidade para explicar os blocos de comandos `setup ()` e `draw ()`. A seção *setup* é utilizada para definir propriedades iniciais de ambiente (por exemplo, o tamanho da janela, cor de fundo, etc.) e a seção *draw* os comandos de desenho (por exemplo, ponto, linha, elipse, etc.).

No Processing, as variáveis globais são declaradas na parte superior do programa, fora dos blocos `setup ()` e `draw ()`. As variáveis locais são variáveis declaradas dentro de um bloco de código. Uma variável local declarada dentro de um bloco de código está disponível apenas para uso dentro desse bloco específico de código onde foi declarada.

Quadro 34- Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 3.

<b>Peculiaridade</b>	<b>Fala dos alunos</b>	<b>Como foi observado</b>
Não linearidade	<b>CA:</b> “ <i>Professor, mas se eu quiser fazer a pergunta se eles são diferentes eu posso, porque o resultado vai ser o mesmo, só vai inverter!</i> ”.	Quando os aprendizes são deixados livres para explorar um espaço de problemas complexos, como ocorre em ambientes permissivos em que se podem executar, eles tendem a alcançar soluções criativas para problemas complexos.
Ênfase no “saber fazer”	<b>MB:</b> “ <i>Com a condicional somos nós que controlamos o programa para fazer alguma coisa quando aquilo é verdade</i> ”.	Os estudantes têm uma real sensação de controle sobre o que fazem. Eles têm um verdadeiro sentido de propriedade em relação ao que estão fazendo, um sentido que é raro na aula tradicional.
Informal	<b>VH:</b> “ <i>Professor, olha o meu programa ficou diferente de todos. Já testei vários valores</i> ”.	Os estudantes em geral podem customizar seus programas para que eles se ajustem aos seus estilos. Assim, percebemos que o método favorece um ritmo próprio de aprendizagem, com intersecções profundas entre o PPP e os interesses, desejos e estilos dos aprendizes.
Interatividade	<b>CA:</b> “ <i>Isso é fácil, nem tem graça de fazer!</i> ”.	Os desafios são percebidos como factíveis, mas ao mesmo tempo desafiadores. Este é um estado altamente motivador para os aprendizes. Muitas vezes a aula é fácil demais para alguns estudantes e difícil demais para outros.
Engajamento	<b>MB:</b> “ <i>Deixa que eu te explico, estou terminando</i> ”	Essa cultura diferenciada com o uso da ferramenta encoraja os estudantes, e possibilita estimular o sucesso uns dos outros de forma deliberada. Nas oficinas, os estudantes reconhecem que parte do seu propósito é certificar-se de que todos os seus amigos tenham sucesso na resolução do desafio.

Fonte: Autor (2015).

Nesta oficina foi possível explorar de forma situada o reconhecimento dos números naturais e ou decimais na declaração das variáveis, comparações numéricas, desigualdades e principalmente o principio de relação.

#### 4.2.4 Oficina 4 – Movimentação

A presente oficina buscou explorar a função `draw()`, que indica o que deve se repetir constantemente. No modo contínuo de execução, todo o código que o programador deseja repetir indefinidamente deve ser colocado dentro da função `draw()`. O código da função `draw()` é submetido num processo conhecido como “loop” e apenas será interrompido quando o usuário requerer. Por padrão, a taxa de repetição de um bloco da função `draw()`, também especificado de `frameRate`, é de 60 frames por segundo (fps). A função `draw` é chamada no máximo 60 vezes, sendo que este valor não é exato, pois, depende das configurações do hardware e do programa em execução.

A função `draw()` é fundamental para a nossa compreensão dos princípios do movimento de programação. Assim, o grupo estava ansioso por realizar a movimentação dos objetos que aprenderam a construir em Processing 2. Os alunos haviam criado um círculo e o desafio consistia em fazê-lo mover da esquerda para direita bem lentamente na área do sketch.

**MB:** *“Professor, para movimentar este círculo eu tenho que ir aumentando o valor da horizontal, mas eu não sei como fazer isso”.*

**P:** *“Como podemos fazer isso, lembrando que a função `draw` repete infinitamente”.*

**KA:** *“Mas então ficou fácil, é só a gente ir aumentando o valor do `x` de pouquinho, tipo 1”.* A aluna **KA** foi a primeira a apresentar o seguinte código (Figura 53) como solução para o desafio.

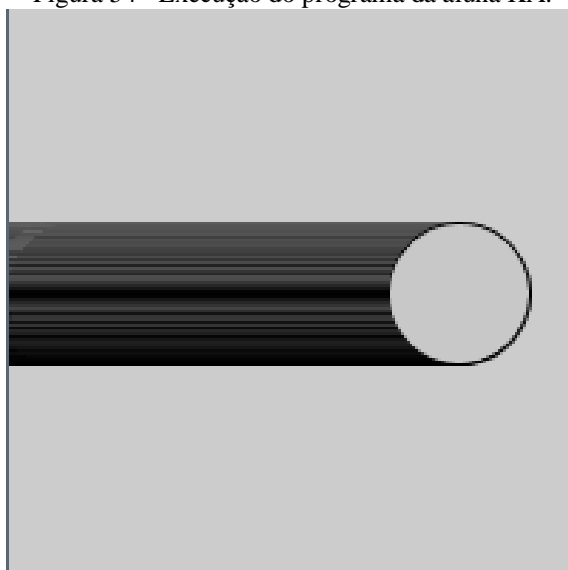
Figura 53 - Estratégia utilizada pela aluna KA no desafio.

```
int xcirculo=0;
int ycirculo=100;
void setup()
{
  size(200,200);
}
void draw()
{
  fill(200);
  ellipse(xcirculo,ycirculo,50,50);
  xcirculo = xcirculo+1;
}
```

Fonte: Primária (2015).

A aluna **KA** não obteve o resultado esperado na execução do programa o que a levou refletir sobre o que poderia estar errado em seu código. O grupo compartilhava da mesma problemática (Figura 54), focado no entendimento do “por que” não estavam obtendo êxito satisfatório.

Figura 54 - Execução do programa da aluna KA.



Fonte: Primária (2015).

**NB:** *“Mas professor, tá ficando o rastro porque a gente não tá limpando. Temos que fazer o fundo ficar branco de novo, porque isso aqui repete infinito não é assim?”.*

**MB:** *“Então é só colocar um background branco antes”.*

Era evidente na fala do aluno **MB** que sabia como fazer o descolamento do círculo quando ele diz “aumentar o valor da horizontal”, mas não sabia como implementar isso em Processing 2. Quando os alunos se depararam com o desafio, precisaram entender e compreender como funciona, quais conceitos estão envolvidos nessa construção, e a programação passa a ser um suporte. A pergunta posterior é como fazer isso em programação, assim, a ela é um suporte e isso foi notório na busca de construção de estratégias para solução dos desafios propostos.

Quando o aluno **MB** disse “background branco”, teve um insight e logo apresentou um código que atendia a sua meta e expressava satisfação. O grupo começou a aumentar a velocidade da movimentação, sem este mediador solicitar, demonstrando capacidade de criatividade e domínio do conceito. Assim, a expressão de novas ideias e úteis em uma situação particular contribuiu para o desenvolvimento da competência crítica, exigida dos estudantes nesses novos tempos.

A seguir, foi proposto um desafio em que, em vez de o círculo mover-se da esquerda para a direita, o círculo cresceria em tamanho. O que vocês mudariam? Como se pode variar a velocidade a que o círculo cresce?

**CA:** *“Professor posso fazer as duas ao mesmo tempo, andar e aumentar?”.*

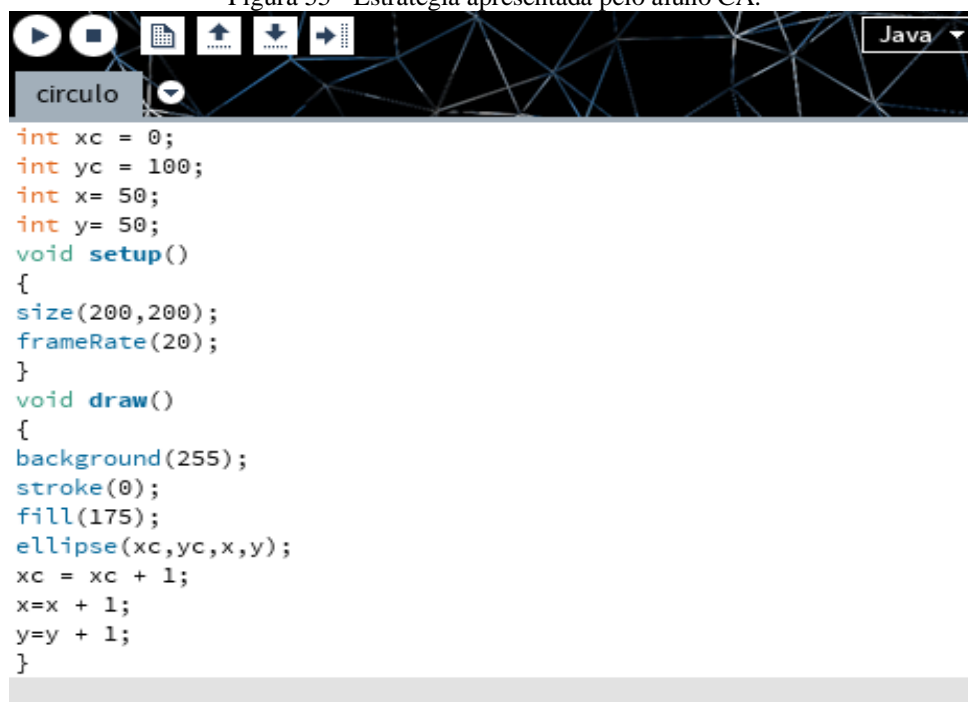
**P:** *“Sem problemas, o importante é que vocês alcancem o objetivo proposto. Não se esqueçam de que vocês têm que variar a velocidade do crescimento do círculo”.*

**MB:** “*Eu acho que eu sei fazer isso!*”.

O aluno **VH** encontrava dificuldade para alterar a dimensão do diâmetro do círculo, o mesmo só trabalhou com a dimensão horizontal e se esqueceu da dimensão vertical. Como feedback da execução do programa uma elipse deformada era apresentada. O aluno **MB** saiu da sua máquina e foi ajudá-lo a concluir o desafio. Mais uma vez evidenciamos a cultura de colaboração, que encoraja os estudantes, e possibilita estimular o sucesso uns dos outros de forma deliberada. Assim, os estudantes reconhecem que parte do seu propósito é certificar-se de que todos os seus amigos tenham sucesso na resolução do desafio.

Para a variação da velocidade, uma estratégia diferente das demais (Figura 55) chamou atenção. Ao invés do aluno aumentar o valor incremental das variáveis *x* e *y*, o aluno **CA** utilizou o comando `frameRate` para diminuir a frequência de 60 frames por segundo (padrão). Neste caso, a cada segundo, a função `draw()` é chamada no máximo 20 vezes. Interessante que o aluno quis inovar e mesclou movimentação horizontal do círculo, juntamente com aumento das dimensões do círculo. Quando os estudantes compartilham variadas estratégias isso contribui para aqueles estudantes que têm dificuldades em considerar ou enxergar outras técnicas e métodos. Assim, o trabalho no Processing 2 contribuiu nesse sentido.

Figura 55 - Estratégia apresentada pelo aluno CA.



```
circulo
int xc = 0;
int yc = 100;
int x= 50;
int y= 50;
void setup()
{
  size(200,200);
  frameRate(20);
}
void draw()
{
  background(255);
  stroke(0);
  fill(175);
  ellipse(xc,yc,x,y);
  xc = xc + 1;
  x=x + 1;
  y=y + 1;
}
```

Fonte: Primária (2015).

Quadro 35 - Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 4.

<b>Peculiaridade</b>	<b>Fala dos alunos</b>	<b>Como foi observado</b>
Não linearidade	<b>KA:</b> “ <i>Vou, começar montando a circunferência, depois eu penso como fazer andar</i> ”.	O conhecimento não ficou fragmentado em módulos, em que o professor esgota determinado conteúdo em um bloco e num próximo momento, aborda da mesma maneira um assunto totalmente novo.
Informal	<b>NB:</b> “ <i>É assim que faz MB?</i> ”.  <b>MB:</b> “ <i>Vou te ajudar!</i> ”.	O desprendimento do modelo tradicional de aula promoveu um relacionamento mais favorável à aprendizagem, com menos hierarquia dentro do ambiente de aprendizagem.
Construção do conhecimento parte da necessidade	<b>CA:</b> “ <i>Professor posso fazer as duas ao mesmo tempo, andar e aumentar?</i> ”.	Quando os alunos percebiam que estavam longe de seus objetivos, passavam a testar novas possibilidades. Nessa busca o interesse é despertado e o conhecimento emana da necessidade do aluno.
Interatividade	<b>CA:</b> “ <i>Roda de novo pra gente vê</i> ”.	Vivemos em tempos de retorno rápido e o conhecimento tem que ser capaz de se adaptar a um ritmo similar. A velocidade acelerada de mudança tem consequências diretas na prática educativa.
Engajamento	<b>TS:</b> “ <i>Vamos deixar mais rápido, sabe como faz? Vou explicar</i> ”.	Alguns alunos se destacavam como líderes. Uma comunidade de prática envolve a criação de presença social. Define a presença social como o grau de importância de uma pessoa numa interação e a sua importância numa relação interpessoal (Tu, 2002).

Fonte: Autor (2015).

#### 4.2.5 Oficina 5 - Variáveis do sistema

Nesta oficina iniciamos abordando as variáveis do sistema comumente usadas. Há muito mais, a que os alunos, posteriormente, podem recorrer à Referência do Processing 2 na web. Entregamos ao grupo o seguinte impresso (Quadro 36) no início da oficina e exemplificamos algumas situações simples.

Quadro 36 - Impresso sobre as variáveis do sistema.

<b>mouseX</b>	Coordenada horizontal atual do mouse.
<b>mouseY</b>	Coordenada vertical atual do mouse.
<b>Width</b>	Largura em pixels da janela do esboço.
<b>Height</b>	Altura em pixels da janela do esboço.
<b>Key</b>	Mostra a última tecla pressionada no teclado.
<b>keyCode</b>	Mostra o último número pressionado no teclado.
<b>keyPressed</b>	É uma tecla pressionada? (verdadeiro ou falso).
<b>mousePressed</b>	É o mouse pressionado? (verdadeiro ou falso)
<b>mouseButton</b>	Qual botão do mouse é pressionado? Esquerdo ou direito?

Fonte: Autor (2015).

Iniciando-se o estudo das variáveis de sistema **mouseX** e **mouseY** que contém a coordenada horizontal e vertical da posição atual do mouse, dúvidas surgiram (Quadro 37).

Quadro 37 - Dúvidas iniciais dos alunos sobre variáveis do sistema.

<b>TS</b>	<i>“Mas ele sabe a posição do mouse quando a setinha está em qualquer lugar da tela ou dentro do programa?”.</i>
<b>P</b>	<i>“Só na janela de execução do Processing, fora não!”.</i>
<b>MB</b>	<i>“Professor, mostra na prática fica muito mais fácil do que você ficar mostrando ai no quadro”.</i>
<b>VH</b>	<i>“Verdade professor”.</i>

Fonte: Primária (2015).

Era notável uma insatisfação dos alunos, toda vez que dirigíamos ao quadro para explicar um comando. Quando o aluno diz “ficar ai no quadro” ele está querendo dizer que o mediador estava “enrolando” a aula, e que diretamente no Processing 2 é mais interessante por ser visual. Sua fala foi ratificada pelo aluno **VH**, quando diz “verdade professor” o aluno espera algo mais interessante do que quadro e pincel.

Momentos expositivos são importantes, mas percebemos que os alunos demonstravam inquietudes toda vez que dirigíamos ao quadro com um pincel da mão, eles preferiam aprender na interação prática, e com recorrência aos amigos.

A seguir foi lançado um desafio que faz uso de algumas das variáveis contextualizadas na presente oficina.

**NB** *“Nossa professor a gente vai precisar usar tudo isso de uma vez!”.*

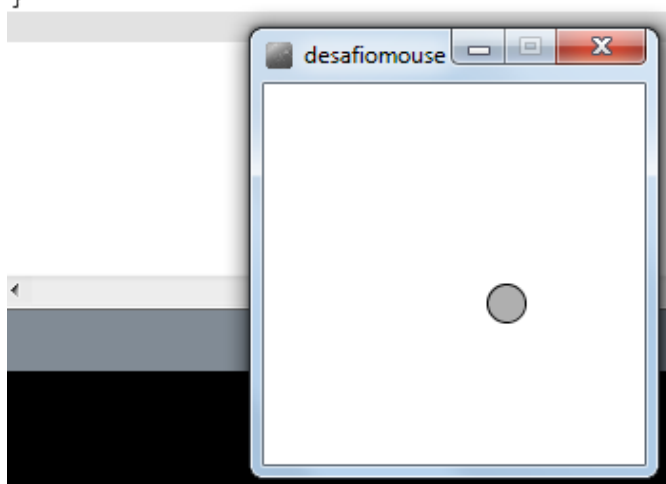
**P:** *“Nós não estamos prontos para usá-los todos ainda assim, vamos devagar, precisamos de alguns conceitos mais elaborados para fazermos uso”.*

Assim, foi proposto um desafio no qual uma circunferência acompanhasse o ponteiro do mouse por toda a janela.

Num primeiro momento os alunos não compreenderam o objetivo do desafio proposto. Elucidamos novamente o objetivo e os alunos partiram para construção. A aluna **TS** já demonstrava indícios de compreensão e liderava o grupo nesse momento apresentando uma estratégia coerente para solução do desafio (Figura 56).

Figura 56 - Código da aluna TS.

```
void setup() {
  size(200,200);
}
void draw() {
  background(255);
  fill(175);
  ellipse(mouseX,mouseY,20,20);
}
```



Fonte: Primária (2015).

O aluno **VH** consegue observar um fato implícito:

**VH:** “*Professor, sempre que a gente roda, a circunferência aparece no canto*”.

O aluno **VH** conseguiu perceber que as variáveis utilizadas são inicializadas com zero por padrão, por isso quando executado sempre desloca o ponteiro do mouse para a origem do plano.

Os alunos estavam discutindo sobre por que vemos o rastro da circunferência se movermos o background (255), deixando-o fora de draw().

Aproveitou-se da discussão, para encaminhar o desafio que ao ser pressionado o mouse construísse uma circunferência qualquer de centro (posição do mouse pressionado) e que apresentasse na área de mensagem as coordenadas x e y do centro da circunferência.

O desafio foi construído junto com os alunos no sentido de andaime. À medida que as habilidades vão sendo apropriadas pelos alunos, o mediador vai aumentando o andaime até que o estudante consiga proficiência na aplicação de um conhecimento já consolidado.

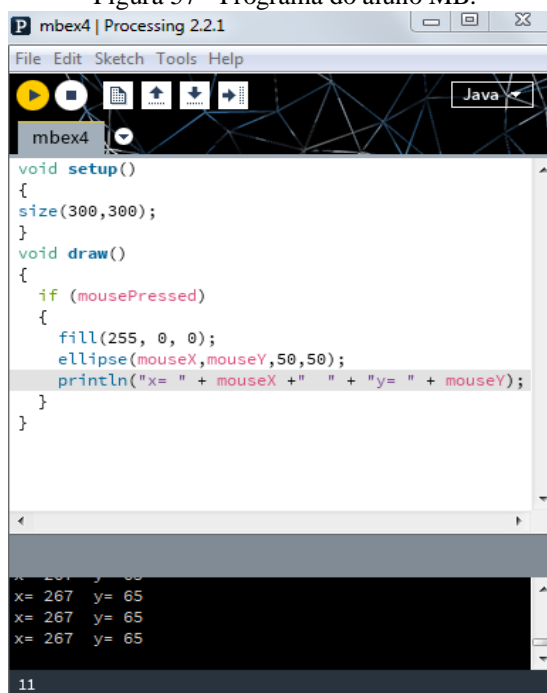
Os estudantes apresentaram dúvidas, em destaque na sintaxe do comando `println()`. Compreendiam que essa função é frequentemente útil para olhar para os dados que o

programa está produzindo (saída), mas apresentaram dificuldades em compreender que esta função pode receber mais do que um parâmetro e concatená-los com o operador (+).

**NB:** “*mouseX é uma variável?*”. Pela pergunta, a aluna ainda não tinha compreendido que estas são variáveis do sistema e que traziam o valor atual da coordenada do mouse.

O aluno **MB** apresentou uma codificação bem organizada (Figura 57) que atendeu ao propósito do desafio, seguida da sua execução (Figura 58).

Figura 57 - Programa do aluno MB.



```

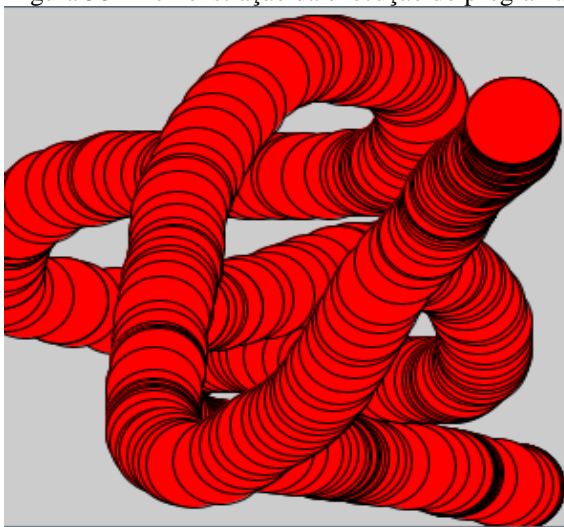
void setup()
{
  size(300,300);
}
void draw()
{
  if (mousePressed)
  {
    fill(255, 0, 0);
    ellipse(mouseX,mouseY,50,50);
    println("x= " + mouseX + " " + "y= " + mouseY);
  }
}

```

x= 267 y= 65  
x= 267 y= 65  
x= 267 y= 65

Fonte: Primária (2015).

Figura 58 - Demonstração da execução do programa.



Fonte: Autor (2015).

Para explorar as variáveis fomos resgatar o que os alunos traziam de conhecimento prévio sobre o conceito de aleatoriedade. Assim, iniciamos um trabalho para compreensão do conceito (Quadro 38).

Quadro 38 - Compreendendo a função random.

<b>P</b>	<i>Você poderia me dar um número aleatório entre 1 e 100?</i>
<b>CA</b>	<i>“54”.</i>
<b>P</b>	<i>“Então tenho que desenhar um retângulo 54 pixels de largura!”</i>
<b>MB</b>	<i>“Vai ser um número qualquer que o computador vai colocar”.</i>
<b>CA</b>	<i>“Como ele faz isso professor?”</i>
<b>P</b>	<i>“O Processing possui uma função que se encarrega de fazer isso”</i>

Fonte: Primária (2015).

Percebemos que no questionamento do aluno **CA** em querer entender o “porquê” das coisas, não aceita o conhecimento como algo pronto e acabado. Notamos que a dinâmica das oficinas favorecia esse tipo de postura.

Apresentamos no quadro a seguinte sintaxe aos alunos:

```
float x = random (500);
rect (100,100, x, 50);
```

Antes mesmo que começasse, o aluno **MB** questionou o motivo de ter criado uma variável do tipo float.

**P:** *“Professor, é só número quebrado? Eu estava pensando em número sem vírgula”.*

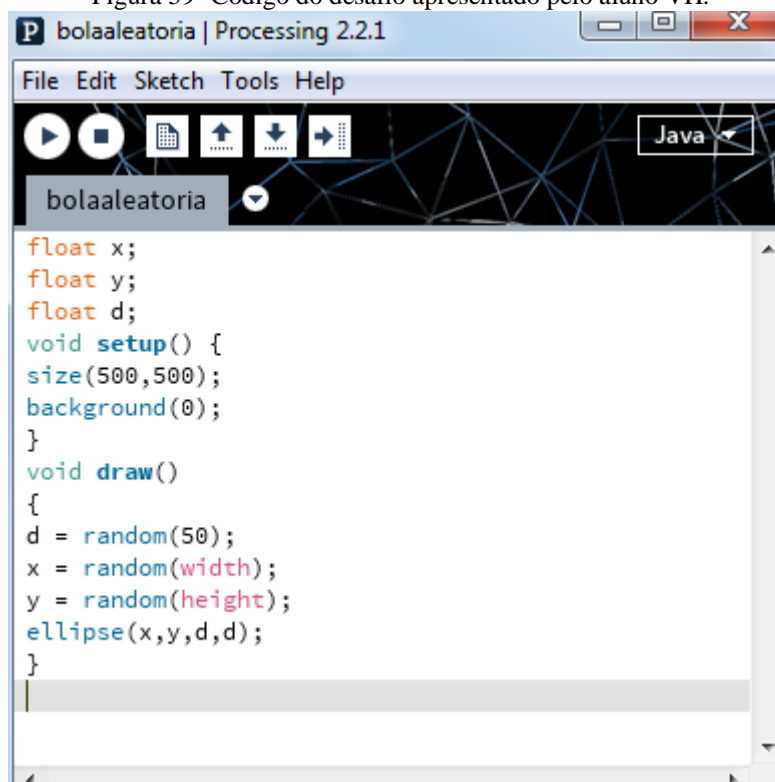
A fala do aluno indica que ele estava pensando em atribuir números naturais, quando ele diz número sem vírgula. Assim, ele queria saber se daria para utilizar números inteiros. Com isso, discutimos a sintaxe que possibilita converter o resultado da função de reprodução aleatória para um int.

```
int x = int (random (100));
rect (100,100, x, 50);
```

**TS:** *“Professor, os parênteses precisam ser colocados daquele jeito? Eu fico confusa!”.*

A dificuldade de trabalhar com chaves e parênteses foi uma característica peculiar do grupo. Assim, quando percebemos que os alunos estavam prontos para testar a função random(), propusemos a construção de uma elipse de localização (máximo possível, ou seja qualquer ponto da janela) e diâmetro aleatório (máximo 50). Como solução do desafio **VH** apresentou o programa (Figura 59).

Figura 59- Código do desafio apresentado pelo aluno VH.



```

float x;
float y;
float d;
void setup() {
  size(500,500);
  background(0);
}
void draw()
{
  d = random(50);
  x = random(width);
  y = random(height);
  ellipse(x,y,d,d);
}

```

Fonte: Primária (2015).

Para aproveitar o mesmo desafio solicitou-se aos alunos que colocassem cor (aleatória) no preenchimento das elipses. Surge a pergunta, como fazer isso? Os alunos se deparam com um desafio e dizem não saber como proceder (Quadro 39).

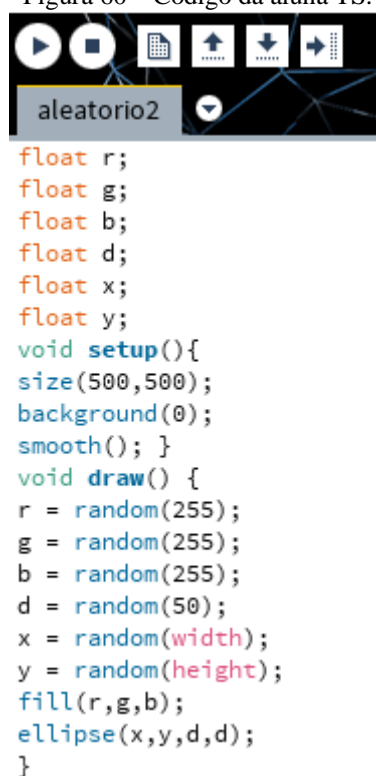
Quadro 39 - Dúvidas apresentadas no desafio das elipses.

<b>P</b>	<i>“Vocês tem que pensar no padrão de cor RGB e no comando Fill para preenchimento”.</i>
<b>KA</b>	<i>“O máximo do RGB é 255? Então vamos criar três variáveis aleatórias pra cada cor?”</i>
<b>NB</b>	<i>“Eu não entendi”.</i>

Fonte: Primária (2015).

A aluna **NB** observava a construção da aluna **KA** e tentava entender a estratégia empregada pela amiga. O aluno **VH** apresentou uma solução para o desafio, porém as elipses não apresentavam as mesmas dimensões para a horizontal e vertical. O mesmo criou duas variáveis, uma para cada dimensão da elipse, percebendo seu erro com a execução do programa. A seguir a codificação realizada pela aluna **TS** (Figura 60), apresentada como solução ao desafio.

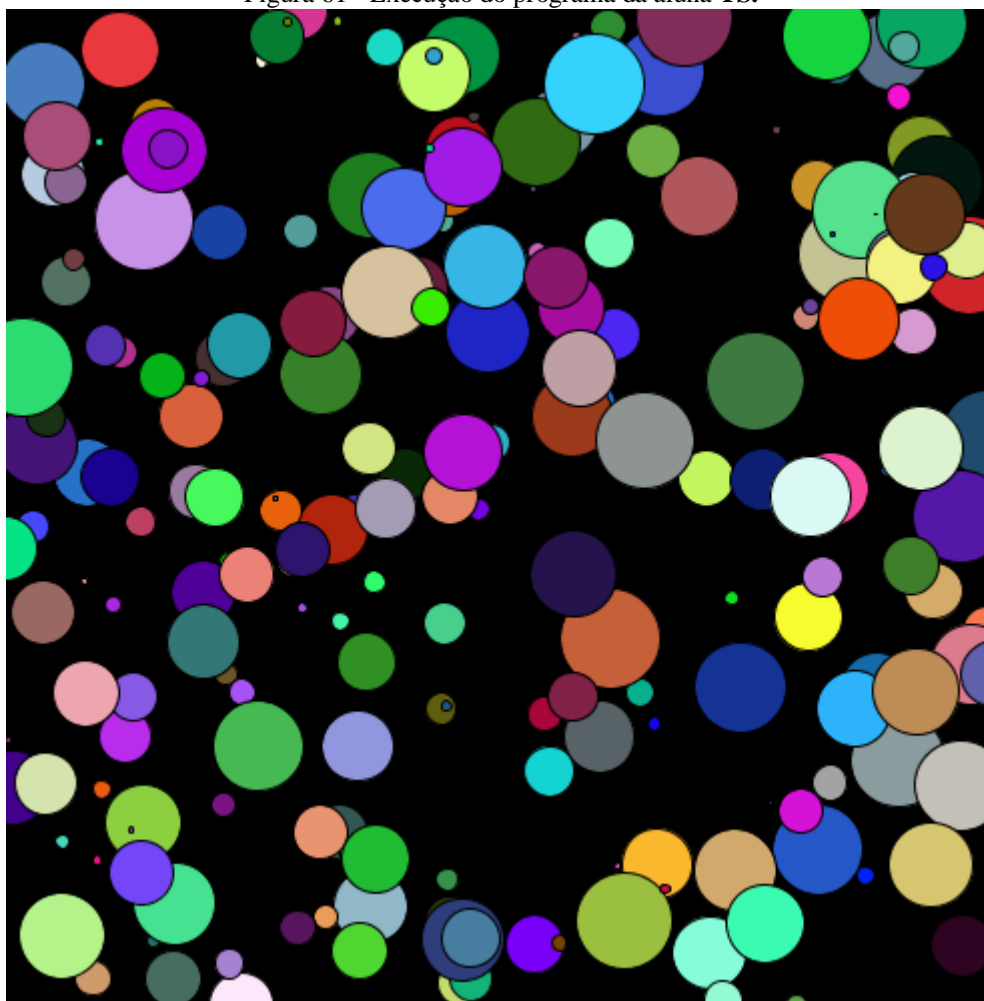
Figura 60 – Código da aluna TS.

A screenshot of a code editor interface. At the top, there is a toolbar with icons for play, stop, refresh, and other functions. Below the toolbar, the text 'aleatorio2' is displayed in a grey box. The main area contains the following code:

```
float r;  
float g;  
float b;  
float d;  
float x;  
float y;  
void setup(){  
  size(500,500);  
  background(0);  
  smooth(); }  
void draw() {  
  r = random(255);  
  g = random(255);  
  b = random(255);  
  d = random(50);  
  x = random(width);  
  y = random(height);  
  fill(r,g,b);  
  ellipse(x,y,d,d);  
}
```

Fonte: Primária (2015).

Ao término desse desafio o resultado foi surpreendente (Figura 61). Os alunos adoraram o resultado final. Pudemos evidenciar momentos de diversão e olhares de encantamento com o resultado de seus programas, expressos no discurso da aluna **TS** “*Ficou lindo, quero fazer mais deste tipo*”. Os alunos estavam se divertindo enquanto criavam seus programas coloridos e dinâmicos, aumentando o fator motivacional.

Figura 61 - Execução do programa da aluna **TS**.

Fonte: Autor (2015).

Observando a execução de seu programa a aluna **NB** diz: *“isso não tem fim!”*.

A aluna refere-se que ao executar o programa, as elipses só aumentavam na janela, e após deixar executando o programa por um período de tempo, o aluno **VH**, completou *“...é porque o código está dentro do draw”*.

Encaminha-se para o final de mais uma oficina, em que a matemática foi sendo percebida direta e indiretamente por meio dessas ações de interação, colaboração, motivação dos alunos num processo social, em consonância com os pressupostos apresentados por Lave e Wenger (1991).

Quadro 40- Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 5.

<b>Peculiaridade</b>	<b>Fala dos alunos</b>	<b>Como foi observado</b>
Não linearidade	<b>NB</b> <i>“Nossa professor a gente vai precisar usar tudo isso de uma vez!”.</i>	O método veio contribuir para superar as fragmentações curriculares, por meio da valorização de uma visão holística, que conecta as partes, possibilitando ao aluno permear e relacionar as diferentes habilidades apropriadas.
Ênfase no “saber fazer”	<b>MB:</b> <i>“Professor, mostra na prática fica muito, mais fácil do que você ficar mostrando ai no quadro”.</i>	A Lei de Diretrizes e Bases da Educação Nacional (LDB 9394/96) faz referência ao trabalho, simultaneamente com as práticas sociais, como elemento que relaciona o ensino à prática profissional. O método minimizou a distância entre saber e fazer. A compreensão dos fundamentos científicos e tecnológicos oportunizou relacionar o currículo ao mundo da produção de bens e serviços, envolvidos numa prática profissional, o que possibilitou aos alunos verem sentido entre o que a escola ensina e o mundo fora dela, e isso foi possível com o fazer programação.
Informal	<b>NB</b> <i>“Este programa ficou a minha cara”</i>	O computador torna possível a criação de micromundos que podem transformar o processo educacional numa forma mais natural e espontânea de aprender. Dentro deste micromundo, uma criança pode explorar através da programação a sua imaginação. Assim, evidenciamos que a aprendizagem situada contribui para a criação de micromundo. A interface se transforma num mundo no qual os estudantes interagem a todo instante e esse mundo se comporta de acordo com as ações dos estudantes.
Engajamento	<b>TS:</b> <i>“Ficou lindo, quero fazer mais desse tipo”</i>	Nenhuma aprendizagem ocorre se os estudantes não firmarem um compromisso com ela. Aprender alguma coisa em um novo campo, como em programação, requer que o aluno assuma uma nova identidade, assumo o compromisso de ver e de valorizar o trabalho e o mundo da forma como fazem os programadores. Este compromisso foi aumentando gradualmente a cada oficina.

Fonte: Autor (2015).

#### 4.2.6 Oficina 6 - Operadores lógicos

A presente oficina iniciou-se com um programa (Figura 62) desenvolvido pelo aluno CA em sua casa e apresentou um plano de fundo contendo uma imagem e todos se levantaram e se dirigiram para sua máquina (Figura 63).

O aluno explicava aos amigos que copiou a imagem para dentro da pasta gerada quando salvou o programa e alterou o caminho da imagem e compartilhava o erro que apareceu e como ele conseguiu resolvê-lo.

Figura 62 - Código do programa do aluno CA.

```
size(1280,853);
PImage img;
img = loadImage("mb002.jpg");
background(img);
```

Fonte: Primária (2015).

Figura 63 - Execução do programa do aluno CA.



Fonte: Primária (2015).

**P:** “Então você curte carros importados?”.

**CA:** “Tenho essa imagem de plano de fundo no meu celular, tenho muita foto aqui, quer ver?”.

Assim, percebe-se que o aluno é um apaixonado por carros importados e traz essa paixão embutida em suas criações em Processing.

Os alunos começam a testar a novidade apresentada pelo amigo, colocando uma imagem como plano de fundo, assim como o amigo tinha realizado. A aluna **TS** não estava conseguindo, e não hesitou em pedir auxílio ao aluno **CA**. Os estudantes se depararam com a mensagem de erro “*background image must be the same size as your application*”, o mesmo erro que o aluno **CA** tinha superado e, portanto pôde compartilhar seus conhecimentos tácitos empregados na tarefa.

Os alunos não entendem ao certo a estrutura do bloco de comando para trabalhar com imagem, e aproveitando o direcionamento, resolvemos explorar no quadro cada detalhe dos

comandos. O Processing 2 pode carregar imagens GIF, JPEG e PNG. (Ressalta-se que essas extensões de arquivos de imagem já eram conhecidas por todos os alunos do grupo).

Da mesma forma que números inteiros podem ser armazenados em variáveis do tipo `int`, as imagens podem ser armazenadas em variáveis do tipo **PImage**. Antes de ser mostrada na tela, uma imagem deve ser previamente carregada com a função **loadImage()**. A função **image()** mostra a imagem armazenada na variável. O arquivo da imagem deve se encontrar na pasta.

Após o episódio inicial, voltamos ao foco da oficina com alguns exemplos.

**P:** “*Se o mouse está no lado esquerdo da tela e o mouse está na parte inferior da tela, desenhar uma elipse na parte inferior esquerda*”.

Abordou-se que questões como esta podem ser realizadas de uma forma mais simples usando o operador lógico AND representado por (&&). Assim, foram apresentadas ao grupo os operadores lógicos || (OR), && (AND) e ! (NOT) e a tabela verdade de cada um deles (Quadro 41).

Quadro 41- Tabela verdade AND e OR.

A	B	A AND B	A OR B
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F

Fonte: Autor (2015).

Inicialmente os alunos tiveram dificuldades em encontrar no teclado as duas barras verticais "||" que geralmente no teclado são a segunda função (shift-barra invertida), mas descobriram rapidamente.

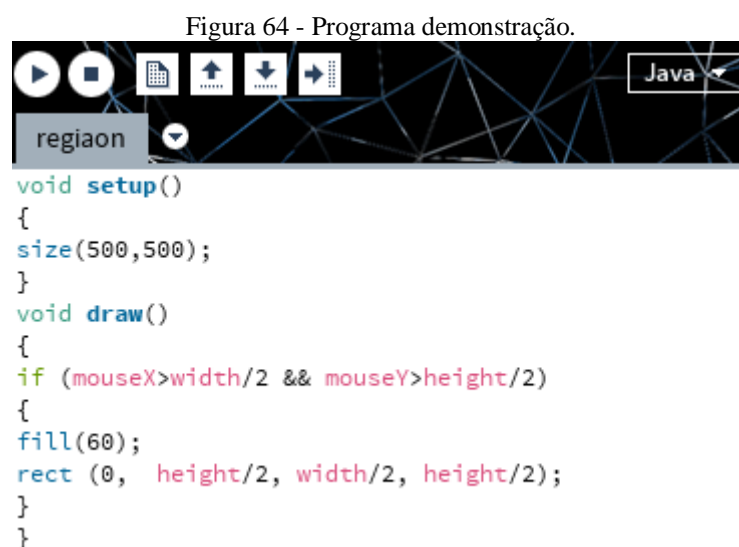
Assim, foi apresentado um desafio de andaime, para que os alunos se familiarizassem com a estrutura e sintaxe dos comandos e operadores empregados. Com uma demonstração no quadro, dividiu-se a janela em quatro partes imaginárias e quando o ponteiro do mouse estivesse sobre na região (quadrante) especificados, realizasse a construção desejada. Assim, foram definidos conjuntamente as regiões e sistematizados no quadro. Observe as falas durante a apresentação do desafio (Quadro 42).

Quadro 42 - Elaboração do desafio que envolve áreas específicas do plano.

<b>NB</b>	<i>“Olhando assim parece tão fácil!”.</i>
<b>KA</b>	<i>“Não é difícil esse. Só entender montar na cabeça a divisão da tela pra saber as partes”.</i>
<b>KA</b>	<i>“Professor, só usa variáveis do sistema!”.</i>

Fonte: Primária (2015).

O pesquisador não tinha atentado para esse detalhe, que o desafio envolve apenas variáveis do sistema, mas a aluna **KA** consegue distinguir perfeitamente variáveis do sistema de variáveis comuns (Figura 64).



Fonte: Autor (2015).

O aluno **VH** solicitou um exemplo cotidiano (Quadro 43) para mostrar o operador lógico NOT. Além de && e ||, também temos acesso ao operador lógico "não", escrito com um ponto de exclamação.

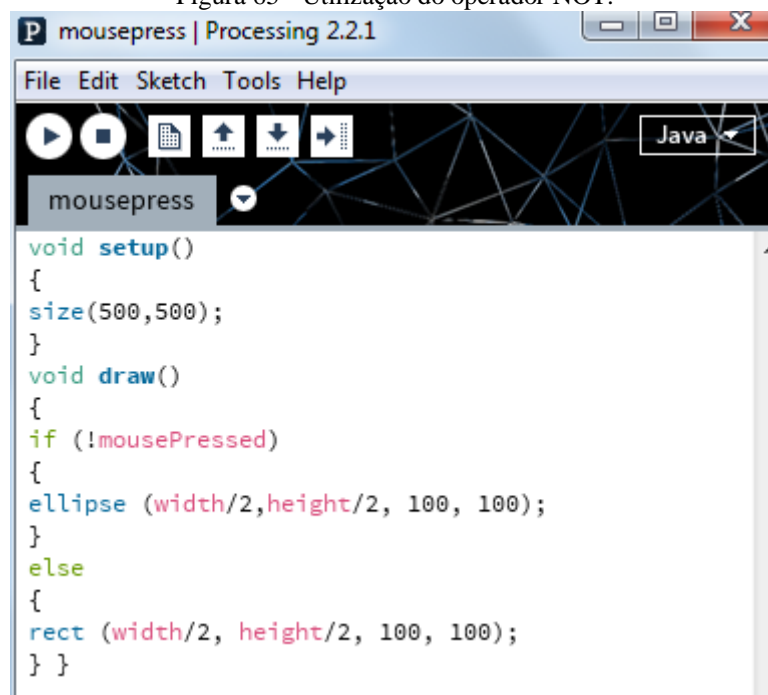
Quadro 43 - Exemplo do cotidiano envolvendo operadores lógicos.

<b>P</b>	<i>“No Processing seria algo tipo: Se o mouse não for pressionado, desenhar um círculo, caso contrário, desenhar um quadrado. Percebeu?”.</i>
<b>VH</b>	<i>“Vamos fazer!”.</i>
<b>VH</b>	<i>“Professor é só colocar o comando com a exclamação antes?”.</i>

Fonte: Primária (2015).

Mais uma vez o desafio partiu da própria curiosidade dos alunos em testar os comandos. O aluno **VH** apresentou o programa (Figura 65).

Figura 65 - Utilização do operador NOT.



```

void setup()
{
  size(500,500);
}
void draw()
{
  if (!mousePressed)
  {
    ellipse (width/2,height/2, 100, 100);
  }
  else
  {
    rect (width/2, height/2, 100, 100);
  } }

```

Fonte: Primária (2015).

**TS:** “Professor, o comando não faz diferença de botão”.

**MB:** “Para resolver isso daí a gente tinha que usar o MouseButton, olha na folhinha”

Realizando os testes a aluna percebeu que independentemente do botão pressionado (direito ou esquerdo), o quadrado era construído. Completando o aluno **MB** já encaminhava uma solução para o problema em Processing 2.

**MB:** “Professor, mas se eu quisesse fazer a pergunta de outro jeito, se o mouse for pressionado, desenhe um quadrado, caso contrário, desenhar um círculo, não é muito mais fácil e vai dar na mesma coisa?”.

O aluno percebe e reconhece várias possibilidades de chegar ao mesmo resultado. O ambiente possibilita testar ideias tornando-as concretas. Aquilo que antes era abstrato pode ser agora reificado por meio da ferramenta.

A seguir, foi lançado um desafio aos alunos, subsidiados pelos comandos apropriados no desafio anterior, que consistia em dividir a área de execução em quatro partes (quadrantes) e conforme o mouse se posicionasse em qualquer uma das partes ela ficasse colorida.

**NB:** “Não precisa clicar?”

**KA:** “Deixa eu pensar, como vou fazer isso. Se a horizontal for menor que a metade e a vertical também, vai ter que fazer o quadrado na primeira parte...”

Durante a construção do desafio notamos que o ambiente de aprendizagem situada colaborou para a promoção de uma reflexão que possibilitou abstrações. Exigiu que os alunos

refletissem sobre uma base muito mais ampla de conhecimento para resolver seu problema. A seguir (Figura 66) o programa apresentado pela aluna TS como solução ao desafio.



Fonte: Primária (2015).

Após este desafio o aluno CA teve a ideia de generalizar em outro contexto. Nota-se que quando os alunos são desafiados, transferem o conhecimento para situações diferentes daquela na qual aprenderam um determinado conteúdo.

**CA:** “*Professor, vou tentar fazer algo diferente, que muda de cor, conforme a gente passa o mouse*”.

Assim, surgiu outro desafio a partir da necessidade do aluno. Ele queria construir uma elipse que mudasse de cor, conforme o ponteiro do mouse se posicionasse em uma das quatro regiões. Assim foi lançado o desafio ao grupo.



pesquisador e alunos, contribuindo para uma ligação mais profunda entre os estudantes e situação de aprendizagem, possibilitando a eles formalizarem suas construções de conhecimentos por meio da interferência do ambiente, que é sempre cultural e torna-se explícito nas relações sociais e conseqüentemente contribuiu para que os alunos viessem perceber a importância da matemática.

Quadro 44- Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 6.

<b>Peculiaridade</b>	<b>Fala dos alunos</b>	<b>Como foi observado</b>
Não linearidade	<b>MB:</b> <i>“Professor, mas se eu quisesse fazer a pergunta de outro jeito, se o mouse for pressionado, desenhe um quadrado, caso contrário, desenhar um círculo, não é muito mais fácil e vai dar na mesma coisa?”.</i>	Para resolver um problema ou completar a tarefa, o aluno era levado a refletir para produzir uma solução. Esta ausência de um caminho linear e a possibilidade de comparar várias performances significa que o interativo Programa multimídia pode ajudar os alunos a concentrar a atenção diretamente em seus próprios processos de pensamento.
Informal	<b>CA:</b> <i>“Tenho essa imagem de plano de fundo no meu celular, tenho muita foto aqui, quer ver?”.</i>	Um ambiente de aprendizagem situado fornece um contexto autêntico que reflete a forma como o conhecimento vai ser usado na vida real, que preserva o contexto completo da situação sem fragmentação e decomposição, que convida a exploração e permite a complexidade natural do mundo real.
Interatividade	<b>KA:</b> <i>“Deixa eu pensar, como vou fazer isso. Se a horizontal for menor que a metade e a vertical também, vai ter que fazer o quadrado na primeira parte...”</i>	O emprego de recursos computacionais possibilitou aos alunos, entre outros fatores, reaproveitar seu código, melhorar seus programas sem a preocupação com o erro. No método tradicional se o erro ocorre o aluno é levado a rabiscar, apagar e a fazer novamente. Com o uso da tecnologia, por meio de uma adequada orientação, tornou o processo mais colaborativo, interativo e desafiador.
Engajamento	<b>CA:</b> <i>“Professor, vou tentar fazer algo diferente, que muda de cor, conforme a gente passa o mouse”.</i>	As habilidades desenvolvidas pelos alunos demonstram que o conhecimento não é mais um domínio exclusivo do indivíduo com maior experiência (expert), a participação periférica legítima colaborou para tornar a trajetória mais difusa e não hierárquica.

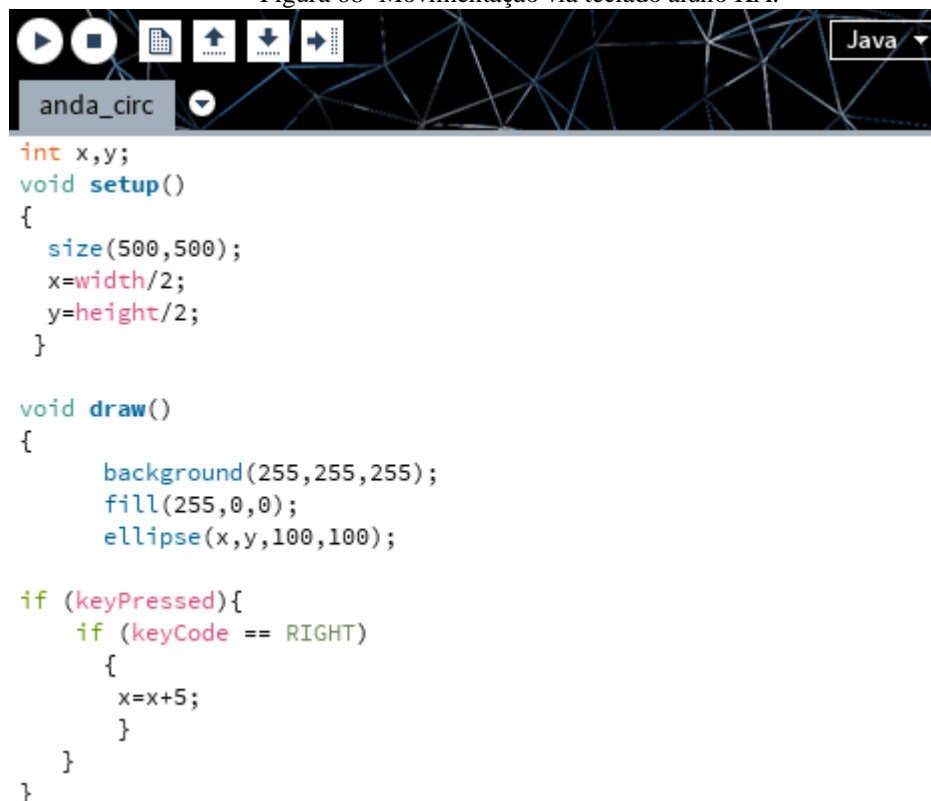
Fonte: Autor (2015).

#### 4.2.7 Oficina 7 – Utilização dos comandos *KeyPressed* e *KeyCode*

Iniciou-se a penúltima oficina com um desafio que consistia em movimentar um círculo para a direita. Para isso era necessário colocar o centro do círculo no ponto central da janela e movimentá-lo para direita com a seta indicativa do teclado. Este desafio teve como objetivo explorar os comandos *KeyPressed* e *KeyCode* e num primeiro momento, direcionamos a resolução do desafio como forma de andaimes.

O grupo em geral apresentou problemas em relação à abertura e fechamento das chaves no ponto crítico do desenvolvimento do programa, no qual testa, se uma tecla foi pressionada e se esta tecla é a seta para a direita. Precisamos recorrer ao quadro para expor novamente a estrutura. A aluna **KA** apresentou a codificação que atendia ao proposto (Figura 68).

Figura 68- Movimentação via teclado aluno KA.



```

anda_circ
int x,y;
void setup()
{
  size(500,500);
  x=width/2;
  y=height/2;
}

void draw()
{
  background(255,255,255);
  fill(255,0,0);
  ellipse(x,y,100,100);

  if (keyPressed){
    if (keyCode == RIGHT)
    {
      x=x+5;
    }
  }
}

```

Fonte: Primária (2015).

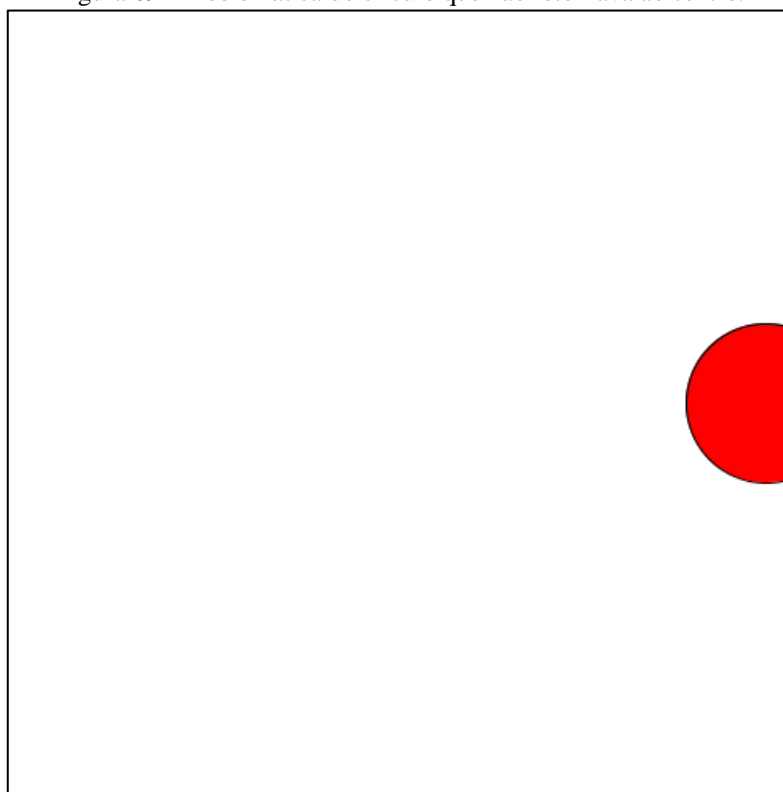
Quando os alunos implementaram o programa e começaram os testes, os mesmos apertavam a seta para a esquerda para que o círculo retornasse ao centro (Figura 69).

**NB:** “Não volta!”.

**TS:** “Tem que programar pra voltar, tudo é a gente que tem que montar”.

Os alunos perceberam que em programação, tudo tem que ser indicado passo a passo numa sequência lógica, construindo o algoritmo.

Figura 69 - Problemática do círculo que não retornava ao centro.



Fonte: Autor (2015).

Depois de apropriados os conceitos anteriormente formalizados, apresentamos novo desafio aos estudantes que tinham como objetivo colocar um retângulo central na parte inferior do sketch e fazê-lo movimentar para a esquerda e para a direita por meio das setas de movimentação.

Para esse desafio, o dividimos em duas partes. Inicialmente criar o retângulo na cor vermelha, central, independentemente do tamanho da área do sketch e na segunda parte da oficina fazê-lo movimentar. Veja as discussões no Quadro 45.

Quadro 45 – Discussões sobre a criação da plataforma móvel inferior

<b>VH</b>	<i>“Professor, se independente do size, então temos que trabalhar com a altura e largura atual, vamos precisar utilizar o width e o heigth”.</i>
<b>P</b>	<i>“Muito boa estratégia. É por ai o começo”.</i>
<b>NB</b>	<i>“Mas professor como vamos colocar ele no meio. Eu tenho que colocar a coordenada onde começa o retângulo”.</i>
<b>P</b>	<i>“Usa as dimensões do size 500 por 500, só pra vocês pensarem. Vou exigir mais, eu quero um retângulo que tenha 100 pixels na horizontal e 20 na vertical”.</i>
<b>MB</b>	<i>“Vai ter que ficar 50 pixels para cada lado”.</i>

Fonte: Primária (2015).

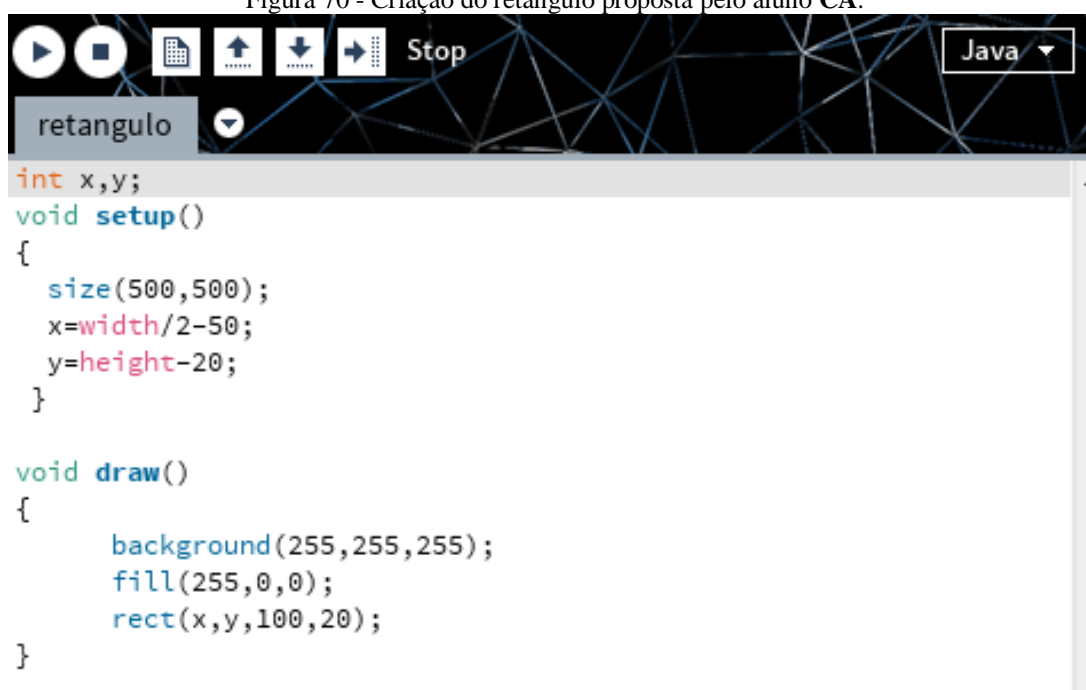
No momento da oficina não conseguimos compreender a fala do aluno **MB**. Mas na análise do vídeo, percebemos que o aluno pensava corretamente. Pois se solicitamos 100 pixels na horizontal, para que ele fosse central, teríamos que obedecer a essas condições.

Os alunos ficaram aproximadamente 15 minutos tentando encontrar uma solução para o desafio. Eles compreenderam o objetivo do desafio, mas todos estavam diante de um impasse, não sabiam onde colocar a coordenada de início do retângulo. Assim, foi desenhada no quadro branco a situação, e pensada conjuntamente como resolvê-la.

**CA:** “Agora a gente já sabe, pode parar”.

O insight que os alunos tiveram foi no momento em que subtraímos 50 pixels da metade do eixo horizontal ( $x = \text{width}/2 - 50$ ). A partir desse momento o aluno **CA** pediu para que parássemos a orientação e que eles já haviam enxergado o caminho para solução do desafio (Figura 70).

Figura 70 - Criação do retângulo proposta pelo aluno **CA**.



```

retangulo
int x,y;
void setup()
{
  size(500,500);
  x=width/2-50;
  y=height-20;
}

void draw()
{
  background(255,255,255);
  fill(255,0,0);
  rect(x,y,100,20);
}

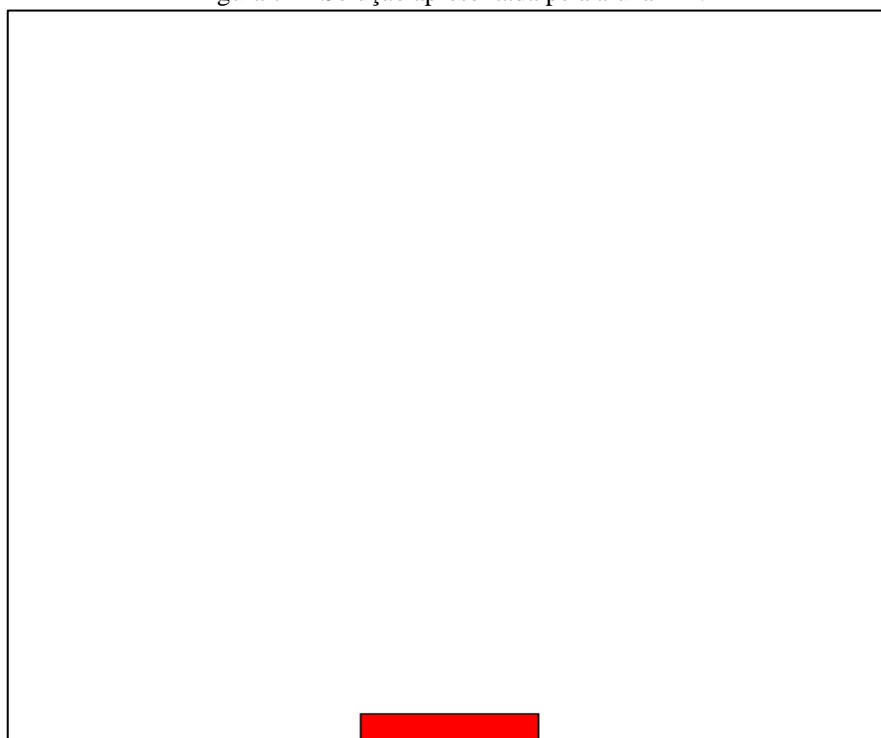
```

Fonte: Primária (2015).

Durante a resolução desse desafio o aluno **CA** disse: “*Professor, programar é matemática, isso aqui tudo parece matemática*”.

Foi a primeira vez, durante as oficinas que os alunos reconheceram a importância da matemática explicitamente na aprendizagem situada numa comunidade de prática de programadores, ratificando os pressuposto de Lave e Wenger (1991), em que os compromissos sociais forneceram o contexto adequado para que a aprendizagem ocorresse. Por meio desses compromissos sociais os estudantes reconhecem a importância da matemática e rompem com a ideia de que a matemática tem utilidade prática apenas nas profissões que lidam com números, como as Engenharias, e que o raciocínio lógico se faz presente nas diferentes profissões. A seguir o resultado da execução do desafio (Figura 71).

Figura 71 - Solução apresentada pela aluna KA.



Fonte: Primária (2015).

No segundo momento da oficina, todos os esforços se voltavam para a movimentação do retângulo por meio das setas do teclado. Os alunos estavam ansiosos para conseguirem movimentar o retângulo.

Consideramos que o grau de dificuldade para a resolução dessa segunda etapa era elevado e exigiria dos alunos muita reflexão para consecução dos objetivos. Os alunos já dominavam a estrutura de movimentação por meio das setas.

**KA:** “*Professor agora dá pra gente aproveitar o código que fizemos ontem para movimentar o círculo, pois é bem parecido, não é?*”.

**P:** “*Mas agora com um diferencial, teremos que traçar uma estratégia para que a plataforma não ultrapasse os limites da janela*”.

**MB:** “*O limite máximo para a direita não pode ultrapassar o width*”.

O aluno **MB** apresentou estratégia (Figura 72), mas percebia que o retângulo ultrapassava o limite da região. O aluno realizava testes seguidamente, promovendo alterações até que percebeu que teria que comparar o valor de  $x$  adicionado do tamanho da dimensão horizontal do retângulo, para testar se ainda havia espaço suficiente para movimentação.

Figura 72 - Problemática dos limites da janela.

```
int x,y;
void setup()
{
  size(400,300);
  x=width/2-50;
  y=height-20;
}
void draw()
{
  background(255);
  fill(255,0,0);
  rect(x,y,100,20);
  if (keyPressed)
  {
    if (keyPressed)
    {
      if (keyCode == RIGHT)
      {
        if (x < width)
        {
          x=x+3;
        }
      }
    }
  }
}
```

Fonte: Primária (2015).

Agora a atenção se voltava para barrar a plataforma até o limite esquerdo. Os estudantes resolveram com maior facilidade a problemática dos limites esquerdo da janela, uma vez que já haviam refletido anteriormente. Os estudantes **CA** e **MB** foram os primeiros que apresentaram uma estratégia coerente. Solicitamos que os alunos socializassem com os demais integrantes do grupo que apresentaram dificuldades para solucionar a problemática.

É apresentado o código completo e funcional elaborado pelo aluno **CA** (Figura 73).

Figura 73 - Programa do aluno CA, completo e funcional.

```

int x,y;
void setup()
{
  size(500,500);
  x=width/2-50;
  y=height-20;
}
void draw()
{
  background(255,255,255);
  fill(255,0,0);
  rect(x,y,100,20);
  if (keyPressed)
  {
    if (keyCode == RIGHT)
    {
      if (x+100 < width)
      {
        x=x+3;
      }
    }
    if (keyCode == LEFT)
    {
      if(x>=1)
      {
        x=x-3;
      }
    }
  }
}

```

Fonte: Primária (2015).

A aprendizagem situada numa comunidade de prática de programação encorajou os estudantes a pensarem sobre as relações formalizadas no processo de construção. O pensamento sistemático foi crucial para todos, que assim, puderam repensar seus objetivos. Nota-se que a aprendizagem situada encoraja atitudes diferentes em sintonia com as exigências atuais do mercado de trabalho, profissões modernas, globais que necessitam de alta tecnologia.

Foi possível trabalhar de forma situada, habilidades que contemplam expressões numéricas, contextos geométricos que exercitam a capacidade de levantar hipóteses e validá-las e habilidades de construção de argumentos matemáticos.

Quadro 46 - Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 7.

<b>Peculiaridade</b>	<b>Fala dos alunos</b>	<b>Como foi observado</b>
Ênfase no “saber fazer”	<b>NB:</b> “ <i>Mas professor como vamos colocar ele no meio. Eu tenho que colocar a coordenada onde começa o retângulo</i> ”.	Enquanto o aluno programava em Processing era possível indagar: por que você pensa assim? Como você chegou a essa conclusão? É possível generalizar isso? Como é possível representar essa situação? Existe outra possibilidade mais eficiente?
Informal	<b>CA</b> “ <i>Professor, programar é matemática, isso aqui tudo parece matemática</i> ”.	Nas oficinas respondiam as questões com muita naturalidade, sem se preocupar com a nomenclatura e/ou simbologia própria da matemática. Não houve preocupação com o ritmo da aprendizagem, enquanto o sistema apostilado impõe ritmos de aprendizagem. Quando o aluno tinha dificuldades ele recorria a um membro mais experiente e em grande parte não se dirigia ao mediador. A pergunta recorrente era: Você está com dificuldades em qual parte do código? E assim a matemática foi sendo percebida no contexto social.
Construção do conhecimento parte da necessidade	<b>VH:</b> “ <i>Eu preciso tirar o contorno como que eu faço isso mesmo?</i> ”.	Os alunos têm dificuldade em lidar com palavras fora de contexto, como no caso das apostilas. Na aprendizagem situada com o uso do Processing 2 fornecem-se informações no tempo certo quando o aluno sente necessidade dela.

Fonte: Autor (2015).

### 7.2.8 Oficina 8 – Desafio da bolinha

Chega-se à última oficina prevista para a concretização de um desafio que envolva as habilidades trabalhadas anteriormente. Nesta oficina, conforme planejado, não se aborda nenhum comando inédito. Para sua solução, os alunos teriam que criar estratégias com os conceitos e comando apropriados. Ressalta-se que não teve ausência em toda a trajetória de participação na comunidade de prática de programação.

A oitava oficina propôs a continuação dos trabalhos da oficina anterior (oficina 7). Para isso os alunos teriam que fazer uma bolinha cair em posições diferentes do eixo horizontal.

**CA:** “*Vai ficar parecendo um joguinho, aquele que não podemos deixar a bolinha cair*”.

**MB:** *“Que legal professor, da pra fazer qualquer coisa no Processing, inclusive jogos. Eu quero fazer faculdade de jogos digitais, meu primo faz, eu também pretendo”.*

Quando os alunos se viram diante de um projeto funcional a motivação aumentou. Era perceptível o clima de determinação logo no início da oficina.

Solicitamos que os alunos abrissem o código da oficina anterior.

**P:** *“Agora, como podemos fazer um bolinha descer na vertical?”.*

**CA:** *“Vamos ter que variar a posição do y”.*

Como foi atribuído x para o eixo horizontal e y para o vertical, explica a resposta utilizando o y como eixo vertical.

**P:** *“Então vamos lá, eu quero que a bolinha desça bem no centro. Como fazer isso?”.*

**CA:** *“Vamos criar a bolinha primeiro, qualquer tamanho?”.*

**P:** *“A dimensão pode ser 30 pixels”.*

Foram sugeridos os nomes das variáveis para que se chegue num resultado mais padronizado. Como resultado, obtém-se conjuntamente ao código apresentado (Figura 74) pelo estudante **VH**.

Figura 74 - Programa do aluno VH.

```

int x,y, bolax, bolay;
void setup()
{
  size(400,300);
  x=width/2-50;
  y=height-20;
  bolax=width/2;
  bolay=0;
}
void draw()
{
  background(255,255,255);
  fill(0,0,255);
  ellipse(bolax,bolay,30,30);
  bolay=bolay+2;
  if (bolay+20>=height)
  {
    bolax= int (random(0, width));
    bolay=0;
  }
  fill(255,0,0);
  rect(x,y,100,20);
  if (keyPressed)
  {
    if (keyCode == RIGHT)
    {
      if (x+100 < width)
      {
        x=x+3;
      }
    }
  }
}

```

Fonte: Primária (2015).

Os alunos percebem facilmente o uso da função randômica e não tiveram problemas em suas construções pelas experiências anteriores.

A construção da codificação de descida da bolinha foi tranquila. O grupo discutia uma solução para que a bolinha retornasse novamente em outra posição horizontal quando tocasse a extremidade inferior da janela, sendo esse ponto o que mais gerou divergências entre os estudantes. A condicional empregada pelos alunos foi fruto de discussão deliberada.

Em continuidade propusemos que os alunos criassem uma estratégia para que alterasse a velocidade de descida da bolinha.

Os estudantes reconheciam que para solucionar o problema a mudança ia acontecer com o valor incremental ao eixo y.

O aluno **MB** deu a sugestão para que criássemos mais uma variável aleatória, em substituição ao +2 que a variável *bolay* estava recebendo (Figura 75).

Figura 75 - Solução apresentada pelo aluno para resolver o problema da velocidade.

```
int x,y, bolax, bolay, vel;
void setup()
{
  size(400,300);
  x=width/2-50;
  y=height-20;
  bolax=width/2;
  bolay=0;
  vel = int (random(0,5));
}
void draw()
{
  background(255,255,255);
  fill(0,0,255);
  ellipse(bolax,bolay,30,30);
  bolay=bolay+vel;
  if (bolay+20>=height)
  {
    bolax= int (random(0, width));
    bolay=0;
  }
  fill(255,0,0);
  rect(x,y,100,20);
  if (keyPressed)
  {
    if (keyCode == RIGHT)
    {
      if (x+100 < width)
```

Fonte: Primária (2015).

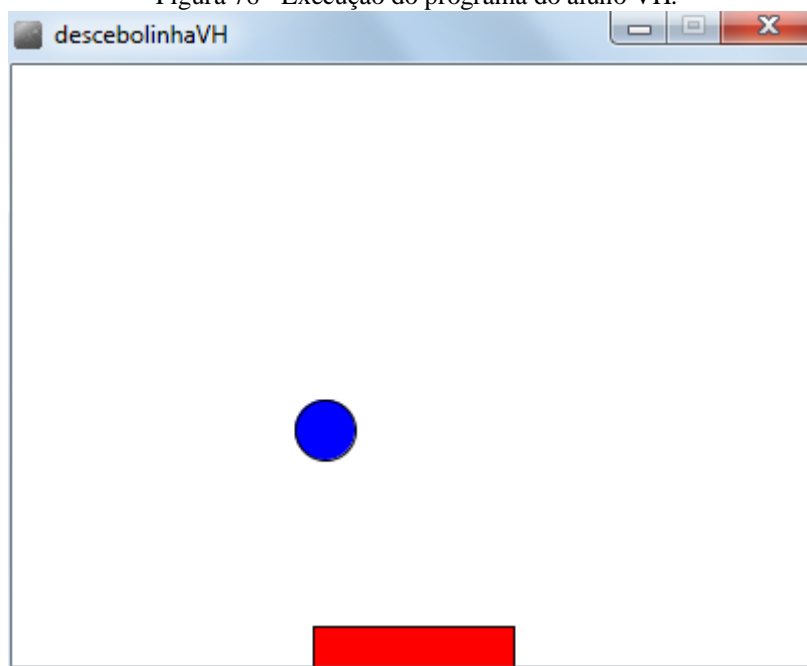
**TS:** “Professor, dividir o problema em partes, as menores possíveis, fica bem mais fácil”.

**MB:** “Como faço para passar isso no meu androide?”.

**P:** “MB sinceramente eu nunca fiz isso. Eu não sei fazer, mas podemos descobrir”.

Nesse momento assumimos que não sabíamos como fazer este processo. Mas que para uma próxima oportunidade, poderíamos pesquisar a respeito e tentarmos realizar este processo. Alguns se entusiasmavam, trocando ideias sobre o que as estratégias da resolução. Aos poucos forma surgindo ideias e ao final perguntavam se podiam recriar em cima do que tinham construído. O pesquisador reforçou que isso é fantástico! Veja a execução do programa do aluno **VH** (Figura 76).

Figura 76 - Execução do programa do aluno VH.



Fonte: Autor (2015).

Quadro 47 - Peculiaridades da aprendizagem situada evidenciadas nas falas dos alunos durante a oficina 8.

Peculiaridade	Fala dos alunos	Como foi observado
Não linearidade	<b>MB:</b> “Como faço para passar isso no meu androide?”.	A mediação do professor desempenha um papel determinante à medida que, ao trabalhar com Processing 2, pode-se criar situações desafiantes, em que o aluno pode permear em outros contextos, reavaliar suas estratégias e objetivos, envolvendo-se no processo de construção de conhecimento.
Ênfase no “saber fazer”	<b>CA:</b> “Vamos criar a bolinha primeiro, qualquer tamanho?”.	Os alunos aprendem de maneiras diferentes e o conhecimento do aluno foi percebido pelo processo social de interagir e fazer. A aprendizagem situada possibilitou aos alunos a aprender naturalmente como resultado de um comportamento social. O Processing 2 atuou como um guia enquanto os alunos aprendem naturalmente a partir de sua própria experiência.
Informal	<b>TS:</b> “Professor, dividir o problema em partes as menores possíveis, fica bem mais fácil”.	Essa visão do ensinar matemática e do aprender matemática é imprescindível. Os alunos passaram a ter autonomia a partir do momento que puderam direcionar o andamento das aulas e os conteúdos a serem trabalhados para possibilitar

		a resolução de seus desafios. O papel mais ativo se configura como sujeitos de sua aprendizagem, pois tiveram que pensar, elaborar estratégias, rever suas ações e descobrir como se implementavam suas ideias em Processing 2.
Engajamento	<i>MB: “Que legal professor, da pra fazer qualquer coisa no Processing, inclusive jogos. Eu quero fazer faculdade de jogos digitais, meu primo faz, eu também pretendo”.</i>	Notou-se que o Processing mudou a forma pela qual os estudantes se relacionavam com a matemática, pois ofereceu novas perspectivas ao uso da linguagem matemática. O Processing permitiu a formação de micromundos, pois seriam situações impossíveis de serem criadas em sala de aula, o que com o uso do Processing foi possível, lembrando que o Processing por si não funciona como estímulo à aprendizagem. O sucesso depende da integração com o currículo e com os desafios propostos.

Fonte: Autor (2015).

O último desafio possibilitou que os alunos enxergassem que o importante é quebrar os conteúdos em pequenas partes e explorar cada parte lentamente, uma de cada vez. Dividir um problema de programação em etapas é um bom começo para que ao final tenha algo completo e funcional.

Os alunos perceberam que a cada passo adicionavam uma pequena melhora ao programa: primeiro criaram o retângulo, depois resolveram o problema da movimentação, para então trabalharem na problemática das bolinhas, realizando as devidas animações. Claro que isso é apenas uma pequena fração de todas as etapas de que precisamos para um jogo completo e funcional, mas ele demonstra uma forma importante de pensar.

Essa quebra em etapas facilitou o processo de depuração. Cada vez que testamos os programas, muito provavelmente algo pode não funcionar como o esperado. Assim se começarmos a testar um programa completo, o processo de depuração ficará mais complicado.

Aprender o básico em programação ajudou iniciar os alunos na nova alfabetização do século e treiná-los em projetos que envolvam colaboração.

O Processing 2 se mostrou adequado aos novatos numa comunidade de prática de programação, pela sua simplicidade e por sua dinâmica visual, permitindo aos estudantes transformar suas ideias em projetos reais de software.

O emprego do método não favoreceu somente a dimensão técnica, mas a dimensão humana, política e social evidenciando a multidimensionalidade do processo de ensino e aprendizagem. Evidenciamos também que exigiu do professor uma boa formação. É obrigação estar preparado para propiciar aprendizagem matemática a todos os alunos. Além disso, todas as metodologias de ensino exigem uma boa formação matemática. Nenhum método produz aprendizagem com professores despreparados. Os alunos passaram a fazer indagações fora do planejamento da oficina, e isso pode ser um complicador para os professores.

O ambiente Processing 2 não é um caminho para todas as práticas pedagógicas, uma vez que deve ser ofertado aos alunos um pluralismo metodológico, pois eles aprendem de diferentes formas, e uma diversificação de meios é requisito essencial.

Outro ponto, observado é em relação ao tempo. Trabalhar com a ferramenta, requer, com certeza, mais tempo para ensinar. Mas, antes de considerar o tempo dispendido para que os alunos aprendam é preciso considerar a qualidade da aprendizagem. A aprendizagem situada por meio da ferramenta tem característica não-linear e gerou indagações não previstas pelo professor e, nesse sentido, o ensino poderá demandar mais tempo que o previsto. Em contrapartida, muitas vezes, o uso do Processing 2, por facilitar a aprendizagem gera ganho de tempo. A reutilização do código também foi fundamental pois conseguimos agilizar muitos momentos do processo de aprendizagem.

Os alunos dispunham de uma pasta individual para guardarem seus programas e outra pasta compartilhada com todos. Assim, os alunos poderiam acessar a qualquer instante, e também potencializar o tempo de aprendizagem.

A aprendizagem situada numa comunidade de prática possibilitou ampliar a linguagem e promover a comunicação de ideias matemáticas; apropriar estratégias de resolução de problemas; gerar a capacidade de fazer estimativas, cálculos mentais; abstrair, organizar e planejar ações. Também estimulou a concentração (foco), perseverança e a criatividade. Promoveu a troca de ideias e contribuiu para a compreensão de regras, percepções no plano cartesiano, e principalmente para a formação de conceitos.

Percebe-se que, à medida que avançam nas oficinas, os alunos aumentam sua atitude reflexiva e questionadora. Uma observação do processo de execução permitiu constatar que os alunos foram levados à análise, de modo que eles construía (ênfase no fazer), encontravam seus erros, procuravam corrigi-los, testavam e aprendiam os conceitos envolvidos na solução dos problemas, além de trabalhar a dimensão dos sentimentos, da afetividade e da criatividade.

Notamos que os estudantes persistem mais quando são colocados diante de objetivos de curto prazo, específicos e moderadamente desafiadores, o que favoreceu a motivação na busca de soluções, uma vez que a motivação dos alunos é mais difícil diante de objetivos genéricos, distantes e altamente desafiadores, o que contribuiu para o que Csikszentmihalyi (2009) chama de “flow”, um canal que equilibra habilidades e desafios.

A construção do saber matemático na superação dos baixos índices de desempenhos de nossos alunos requer também conhecimentos externos à matemática. As novas demandas sociais educativas apontam para a necessidade de um ensino voltado para a promoção do desenvolvimento da capacidade de ação e reflexão e o Processing 2 veio colaborar com tais desenvolvimentos, conforme apontados por Almeida (2000).

Além disso, conseguimos ratificar algumas das dimensões dos pressupostos teóricos apresentados por Lave e Wenger (1991) e Wenger (1998) na aprendizagem situada, quando se trabalha com Processing 2, entre eles:

- Alta interatividade dos alunos e mediadores (pesquisador e computador).
- Uma aprendizagem mais colaborativa voltada à construção do conhecimento compartilhado.
- Presença social (líderes).
- Os estudantes procuram problemas desafiantes e habilidades de que necessitam para avançar no momento certo e preciso.
- Estado reflexivo processual. Passamos algum tempo pensando como realizar a construção, depois como realizarmos isso em Processing, e após implementação, outro processo reflexivo se projeta sobre os resultados das ações.
- A formação de uma cultura de sala de aula onde todos os estudantes são tratados como talentosos e com potencial.
- Formação de uma identidade, por meio de alunos apaixonados por programação e fascinados pelas possibilidades de criação que Processing 2 possibilita.

Quando os estudantes não obtinham sucesso em suas estratégias, direcionávamos o desafio. Assim, na resolução dos desafios quando um aluno não chegava próximo ao objetivo, não havia em nenhum momento a expressão “você está errado”. Os questionamentos recorrentes eram: “De que outra maneira podemos olhar para isso, de forma a atender a meta?” ou até mesmo: “Essa é um boa resposta, mas não é exatamente o objetivo que temos que alcançar no desafio”

O ambiente de aprendizagem situada, utilizando Processing 2 possibilitou também uma avaliação integrada de aprendizagem, por meio de uma avaliação autêntica, que está relacionada com o processo e com engajamento dos estudantes na comunidade de prática.

Algumas dimensões foram observadas e representadas no diagrama (Figura 77).

Figura 77- Dimensões observadas.

Autonomia	<b>CA</b> : <i>"Agora a gente já sabe, pode parar".</i> <i>"Professor, tá dando erro, mas não fala eu vou descobrir".</i>
Cooperação	<b>TS</b> : <i>"Vamos deixar mais rápido, sabe como faz? Vou explicar". (líderes)</i>
Erro como elemento natural	<b>MB</b> <i>"Eu fiz um de cada, mas olha aqui vou arrumar porque o triângulo ficou sobre o meu retângulo".</i>
Imediatismo	<b>CA</b> : <i>"Roda de novo pra gente vê".</i>
Assimilação de conceitos Matemáticos	<b>TS</b> : <i>"[..]isso aqui é duas circunferências tangentes, porque ela se encontra só nesse ponto".</i>

Fonte: Autor (2015).

Assim, a *autonomia*, *cooperação*, o *erro como elemento natural*, *imediatismo*, *assimilação de conceitos matemáticos* foram dimensões evidenciadas quando se utiliza a programação de computadores como instrumento didático-pedagógico.

### 4.3 Grupo focal final

Após o término da oficina, foram realizados as cópias das pastas contendo todos os programas construídos pelos alunos ao longo das oito oficinas. Assim, dirigimos à biblioteca da unidade escolar para realização do grupo focal final e posteriormente para entrevista com a professora de matemática dos participantes envolvidos no trabalho de pesquisa (Quadro 48).

Quadro 48 - Percepções dos alunos na aprendizagem situada.

<b>P</b>	<i>“Vocês ensinaram o computador a fazer coisas, o que vocês perceberam durante essa construção?”.</i>
<b>MB</b>	<i>“O segredo de tudo eu acho que é dominar as coordenadas”.</i>
<b>CA</b>	<i>“Saber o comando também é importante”.</i>
<b>P</b>	<i>“Mas você acharam os comandos difícil de guardar?”.</i>
<b>CA</b>	<i>“Não, foi fácil, e quando a gente erra ele não coloca na cor certa”.</i>
<b>VH</b>	<i>“Eu não imaginava que programar computador seria dessa forma”.</i>
<b>P</b>	<i>“Que forma?”.</i>
<b>VH</b>	<i>“A gente precisa pensar muito”.</i>
<b>CA</b>	<i>“Saber programar vai muito além de saber comandos, a gente precisa organizar as nossas ideias”.</i>
<b>P</b>	<i>“Interessante CA, mas o que é necessário nesse processo de organizar ideias?”.</i>
<b>NB</b>	<i>“Pra mim, tem que saber matemática”.</i>
<b>TS</b>	<i>“Nossa professor se você tivesse falado que envolvia matemática todo mundo tinha faltado”.</i>
<b>KA</b>	<i>“É verdade professor, saber um comando não basta, porque eu tenho que pensar matematicamente para resolver o problema, só que lá parece ser mais fácil e a gente acabava nem notando que era matemática”.</i>
<b>P</b>	<i>“Enquanto vocês estavam programando vocês notaram a matemática ou ela passou assim mesmo despercebida”.</i>
<b>KA</b>	<i>“No começo não, mas com o passar dos desafios fomos conseguindo perceber que a matemática estava presente ali”.</i>
<b>MB</b>	<i>“Eu aprendi muito sobre ângulos quando eu fiz o come come”.</i>
<b>TS</b>	<i>“Eu aprendi a gostar mais da matemática. Pra mim a matemática é um quebra cabeça, às vezes acertamos e às vezes erramos, mas temos que prestar atenção para nós próprio nos desafiar, assim fica mais interessante na hora do cálculo”.</i>
<b>P</b>	<i>“O que motivou vocês a frequentarem as oficinas?”</i>
<b>MB</b>	<i>“Pra mim foi aprender Processing cada dia mais”.</i>
<b>VH</b>	<i>“Foi uma experiência diferente daquilo que a gente faz na escola, pois a gente tá aprendendo uma profissão. Lá no futuro eu posso ser programador”.</i>

Fonte: Primária (2015).

Assim, por meio dos discursos dos alunos podemos identificar que a matemática foi sendo percebida direta e indiretamente por meio dessas ações de interação, colaboração e motivação. A frequência do grupo foi de 100%, mesmo no contraturno e facultativo.

Aproveitou-se a oportunidade para conversar com a professora de matemática dos estudantes que estava em seu horário de estudos (Quadro 49). Descobriu-se que os alunos, durante as aulas regulares compartilhavam suas experiências com ela, para enriquecer mais suas atividades em sala de aula e isso foi surpreendente para este pesquisador.

Quadro 49 - Percepções da professora de matemática da turma.

<b>P</b>	<i>“Professora, a senhora notou alguma diferença no grupo de alunos participantes deste projeto?”</i>
<b>AC</b>	<i>“Eu percebi que os alunos passaram a se dedicar mais nas aulas de matemática, e o MB, o CA, e a TS mudaram muito sua postura em sala de aula, estão mais envolvidos com o processo, principalmente no retorno da tarefa de casa”.</i>
<b>P</b>	<i>“Mais alguma coisa que a senhora queira comentar, sinta-se à vontade.”</i>
<b>AC</b>	<i>“Outra coisa, eles comentavam nas aulas sobre o que estavam aprendendo no curso. Até na reunião de pais a mãe do M perguntou como que fazia para o filho participar, porque os amiguinhos disseram que estavam gostando muito do curso. Mas quando selecionei o grupo, apontei outros alunos que se enquadravam no perfil que você solicitou, mas não quiseram, agora estão querendo participar”.</i>

Fonte: Primária (2015).

Quando o aluno esquece o dever de casa, nós enquanto docentes, exclamamos “Então você esqueceu o dever de casa de novo” e tachamos de irresponsáveis. Mas dificilmente perguntamos qual estratégia podemos tentar para ajudá-lo a entregar o dever de casa na data solicitada. Com o Processing foi diferente como pudemos evidenciar na fala da professora.

Fomos buscar uma relação para os nomes citados pela professora e notamos que são justamente aqueles alunos que se transformaram numa espécie de mentores (líderes) durante as oficinas para ajudar os outros integrantes do grupo, que estenderam essa postura para além das atividades no LIE.

Percebemos pela fala da professora que o número de alunos que quer participar das oficinas de programação aumentava à medida que os alunos comentavam sobre suas experiências no Processing.

Outro ponto que merece destacar é que a professora de matemática da outra turma, não envolvidas nas oficinas, queria fazer parte das atividades no LIE. A mesma queria entender e saber sobre as oficinas de programação.

A condução das oficinas estava sempre cheia de novos pensamentos, conversas, novas ideias. Nas pequenas pausas realizadas, os alunos mantinham o foco no assunto e objetivos. A pesquisa demonstra um incentivo a investir em métodos pedagógicos assistidos por tecnologia, que quando usados e avaliados de forma apropriada, melhoram a aprendizagem de maneira significativa.

Outro fator notável é a potencialização do tempo gasto para os alunos receberem suas correções. Durante as oficinais percebeu-se a motivação dos alunos em receberem um feedback rápido. Com a apostila, um assunto é explorado e solicita-se a realização da tarefa de casa e só então num outro momento é realizada a correção. Com o uso do Processing 2 o retorno rápido aos alunos foi um diferencial.

Os alunos se sentem frustrados porque geralmente demoram em média algumas aulas para receberem as correções das atividades da apostila. Isso significa que o professor não percebe de imediato se o conteúdo de um módulo teve seus conceitos e habilidades dominados pelos alunos, não podendo sanar suas dúvidas.

A avaliação passou a ser dinâmica e o professor pode concentrar sua aula nas maiores dúvidas e passar mais rapidamente pelas habilidades já contempladas. A aprendizagem situada por meio das oficinas de programação contribuiu para o fluxo de acordo com as necessidades reais dos alunos mantendo a dosagem entre os eixos habilidades e dificuldades conforme apontados por Csikszentmihalyi (2009). Esse canal exige do professor formação adequada e capacidade em atender às necessidades dos alunos, seus anseios e suas criações.

A aversão à língua inglesa evidenciada no grupo focal inicial, na codificação se tornou interessante e prazerosa, uma vez que os alunos ampliavam seus vocabulários naturalmente na comunidade de prática de programação em Processing 2 e a aprendizagem de programação contribuiu para que os alunos reconhecessem a importância do idioma no mercado de computação, onde as ferramentas de programação, manuais, referências são desenvolvidas na sua grande parte em língua inglesa. Assim, as oficinas de programação contribuíram para que os estudantes ampliassem suas visões no tocante à importância do domínio da língua inglesa nos dias de hoje.

A ferramenta Processing 2 proporcionou ricos momentos de aprendizagem da geometria, colaborando para construções dinâmicas, em que possibilitou aos alunos visualizarem suas construções. Assim, é possível entender que a aprendizagem situada da matemática, inserida numa comunidade de prática de programadores contribuiu para formar uma comunidade escolar que realmente prepara alunos para a cultura global do século XXI, envolvendo a nova alfabetização apontada por Prensky (2008), Resnick (2013). A educação do caráter, da cidadania, da comunicação, do pensamento crítico e resolução de problemas, além de promover maior colaboração e criatividade.

## CONCLUSÃO

Este trabalho teve como objetivo mostrar as relações formalizadas por alunos do ensino fundamental com baixo rendimento em matemática em uma comunidade de prática de iniciação em programação.

Conforme a discussão dos dados já mostrou, a programação de computadores como instrumento didático-pedagógico, no emprego da ferramenta Processing 2 aponta avanços na aprendizagem dos alunos, no tocante à promoção de trabalho mais cooperativo que coloca o erro como elemento natural no processo de aprendizagem, e torna a aprendizagem mais interativa, contribuindo para um feedback imediato, e avança, sobretudo, na apropriação de conceitos matemáticos adjacentes da atividade de programação.

Foi possível evidenciar que no transcorrer das oficinas os alunos passaram a perceber a necessidade e a importância da matemática enquanto constroem programas que representam seus anseios e desejos. Assim, o conhecimento parte da necessidade pontual e com isso explora uma aprendizagem mais ativa. A programação de computadores articulada com a resolução de problemas possibilitou a sistematização e apropriação de conceitos matemáticos. Na construção de algoritmos para solução dos desafios, precisaram entender e compreender os conceitos matemáticos envolvidos nessa construção e a programação passou a ser um suporte. Os estudantes deixaram de ser meramente receptores de informações, e se engajaram de maneira ativa nas práticas da comunidade, o que contribuiu para uma aprendizagem ativa.

A discussão dos dados mostra, em vários momentos, vantagens de aulas de matemática utilizando ferramentas de programação como o Processing 2, com ricas situações de aprendizagem. O trabalho realizado nas oficinas de programação foi articulado à solução de problemas, assim, o método de ensino foi modificado. Não há aluno ruim que não avance, mas uma escola e métodos talvez ineficientes e incompatíveis com os objetivos desse século. Para melhorar essa situação acredita-se que o método pode ser uma das respostas.

O desafio é escolher ou desenvolver métodos interessantes e lembrar que não existe um método bom para todos os alunos, bem como um aluno adequado para qualquer método; o correto é a oferta de um pluralismo metodológico. Pensa-se que há uma relação método/aluno, o mais difícil é saber exatamente qual. Assim, acredita-se que um dos fatores está relacionado à interatividade e no tempo rápido de respostas conforme evidenciado neste trabalho.

As dificuldades ao desenvolver aulas de matemática utilizando ambiente de programação Processing 2 foram elencadas. Entre elas, a mudança filosófica na postura do

professor, juntamente com os tipos de mudanças que o uso de ferramentas de programação provocou na dinâmica das aulas de matemática, com a formação de líderes.

Os ganhos na aprendizagem dos alunos são efetivos e evidenciados nesses estudos. Os alunos não saíram experts em programação após as oito oficinas, mas tiveram uma ideia mais precisa do que é e como deve ser construído um programa de computador. Enquanto os alunos trabalhavam no desenvolvimento dos projetos, como eles mesmos declararam, criando as sequências de comandos, eles estavam a aprender sobre o processo de construção; de como começar uma ideia; de como discuti-la com o grupo e como transformá-la num projeto completo e funcional.

O Processing 2, como ambiente de aprendizagem, oportunizou uma aprendizagem mais colaborativa. Como consequência, ocorreu uma melhora no rendimento na aula. Segundo a professora de matemática o Processing 2 auxiliou na compreensão da programação e tornou o ensino da matemática mais atraente e acessível. A cada experiência bem-sucedida, os alunos desenvolviam gosto pela descoberta, coragem para enfrentar desafios e para superá-los, desenvolvendo conhecimentos na direção de uma melhor reflexão sobre o pensamento matemático. Desta forma, atividades que envolvem aprendizagem da programação para resolução de problemas permitem que os alunos aprendam participando ativamente da experiência de aprendizagem.

A interação por parte dos alunos entre a teoria e prática, ultrapassou o simples “aprender fazendo”. Os alunos passaram a ter autonomia a partir do momento que puderam direcionar o andamento das aulas e os conteúdos a serem trabalhados pelo professor para possibilitar a resolução de seus desafios. Alguns estudantes se destacaram como líderes, e se transformaram numa espécie de mentor para ajudar outros integrantes do grupo, com o compromisso mútuo de equivalência entre todas as partes envolvidas nos processos sociais, construindo uma identidade que direciona para a solidariedade e ajuda mútua, fruto de uma prática que propicia a horizontalidade.

A programação é considerada uma habilidade difícil de aprender. No entanto, se for apoiada por estratégias de ensino adequadas e ferramentas de TI podem ser dominadas pelos alunos. Assim, este trabalho conseguiu identificar talentos naqueles que são ignorados ou taxados de ruins (baixo rendimento em matemática) e provou a tese de que com um pouco de coerência, dedicação e vontade podemos transformar a educação matemática, na busca da superação dos baixos índices de desempenho.

A programação e matemática são áreas fundamentais da educação em ciência da computação, e assim promovemos uma conexão entre o ensino de programação e a

aprendizagem matemática, possibilitando aos estudantes enxergarem relevância no conhecimento matemático. A linguagem de programação Processing 2 por possuir uma sintaxe simples, se comparada com muitas outras linguagens populares, veio viabilizar sua utilização com crianças do ensino fundamental com baixo rendimento na disciplina.

Os resultados indicam vários benefícios da introdução do ensino de programação na aprendizagem matemática para alunos do ensino fundamental com baixo rendimento na disciplina e neste final deixamos-lhes uma sugestão, a de usarem uma linguagem de programação visual, como o Processing 2, mas também o professor pode se atentar a desenvolver seus próprios métodos que levem em consideração a forma interativa, não linear, informal e reflexiva.

Este estudo abordou questões que ainda necessitam ser aprofundadas na área do desenvolvimento do raciocínio matemático, e, possivelmente, trará significativas contribuições para área de Educação Matemática em futuros trabalhos relacionados ao ensino de programação de computadores e aprendizagem matemática.

Alguns questionamentos ainda permanecem, e como continuidade para este estudo, propõe-se investigar a articulação entre o pensamento matemático e o pensamento computacional, uma vez que essa ponte parece não estar bem compreendida, em que encontramos alunos com carência ou ausência de pensamento computacional, mas com bom rendimento em matemática e, por outro lado, alunos com baixo rendimento em matemática, mas que são bons em programação. Recomendam-se também, replicações em outras escolas, com amostras maiores, com faixa etária diferente da empregada nesse estudo.

## REFERÊNCIAS

ADRIÃO, Theresa; GARCIA, Teise; BORGHI, Garcia; ARELARO, Lisete. Uma modalidade peculiar de privatização da educação pública: a aquisição de sistemas de ensino por municípios paulistas. **Educação e Sociedade**. Campinas: Educ. Soc, 2009. v 30. n. 108. Disponível em [www.scielo.br/pdf/es/v30n108/a0930108.pdf](http://www.scielo.br/pdf/es/v30n108/a0930108.pdf)>. Acesso em 10 mar. 2015.

ALMEIDA, Maria Elizabeth Bianconcini. **Informática e formação de professores**. Brasília: Ministério da Educação, 2000a, 1 v.

ALMEIDA, Maria Elizabeth Bianconcini. **Informática e formação de professores**. Brasília: Ministério da Educação, 2000b, 2 v.

ALVES, Leonardo M. **Educação Liberal**: formação além dos limites profissionais. Disponível em: <<https://ensaiosnotas.wordpress.com/2012/12/07/educacao-liberal-interdisciplinaridade-para-a-vida>>. Acesso em 03 jun. 2015.

ANDERSON. John R.; REDER. Lynne M.; SIMON. Herbert A. **Situated Learning and Educational**. 1996. Disponível em <<http://sites.harvard.edu/fs/docs/icb.topic122288.files/Andersonetal1996.pdf>>. Acesso em: 10 mar. 2015.

ARDICHVILLI, Alexander; WENTLING, Tim. **Motivation and barriers to participation in virtual knowledge sharing in communities of practice**. 2003. Disponível em <<http://www2.warwick.ac.uk/fac/soc/wbs/conf/olkc/archive/oklc3/id78.pdf>>. Acesso em: 12 jul. 2015.

BARDIN, Laurence. **Análise de Conteúdo**. Lisboa, Portugal: Edições 70, 2011.

BATISTA, Adriana Correia Almeida. **Aprendizagem Situada em uma comunidade de aprendizes de matemática de uma escola pública**. 2014, 223p. Tese (Doutorado em Educação). Programa de Pós-Graduação em Educação da Unicamp, Área de concentração: Ensino e Práticas Culturais, Eixo: Prática Pedagógica em Matemática. Campinas, SP; Universidade Estadual de Campinas.

BLOOMBERG. **Why Schools in England Are Teaching 5-Year-Olds How to Code**. (2014). Disponível em: <<http://www.bloomberg.com/news/print/2014-10-15/why-schools-in-england-are-teaching-5-year-olds-how-to-code.html>>. Acesso em: 10 jan. 2015.

BORBA, Marcelo de Carvalho; PENTEADO, Miriam Godoy. **Informática e educação matemática**. Belo Horizonte: Autêntica, 2001.

BRASIL. **Parâmetros curriculares nacionais: Matemática Ensino Fundamental**. 3. ed. Brasília: MEC/SEF, 2001.

CARGOBOT. (2015). **Cargo-Bot**: o primeiro jogo programando inteiramente em Ipad. Disponível em: < <http://twolivesleft.com/CargoBot/>>. Acesso em: 21 jan. 2015.

CARVALHO, Isamir M.; STEIL, Andrea Valéria. **Aprendizagem Tradicional e Aprendizagem Situada em Organizações**: uma Reflexão relacionada aos Tipos de Aprendizagem. In: Eneo 2012 - Encontro de Estudos Organizacionais, 2012, Curitiba/PR. Anais EnEO 2012. Rio de Janeiro: ANPAD, 2012. v. 1.

CODE. (2015). **Organização sem fins lucrativos que ensina programação a pessoas de todas as idades**. Disponível em < <https://code.org/>> . Acesso em: 22 de jul. 2015.

CODECADEMY. (2015). **Aulas gratuitas de codificação em linguagens de programação**. Disponível em < <https://www.codecademy.com/pt/>> . Acesso em: 22 jul. 2015.

CODECLUBBRASIL. (2015). **Site sem fins lucrativos que dá a crianças a oportunidade de aprender a programar**. Disponível em <<http://codeclubbrasil.org/>>. Acesso em: 22 jul. 2015.

CODEMONKEYISLAND. (2015). **Monkey Island**. Disponível em: <<http://codemonkeyplanet.com/>>. Acesso em: 22 jul. 2015.

CORTELLA, Mário Sérgio. **Educação, Escola e Docência**: novos tempos, novas atitudes. São Paulo: Cortez, 2014, 126 p.

CSIKSZENTMIHALYI, Mihaly. **Flow The Psychology Of Optimal Experience**. New York: Collins Publishers, 2009.

DAVENPORT, Thomas H.; PRUSAK, Laurence. **Working knowledge: How organizations manage what they know**. Boston: Harvard Business School Press, 2000.

DUGUID, Paul. **The Art of Knowing**: Social and Tacit Dimensions of Knowledge and the Limits of the Community of Practice. 2005. Disponível em: < [http://people.ischool.berkeley.edu/~duguid/articles/Art\\_of\\_knowing.pdf](http://people.ischool.berkeley.edu/~duguid/articles/Art_of_knowing.pdf)>. Acesso em: 22 jul. 2015.

EDUSCRATCH. (2015). **Página do Scratch para apoio, formação e partilha de experiências na comunidade educativa**. Disponível em: < <http://eduscratch.dge.mec.pt/>>. Acesso em: 21 fev. 2015.

FERNANDES, Elza Maria dos Santos. **Aprender matemática para viver e trabalhar no nosso mundo**. 2004, 479p. Tese de doutorado: Departamento de Educação da Faculdade de Ciências de Lisboa. APM. Lisboa.

FIGUEIREDO, Antônio Dias. (2012). **Moocs**: virtudes e limitações. Disponível em: <<http://moocad.blogspot.com.br/2012/10/moocs-virtudes-e-limitacoes.html>>. Acesso em: 17 mar. 2015.

GATTI, Bernadete Angelina. **Grupo focal na pesquisa em ciências sociais e humanas**. 1. ed. Brasília: Líber Livro, 2005.

GEE, James Paul. (2008). **Game-Like Learning: an example of situated learning and implications for Opportunity to Learn.** Disponível em: <[http://jamespaulgee.com/pdfs/Situated %20learning%20and%20Games.pdf](http://jamespaulgee.com/pdfs/Situated%20learning%20and%20Games.pdf)>. Acesso em: 17 ago. 2015.

HALVERSON, Richard. **Rethinking education in the age of technology.** New York: Teachers College Press, 2009.

HECKHAUSEN, Heinz. **The anatomy of achievement motivation.** New York, Academic Press, 1967.

HUNG, D. **Situated cognition and problem-based learning: implications for learning and instruction with technology.** 2002. Disponível em < <http://www.editlib.org/noaccess/9176>>. Acesso em: 15 mar. 2015.

INEP. **Relatório Nacional Pisa 2012: Resultados brasileiros.** (2014). Disponível em: <[http://download.inep.gov.br/acoes\\_internacionais/pisa/resultados/2014/relatorio\\_nacional\\_pisa\\_2012\\_resultados\\_brasileiros.pdf](http://download.inep.gov.br/acoes_internacionais/pisa/resultados/2014/relatorio_nacional_pisa_2012_resultados_brasileiros.pdf)>. Acesso em: 10 dez. 2015.

KDEedu. (2015). **Site do projeto educacional KDE.** Disponível em: <<https://edu.kde.org/>>. Acesso em: 05 jun. 2015.

KENSKI, Vani. Moreira. **Educação e tecnologias: o novo ritmo da informação.** Campinas, SP: Papyrus, 2007.

KODABLE. **Site oficial Kodable.** Disponível em: < <https://www.kodable.com/>>. Acesso em 19 jan. 2015.

KRESS, Gunther; LEEUWEN, Van. **Reading Images: The Grammar of Visual Design.** 1996. 2 ed. London: Routledge, 2006.

LABURU, Carlos Eduardo; CARVALHO, Marcelo de. Controvérsias construtivistas e pluralismo metodológico no ensino de ciências naturais. **Revista brasileira de pesquisa em educação em ciências.** v.1, n.1. 2001. Disponível em <<http://revistas.if.usp.br/rbpec/article/view/216/200>>. Acesso em: 10 mar. 2014.

LAVE, Jean.; WENGER, Etienne. **Situated Learning: Legitimate Peripheral Participation.** Cambridge: Cambridge University Press, 1991.

MARTINO, Luiz C. **De qual comunicação estamos falando?** Teoria da comunicação: conceitos, escolas e tendências. 5 ed. Petrópolis: Editora vozes, 2005. p. 11-25.

MARTINS, Amilton Rodrigo de Quadros. **Usando o Scratch para potencializar o pensamento criativo em crianças do ensino fundamental.** 2012, 113 f. Dissertação (Mestrado em Educação), Universidade de Passo Fundo, RS.

MATOS, João Filipe. **Logo na Educação Matemática: um estudo sobre as concepções e atitudes dos alunos.** 1991. 257 f. Dissertação. Universidade de Lisboa.

MATUI, Jiron. **Construtivismo: Teoria construtivista sócio-histórica aplicada ao ensino.** São Paulo: Editora Moderna, 1995.

McGONIGAL, Jane. **A realidade em jogo: porque os games nos tornam melhores e como eles podem mudar o mundo.** 1. ed. Rio de Janeiro: Bestseller, 2012.

MONTEIRO, Mário A. **Introdução à Organização de Computadores.** 5. ed. Rio de Janeiro: LTC, 2007.

OBAMA, Barack. (2013). **Don't Just Play on Your Phone, Program It.** Disponível em: <<http://m.whitehouse.gov/blog/2013/12/09/don-t-just-play-your-phoneprogram-it>>. Acesso em: 02 nov. 2015.

OTEC. (2007). **Situated Learning.** Disponível em <[http://otec.uoregon.edu/learning\\_theory.htm#SituatedLearning](http://otec.uoregon.edu/learning_theory.htm#SituatedLearning)> . Acesso em: 12 mar. 2015.

PAIS, Luiz Carlos. **Tecnologias informáticas e educação escolar.** 2. ed. Belo Horizonte: Autêntica, 2002.

PAPERT, Seymour. **Logo: computadores e educação.** São Paulo: Brasiliense, 1986.

PAPERT, Seymour. **A máquina das crianças: repensando a escola na era da informática.** ed. rev. Porto Alegre: Artmed, 2007.

PAPERT, Seymour. **Microworlds: transforming education.** Cambridge: Published Massachusetts, 1987. p. 79-94.

PAPERT, Seymour. **Mindstorms. Children, Computers and Powerful Ideas.** New York: Inc. Publishers, 1980.

PRENSKY, M. **Teaching digital natives: partnering for real learning.** California: Corwin, 2010. Disponível em <[http://marcprensky.com/wp-content/uploads/2013/04/Prensky-TEACHING\\_DIGITAL\\_NATIVES-Introduction1.pdf](http://marcprensky.com/wp-content/uploads/2013/04/Prensky-TEACHING_DIGITAL_NATIVES-Introduction1.pdf)>. Acesso em: 19 mar. 2015.

PRENSKY, M. **The True Twenty-first Century Literacy is Programming.** In: Edutopia, feb. 2008. Disponível em <<http://www.edutopia.org/literacy-computer-programming>>. Acesso em: 10 fev. 2015.

PRENSKY, Marc. **Listen to the Natives.** In: Educational Leadership. v. 63. n. 4. 2005. p.8-13. Disponível em <[http://www.ascd.org/ASCD/pdf/journals/ed\\_lead/el200512\\_prensky.pdf](http://www.ascd.org/ASCD/pdf/journals/ed_lead/el200512_prensky.pdf)>. Acesso em 15 mar. 2015.

PROCESSING. (2015). **Processing.** Disponível em: <<http://https://www.processing.org/>>. Acesso em: 03 jan. 2015.

REAS, Casey. FRY, Ben. **Getting Started with Processing.** Published by O'Reilly Media, Inc. Gravenstein Highway North, Sebastopol. 2010. Disponível em: <[http://media.espora.org/mgoblin\\_media/media\\_entries/406/Getting.Started.with.Processing.pdf](http://media.espora.org/mgoblin_media/media_entries/406/Getting.Started.with.Processing.pdf)>. Acesso em: 03 ago. 2014.

REAS, Casey. FRY, Ben. **Processing: a programing handbook for visual designers and artists.** Foreword by John Maeda. Massachusetts: MIT Press, 2007.

RESNICK, Mitchel. **Computer as Paintbrush: Technology, Play, and the Creative Society.** 2006. Disponível em: < <http://web.media.mit.edu/~mres/papers/playlearn-handout.pdf>>. Acesso em: 20 dez. 2014

RESNICK, Mitchel. **Learn to Code, Code to Learn.** 2013. Disponível em: < <http://web.media.mit.edu/~mres/papers/L2CC2L-handout.pdf>>. Acesso em: 20 dez. 2014.

RESNICK, Mitchel. **Scratch: Programming for All.** 2009. Disponível em: <<http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf>>. Acesso em: 20 dez. 2014.

RESNICK, Mitchel. **Vamos ensinar as crianças a escrever códigos.** 2012. Disponível em: <[http://www.ted.com/talks/mitch\\_resnick\\_let\\_s\\_teach\\_kids\\_to\\_code?language=pt-br#t-19466](http://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=pt-br#t-19466)>. Acesso em: 22 jan. 2015.

ROBOTTURTLES. (2015). **Site oficial do Robot Turtles.** Disponível em: < <http://www.robotturtles.com/>>. Acesso em: 09 jan. 2015.

ROBOZZLE (2015). **Site oficial do Robozzle.** Disponível em: <<http://www.robozzle.com>>. Acesso em: 22 jan. 2015.

RUSHKOFF, Douglas. **As 10 questões essenciais da era digital.** São Paulo: Editora Saraiva, 2012.

SÃO PAULO. Secretaria da Educação do Estado. **Relatório Pedagógico Matemática SARESP 2014.** 2015. Disponível em < [http://file.fde.sp.gov.br/saresp/saresp2014/Arquivos/RELATORIO\\_PEDAGOGICO\\_MATEMATICA.pdf](http://file.fde.sp.gov.br/saresp/saresp2014/Arquivos/RELATORIO_PEDAGOGICO_MATEMATICA.pdf)>. Acesso em: 28 nov. 2015.

SETTI, M. DE O. G. O Processo de Discretização do Raciocínio Matemático na Tradução para o Raciocínio Computacional: Um Estudo de Caso no Ensino/Aprendizagem de Algoritmos. Tese (Doutorado em Educação). Curitiba: UFPR, 2009.

SILVA, Heloisa V. R. Corrêa. **Introdução à Programação de computadores.** 3. ed. Rio de Janeiro: Campus, 1989.

SOARES, Magda. **Alfabetização e letramento.** 2º ed. São Paulo: Contexto, 2004.

SOLIGO, Ângela. **Cada um aprende de um jeito.** Nova Escola. 159 ed. 2003. Disponível em: < <http://revistaescola.abril.com.br/formacao/cada-aprende-jeito-432311.shtml>>. Acesso em: 07 jan. 2014.

SQUIRE, Kurt D. **Video Game Literacy: A Literacy of Expertise.** New York: Lawrence Erlbaum, 2008.

STEIN, David. **Situated learning in adult education**. 1998. Disponível em <<http://www.ericdigests.org/1998-3/adult-education.html>>. Acesso em: 10 mar. 2015.

SVEIBY, Karl Erik; SIMON, Roland. **Collaborative climate and effectiveness of knowledge work - an empirical study**. 2002. Disponível em <<http://www.sveiby.com/articles/ccs>>. Acesso em: 12 mar. 2015

TENNANT, Mark. **Psychology and Adult Learning**. London: Routledge, 1997.

TECHCRUNCH. (2013). Jonh Biggs: **Light-Bot Teaches Computer Science With A Cute Little Robot And Some Symbol-Based Programming**. Disponível em: <<http://techcrunch.com/2013/06/26/light-bot-teaches-computer-science-with-a-cute-little-robot-and-some-symbol-based-programming/>>. Acesso em: 20 out. 2015.

TU, Chih-Hsiung. **The measurement of social presence in an online learning environment**. 2002. International Journal on E-learning. Disponível em: <<http://jan.ucc.nau.edu/~ct68/Publication/2002/IJEL12Tu.pdf>>. Acesso em: 09 jan. 2015.

VALENTE, José Armando. (org). **O computador na sociedade do conhecimento**. Campinas: Unicamp/NIED, 1999.

VALENTE, José Armando. **Computadores e conhecimento: repensando a educação**. Campinas: NIED, 1993.

VALENTE, José Armando. **O Professor no Ambiente Logo: formação e atuação**. Campinas: Gráfica da UNICAMP, 1996.

VALENTE, José Armando. **O uso inteligente do computador na educação**. Revista Pátio, Ano I, n. 1, mai-jul. 1997.

VAN DIJK, T.A. **Cognição, discurso e interação**. 4ªed. São Paulo: Editora Contexto, 2002.

VAN HIELE, Pierre M. **Structure and Insight: A Theory of Mathematics education**. 1986. Academic Press. KALEFF, A . M.M.R. (1998) Vendo e Entendendo Poliedros. Niterói: EdUFF, 209 p.

WENGER, Etienne. **Communities of practice and social learning systems: the career of a concept**. 2000. Disponível em < <http://wenger-trayner.com/wp-content/uploads/2012/01/09-10-27-CoPs-and-systems-v2.01.pdf>>. Acesso em: 10 mar. 2015.

WENGER, Etienne. **Communities of practice: Learning, meaning and identity**. New York: Cambridge University Press, 1998.

WILEY, David. **Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy**. 2000. Disponível em <<http://penta3.ufrgs.br/objetosaprendizagem>>. Acesso em: 20 dez. 2015.

WILEY, David. **The MOOC Misnomer**. 2012. Disponível em: <<http://opencontent.org/blog/archives/2436>>. Acesso em: 13 jul. 2015.

WILLINGHAM, Daniel T. **Por que os alunos não gostam da escola?** Respostas da ciência cognitiva para tornar a sala de aula atrativa e efetiva. Porto Alegre: Artmed, 2011.

WING, Jeannette M. Computational thinking and thinking about computing. **Philosophical Transactions of the Royal Society**, 2008. Disponível em: < <https://www.cs.cmu.edu/~CompThink/papers/Wing08a.pdf>>. Acesso em: 18 dez. 2015.

WING, Jeannette M. Computational thinking. **Communications of the acm**. v. 49. n. 3, 2006. Disponível em: < <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>>. Acesso em: 18 dez. 2015.

YOUNG, Michael. F. **Instructional design for situated learning**. Educational Technology Research and Development. 1993. p. 43-58. Disponível em: < <http://link.springer.com/article/10.1007/BF02297091>>. Acesso em: 20 jun. 2015.

## APÊNDICES

### Apêndice A – Termo de consentimento livre e esclarecido para pais de alunos



#### TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO PARA PAIS DE ALUNOS

Senhores Pais e/ou responsáveis

Gostaríamos de solicitar a colaboração de seu filho (a) para realizar a pesquisa de mestrado intitulada “**Programação Visual no Ensino de Matemática**” que será desenvolvida no Programa de Pós-Graduação em Educação para Ciência da UNESP de Bauru, tendo como orientador o prof. Dr. Wilson Massashiro Yonezawa do Departamento de Computação da Faculdade de Ciência.

O objetivo da pesquisa é realizar um estudo sobre questões relacionadas à aprendizagem dos alunos em matemática, tendo como instrumento didático-pedagógico a programação de computadores.

A participação nas oficinas agregará conhecimento extracurricular ao participante e estará contribuindo para pesquisas acerca da aprendizagem dos alunos.

As oficinas serão gravadas em áudio e vídeo, apenas para análise posterior do pesquisador. A identificação do aluno, imagens, áudios e vídeos serão mantidos em sigilo, estando apenas presentes na publicação do trabalho os resultados obtidos nesta pesquisa.

A participação é voluntária e a recusa em participar não irá acarretar em nenhuma forma de penalidade.

Colocamos à disposição para esclarecer eventuais dúvidas.

Atenciosamente,

**Eduardo Cardoso de Souza**  
Mestrando responsável pela pesquisa  
e-mail: eduardoc@fc.unesp.br  
telefone: (14) 3372-3308

**Prof. Dr. Wilson Massashiro Yonezawa**  
Orientador responsável  
email: yonezawa@fc.unesp.br  
telefone: (14) 3103-6079



Eu, ....., responsável pelo aluno (a)..... do 7º ano ....., concordo com a sua participação nas oficinas de programação em “Processing 2”, que será realizada no Laboratório de Informática Educacional da EMEF Prof. “Arnaldo Moraes Ribeiro” e me comprometo a enviá-lo no contraturno de aula nas datas e horários definidos.

As oficinas ocorrerão no período da manhã, com duração de 3 horas cada. Está prevista a realização de 08 (oito) oficinas, nas seguintes datas:

1ª oficina	23/02/2015	09:00 as 11:30 h
2ª oficina	24/02/2015	08:00 as 10:30 h
3ª oficina	25/02/2015	08:00 as 10:30 h
4ª oficina	27/02/2015	08:00 as 10:30 h
5ª oficina	02/03/2015	09:00 as 11:30 h
6ª oficina	03/03/2015	08:00 as 10:30 h
7ª oficina	04/03/2015	08:00 as 10:30 h
8ª oficina	06/03/2015	08:00 as 10:30 h

OBS: Em caso de alteração nas datas e horários será comunicado com antecedência.  
Santa Cruz do Rio Pardo, 20 de fevereiro de 2015.

.....  
Assinatura

RG .....



Eu, ....., responsável pelo aluno (a)..... do 7º ano ....., concordo com a sua participação nas oficinas de programação em “Processing 2”, que será realizada no Laboratório de Informática Educacional da EMEF Prof. “Arnaldo Moraes Ribeiro” e me comprometo a enviá-lo no contraturno de aula nas datas e horários definidos.

As oficinas ocorrerão no período da manhã, com duração de 3 horas cada. Está prevista a realização de 08 (oito) oficinas, nas seguintes datas:

1ª oficina	23/02/2015	09:00 as 11:30 h
2ª oficina	24/02/2015	08:00 as 10:30 h
3ª oficina	25/02/2015	08:00 as 10:30 h
4ª oficina	27/02/2015	08:00 as 10:30 h
5ª oficina	02/03/2015	09:00 as 11:30 h
6ª oficina	03/03/2015	08:00 as 10:30 h
7ª oficina	04/03/2015	08:00 as 10:30 h
8ª oficina	06/03/2015	08:00 as 10:30 h

OBS: Em caso de alteração nas datas e horários será comunicado com antecedência.  
Santa Cruz do Rio Pardo, 20 de fevereiro de 2015.

.....  
Assinatura

RG .....

## Apêndice B – Questionário semiestruturado

### QUESTIONÁRIO

Q1) INICIAIS DO NOME:	Q11) QUANDO VÃO AO LABORATÓRIO DE INFORMÁTICA, QUAIS AS ATIVIDADES QUE FREQUENTEMENTE REALIZAM? QUAIS APLICATIVOS UTILIZAM?
Q2) IDADE: _____ anos	Q12) VOCÊ GOSTA DE MATEMÁTICA? ( )SIM ( )NÃO
Q3) SÉRIE/ANO: _____	Q13) COMO VOCÊ GOSTA DE APRENDER MATEMÁTICA?
Q4) QUANTO TEMPO VOCÊ ESTUDA NESTA ESCOLA? _____	Q14)QUAL O CONCEITO VOCÊ ATRIBUI PARA SEU CONHECIMENTO EM MATEMÁTICA NA RESPECTIVA SÉRIE. ( ) INSUFICIENTE ( ) BOM ( ) MUITO BOM
Q5) SEXO: ( )MASCULINO ( )FEMININO	Q15) QUAL FOI SUA MÉDIA EM MATEMÁTICA? _____
Q6) MORA COM OS PAIS? ( )SIM ( )NÃO	Q16) VOCÊ GOSTA DE INGLÊS? ( )SIM ( )NÃO
Q7) VOCÊ TEM COMPUTADOR COM ACESSO À INTERNET EM CASA? ( )SIM ( )NÃO ( ) SÓ TENHO COMPUTADOR	Q17) O QUE VOCÊ ENTENDE POR PROGRAMAÇÃO DE COMPUTADORES?
Q8) QUANTAS HORAS EM MÉDIA VOCÊ UTILIZA COMPUTADOR DIARIAMENTE?	Q18) VOCÊ ACHA IMPORTANTE TER CONHECIMENTO EM PROGRAMAÇÃO? POR QUE?
Q9) QUANTAS PESSOAS QUE MORAM EM SUA CASA COMPARTILHAM O USO DO COMPUTADOR?	Q19) VOCÊ JÁ TEVE CONTATO COM ALGUMA LINGUAGEM DE PROGRAMAÇÃO ANTERIORMENTE? ( ) SIM (.....) ( ) NÃO
Q10) COM QUAL FREQÜÊNCIA SEU PROFESSOR DE MATEMÁTICA LEVAM VOCÊS PARA O LABORATÓRIO DE INFORMÁTICA EDUCACIONAL?	Q20) QUAL SUA OPINIÃO SOBRE PROFESSORES QUE NÃO DOMINAM O USO DE NOVAS TECNOLOGIAS?

## Apêndice C – Dados tabulados sobre o perfil dos estudantes

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	*Q15
CA	12	7	7	M	S	S	5	4	NUNCA	ESCREVER TEXTO	S	USANDO CALCULADORA	B	5
KA	12	7	7	F	N	S	1	5	NUNCA	ESCREVER TEXTO	S	COM EXERCÍCIOS DE DESAFIOS	B	5
MB	12	7	7	M	N	S	5	5	NUNCA	ESCREVER TEXTO	S	QUANDO A PROFESSORA MOSTRA ALGO ÚTIL	B	5
NB	12	7	7	F	N	S	3	5	NUNCA	ESCREVER TEXTO	S	NO COMPUTADOR	B	**
TS	12	7	2	F	S	S	2	3	NUNCA	ESCREVER TEXTO	S	COM JOGOS DA APOSTILA	B	5
VH	13	7	8	M	N	S	2	4	NUNCA	ESCREVER TEXTO	S	USANDO A BARRINHA CUISENAIRE	B	5

\* Notas informadas pelos estudantes

\*\* Não informado

	Q16	Q17	Q18	Q19	Q20
CA	N	CRIAÇÃO DE UTILIDADE	AJUDA EM MUITA COISA	N	INCAPACITADO
KA	N	DAR COMANDOS AO COMPUTADOR	UM CONHECIMENTO DIFERENTE	N	MUITO BURRO
MB	N	CONFIGURAR	PARA SABER CRIAR	N	"NORMAL" PORQUE NÃO É DA SUA GERAÇÃO
NB	N	ENSINAR O COMPUTADOR	CONHECER MAIS SOBRE COMPUTAÇÃO	N	RUIM
TS	N	FAZER UM MONTE DE CÓDIGO	TEM MUITA COISA QUE USA PROGRAMAÇÃO	N	DEVERIA SE CAPACITAR, FAZER CURSOS
VH	N	CONSTRUIR ALGO QUE QUERO	O FUTURO VAI EXIGIR	N	COM POUCA INTELIGÊNCIA

Q1-Iniciais do nome

Q2-Idade

Q3-Série/ano

Q4-Quanto tempo estuda nessa escola?

Q5-Sexo

Q6-Mora com os pais?

Q7-Tem computador com acesso à internet em casa?

Q8-Quantas horas em média você utiliza computador diariamente?

Q9-Quantas pessoas que moram em sua casa compartilham o uso do computador?

Q10-Com qual frequência seu professor de matemática leva sua classe para o laboratório de Informática educacional?

Q11-Quando vão ao laboratório de informática com outros professores, quais as atividades que frequentemente realizam?

Q12-Você gosta de matemática?

Q13-Como você gosta de aprender matemática?

Q14- Qual conceito você atribui para seu conhecimento em matemática?

Q15-Qual é sua média em matemática?

Q16-Você gosta de inglês?

Q17-O que você entende por programação de computadores?

Q18-Você acha importante ter conhecimento em programação? Por quê?

Q19-Você já teve contato com algum tipo de linguagem de programação anteriormente?

Q20-Qual sua opinião sobre professores que não dominam o uso de novas tecnologias?