

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA NA EDUCAÇÃO

ANUAR DAIAN DE MORAIS

O DESENVOLVIMENTO DO RACIOCÍNIO CONDICIONAL A PARTIR DO
USO DE TESTE NO *SQUEAK ETOYS*

Porto Alegre

2016

ANUAR DAIAN DE MORAIS

**O DESENVOLVIMENTO DO RACIOCÍNIO CONDICIONAL A PARTIR DO
USO DE TESTE NO *SQUEAK ETOYS***

Tese apresentada ao Programa de Pós-Graduação em Informática na Educação da Universidade Federal do Rio Grande do Sul como requisito obrigatório para obtenção do título de Doutor em Informática na Educação.

Orientadora: Léa da Cruz Fagundes

Coorientador: Marcus Vinícius de Azevedo Basso

Linha de Pesquisa: Interfaces Digitais em Educação, Arte, Linguagem e Cognição.

Porto Alegre

2016

DEDICATÓRIA

Dedico este trabalho às mulheres da minha vida: Cora, Larissa, Bianca e Nílce (in memoriam) e ao meu pai Osmar por me ensinarem os segredos da vida.

AGRADECIMENTOS

À minha orientadora Léa da Cruz Fagundes que acreditou em mim desde a minha graduação, possibilitando diversas oportunidades de aprendizagens ao longo da minha formação profissional e acadêmica. Agradeço pela confiança e liberdade depositadas em mim, por partilhar comigo suas ideias, conhecimentos e experiências que sempre me rejuvenescem e me inspiraram. Manifesto minha gratidão pelo carinho dirigido a mim e a minha família.

Ao meu professor e coorientador Marcus Vinícius de Azevedo Basso, por sempre me indicar caminhos, por fornecer diferentes parcerias que sempre me ajudaram na minha formação profissional e acadêmica. Por me apresentar a Léa. Quero expressar a minha admiração pela sua competência profissional, tu és a melhor tradução da palavra professor e, por último, agradeço pela tua amizade e por seres quem és.

Aos colegas do PPGIE/UFRGS e do LEC/UFRGS, em especial à Aline Silva de Bona, ao Eduardo Britto Velho de Mattos e ao Bruno Fagundes Sperb por estarem sempre dispostos a discutir, dividir ideias e colaborar em trabalhos, atividades e artigos, além do afeto e da amizade.

Ao meu sogro Ivan e a minha sogra Denise pelo apoio e confiança que possibilitaram o meu desenvolvimento pessoal e profissional.

Aos colegas de trabalho que de alguma forma se interessaram e acompanharam este meu processo de aprendizagem.

Aos meus queridos alunos que me ensinam e aprendem junto comigo todos os dias.

Aos amigos e familiares pela compreensão nos momentos que não pude estar presente, presencialmente, em função deste trabalho.

RESUMO

A presente tese apresenta uma investigação acerca do desenvolvimento do raciocínio condicional, considerado um componente chave do pensamento lógico-dedutivo, em crianças e adolescentes que participaram de uma experiência de programação com o *software Squeak Etoys*. O desenvolvimento do raciocínio condicional é classificado em etapas relacionadas à composição e reversão de transformações que operam sobre a implicação, culminando com a plena reversibilidade que corresponde, na teoria piagetiana, à construção e mobilização do grupo de transformações INRC (Identidade, Negação, Recíproca, Correlativa). Tais etapas são identificadas a partir de entrevistas realizadas segundo o método clínico piagetiano, através da aplicação de três desafios de programação com complexidade crescente, cuja solução envolvia o uso da operação lógica da implicação. As entrevistas foram realizadas com oito crianças, com idades entre 10 e 16 anos, que cursavam as séries finais do Ensino Fundamental de duas escolas públicas. Com base nos dados, a análise revela a importância do pensamento combinatório, que permite aos adolescentes testarem, sistematicamente, todas as possibilidades de ordenamento e inclusão dos comandos sugeridos, e a obterem as conclusões lógicas adequadas, enquanto que as crianças mais novas não obtêm o mesmo êxito. Além disso, na tese é realizada uma discussão sobre a inclusão da escola numa cultura digital sob uma perspectiva construtivista de construção do conhecimento. Nesse contexto, a metodologia de projetos de aprendizagem foi apresentada como sendo adequada e o *software Squeak Etoys* despontou como uma possibilidade interessante de se desenvolver projetos e de promover a aprendizagem de matemática. Por último, neste trabalho também é realizado um debate sobre a importância de se aprender a programar na escola.

Palavras chave: *Squeak Etoys*, *Scratch*, Agrupamento, INRC, Epistemologia Genética, Educação Matemática, Aprendizagem de Lógica, Raciocínio Condicional, Pensamento Computacional, Ciência da Computação na Escola. Algoritmo, Cultura digital, Projetos de Aprendizagem.

ABSTRACT

The present thesis presents an investigation into the development of conditional reasoning, considered a key component of logical-deductive thinking, in children and adolescents who participated in a programming experience with the *software Squeak Etoys*. The development of conditional reasoning is classified into stages related to the composition and reversal of transformations that operate on the implication, culminating in the full reversibility that corresponds, in Piaget's theory, to the construction and mobilization of the Transformations INRC (Identity, Negation, Reciprocity and Correlation). These steps are identified from interviews conducted according to Piaget's clinical method, through the application of three programming challenges with increasing complexity, whose solution involved the use of the logical operation of the implication. The interviews were conducted with eight children aged 10-16, who attended the final series of the Elementary School of two public schools. Based on the data, the analysis revealed the importance of combining thinking, which allows teenagers to systematically test all the possibilities for ordering and inclusion of the suggested commands, and to obtain the appropriate logical conclusions, while younger children do not achieve the same results. Moreover, in the thesis a discussion is conducted on the inclusion of the school in a digital culture under a constructivist perspective of building knowledge. In this context, the methodology of learning through projects has been presented as being appropriate and the *Squeak Etoys software* has appeared as an interesting possibility of developing projects and promoting the learning of mathematics. Finally, in this study a debate is also conducted on the importance of learning to plan in the school.

Keywords: *Squeak Etoys*, *Scratch*, Collation, INRC, Genetic Epistemology, Mathematics Education, Logic Learning, Conditional Reasoning, Computational Thinking, Computer Science at School. Algorithm, Digital Culture, Learning Projects.

Índice de Siglas e Abreviaturas:

EAD – Ensino à Distância;

LEC/UFRGS – Laboratório de Estudos Cognitivos da Universidade Federal do Rio Grande do Sul;

NTD – Novas Tecnologia Digital;

OLPC – One Laptop per Child;

PCN – Parâmetros Curriculares Nacionais.

PIPA – Projeto Interdisciplinar de Pesquisa e Aprendizagem da Escola La Salle Pão dos Pobres.

PIXEL – Projeto do Colégio de Aplicação da Universidade Federal do Rio Grande do Sul;

UCA – Projeto um Computador por Aluno;

XO – Laptop de baixo custo e conhecido como “a máquina das crianças” fornecido pela OLPC;

Índice de tabelas

Tabela 1: Diferenças entre projetos de ensino, trabalho e aprendizagem.....	28
Tabela 2: Tabela verdade da operação ($p \cdot q$).....	59
Tabela 3: Tabela verdade das combinações possíveis de p	60
Tabela 4: Tabela verdade das possibilidades lógicas de p e q	61
Tabela 5: Diferentes representações das 16 possibilidades lógicas.....	62
Tabela 6: Transformações no INRC.....	65
Tabela 7: Desempenho dos entrevistados nos desafios.....	83

Índice de ilustrações

Figura 1: Projeto do Dynabook de Alan Kay.....	32
Figura 2: Crianças com o Dynabook.....	32
Figura 3: Demonstração do teorema de Pitágoras.....	33
Figura 4: Tela inicial do Etoys e seus objetos programáveis.....	36
Figura 5: Script para o carro reconhecer a pista rosa.....	42
Figura 6: Diagrama das áreas de conhecimento das teses e dissertações encontradas na revisão de literatura.....	45
Figura 7: Teorias de Conhecimento.....	52
Figura 8: Grupo de Deslocamentos.....	54
Figura 9: Inclusão hierárquica.....	57
Figura 10: As quatro combinações lógicas possíveis para a proposição p	60
Figura 11: Diagrama de Venn das três proposições lógicas.....	63
Figura 12: INRC da operação de conjunção.....	65
Figura 13: Projeto do tanque disparando um projétil.....	68
Figura 14: Script do projeto do tanque.....	70
Figura 15: Experimento de pesquisa: Desafio 1 A e B.....	71
Figura 16: Experimento de pesquisa: Desafio 2 A e B.....	72
Figura 17: Experimento de pesquisa: Desafio 3 A e B.....	73

Figura 18: INRC da solução do Desafio 1.....	75
Figura 19: Aplicando as transformações correlativa e recíprocas.....	75
Figura 20: Aplicando as transformações: comutativa de R e C.....	75
Figura 21: Aplicando as transformações: comutativa de N e R.....	76
Figura 22: Aplicando as transformações C e N.....	76
Figura 23: Aplicando três transformações do INRC seguidas.....	76
Figura 24: A solução do desafio 2 não comporta o INRC.....	78
Figura 25: INRC da operação de implicação com conjunção.....	81
Figura 26: Script 1 da estudante L.....	85
Figura 27: Scripts do desafio 01 do estudante M.....	87
Figura 28: Estudante K procurando onde colocar o comando avançar.....	90
Figura 29: Estudante K com dúvidas sobre lugar do comando avançar.....	91
Figura 30: Scripts para o desafio 01 do Estudante J.....	92
Figura 31: Solução do desafio 1A do estudante N.....	93
Figura 32: Scripts de solução do desafio 1B do estudante N.....	95
Figura 33: Situações propostas ao estudante N.....	97
Figura 34: Script da solução do desafio 2 do estudante K.....	99
Figura 35: Scripts criados pelo estudante J para o desafio 02.....	99
Figura 36: Scripts criados pelo estudante N para o desafio 02.....	100
Figura 37: Scripts criados pelo estudante K para o desafio 03.....	104
Figura 38: Scripts criados pelo estudante J para o desafio 03.....	106
Figura 39: Entrevistador mostrando que é possível inserir um Teste dentro do outro.....	107
Figura 40: Scripts criados pelo estudante N para o desafio 03.....	108
Figura 41: Primeira versão do experimento com comidas de apenas duas cores.....	109
Figura 42: Primeira tentativa do estudante G.....	110
Figura 43: Scripts A e B que o estudante G criou para o desafio 01.....	111
Figura 44: Demais Scripts criados pelo Estudante G para o desafio 01.....	112
Figura 45: Primeiro script criado pela estudante H.....	114
Figura 46: Segundo script criado pela estudante H.....	114
Figura 47: Primeiro script criado pela estudante H para resolver o desafio 02.....	116
Figura 48: Scripts criados pela estudante H para o desafio 02B.....	117
Figura 49: Primeiros scripts criados pelo estudante G para o desafio 02B.....	119

Figura 50: Scripts C e D criados pelo estudante G para o desafio 02B.....	120
Figura 51: Scripts E e F criados pelo estudante G para o desafio 2B.....	121
Figura 52: Primeiro script criado pelo estudante G para o desafio 03.....	122
Figura 53: Segundo script construído pelo estudante G para o desafio 03.....	123
Figura 54: Estudante G explora diferentes posições para o segundo Teste.....	124
Figura 55: Scripts E e F criados pelo estudante G para o desafio 03.....	125
Figura 56: Script G criado pelo estudante G para o desafio 03.....	126
Figura 57: Scripts H e I construídos pelo estudante G para o desafio 03.....	127
Figura 58: Primeiro Script criado pela estudante H para o desafio 03.....	128
Figura 59: Segundo script construído pela estudante H.....	129
Figura 60: Script proposto pelo entrevistador à estudante H.....	130
Figura 61: A estudante H explora os comandos sugeridos no desafio 03B.....	131
Figura 62: Estudante H estuda a possibilidade de colocar um Teste dentro do outro.....	133
Figura 63: Primeira sequência de script criada pelo estudante F para o desafio 03.....	136
Figura 64: Scripts F, G e i criados pelo estudante G para o desafio 03.....	137
Figura 65: Scripts que representam o INRC no desafio 01.....	144
Figura 66: Diagramas de Venn da Negação e da Incompatibilidade gerada pelo script.....	144
Figura 67: Diagramas de Venn da Correlação e da disjunção gerada pelo script.....	144
Figura 68: Scripts que traduzem o INRC no desafio 03.....	145
Figura 69: Polígonos e desenhos geométricos ao estilo LOGO.....	159
Figura 70: Script de um quadrado.....	160
Figura 71: Construtor de um polígono genérico.....	160
Figura 72: Construtor de cataventos.....	161
Figura 73: Algoritmos recursivos: árvore fractal.....	162
Figura 74: A matemática ao brincar com carros virtuais.....	164
Figura 75: A matemática dos carros inteligentes.....	164
Figura 76: Janela do comando Teste.....	165
Figura 77: Aceleração e velocidade no Etoys.....	165
Figura 78: Modelo computacional no Etoys da experiência de gravidade.....	166
Figura 79: Construções geométricas feitas no Dr. Geo II do Etoys.....	167
Figura 80: Três imagens em sequência da animação Avião na Pista.....	176
Figura 81: Scripts da animação Avião na Pista.....	177

Figura 82: Duas animações de relógios diferentes.....	179
Figura 83: Script para o carro reconhecer a pista rosa.....	181
Figura 84: Script para reconhecer a pista rosa.....	182
Figura 85: Ordem dos comandos no script.....	183
Figura 86: Script estudante B.....	184
Figura 87: Teste apresentado para os estudantes.....	185
Figura 88: Bug apresentado pelos estudantes.....	186
Figura 89: Script 1 proposto pelo primeiro professor.....	187
Figura 90: Script 2 proposto pelo segundo professor.....	187
Figura 91: Scripts equivalentes aos anteriores.....	188
Figura 92: Script para controlar o carro com o teclado.....	189
Figura 93: Script para controlar o carro com o teclado.....	189
Figura 94: Modificação no script feita pelo estudante.....	191
Figura 95: Script com a operação lógica de conjunção.....	193
Figura 96: Script do Jogo da Cobra.....	194

SUMÁRIO

1. INTRODUÇÃO.....	15
2. A CULTURA DIGITAL.....	18
2.1 A CULTURA EM REDE.....	19
2.2 O DIGITAL NA ESCOLA.....	20
2.3 DA CULTURA INDUSTRIAL À CULTURA DIGITAL.....	22
3. PROJETOS DE APRENDIZAGEM.....	25
3.1 PROBLEMATIZANDO OS PAs.....	29
4. O SQUEAK ETOYS: O DIGITAL NA ESCOLA.....	31
4.1 ETOYS E MATEMÁTICA.....	38
4.2 EXPERIÊNCIA-PILOTO.....	40
5. DEFINIÇÃO DO PROBLEMA DE PESQUISA E REVISÃO DA LITERATURA....	44
5.1 PROBLEMA DE PESQUISA.....	44
5.2 REVISÃO DA LITERATURA.....	44
5.3 METODOLOGIA DA PESQUISA.....	49
6. A LÓGICA NA TEORIA PIAGETIANA.....	51
6.1 A TEORIA PIAGETIANA.....	51
6.2 OS ESTÁDIOS DO DESENVOLVIMENTO E AS ESTRUTURAS LÓGICAS.....	53
6.2.1 O sensório-motor e o grupo de Deslocamentos.....	54
6.2.2 <i>O pré-operatório: o período pré-lógico.....</i>	<i>55</i>
6.2.3 Operatório concreto e a estrutura de agrupamento.....	55
6.2.4 O operatório formal – Da lógica das proposições ao INRC.....	58
6.2.4.1 As 16 possibilidades lógicas.....	61
6.2.4.2 Grupo INRC.....	64
7. EXPERIMENTO LÓGICO NO ETOYS.....	67
7.1 DESCREVENDO O EXPERIMENTO.....	71
7.2 O INRC NOS DESAFIOS DO EXPERIMENTO DE PESQUISA.....	73
7.2.1 Desafio 1: A implicação simples.....	73
7.2.2 Desafio 2: Implicação com disjunção.....	76
7.2.3 Desafio 3: Implicação com conjunção.....	78
8. ANÁLISE DOS RESULTADOS.....	82
8.1 OBSERVAÇÕES INICIAIS SOBRE OS DESAFIOS.....	82

8.2 NÍVEL I.....	84
8.2.1 Entrevista com estudante L (10 anos).....	84
8.3 NÍVEL II A.....	86
8.3.1 Entrevista com o estudante M (10 anos).....	86
8.3.1.1 Desafio 01.....	86
8.3.1.2 Desafio 02.....	89
8.4 NÍVEL II B.....	89
8.4.1. Desafio 01.....	89
8.4.1.1 Estudante K (10 anos).....	89
8.4.1.2 Estudante J (10 anos).....	91
8.4.1.2 Estudante N (10 anos).....	93
8.4.2. Desafio 02.....	98
8.4.2.1 Estudante K (10 anos).....	98
8.4.2.2 Estudante J (10 anos).....	99
8.4.2.3 Estudante N (10 anos).....	100
8.4.3 Desafio 03.....	101
8.4.3.1 Estudante K (10 anos).....	101
8.4.3.2 Estudante J (10 anos).....	104
8.4.3.3 Estudante N (10 anos).....	108
8.5 NÍVEL III A.....	108
8.5.1 Desafio 01.....	109
8.5.1.1 Estudante G (16 anos).....	109
8.5.1.2 Estudante H (15 anos).....	113
8.5.2 Desafio 02.....	115
8.5.2.1 Estudante H (15 anos).....	115
8.5.2.2 Estudante G (16 anos).....	118
8.5.3 Desafio 03.....	121
8.5.3.1 Estudante G (16 anos).....	122
8.5.3.2 Estudante H (15 anos).....	128
8.6 NÍVEL III B.....	135
8.6.1 Entrevista com o Estudante F (14 anos).....	135
8.7 O DESENVOLVIMENTO DO RACIOCÍNIO CONDICIONAL QUANDO SE	

APRENDE A PROGRAMAR.....	139
8.8 REPENSANDO O EXPERIMENTO.....	142
9. CONSIDERAÇÕES FINAIS.....	146
REFERÊNCIAS.....	149
ANEXOS.....	157
ANEXO A.....	158
1. <i>ETOYS</i> E MATEMÁTICA.....	158
1.1 Tartarugas no <i>Etoys</i>	158
1.2 Introdução à Álgebra.....	159
1.3 Triângulos e trigonometria.....	161
1.4 Fractais e algoritmos recursivos.....	161
1.5 Matemática dos movimentos.....	162
1.6 Geometria dinâmica.....	166
ANEXO B.....	169
1. PILOTO: OFICINA DE <i>ETOYS</i> NO CAP.....	169
1.1 ALFABETIZAÇÃO BÁSICA DE <i>ETOYS</i>	170
1.2 A AUTORIA NA CULTURA DIGITAL: GAMES E ANIMAÇÕES.....	171
1.2.1 Os Games e a Cultura Digital.....	171
1.2.2 Autoria e as Animações Quadro a Quadro.....	172
1.3 CONCEITOS DE MATEMÁTICA NO <i>ETOYS</i>	175
1.3.1 Avião na Pista.....	175
1.3.2 Dança das Mimosas.....	178
1.4 DESENVOLVIMENTO DO RACIOCÍNIO LÓGICO-MATEMÁTICO ATRAVÉS DO USO DE TESTES NO <i>ETOYS</i>	180

1. INTRODUÇÃO

Num primeiro momento esta pesquisa emergiu de dois desejos: o primeiro era o de produzir estudos que pudessem contribuir para a inclusão da escola na cultura digital, cujas práticas pedagógicas promovessem a formação de cidadãos capazes de compreender e resolver problemas do seu dia a dia. No contexto da cultura digital, isso significa formar pessoas capazes de compreender e produzir tecnologia ao invés de apenas consumi-la. Diante de tal demanda defende-se que a concepção construtivista de construção de conhecimento atende às necessidades dessa cultura. Por conseguinte, a metodologia de Projetos de Aprendizagem desponta como proposta pedagógica adequada a tal paradigma.

O segundo era o desejo de investigar como se dá a aprendizagem de matemática nesse contexto da cultura digital. A partir disso, a utilização do *Squeak Etoys* surgiu como uma possibilidade pedagógica interessante, visto que tal *software* foi criado para ser um micromundo no qual a matemática seria a sua principal linguagem. Mas de que maneira se pode aprender matemática com o *Squeak Etoys*? Quais seriam as possibilidades de aprendizagens que o seu uso promove?

Na busca por indícios que ajudassem a responder tais questões desenvolveu-se um estudo sobre as diferentes formas de uso do *Etoys* para se aprender matemática e que estão disponíveis em livros, artigos, dissertações e teses na forma digital ou impressa. Sendo assim, identificou-se duas maneiras de promover a aprendizagem da matemática pelos estudantes através do *Etoys*:

a) A Matemática *como objeto de estudo*: a partir de atividades que exploram diretamente objetos matemáticos, cujo tema é a própria matemática. Tem-se, como exemplos, as atividades de desenhar triângulos, polígonos, mosaicos, criar figuras simétricas, medianas no Dr. Geo, etc.

b) A Matemática *como um meio para se aprender*: que seria aquela que possibilita ao estudante produzir algo quando programa no *Etoys*, ou seja, um projeto cujo tema não seja matemático: uma simulação de decolagem de um avião, ou um game, etc.

Diante de tais possibilidades, foi desenvolvida uma experiência-piloto com uma turma da oitava série do ensino fundamental do Colégio de Aplicação da UFRGS, tendo como referência a matemática como um meio para se aprender e desenvolver projetos de aprendizagem no *Etoys*. A análise de tais dados indicou que era pertinente investigar sobre o processo de apropriação da lógica quando se aprende a programar. No caso do *Squeak Etoys*, isso significava investigar o desenvolvimento do raciocínio condicional (através da operação lógica da implicação) possibilitado

pelo uso do comando *Teste* do *Etoys*. Dessa maneira, após a experiência-piloto, esta pesquisa também visa contribuir para a discussão sobre a inserção da programação na escola básica e na compreensão desse processo de aprendizagem.

Nesses últimos anos observa-se que a importância de inserir a programação na escola básica tem ganhado força. Tal movimento não é novo, pois desde o fim da década de 1960 a filosofia LOGO, de Papert, tem inspirado uma série de pesquisadores, professores e escolas a investir em tal tema. No entanto, tem-se a impressão que desde o fim da década de 1990 até 2005 o interesse pela programação na escola básica foi ficando num segundo plano, pois a popularização da internet e o impacto das redes sociais no mundo têm norteado as pesquisas desde então.

Por outro lado, um tema é intrínseco ao outro, à medida que cresce a cultura em rede, torna-se necessário compreender e desenvolver tecnologias digitais. Assim nos últimos anos a programação escolar volta a ganhar força e identificam-se dois motivos para tal fato: a escassez de programadores no mercado de trabalho e, também, o desenvolvimento de novas linguagens de programação visual voltadas para crianças – com recursos multimídias mais avançados – tais como o *Squeak Etoys*, *Scratch*, etc. Tais linguagens foram e são disseminadas através do projeto OLPC realizado em diversos países e, no Brasil, pelo projeto UCA.

Como consequência desses dois fatores, começam a surgir uma série de clubes de programação no mundo inteiro, tais como Code Club, CLWB - “Community for Learning With Bits” (Comunidade para Aprender com Bits), etc. Tais clubes são fomentados por instituições de ensino, por governos e por empresas interessadas na formação de programadores. Mas a justificativa para a inserção da programação na escola vai além da formação de trabalhadores, abarcando o fato de que aprender a programar promove o chamado *Pensamento Computacional*, que desenvolve uma série de habilidades importantes para o desenvolvimento cognitivo das crianças tais como a resolução de problemas, a criatividade, a representação, etc.

Além disso, após revisar artigos e livros publicados sobre tal tema, observou-se que a maioria desses estudos volta-se para a análise dos algoritmos dos projetos desenvolvidos pelos estudantes ou dos conceitos de diferentes disciplinas que se pode aprender. Porém não há uma análise específica da lógica envolvida e de como ocorre tal processo. Será que todas as crianças conseguem utilizar as operações lógicas corretamente? Será que elas compreendem da mesma forma? Afinal, como se dá esse desenvolvimento? Dessa maneira, para compreender tal processo, os estudos de lógica piagetiana serviram como fundamentação teórica para esta pesquisa.

Neste contexto, esta tese apresenta o seguinte problema da pesquisa: *Como ocorre o*

desenvolvimento do raciocínio condicional dos estudantes quando se aprende a programar através do uso do comando Teste do Squeak Etoys numa perspectiva construtivista de aprendizagem?

Sendo assim, inspirado nos testes piagetianos, foi desenvolvido um experimento constituído de três desafios que envolviam situações de complexidade gradativa e que exigiam a utilização e compreensão da operação lógica da implicação na sua solução. Tal experimento foi aplicado através de entrevistas, inspiradas no método clínico piagetiano, com oito estudantes de idades variando entre 10 e 16 anos. Nessas entrevistas foi produzido um conjunto de dados, cuja análise contribuiu para a compreensão do desenvolvimento do raciocínio condicional dos estudantes. Pensa-se que qualquer professor ou pesquisador pode reproduzir tal experimento, seja como uma atividade pedagógica ou como instrumento de pesquisa para verificar, validar ou ampliar os dados e resultados obtidos nesta tese.

Para a análise de dados foi necessário utilizar os estudos de Piaget sobre o desenvolvimento das estruturas lógicas no desenvolvimento humano como fundamentação teórica. Embora a análise dos dados esteja centrada na estrutura lógica de Agrupamento e no Grupo INRC que, segundo Piaget, caracterizam a diferença entre o pensamento da criança e do adolescente; neste trabalho os demais estádios do desenvolvimento são apresentados também sobre tal ótica.

Por fim, a partir da análise dos dados foi possível identificar e propor uma categorização em níveis cognitivos do desenvolvimento do raciocínio condicional quando se aprende a programar, seja no *Etoys* ou em qualquer linguagem visual de programação. Nos próximos capítulos serão detalhados os conteúdos apresentados nesta introdução, que teve como objetivo expor de forma geral a trajetória desta pesquisa, a fundamentação teórica, a metodologia, os dados, a análise e os resultados obtidos nesta tese.

2. A CULTURA DIGITAL

Quando se reflete sobre a estrutura escolar vigente, pode-se identificar que ela ainda está baseada num modelo de cultura industrial. Neste contexto, a escola possui a função de transmitir às novas gerações os conhecimentos culturalmente desenvolvidos e acumulados pela humanidade.

Para que essa transmissão seja potencializada, a educação passa a ter uma organização similar à linha de produção industrial, no qual as áreas do conhecimento foram separadas e organizadas em disciplinas. Nessa lógica, tem-se um modelo de educação voltado para o ensino, em que cabe ao professor a função ativa de selecionar, organizar e transmitir o conhecimento; já aos alunos cabe o papel de recepção passiva daquilo que lhes é ensinado.

Em contrapartida a esses fatos, no prefácio do livro “O computador Portátil na Escola”, D'Ambrósio et al. (2011) afirma que a escola é para os alunos um ambiente irreal, visto que os meios digitais estão tão presentes na nossa cultura que o que não é digital, é fantasia. Negroponte (1995) assegura que vivemos numa era digital e que é impossível negar quando se analisa como a humanidade tem se modificado desde o advento dos computadores e da internet. Segundo Battro e Percival (2007) já foi comprovado que existem muitos indivíduos, especialmente as crianças, de diferentes culturas e níveis sociais que possuem uma facilidade impressionante para utilizar computadores, principalmente quando elas têm acesso à internet. Para exemplificar tal fato, podemos citar a experiência “O Buraco no Muro” de Mitra e Rana (2001). De acordo com Battro e Percival (2007), tais habilidades também são observadas em pessoas com deficiências de todos os tipos e, em função desses casos extremos e marginais, tais fatos indicam a existência de uma inteligência digital de alcance universal. Segundo ele, não se trata de um desenvolvimento genético (que venha codificado nos genes da espécie), mas de um desenvolvimento epigenético do potencial humano de cada indivíduo.

Desta forma, para D'Ambrósio et al. (2011) a educação deve ser reconceituada e seguir de dois objetivos maiores:

- 1) Dar oportunidade para que todos desenvolvam seu potencial criativo.
- 2) Preparar as novas gerações para o exercício pleno de cidadania.

Ele ainda acrescenta que nenhum desses objetivos pode ser atingido sem a plena inclusão digital (D'Ambrósio et al., 2011, p. 09).

Mas, afinal, o que significa realizar tal inclusão digital? Apenas distribuir computadores com

acesso à internet é suficiente? Como afirma Morais et al. (2011), não. Para esses autores é preciso que haja uma inclusão da escola na cultura digital, mas para que isso ocorra é necessário que ela passe por uma mudança de paradigma educacional. Para isso é importante compreender o que é a *cultura em rede* e qual a sua atuação na sociedade.

2.1 A CULTURA EM REDE

Já faz tempo que as redes influenciam a nossa cultura. Ao refletir um pouco sobre o assunto, vê-se que nosso cotidiano é suprido por um grande número de redes de serviços, tais como: o abastecimento de água e esgoto, de energia elétrica, meios de transporte e comunicação. Todas essas redes são reais e influenciam a forma de se comunicar, de pensar e de produzir significados. Sendo assim, para Parente (2004), antes mesmo do surgimento do chamado Ciberespaço, as redes já influenciavam na forma como o indivíduo se relaciona com o mundo social, dando significado às crenças e valores compartilhados em tal dimensão e que constituirão a experiência histórica e coletiva dos grupos e populações.

Segundo esse autor, no momento em que as Novas Tecnologias Digitais (NTD) passaram a ter um papel estruturante na nova Ordem Mundial, nada escapa à rede, tudo é definido a partir dela: a sociedade, o capital, o mercado, o trabalho, a arte, a guerra, etc. Vale ressaltar que, para Arruda (2004), as NTDs contemplam as tecnologias de informação e comunicação em rede ou não. De acordo com Lévy (1999) a rede formada pelas NTDs é o chamado Ciberespaço. Para este autor:

Ciberespaço é o novo meio de comunicação que surge da interconexão mundial de computadores, o termo especifica não apenas a infraestrutura material da comunicação digital, mas também o universo oceânico de informações que ela abriga, assim como os seres humanos que navegam e alimentam esse universo. (p. 17).

Além disso, um dos traços mais marcantes do ciberespaço é que, através de sua virtualidade, ele se constitui num espaço que está em todo lugar e em lugar nenhum. Nele a informação pode ser acessada por qualquer pessoa e de qualquer lugar. Além disso, qualquer usuário pode incluir novas informações ao ciberespaço e, dessa forma, ele se constitui num espaço livre, informal e descentrado. Tais características garantem recursos para a produção de conhecimento por parte de seus usuários.

Logo, acompanha-se o surgimento de uma nova cultura a *cultura digital* ou *cibercultura* que, segundo Lévy, é “o conjunto de técnicas (materiais e intelectuais), de práticas, de atitudes, de modos de pensamento e de valores que se desenvolvem juntamente com o crescimento do ciberespaço” (1999, p.17). Na *cultura digital*, as trocas entre os sujeitos são fundamentais para a

existência e manutenção da própria rede, na qual a interação entre as pessoas é mediada por diferentes recursos tecnológicos, que também interagem entre si. Dessa maneira, a interação social é potencializada pelo uso das NTDs e, conseqüentemente, a socialização é uma característica intrínseca à *cultura digital*. É relevante pontuar que a cibercultura, historicamente, é uma sintetização da relação da sociedade com as tecnologias digitais da informação e da comunicação. Segundo Lemos (2002), com a evolução das tecnologias digitais a ideia da cultura digital se sobrepõe, porque a cibercultura é a cultura em rede, é o impacto das NTDs e da conexão em rede da sociedade, além de ser um elemento do paradigma da complexidade, para Morin (2000), e da manifestação cultural da matemática, para D'Ambrosio (1999).

2.2 O DIGITAL NA ESCOLA

Na seção anterior foi discutido que a cultura digital é uma sintetização da relação da sociedade com as NTDs e que, assim, a interação social é potencializada por elas, porém é importante ressaltar que o movimento no sentido inverso também ocorre. Pois, à medida que a interação entre as pessoas é mediada por recursos tecnológicos, a demanda pelo desenvolvimento de novas tecnologias digitais também aumenta, portanto também é potencializada. No final do século passado Castells já dizia que na sociedade em rede a figura do usuário e do criador de NTDs poderiam se fundir.

[...] a difusão da tecnologia amplifica seu poder de forma infinita, à medida que os usuários apropriam-se dela e a redefinem. As novas tecnologias da informação não são simplesmente ferramentas a serem aplicadas, mas processos a serem desenvolvidos. Usuários e criadores podem tornar-se a mesma coisa. Dessa forma, os usuários podem assumir o controle da tecnologia [...]. Segue uma relação muito próxima entre os processos sociais de criação e manipulação de símbolos (a cultura da sociedade) e a capacidade de produzir e distribuir bens e serviços (as forças produtivas). Pela primeira vez na história, a mente humana é uma força direta de produção, não apenas um elemento decisivo no sistema produtivo. (Castells, 1999, p. 51).

De fato isso ocorre em alguns casos, mas não na mesma proporção em que a difusão da tecnologia ocorre. Sendo assim, nossa sociedade precisa cada vez mais que usuários de tecnologia sejam capazes de desenvolver tecnologias, ou seja, de programar. Conseqüentemente, há uma grande demanda por profissionais dessa área. A escassez de programadores no mercado de trabalho incentivou uma série de programadores famosos (como Bill Gates, Mark Zuckerberg, etc) a apoiar organizações como a *Code.org*, que tem como objetivo disseminar o ensino da Ciência da Computação para crianças e que são sem fins lucrativos. Dessa maneira, nesses últimos cinco anos começaram a surgir uma série de grupos ou clubes de programação no mundo inteiro, citando

algumas: *Programaê* (da Fundação Lemann), *Code Adademy*, *Code Club*, *CLWB - Community for Learning With Bits* (Comunidade para Aprender com Bits), etc.

Mas não é apenas a demanda advinda do mercado de trabalho que mobiliza tais movimentos, avalia-se que o surgimento de novas linguagens de programação voltadas para crianças (e que utilizam recursos multimídias mais avançados) também foram motivadoras. Pode-se citar, como exemplo, o *Squeak Etoys* e o *Scratch*. Tais linguagens foram (e são) disseminadas através de programas como *One Laptop per Children* (OLPC) e, no Brasil, pelo projeto UCA. Em tais projetos, o objetivo era o de distribuir *laptops* de baixo custo aos alunos da rede pública de ensino e formar professores para uso na escola. Em parceria com a OLPC, o *laptop* distribuído nas redes públicas era o XO, que trazia o *Squeak Etoys* ou o *Scratch* entre seus recursos.

O surgimento de novas e poderosas linguagens, bem como a inserção na cultura já era prevista por Papert (1986) em *Mindstorms*.

Numa sociedade onde haja computadores em abundância, as linguagens computacionais, que oferecem simultaneamente os meios para controlar os computadores e novas e poderosas linguagens capazes de descrever o pensamento, serão, sem dúvida, incorporadas à cultura (p. 124).

Além disso, ele já dizia que essas novas linguagens teriam um importante efeito na forma como descreveríamos nós mesmos e nossa aprendizagem, portanto usaríamos a programação como uma maneira de expandir a linguagem. Nesse sentido, a justificativa para a inserção da programação na escola vai além da formação profissional.

Para apresentar a discussão sobre porque que é importante aprender a programar, Resnick (2013b) apresenta uma abordagem interessante, através dos seguintes questionamentos: É importante que todas as crianças aprendam a escrever? Afinal, porque todos devem aprender a escrever se quando crescem, muito poucos serão jornalistas, novelistas ou escritores profissionais? Segundo ele, da mesma maneira que o ato de escrever envolve novas formas de pensamento, conforme as pessoas escrevem elas aprendem a organizar e refinar suas ideias, refletindo sobre elas. Portanto, existem poderosas razões para que todo mundo aprenda a escrever. Para o autor “a programação é uma extensão da escrita, saber programar permite “escrever” novos tipos de coisas – contos interativos, jogos, animações e simulações” (2013b).

Além disso, Resnick (2013b) ressalta que nesse processo não só se aprende a programar, mas programa-se para aprender. Pois, compreende-se conceitos, matemáticos e informáticos (tais como variáveis e condicionais), também se aprende estratégias de resolução de problemas, design de projetos e comunicação de ideias. Essas habilidades não são úteis apenas para os informatas, mas sim, para todos, sem importar a idade, história pessoal, interesses ou ocupação.

Para Papert (1986) a presença do computador poderia contribuir para os processos mentais não somente como um instrumento, mas de maneira conceitual, influenciando o pensamento das pessoas mesmo quando essas estiverem fisicamente distantes dele. Segundo ele sua proposta versa sobre o fim da cultura que faz com que a ciência e a tecnologia sejam hostis à vasta maioria dos seres humanos. Dessa maneira, os computadores podem ser os portadores de inúmeras ideias e de sementes de mudança cultural, como podem ajudar na formação de novas relações com o conhecimento como forma de atravessar as tradicionais barreiras que separam a ciência dos seres humanos e do conhecimento que cada indivíduo tem de si mesmo.

Assim as crianças podem desenvolver o que Papert chamou de *pensamento computacional* que, segundo Cuny, Snyder e Wing (2010) e Wing (2007), é a capacidade de promover o processo de formulação de problemas do mundo real e solucioná-los. Sousa e Lencastre (2012) afirmam que além de desenvolver seus próprios sistemas, desenvolvem:

O pensamento abstrato (utilização de diferentes níveis de abstração para perceber os problemas e, passo a passo, solucioná-los), o pensamento algorítmico (expressão de soluções em diferentes passos de forma a encontrar a forma mais eficaz e eficiente de resolver um problema), o pensamento lógico (formulação e exclusão de hipóteses) e o pensamento dimensionável (decomposição de um grande problema em pequenas partes ou composição de pequenas partes para formular uma solução mais complexa) (p. 257)

Além disso, tais pesquisadores chamam atenção que tais capacidades podem ser transpostas para qualquer área de conhecimento.

2.3 DA CULTURA INDUSTRIAL À CULTURA DIGITAL

Dessa maneira, tanto na perspectiva da interação social, quanto na do desenvolvimento de novas tecnologias, a cultura digital impele uma mudança na cultura escolar, ou seja, ela deve se transformar, abandonar o modelo da cultura industrial e adotar um paradigma mais dinâmico, interativo e adequado à cibercultura.

Para abordar tal questão recorre-se, como exemplo, ao fenômeno do Ensino à Distância. Nos últimos anos tem-se acompanhado uma ampla disseminação de cursos na modalidade EaD. Por conseguinte, avalia-se que tal fato é influência da cibercultura na educação, visto que ela possibilita novas modalidades de ensino e de aprendizagem. Segundo Nevado, Carvalho e Menezes (2005, p. 10) na EAD é fundamental que as ações de experimentar, pensar e avaliar estejam presentes em todos os momentos da aprendizagem dos estudantes, assim como a troca entre os sujeitos organizados em comunidades de aprendizagem. Porém, na avaliação deste trabalho, essa pressuposição não é exclusividade da EAD, mas da educação como um todo, pois as trocas entre os

sujeitos em comunidades de aprendizagem são fundamentais, seja na modalidade à distância ou presencial.

Sendo assim, as tecnologias digitais propõem um novo paradigma social e educacional. Para Fagundes, Sato e Maçada “estamos vivendo um processo de rápidas transformações nas formas de ser, viver, relacionar-se. (...) Torna-se quase impossível planejar e definir com antecedência o que deve ser aprendido e que competências são necessárias para habitar esse “mundo novo”.” (1999, p. 13). Indo ao encontro dessa ideia, Veen & Vrakking chamam de *Homo Zappiens* essa nova geração que nasceu e cresceu utilizando as tecnologias, portanto, o *Homo Zappiens* é a “geração da rede”. Para esses autores “aprendizagem é o processo mental pelo qual os indivíduos tentam construir o conhecimento a partir de informações, outorgando significados a elas” (2009, p. 13).

Segundo Papert o fato fundamental sobre a aprendizagem é que “qualquer coisa é simples se a pessoa consegue incorporá-la ao seu arsenal de modelos; caso contrário tudo pode ser extremamente difícil” (1986, p. 13). Portanto o que um indivíduo pode aprender e como ele aprende depende dos modelos que tem disponíveis. Para esse autor, isso impõe, recursivamente, a questão de como ele aprendeu tais modelos. Logo sua concepção de aprendizagem assume uma compreensão genética (no sentido piagetiano), ou seja, refere-se a gênese do conhecimento.

Segundo Papert (1986), o computador é o Proteu¹ das máquinas, sua essência é sua universalidade, seu poder de simulação, pois pode assumir, atrair e servir de milhares formas e finalidades diferentes. Então seus esforços sempre foram voltados para transformar o computador em instrumentos flexíveis o bastante para que muitas crianças possam criar modelos de representação para si.

Por conseguinte, a escola precisa adequar-se a tal contexto e, portanto, passar de um modelo baseado na reprodução de conhecimento (cultura industrial), para um modelo de produção de conhecimento. A ideia de uma educação cuja função seja a transmissão de informações e valores, na qual cabe ao professor a função de transmitir conhecimento deve ser ultrapassada, e de acordo com Costa, Fagundes e Nevado “é necessário, portanto uma proposta heurística e construtiva para a expansão das capacidades individuais e grupais [...]” (1998, p. 02).

De acordo com Valente (1986), as ideias de Papert colocam o aluno numa posição ativa na construção de seu próprio conhecimento: “Segundo a filosofia Logo, o aprendizado acontece através do processo *de uma criança inteligente “ensinar” o computador burro, ao invés de o*

1 É uma divindade da mitologia grega, filho de Poseidon, Proteu protegia o oceano e era procurado por humanos que queriam saber algo sobre o seu destino. Dessa maneira, Proteu assumia uma aparência monstruosa e assustadora com o objetivo de dissuadi-los na sua busca. Caso o humano fosse corajoso suficiente para aproximar-se, então a verdade lhe era revelada.

computador inteligente ensinar a criança burra”. Com esta proposta, Papert inverteu o atual quadro de uso do computador na escola. O computador deixa de ser o meio para transferir informação, e passa a ser a ferramenta com a qual a criança pode formalizar os seus conhecimentos intuitivos. Nessa concepção o programa (a sequência de ações ao computador) que a criança elabora é o espelho que reflete o seu conhecimento sobre um determinado assunto e o seu estilo de pensamento. Este programa, quando usado como objeto de reflexão, se torna uma poderosa fonte de aprendizagem.

Nesse contexto, pensa-se que a concepção educacional construtivista atende às necessidades dessa nova cultura digital. Pois nessa perspectiva o conhecimento não é algo fixo e acabado, ele surge de um contexto de trocas e de um processo de reflexões sobre aquilo que se conhece e o que se quer conhecer. Nesta altura, para a evolução dessa discussão, é preciso diferenciar os termos *inclusão digital* (da escola) e *inclusão da escola na cultura digital*: o primeiro se refere apenas ao acesso às tecnologias digitais; já o segundo, a uma mudança de paradigma educacional.

Viu-se que numa cultura digital a interação social é fundamental, para Piaget essa interação é indispensável para que a criança desenvolva uma lógica, visto que “*a criança procura evitar contradizer-se em presença de outras pessoas*” (Piaget, 1977, p. 163). Além disso, é a partir da interação social que qualquer pessoa entra em contato com diferentes pontos de vista que podem gerar o que Piaget chamou de Conflito Cognitivo. Portanto, o contexto social incentiva o sujeito a pensar sobre os outros pontos de vista em relação ao seu próprio, constituindo-se num processo de descentração que é fundamental para o desenvolvimento lógico-matemático, moral e social.

Ainda é relevante ressaltar que não é apenas a utilização dos computadores e das tecnologias digitais que garantirá uma mudança educacional voltada para o desenvolvimento da autonomia e do potencial criativo dos indivíduos. Mas seu uso deve estar relacionado a uma proposta pedagógica que leve esses aspectos em consideração. Por consequência, torna-se importante pesquisar quais são as metodologias disponíveis que atendem às necessidades dessas novas demandas. Nesse sentido, pensa-se que a metodologia de projetos de aprendizagem como proposta pedagógica possa promover a mudança para um paradigma educacional de aprendizagem, já que é um modelo mais dinâmico de construção de conhecimento.

3. PROJETOS DE APRENDIZAGEM

Um projeto de aprendizagem é uma pesquisa que será desenvolvida a partir de uma questão, problema ou algo a respeito do qual o estudante deseja saber mais. Para ajudá-lo a desenvolver tal pesquisa, o aprendiz é orientado a construir uma lista com as suas *certezas provisórias* e, para cada uma delas, suas *dúvidas temporárias*. Nesse contexto de projetos, as dúvidas são temporárias, pois a intenção é que sejam respondidas durante o desenvolvimento da própria pesquisa. Por outro lado, durante esse processo algumas das certezas que o aprendiz tinha no início tornam-se dúvidas ou o aprendiz descobre que estavam equivocadas, por isso elas são ditas provisórias.

O quadro de *certezas provisórias* e *dúvidas temporárias*, além de ser uma ferramenta que serve tanto para auxiliar o aprendiz a organizar e planejar suas futuras ações na pesquisa, é uma representação dos sistemas de significação do aprendiz. De acordo com Battro (1978), para Piaget significação é um conjunto de ações que um sujeito atribui a um determinado objeto; já um sistema de significação é a união, a intersecção ou a estrutura de esquemas de ações que são aplicadas a este objeto nesta situação. Portanto, no sistema de significação de um sujeito estão incluídas suas crenças, experiências vivenciadas e conhecimentos que podem ser científicos ou não.

Já para o professor que orientará a pesquisa desse aluno, tal quadro pode ajudá-lo a tomar consciência de tais crenças, saberes, dificuldades e hipóteses que o estudante tem a respeito desse objeto em específico. Com isso ele poderá planejar as intervenções que auxiliem o aprendiz a fazer suas próprias descobertas. No entanto Fagundes, Sato e Maçada (1999) ressaltam que é preciso respeitar e orientar a autonomia do aluno para que ele:

- Decida critérios de julgamento sobre relevância em relação a determinado contexto;
- Busque/localize/selecione/recolha informações;
- Defina/escolha/invente procedimentos para testar a relevância das informações escolhidas em relação aos problemas e às questões formuladas.
- Organize e comunique o conhecimento construído.

No entendimento desse trabalho, tal metodologia pretende manter aquilo que é mais precioso na educação (e que é cotidianamente posto de lado no atual sistema educacional): o desejo e a curiosidade de aprender cada vez mais. Nela o estudante deixa de ser “o sem luz” do iluminismo (ou a “tábula rasa” da cultura industrial) e passa a ser, desde o início, um sujeito que possui saberes anteriores e que é produtor de (seu próprio) conhecimento. Portanto é um sujeito ativo e responsável

pela sua aprendizagem.

Por conseguinte, o trabalho com projetos de aprendizagem é uma prática educativa que procura diminuir um dos principais equívocos da educação, citado por Papert (2008): a demasiada valorização do ensino e praticamente o esquecimento da aprendizagem. Diante desse fato, Papert (2008) afirma que a escola deveria desenvolver mais práticas que visem favorecer a aprendizagem de seus estudantes e, assim, procurar equilibrar a “balança” entre o ensino e a aprendizagem. Em tal afirmação, identifica-se uma convicção de Papert que vai ao encontro da ideia expressa por Freire (2005) quando diz que não há ensino sem aprendizagem e que não há aprendizagem sem ensino. Portanto, ambos evidenciam a interdependência entre os processos de ensino e aprendizagem.

Mas, infelizmente, é comum ouvir as seguintes falas dos professores “eu ensino, mas eles não aprendem”. Tal fala é significativa, pois expressa a crença behaviorista de que se o professor apresentar o conteúdo da maneira mais adequada (respeitando uma sequência pré-definida, dentro de um determinado tempo e utilizando os melhores recursos didáticos) o aluno irá aprender. Ou seja, o conhecimento se dá de “fora para dentro”, do meio para o sujeito. Dito de outra maneira: ele é colocado no aluno pelo professor, como se fossem os depósitos da educação bancária citada por Freire (2005) e que segundo este autor “deforma a criatividade”. O próprio autor pensava que se deveria substituir o ensino bancário por práticas que valorizem a criatividade do aluno.

Autores como Papert (2008), Freire (2005), Veen & Vrakking (2009) colocam o desenvolvimento da criatividade como algo imprescindível para educação. No seu livro *Brincando com Robôs*, Lopes (2010) identifica e ressalta a proposta de Piaget de “que se pense a criatividade a partir da ideia de construção e reconstrução do conhecimento, e que este é um processo que se dá a cada geração e em cada indivíduo” (2010, p. 24). Nesse sentido a criança cria o mundo à medida que constrói (e reconstrói) o conhecimento à medida que interage com aquilo que quer conhecer.

Pretende-se ainda chamar a atenção para o significado de aprendizagem para Piaget e Gréco (1974). No sentido estrito (*stricto sensu*), a aprendizagem para Piaget e Gréco (1974) é o resultado do processo de construção do conhecimento, ou seja, é adquirido em função da experiência – seja física, lógico-matemática ou de ambos – e que se desenvolve no tempo, portanto é mediata e não imediata. Já no sentido amplo (*lato sensu*) são aquelas que não se devem à experiência, mas são aquisições devidas ao desenvolvimento de um processo dedutivo de nível operatório e independentes da experiência e por processos de equilíbrio.

Sendo assim, fica claro que em uma perspectiva piagetiana as concepções de criatividade, aprendizagem e conhecimento estão inter-relacionadas ao desenvolvimento da inteligência. Bem

como para a importância do desenvolvimento do raciocínio lógico-matemático na aprendizagem e na criatividade.

Voltando aos projetos de aprendizagem, de acordo com tal metodologia, numa turma convencional de trinta estudantes, poderão existir trinta pesquisas diferentes. Logo em seguida, tais estudantes serão agrupados por assuntos afins. Segundo Fagundes et al (2006) o trabalho com projetos para aprender pode favorecer a aprendizagem da cooperação, as trocas recíprocas e respeito mútuo entre os estudantes. Para esses autores o foco da aprendizagem não é o conteúdo formal descontextualizado, mas “a proposta é aprender conteúdos, por meio de procedimentos que desenvolvam a própria capacidade de continuar aprendendo” (2006, p.30). Sendo assim, essa situação propicia o desenvolvimento de novas competências tanto do quadro conceitual do sujeito, de seus sistemas de valores e de suas condições de tomada de consciência.

Além disso, tal modelo contempla uma organização escolar interdisciplinar em função da curiosidade e do interesse do estudante, portanto assume oposição entre a disciplinarização do conhecimento criticada por Morin (2005) – em que os saberes são separados e compartimentalizados – e procura realidades ou problemas cada vez mais polidisciplinares, transversais, etc.

Dessa forma, o papel do professor e do estudante também deve se transformar, segundo Fagundes, Sato e Maçada:

Ao professor cabe à função de promover a aprendizagem, estimular o diálogo, provocar a emergência de situações de dúvidas (desequilíbrios) e apoiar as reconstruções (novos conhecimentos). Ao aluno cabe uma postura ativa. A ele cabe experimentar, compartilhar, criar, interagir para compreender (FAGUNDES, SATO e MAÇADA, 1999).

Como a pedagogia de projetos foi desenvolvida por John Dewey no início do século XX, é comum que haja confusões e indiferenciações entre diferentes propostas existentes. Neste trabalho serão apresentados as diferenças entre projetos de ensino, de trabalho e de aprendizagem que estão presentes nos estudos de Mattos (2010) e Fagundes, Sato e Maçada (1999). Pensa-se que a primeira diferenciação importante nessas propostas diz respeito à pessoa que propõe a pesquisa. Como foi descrito anteriormente, nos projetos de ensino a pesquisa é desenvolvida a partir de uma questão ou conteúdo determinado arbitrariamente pelo professor e, geralmente, faz parte do currículo formal da escola. Em contraponto a tal proposta, viu-se que nos projetos de aprendizagem a pesquisa é desenvolvida a partir da curiosidade e do desejo de cada estudante. Já nos projetos de trabalho, defendidos por Hernández e Ventura (1998), um grupo de alunos e professores determinam uma

série de assuntos a serem pesquisados; logo em seguida há um momento para argumentação, seleção, exclusão e, por fim, uma votação para eleger um tema que será desenvolvido por todo o grupo.

Como foi visto, a autoria do projeto é que define as diferenças entre as três propostas; podemos ver a sistematização disso na tabela 01, construída por Mattos (2010).

Tabela 1: Diferenças entre projetos de ensino, trabalho e aprendizagem.

	Projetos de Ensino	Projetos de Trabalho	Projetos de Aprendizagem
Autoria Quem escolhe o tema?	Professor, coordenação pedagógica	Turma de alunos e professores por argumentação, seleção, exclusão e votação	Alunos e professores, individualmente, e em cooperação.
Contextos	Arbitrado por critérios externos e formais.	Trabalhos e temas precedentes e propostas de professores.	Realidade de vida dos alunos.
A quem satisfaz?	Arbitrio da sequência de conteúdos do currículo.	Interesse do professor e da maioria dos estudantes.	Curiosidade, desejo e vontade do aprendiz.
Decisões	Hierárquicas.	Oscilam entre hierárquicas e heterárquicas.	Heterárquicas
Definições de regras, direções e atividades.	Impostas pelo sistema, cumpre determinações sem optar.	Elaboradas pelo grupo, consenso de alunos e professores.	Elaboradas pelo grupo, consenso de alunos e professores.
Paradigmas	Transmissão de conhecimento.	Construção de conhecimentos.	Construção de conhecimentos.
Papel do professor	Agente.	Oscila entre agente e orientador.	Estimulador/ orientador.
Papel do aluno	Receptivo.	Oscila entre receptivo e agente.	Agente.

Vale ressaltar que a proposta de projetos de trabalho representa um avanço em relação aos projetos de ensino, visto que seu paradigma educacional é a construção de conhecimento, portanto o conteúdo curricular é construído a partir dos interesses dos estudantes e, conseqüentemente, visa a desfragmentação e hierarquização dos conteúdos. Além disso, segundo Hernández e Ventura (1998), tal metodologia promove no estudante a consciência da responsabilidade sobre a sua própria aprendizagem.

Apesar de tais semelhanças com os projetos de aprendizagem, Mattos cita algumas diferenças significativas entre as duas propostas:

A escolha desse tema e a formulação das perguntas, no entanto, não se dá individualmente pelos alunos, mas pela turma inteira pelo professor, o que significa, muitas vezes, o privilégio dos interesses de alguns em detrimento de outros. [...] Os Projetos de Aprendizagem, portanto, ao considerar e aceitar as propostas de todos os alunos – sem seleção ou exclusão – configuram-se em um avanço em relação aos Projetos de Trabalho e trazem efetivamente o aluno – seus interesses, anseios, curiosidade e desejos – para a escola. (2010, p. 52)

Outra diferença significativa apontada por Mattos é que é possível trabalhar com matemática

(de forma contextualizada) em qualquer pesquisa desenvolvida no âmbito dos projetos de aprendizagem. Porém o mesmo não ocorre nos projetos de trabalho, que segundo Hernández e Ventura (1998), “os temas dos projetos de trabalho costumam corresponder, como já disse, às Ciências Sociais e Naturais” (1998, p.181).

3.1 PROBLEMATIZANDO OS PAs

A partir da experiência com projetos de aprendizagem em diferentes locais – formação de professores no ECSIC, no projeto PIPA, nas oficinas da fundação Pensamento Digital e em escolas de educação básica – o autor deste trabalho observou que alguns projetos de aprendizagem desenvolvidos restringe-se à construção de páginas (html, wikis, etc). Tal restrição promove uma pesquisa estritamente bibliográfica (em livros e sites), deixando de explorar diversos recursos interessantes que enriqueceriam e qualificariam o projeto e a aprendizagem do estudante.

Tem-se como hipótese que isso é consequência de uma visão estática da proposta de projetos de aprendizagem por parte daqueles que a aplicam e uma alternativa para superá-la seria incorporar novas formas de projetos. Por exemplo, a criação de um grafite, de um filme, uma animação, um jogo eletrônico, uma peça de teatro, a própria robótica, etc. podem ser um projeto de aprendizagem. Sabe-se que muitas dessas atividades já são desenvolvidas nas salas de aula de muitas escolas, no entanto é comum que a forma como são propostas aos estudantes siga os princípios dos projetos de ensino, em que o professor determina o tema ou os estudantes executam uma série de ações pré-determinadas e organizadas pelo professor ou, então, são realizadas de forma exploratória e intuitiva pelos estudantes.

Para que essas atividades escolares possam ser consideradas um projeto para aprender, primeiramente é necessário que os estudantes tenham o direito de decidir o tema de sua produção. Logo em seguida é preciso adicionar a ela a ação de *projetar*, isso para que o estudante reflita sobre o que ele deseja realizar, o que já sabe e o que precisa aprender. Podemos utilizar diversos instrumentos que auxiliem aos estudantes a organizar e planejar suas ações futuras. Nesse sentido, a construção do quadro de *certezas provisórias* e *dúvidas temporárias* pode ser utilizado, assim como Dutra (2006) utiliza mapas conceituais, ou Lopes (2010) utiliza as atividades de *design* desenvolvidas por Resnick (1991) e (2006) em projetos de robótica. O fundamental é que esses instrumentos sirvam como uma representação dos sistemas de significação dos estudantes e também auxiliem o professor a tomar consciência de crenças, saberes, dificuldades e hipóteses que o estudante tem a respeito desse objeto em específico. Dessa maneira, ele poderá planejar as intervenções que auxiliem o aprendiz a fazer suas próprias descobertas.

Nessa perspectiva, esta pesquisa assume como hipótese que se pode ampliar a ocorrência de projetos de aprendizagem de matemática e ciências, quando estes estão ligados à criação de jogos eletrônicos, animações, simulações, etc. através de *softwares* que utilizam linguagens de programação visual voltadas para as crianças. Dessa forma, investe-se numa proposta de uso do *software Squeak Etoys* com a intenção de promover a aprendizagem de matemática. Vale ressaltar que a escolha desse *software* se deu em função de ser um *software* livre e de estar presente nos *laptops* que foram distribuídos a escolas da rede pública no projeto UCA e no *software* do sistema XO no projeto OLPC de Nicolas Negroponte, o que contribuiu para a formação de diversos grupos de investigação no mundo.

4. O *Squeak Etoys*: O DIGITAL NA ESCOLA

Nesta secção o *software Squeak Etoys* será apresentado, pois a história de sua concepção reflete aspectos fundamentais da cultura digital e da importância do digital na escola, ou seja, da compreensão de como as crianças podem utilizar o computador como um objeto de aprender com. É pertinente ressaltar que, embora o foco deste trabalho esteja no *Squeak Etoys* e em algumas ideias de Alan Kay, a discussão proposta aqui também está presente nas pesquisas de Resnick com o *Scratch*, cuja inspiração foi o próprio *Squeak*. Além disso, é pertinente destacar que tais linguagens são importantes e relevantes, pois ambas estão fundamentadas nas ideias concebidas por Papert. Avalia-se que a filosofia LOGO, apresentada por Papert, a mais de 40 anos, foi revolucionária e mantém-se atual e necessária para uma modificação de paradigma escolar. Nesse sentido, o autor desta pesquisa, divide o mesmo sentimento com Resnick quando afirma “*Eu serei feliz e orgulhoso se passar o resto da minha vida tentando tornar o sonho de Papert em realidade*” (2012, pg. 46).

O *software Squeak* foi concebido na Apple em 1995 quando Alan Kay, Dan Ingalls e Ted Kaehler se reencontraram e ressuscitaram as ideias do *Dynabook* (Kay, 1972). Segundo Dan Ingalls et al. (1997) o “*Squeak é uma implementação aberta e altamente portátil do Smalltalk*”. O Smalltalk foi a primeira linguagem de programação orientada a objetos inventada por Alan Kay (1993). Dessa maneira, antes de descrever o *Squeak Etoys* é fundamental compreender o processo de sua criação, os desafios, as ideias e hipóteses que afligiam o seu criador quando este idealizou o *Dynabook*.

Alan Kay, quando era pesquisador do Xerox Palo Alto Research Center (PARC), idealizou um computador portátil voltado para crianças de todas as idades, chamado de *Dynabook*. Embora sua idealização tenha sido em 1968, Kay só o descreveu em 1972 (ver as figuras 1 e 2). É interessante observar que sua gênese antecede a criação dos próprios computadores pessoais. Em função desse fato, hoje, o *Dynabook* é tido como o pioneiro das tecnologias móveis tais como os notebooks, tablets, etc.

Figura 1: Projeto do Dynabook de Alan Kay

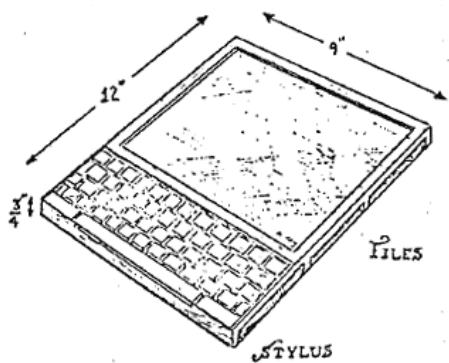
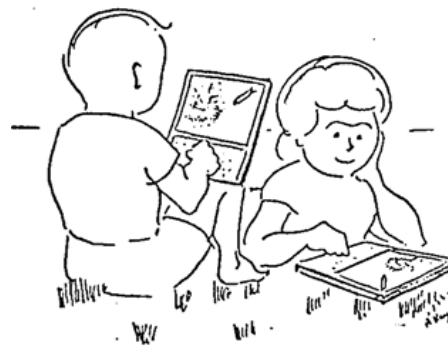


Figura 2: Crianças com o Dynabook

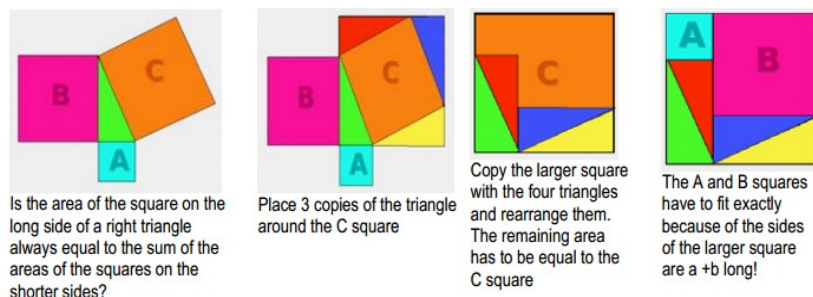


Porém, a proposta por trás do *Dynabook* ia além dos aspectos portáteis do computador. Ao conhecer o trabalho de Papert com o LOGO, Allan Kay (2002) ficou impressionado ao constatar que as crianças começavam a entender algumas ideias poderosas da matemática e, assim, aprendiam o seu significado. Na sua avaliação, tal aprendizagem foi possibilitada quando elas começaram a realizar algumas programações simples no LOGO. Desde então, ele buscou desenvolver um computador voltado para crianças, no qual seu grande potencial seria auxiliar na aprendizagem de seus usuários, em especial das crianças.

Ao assumir tal desafio, Kay (2002) procurou identificar qual seria o novo tipo de argumento que seríamos capazes de fazer com o computador. Afinal, como o uso dos computadores poderia mudar a maneira que as ideias são apresentadas e testadas? Essa era a questão em que se concentrava. Para ele, a simulação seria esse novo tipo de argumento, pois “se você quer falar sobre algo realmente complexo, a simulação é a maneira mais eficiente de argumentação, do que uma equação matemática” (2002, p.2).

Como exemplo dessa ideia, anos depois, Kay (2003) explora o teorema de Pitágoras. Ele argumenta que a maioria das crianças não entende o significado do teorema de Pitágoras quando ele é ensinado a partir da aplicação da sua equação ($a^2=b^2+c^2$). No entanto, a ideia por trás de tal teorema é facilmente compreendida quando a criança constrói uma simulação/animação da demonstração geométrica do teorema de Pitágoras, figura 3, abaixo.

Figura 3: Demonstração do teorema de Pitágoras



Isso indica que, ao propor o *Dynabook* o autor já pensava no computador como um meio que poderia comunicar coisas importantes, ideias poderosas e não só como hardware e *software*. Para ele, “o computador é um instrumento cuja música são as ideias” (2002, p.3). Além disso, para apresentar a proposta do *Dynabook*, Kay (2002) faz uma analogia com o desenvolvimento tecnológico que a Impressão de Gutemberg proporcionou em relação à cultura precedente. Na sua concepção, assim como o livro era uma extensão do meio oral, o computador é uma extensão do meio de impressão. Portanto, via que há muitas coisas que os livros podem fazer, mas os computadores têm uma dimensão extra que lhe parecia extremamente importante e isso foi a chave para a ideia do *Dynabook*.

Aprofundando a analogia entre *livros x computadores*, para o autor a literatura é uma conversa por escrito sobre ideias importantes, portanto as obras *Elementos* de Euclides e o *Principia Mathematica* de Newton pertencem à cultura ocidental de grandes livros, tais como o *Dialogues de Platão*. No entanto, segundo ele, comete-se o equívoco de pensar que as obras de matemática e de física não fazem parte da literatura. Segundo Kay (2002), tal fato indica uma contradição, já que a sociedade está amplamente construída por tecnologias que vêm da compreensão de matemática e ciências.

Para Kay, a alfabetização não é só sobre a capacidade para ler sinais de trânsito ou receitas médicas. Além disso, significa ser capaz de abordar o mundo das ideias. Numa sociedade democrática precisa-se de pessoas que dialoguem com importantes ideias do passado e do presente, ou seja, elas precisam ser capaz de ler, escrever e falar sobre. Portanto, para propor o *Dynabook*, o autor refletiu sobre o ponto de vista de uma nova literatura que deveria incluir coisas novas à literatura antiga. Segundo Kay “nem toda ideia pode ser expressa através da linguagem falada. Eu acho que foi Stravinsky que disse: *falar sobre música é como dançar sobre arquitetura*, Música já é um meio de expressão em si mesmo”. (2002, p. 5). Segundo o autor o que é mais especial na música, nas artes visuais e cinéticas é justamente aquilo que não se pode pôr em palavras. Para Kay todas essas formas de expressão são parte da nossa paisagem de ideias, e por isso, nesse sentido,

também são literatura.

Sendo assim, de acordo com o autor o computador pode ser utilizado como um meio de expressão. Pois num livro pode-se imprimir uma partitura, mas num computador, além de imprimi-la, é possível mover as coisas, experimentar com conjuntos inteiros de linguagens musicais e tocá-la. Dessa forma, Alan Kay (2002) diz que utilizou o termo "Metamedia" para descrever o computador, pois ele é um detentor de todos os meios que uma pessoa pode utilizar para pensar sobre as coisas que já existem ou não. "O computador permite capturar ideias importantes, qualquer que seja a forma de sua expressão, e transmiti-los de uma maneira que ajudará outras pessoas a entendê-los, e (talvez) até mesmo expandi-los." (2002, p.6). Outra analogia utilizada pelo autor: para realmente usar um computador é preciso ser autor e também leitor. Ou, em termos de música, um computador é algo através do qual você pode compor e tocar. Assim, Kay (2002) conclui que a relação que um usuário tem com o computador é diferente do seu relacionamento com um livro. Ou, pelo menos, tem essa possibilidade.

Desde que o Allan Kay propôs o *Dynabook*, muitas tecnologias que tem o propósito de facilitar as interações dinâmicas entre o usuário e o computador foram desenvolvidas, principalmente no que diz respeito em relação ao hardware. No entanto, para o autor, a ideia mais significativa do *Dynabook* – que está nas construções que as pessoas podem (ou só podem) fazer com o uso do computador – ainda não foram concretizadas, entendidas ou foram distorcidas.

"Obviamente, um monte de alta tecnologia foi desenvolvida desde que comecei a falar sobre o *Dynabook*. Mas o que é realmente mais significativo sobre a ideia do *Dynabook* está nas construções que as pessoas, especialmente as crianças, podem fazer com ele, e que eles não podem fazer de outra maneira. Por outras palavras, trata-se da relação entre o computador e o usuário". (2002, p. 3)

Centrando seus esforços nessas ideias centrais do *Dynabook*, com o passar dos anos Kay e seus colegas de pesquisa do View Point Research Institute, desenvolveram um ambiente de autoria chamado *Squeak*, cuja intenção era permitir uma maior variedade de interações dinâmicas. O *Squeak* foi desenvolvido numa linguagem de programação chamada de Smaltalk.

Nestes últimos anos, o autor está procurando entender o que as crianças fazem com o computador e o que se pode fazer para ajudá-las. Nesse sentido, tal pensador, diz que não é sempre claro entender as coisas que as crianças precisam em determinada idade. Segundo ele, quando as coisas estão dando errado, fica difícil de avaliar se é um problema da interface do *software* que não está boa ou se é uma determinada versão do sistema de computador é lenta para manter o interesse das crianças ou, ainda, se tal dificuldade está relacionada à idade da criança. Já quando as coisas

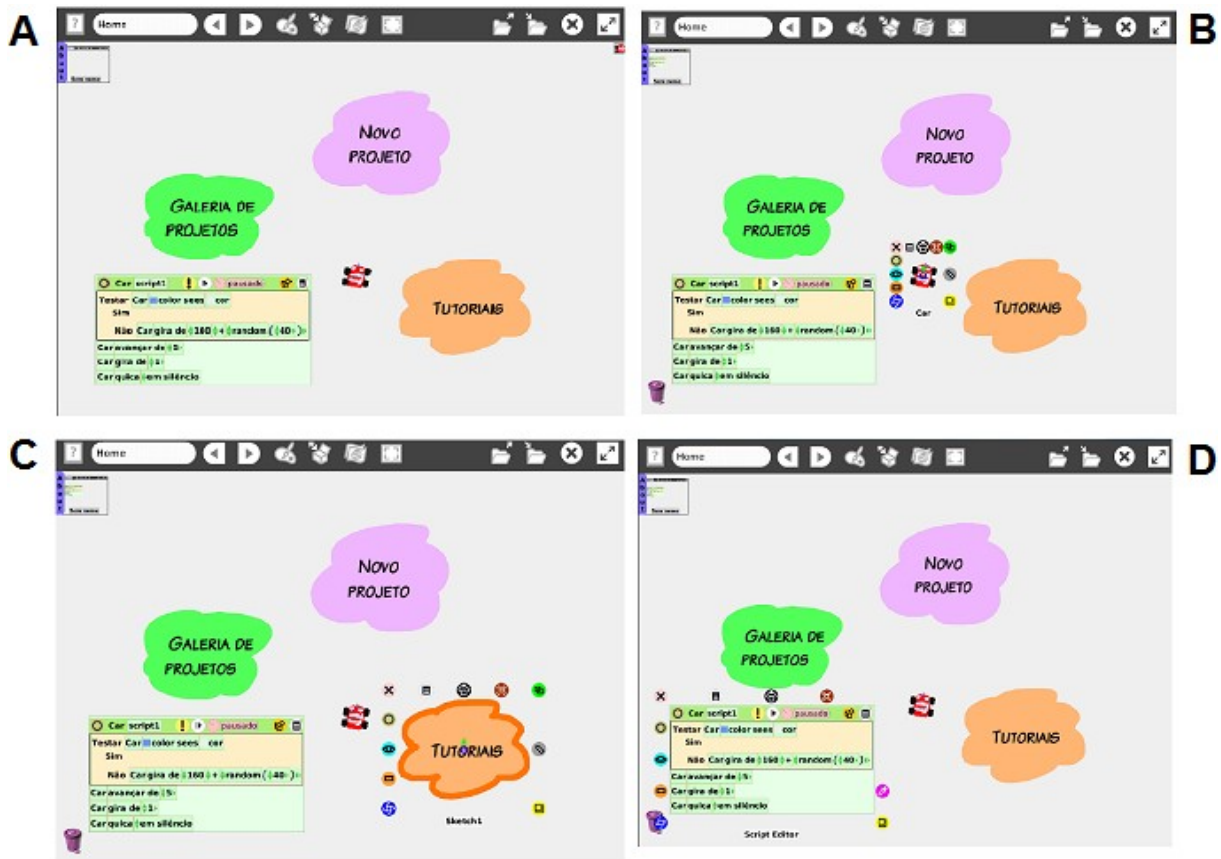
estão dando certo, também não ficam claros os motivos que levaram a tal êxito ou algo que a criança tenha feito.

Procurando minimizar cada vez mais os problemas de interface do *software* para as crianças, foi desenvolvido o componente *Etoys*, que é introdutório ao *Squeak*. À medida que o aluno vai evoluindo, passa a utilizar interfaces do *Squeak* mais avançadas e adequadas ao seu nível. Allen-Conn e Kim Rose, o definem da seguinte maneira: “Os *Etoys* são modelos, simulações e jogos construídos pela montagem de mosaicos em *scripts*, que enviam comandos para os objetos desenhados, para que o aluno obtenha uma melhor percepção de uma área de investigação”. (2003, p.10). Já para Teixeira, o *Squeak Etoys* “funciona como um simulador de mundos virtuais, onde é possível experimentar, reproduzindo fenômenos e processos reais ou inventados”. (2011, p.86).

Sendo assim, o *Squeak Etoys* é um ambiente de autoria multimídia e um sistema de programação visual que foi inspirado na filosofia Logo de Seymour Papert. Uma boa maneira de apresentá-lo é explorando sua página inicial.

Na figura 4A ao abaixo pode-se observar a tela inicial do *Etoys*, ao abri-la vemos o carro se locomover sem parar e mudando de direção quando esbarra nas três nuvens. As três nuvens são botões que levam respectivamente à duas galerias, uma de tutoriais outra de projetos desenvolvidos, e a um novo projeto.

Figura 4: Tela inicial do Etoys e seus objetos programáveis.



A área cinza clara da tela (local onde o carro se desloca) é chamada de mundo, que é o palco onde os objetos realizarão suas simulações, animações etc. Já na parte superior há um menu preto fixo com uma série de funções (abrir, salvar, fechar etc.) e outras ferramentas como editor de imagens. Na imagem acima, no mundo, ainda há uma janela verde chamada *script*, que é o local onde se faz um roteiro de comandos que definem a ação do objeto, ou seja, em que se constrói o algoritmo de programação. Neste caso, o *script* faz o carro se movimentar somente na parte cinza.

Mas como se programar um objeto? Basta clicar com o botão direito do *mouse* em qualquer objeto que está no mundo. Feito isso, o programa abrirá um conjunto de ícones em volta do objeto, que forma um menu que contém as ferramentas necessárias para programá-lo, duplicá-lo, apagá-lo, editar sua aparência etc. Na figura 4B pode-se observar tal menu em volta do carro.

Contudo, uma das propriedades mais interessantes do *Squeak Etoys* é a possibilidade de programar qualquer objeto. Como mostra a figura 4C, com um simples clique pode-se editar cada uma das nuvens (que nesse projeto são botões). Através do mesmo menu, se têm acesso fácil ao *script* que contém a programação por trás deste botão e o usuário pode modificá-lo quando quiser.

Isso significa dizer que tudo que aparece no mundo é programável, tudo mesmo: um texto,

uma equação, uma linha, um retângulo são objetos ou até mesmo uma tela de *script*. Como exemplo, na figura 4D pode-se ver que o menu está em volta da tela de *script* do carro. É interessante observar a reação das crianças quando elas se dão conta dessa possibilidade. Geralmente isso ocorre sem querer e espontaneamente, com a intenção de movimentar o carro, elas programam sem perceber a tela do *script* ao invés do *script* do carro; ao executá-lo, ficam surpresas ao ver a tela do *script* girando ou “andando” pelo mundo.

Nessa ligeira descrição da tela inicial do *Squeak Etoys* ficam claras algumas de suas características mais importantes. O fato de tudo ser um objeto programável indica o seu paradigma de programação: que é a programação voltada a objetos. Portanto, ele explora a ideia de que um programa de computador simula um mundo real, no qual “vivem” múltiplos objetos que se comunicam entre si. Segundo Baranauskas “esse conceito é baseado na ideia de que no mundo real frequentemente usamos objetos sem precisarmos conhecer como eles realmente funcionam” (1993, p.2). Tal paradigma surgiu paralelamente à linguagem de programação Smaltalk que foi criada por Alan Kay em 1972.

Pode-se classificar o *Etoys* como um micromundo, segundo Niquini (1997) um micromundo apresenta um conjunto de elementos primitivos que permite a produção de diferentes efeitos a partir de sua combinação. Menezes (2003) explora a analogia de Rieber (1997) o micromundo é como um “conjunto composto de um balde de areia e pás usado numa praia, pois a criança não precisa ser treinada para brincar com tais elementos”. Dessa forma Menezes (2003) afirma que “O aprendiz pode até explorar ideias e conceitos mais complexos deste domínio, mas sempre de forma crescente a partir de primitivas”. Dessa maneira, todos os autores afirmam que os *softwares* que assumem o paradigma de micromundo – assim como Allan Kay em relação ao *Etoys* – favorecem a aprendizagem de representações mais simples de uma realidade. Teixeira comenta que com o *Etoys* “pode construir-se um Mundo destinado a experimentar, analisar, refletir e tirar conclusões e pode gerar-se informação e testar a sua apropriação por outros” (2011, p. 86).

Além disso, o autor desta pesquisa entende que o uso do *Etoys* pode contribuir para que os estudantes entendam melhor a cultura digital a qual pertence. Pois, ao construir jogos, animações e simulações os estudantes tornam-se produtores de tecnologia ao invés de apenas consumi-la. Nesse sentido, Kay (2002) quando compara o que as novas gerações têm feito em relação às gerações anteriores, pensa que sua geração fazia coisas mais interessantes, já que hoje poucas pessoas sabem como um computador funciona. Ainda nesse sentido, ele critica a ideia de que as pessoas se alfabetizam em informática quando começam a realizar atividades triviais com o computador.

Resnick também apresenta a mesma inquietação, segundo ele, a maioria dos chamados nativos digitais, utilizam os recursos computacionais apenas para navegar na internet, entrar em chats, rodar aplicativos e jogar games. “É como se eles só conseguissem ler, mas não escrever” (2012), ou seja, nossas crianças não são capazes de desenvolver, criar e se expressar através das novas tecnologias. Isso indica que ainda hoje, apesar da grande disseminação e acesso aos computadores, que as crianças não se tornaram verdadeiramente fluentes em novas tecnologias, não concretizando, assim, um dos principais aspectos das ideias de Papert.

4.1 ETOYS E MATEMÁTICA

Alan Kay ressalta o fato de não se precisar de tecnologia para se aprender ciências e matemática. Mas, assim como na música, ela é um instrumento que amplifica as ideias dos estudantes. Porém, Kay (2002) afirma que uma das ações de um professor é criar as melhores condições para cada criança aprender. Uma vez que se tem isso, então o computador pode ajudar imensamente. Além disso, os princípios pedagógicos que fundamentam o *Etoys* confluem com os Parâmetros Curriculares Nacionais (1997), que indicam como objetivos gerais do Ensino Fundamental que os alunos sejam capazes de: a) Utilizar as diferentes linguagens (verbal, matemática, gráfica, plástica e corporal) como meio para produzir, expressar e comunicar ideias, interpretar e usufruir das produções culturais, em contextos públicos e privados, atendendo a diferentes intenções e situações de comunicação. b) Saber utilizar diferentes fontes de informação e recursos tecnológicos para adquirir e construir conhecimentos; c) Questionar a realidade formulando-se problemas e tratando de resolvê-los, utilizando para isso o pensamento lógico, a criatividade, a intuição, a capacidade de análise crítica, selecionando procedimentos e verificando sua adequação.

Portanto, após este momento emergiu a necessidade de se refletir sobre quais tipos de aprendizagem de matemática são possibilitadas pelo uso do *Etoys*. Dessa maneira, foi realizada uma pesquisa em artigos, livros, dissertações e teses (seja na forma impressa ou digital) de atividades e projetos de matemática no *Squeak Etoys*. Nela, se identificou as seguintes possibilidades: a) *Matemática da Tartaruga* – como o *Etoys* é um *software* que se inspira na filosofia Logo, ele mantém todas as possibilidades de desenvolver trabalhos bem ao estilo da tartaruga, como vemos no trabalho de Newell (1988). Dessa maneira, pode-se dizer que o *Etoys* herda todas as atividades de geometria que são próprias do Logo; b) *Introdução à Álgebra* – É possível realizar uma

introdução à álgebra através da construção de polígonos regulares com algoritmos que envolvem o uso de variáveis e o uso do sistema de coordenadas cartesianas; c) *Triângulos e Trigonometria* – A partir do trabalho realizado com polígonos é comum explorar o desenho de triângulos não equiláteros. Esse tipo de desenho se torna mais complexo que os polígonos regulares, visto que é preciso o conhecimento de trigonometria. Sem ele, os estudantes realizam tal tarefa num exaustivo processo de tentativa e erro; d) *Fractais e Algoritmos Recursivos* – Num nível avançado, é possível utilizar algoritmos recursivos no *Etoys* e, assim, abordar a construção de fractais, como no projeto de Pitt (S/D) no; e) *Matemática dos Movimentos* – Uma das primeiras atividades desenvolvidas no *Etoys*, apresentadas por Allen-Conn e Rose (2003), é a construção de um carro que se movimenta de forma autônoma. Tal ação é possível através do botão relógio que fica executando qualquer *script* em loop. Só a existência desse botão agrega novos significados e possibilidades para as atividades ao estilo LOGO. Conceitos como velocidade e aceleração possibilitam a compreensão de incrementos de um escalar (positivo, negativo ou nulo) a uma variável. f) *Geometria dinâmica* – No *Etoys* é possível fazer construções com régua e compasso virtual, pois nele está incorporado o *software* de geometria dinâmica Dr. Geo II; g) *Lógica Proposicional* – a aprendizagem da lógica é um tipo de matemática que é inerente ao ato de programar.

Essas foram as atividades encontradas que possibilitam a aprendizagem de matemática no *Etoys*, pensa-se que tais atividades contemplam a aprendizagem dos principais conceitos-chave da matemática do ensino básico. Ao refletir sobre elas, identificam-se duas maneiras de promover a aprendizagem da matemática pelos estudantes através do uso do *Etoys*. A primeira maneira seria a partir de atividades que exploram diretamente objetos matemáticos, cujo tema é a própria matemática. Temos como exemplos atividades tais como: desenhar triângulos, polígonos, mosaicos, criar figuras simétricas, medianas no Dr. Geo.

Já a segunda maneira, seria explorar a matemática que se aprende ao se programar no *Etoys*, no qual o objeto criado pelo estudante não seja necessariamente matemático. Sendo assim, a aprendizagem de matemática se dá como um meio para se aprender. Já que o objetivo do estudante não seria aprender matemática, mas sim produzir algo no *Etoys* e “de brinde” aprende-se outras coisas tais como matemática, física, biologia e etc. Os projetos desenvolvidos e apresentados por Allen-Conn e Rose (2003) são exemplos deste tipo de aprendizagem de matemática.

No contexto deste estudo, utilizaremos *Matemática como Objeto de estudo* para designar a primeira e *Matemática como um meio para aprender*, a segunda. Caso o leitor, professor de matemática (ou não), se interessar por tais atividades e propostas de trabalho apresentadas nesta seção, no **Anexo A** desta pesquisa elas são descritas com mais detalhes e exemplos, incluindo suas

referências.

Para o autor desta pesquisa, o estudo da aprendizagem da matemática como meio para se aprender se configura como um desafio, uma vez que gera uma série de dúvidas e incertezas, por exemplo: a) Será que é possível aprender matemática em qualquer produção no *Squeak Etoys*? Num primeiro momento, supõe-se que não, e nesse caso, torna-se interessante investigar (b) quais seriam as produções que promovam tal aprendizagem? Ainda há aquelas dúvidas relacionadas às questões práticas da atividade profissional de professor: (c) Será que tais aprendizagens contemplarão os conteúdos curriculares? (d) Será que elas promoverão a aprendizagem de conteúdos que estão além daqueles previstos no currículo?

4. 2 EXPERIÊNCIA-PILOTO

Numa tentativa de delinear novas hipóteses e encaminhamentos para as questões apresentadas na seção anterior, no início de segundo semestre de 2012 teve-se a oportunidade de realizar uma experiência de utilização do *Squeak Etoys* numa turma do nono ano do ensino fundamental do Colégio de Aplicação da Universidade Federal do Rio Grande do Sul (CAP/UFRGS). O professor Eduardo B. V. de Mattos, regente da turma, aceitou desenvolver tal trabalho ao tomar conhecimento da intenção desta pesquisa, que seria promover a aprendizagem de matemática através da utilização do *Squeak Etoys* numa perspectiva de projetos de aprendizagem, visto que ele próprio já havia desenvolvido uma pesquisa de dissertação intitulada “Construção de conceitos de matemática via projetos de aprendizagem” (2010).

Dessa maneira foram realizados encontros semanais com duração de duas horas-aula e no horário da disciplina de matemática. A equipe dessa experiência era formada por dois professores de matemática, e pelo psicólogo e pesquisador do LEC/UFRGS Bruno F. Sperb, que estava desenvolvendo uma pesquisa sobre a *Construção de jogos digitais em Etoys na modalidade 1:1*.

Desde o início, a intenção foi realizar uma investigação bem aberta e que ajudasse a equipe a refletir sobre suas próprias pesquisas. A ideia era criar para os estudantes um espaço de criação livre em que os pesquisadores pudessem observar, formular e testar suas hipóteses. Particularmente para esta tese, a intenção deste piloto era promover os dois tipos de aprendizagem de matemática (como objeto de estudo e como meio para se aprender), sendo a primeira realizada durante os encontros de programação e a segunda, em produções dos alunos voltadas para seus trabalhos desenvolvidos no Projeto PIXEL. O Pixel é um projeto curricular das séries finais do Ensino Fundamental do

CAP/UFRGS. Segundo o site do Colégio de Aplicação, com o Pixel¹ “pretende-se colocar em prática atividades multidisciplinares que permitam ir além do conhecimento específico de cada disciplina, não explorando apenas os pontos comuns entre elas, mas possibilitando experiências reais de interação com o mundo” (PIXEL, 2011). Tal projeto procura realizar uma iniciação científica dos estudantes e suas atividades são inspiradas nos projetos de aprendizagem desenvolvidos no Projeto Amora.

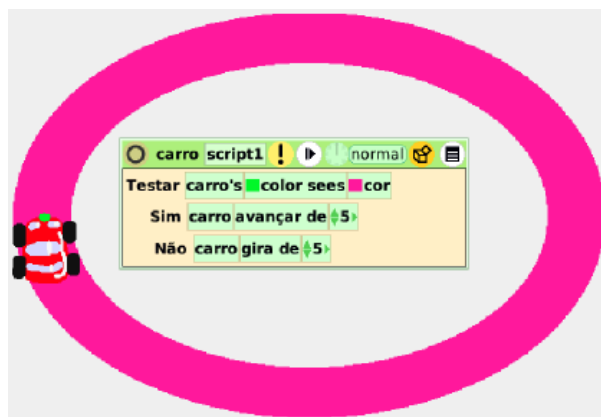
Realmente tal experiência foi significativa e intrigante, pois promoveu uma série de questionamentos e reflexões que, certamente, redirecionaram o rumo desta pesquisa. Nesse sentido, reforça-se a convicção do autor deste trabalho de que uma pesquisa voltada para a educação precisa ser oriunda da prática de sala de aula. Da mesma maneira, quando Piaget foi até as crianças para entender o desenvolvimento cognitivo, professores e pesquisadores precisam ir até elas para interagir e ouvi-las.

Assim como, na seção anterior, a experiência-piloto está detalhadamente descrita no **Anexo B** desta pesquisa para aqueles leitores que se interessarem em aprofundar a leitura sobre essas questões. Porém é importante apresentar os questionamentos que surgiram na experiência-piloto e que ajudaram a definir o problema deste estudo. Abaixo apresenta-se o resultado de algumas delas.

No *Squeak Etoys*, as crianças exploram a lógica, de uma maneira mais explícita, através do uso do comando *Teste*. Tal recurso já pode ser utilizado num dos primeiros projetos desenvolvidos com as crianças, que é a criação de “carros inteligentes” que dirigem sozinhos sobre uma pista e que foram desenvolvidos por Allen-Conn e Rose (2003). Para isso é necessário criar um algoritmo que permita reconhecer quando esse objeto está (ou não) sobre a pista. Uma das formas de conseguir tal efeito é criar um sensor virtual (ponto verde na frente do carro), que, através do comando *Teste*, aplica uma sentença condicional cada vez que o *script* é executado, como mostra a figura 5.

1 Site do projeto PIXEL: <http://www.ufrgs.br/projetopixel>

Figura 5: Script para o carro reconhecer a pista rosa



Pode-se interpretar tal *script* da seguinte forma: na linha “Testar” insere-se a condição a ser testada: “o verde vê a cor rosa”. Se tal condição é verdadeira, ele executará o comando que está na linha “Sim”: carro avançar de 5”. Caso contrário, se for falsa, executa-se o comando da linha “Não” do *Teste*: “girar de 5”. Quando tal *script* foi apresentado, parecia que os estudantes o compreendiam, no entanto quando eles começavam a utilizá-lo em diferentes situações, foi observado que a lógica não era de fácil compreensão. Segundo Moraes, Fagundes e Basso (2013b e 2013c) na aprendizagem da lógica observou-se que as crianças apresentavam:

a) Indiferenciação em relação à ordem dos comandos dentro do *script*; b) Indiferenciação dos elementos do comando *Teste*: qual condição que estava sendo testada, qual ação seria realizada caso a condição fosse verdadeira (linha sim) ou falsa (linha não). c) Dificuldade de utilizá-lo quando envolvia as seguintes operações lógicas: disjunção, conjunção e negação. d) Compreensão de *scripts* distintos, porém análogos.

Ao relembrar a pergunta que norteia esta pesquisa – como pode-se aprender matemática a partir dos projetos desenvolvidos no *Squeak Etoys*? – Sendo assim, após a experiência-piloto, destaca-se a aprendizagem da lógica presente no comando *Teste* como uma questão pertinente a ser investigada. Avalia-se que isso ocorra em função de sua onipresença no ato de programar. Embora o estudo da lógica não faça parte dos currículos educacionais de matemática há algum tempo, tal conhecimento é fundamental para o desenvolvimento da própria matemática demonstrativa, das ciências e da informática. Além disso, num contexto em que a escola esteja incluída na cultura digital, foram observados indícios de que tal conhecimento se torna necessário, pois serve de base para que os estudantes consigam entender e produzir tecnologia.

É importante ressaltar que a expressão “produzir tecnologia” não é utilizada no sentido de

que os estudantes produzirão inovações tecnológicas, ou seja, inventarão aquilo que ainda não existe no mercado de tecnologia; mas, é empregada numa perspectiva piagetiana de construção de conhecimento. Dessa maneira, quando um estudante cria (ou recria um jogo clássico como o Pong), está criando algo, genuinamente, novo para si. Ampliando, assim, seu conhecimento sobre tecnologia e sobre a cultura digital em que está imerso.

5. DEFINIÇÃO DO PROBLEMA DE PESQUISA E REVISÃO DA LITERATURA

Após observar a forma como o grupo de estudantes se apropriou de tal *software* durante a experiência-piloto, pareceu inevitável estudar o desenvolvimento do raciocínio lógico-matemático dos estudantes quando se pretende investigar qual matemática se pode aprender com o uso do *Etoys*. Para muitos, talvez, tal conclusão seja óbvia em função das dificuldades inerentes à aprendizagem de uma linguagem de programação. Como já foi dito, uma das dificuldades citadas por Alan Kay (2002) é que não é sempre claro entender as coisas que as crianças precisam em determinada idade. Segundo ele, quando a criança está aprendendo a programar é difícil de identificar as causas e motivos do erro ou do êxito.

5.1 PROBLEMA DE PESQUISA

Logo, com o objetivo de contribuir na compreensão deste processo, é relevante pesquisar como se dá aprendizagem da lógica ao se aprender a programar. Sendo mais específico, neste estudo procura-se investigar *Como ocorre o desenvolvimento do raciocínio condicional dos estudantes quando se aprende a programar através do uso do comando Teste do Squeak Etoys?*

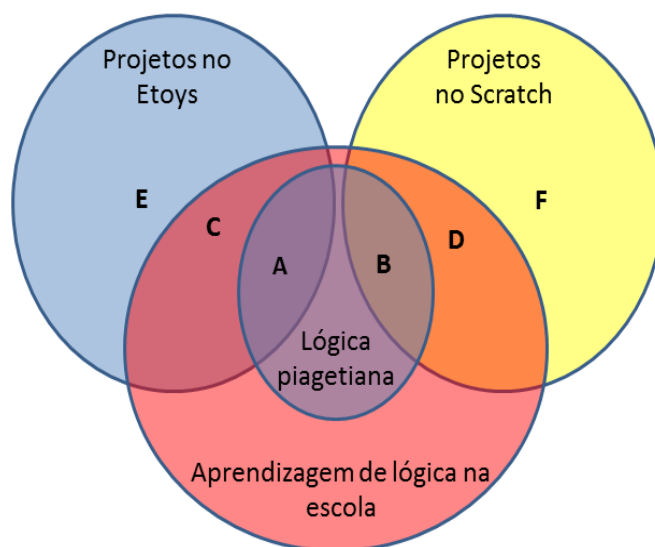
Essa especificação é adotada, pois, até o momento, foi investigado o uso das operações lógicas “dentro” do comando *Teste* do *Etoys*, portanto tal pesquisa está centrada na operação lógica da implicação. No entanto, é relevante ressaltar que não foi a dificuldade dos estudantes que incentiva a compreensão de tal desenvolvimento, mas a riqueza de interpretações e algoritmos que podem ser desenvolvidos nesse tipo de experiência.

Como foi defendido que a adoção de uma concepção construtivista de construção de conhecimento é adequada ao paradigma educacional da cultura digital, pensa-se que é coerente utilizar os estudos de lógica de Piaget et al. (1974), (1975), (1976a) e (1976b) como fundamentação teórica desta pesquisa e que será apresentado no próximo capítulo. Além disso, tal autor desenvolveu uma vasta pesquisa sobre esse tema, cujas descobertas foram essenciais para a construção de sua teoria.

5.2 REVISÃO DA LITERATURA

Dessa maneira foi realizada uma revisão da literatura, em teses e dissertações que investigassem o uso do *Squeak Etoys* ou do *Scratch* em sala de aula (o conjunto $E \cup F$ da figura 6 abaixo). Nessa revisão foram encontradas pesquisas voltadas para a formação de currículos e propostas pedagógicas (Beug, 2012; Ferreira, 2009) e outros estudos que apresentavam alguma implementação de novos recursos para os *softwares* (Silva, 2012 e Gonçalves, 2012). No entanto, a maioria das pesquisas encontradas investigava o desenvolvimento do pensamento computacional ou da criatividade e/ou da autonomia em diferentes disciplinas, tais como:

Figura 6: Diagrama das áreas de conhecimento das teses e dissertações encontradas na revisão de literatura.



Pesquisas voltadas para a disciplina de Matemática

A Tese de Dalla Vecchia (2012) foi a única pesquisa encontrada que apresenta uma análise lógica da programação de um algoritmo. Nela o autor procura investigar a Modelagem Matemática em ambientes produzidos com as Tecnologias Digitais, tendo como interesse a construção de uma visão teórica de Modelagem Matemática que permita compreender as ações dos sujeitos quando estão construindo modelos que se atualizam nesse espaço específico. Nesse estudo, uma parte dos dados coletados e analisados diz respeito aos jogos (feitos no *Scratch*), que foram criados por oito estudantes de um curso de Licenciatura em Matemática. Neles o autor analisa através da lógica proposicional os algoritmos de programação criados para fazer objetos se movimentar na tela: um utilizando coordenadas cartesianas e/ou polares. Segundo Dalla Vecchia (2012), tal fato indica que a linguagem está associada a conceitos matemáticos, pois tais construções podem ser avaliadas em

termos de cálculo proposicional. Logo a linguagem de programação *Scratch* possui uma base notadamente matemática. Embora seja o único trabalho encontrado que faz uso da lógica proposicional para analisar as produções de estudantes, tal trabalho difere deste estudo, pois não foi realizado com estudantes da escola básica e não se faz uma análise genética (no sentido piagetiano) da produção dos estudantes, ou seja, sobre a aprendizagem da lógica proposicional apresentado pelos estudantes. O autor utiliza a lógica proposicional formal com o objetivo de construir ou esboçar um modelo matemático que envolva conteúdos de tal disciplina. Já na pesquisa que se pretende desenvolver a intenção é compreender como ocorre a compreensão das operações lógicas a partir das ações dos estudantes.

O estudo de Capela (2013) procurou investigar se a utilização e construção de recursos digitais pode ter um papel motivador na aprendizagem da disciplina de matemática, potenciador da aquisição e consolidação de conceitos, da autonomia e da autoconfiança do aluno. Nesse estudo o *Scratch* foi utilizado numa tarefa em que os estudantes tinham que criar um jogo, vídeo, etc. que abordasse um conteúdo de matemática. Segundo a autora os resultados indicam que a utilização e criação de recursos digitais são fatores de motivação e que contribuem para uma consolidação de conceitos já adquiridos. No entanto, nem todo o tipo de recurso parece ter a mesma influência motivacional assim como, o contexto em que é utilizado ou produzido, influencia a apreciação dos aprendentes.

Já o estudo de Quintanilla Córdor (2012) investigou como se dá a construção do conceito de fração em crianças do quinto ano da escola primária (na Espanha) utilizando o contexto tecnológico construtivista, representada pelo *Squeak Etoys*. Silva (2009) realizou uma pesquisa, tendo em vista a compreensão da contribuição e o aprofundamento de competências numéricas, em crianças de 6 anos, através de atividades com o *Squeak Etoys*. Pinto (2010) estudou a utilização da tecnologia na aprendizagem de cálculo mental de crianças do 4º ano da escola básica. Nessa investigação os estudantes resolviam, por cálculo mental, problemas matemáticos que eram apresentados pelo *Scratch* (previamente elaborados pela professora). A pesquisadora conversava com os estudantes e registrava tais diálogos em áudio para, após isso, analisar as estratégias utilizadas pelas crianças.

Pesquisa voltada para a disciplina de Química

Lindner (2009) investigou os processos cognitivos que envolvem aspectos da aprendizagem da Química com o uso das NTDs. Para isso, o autor desenvolveu uma arquitetura pedagógica, na qual estudantes do Ensino Médio, utilizaram o *Squeak Etoys* para compreender as experiências químicas realizadas em laboratório. Para esse fim, o *Etoys* foi utilizado para construir mapas conceituais, confeccionar apresentações de estudos (através do recurso livro) e criação de um

modelo digital da experiência.

Pesquisa voltada para a disciplina de Biologia

Lima Filho (2013) pesquisou a aprendizagem colaborativa dos conteúdos de Biologia mediada pelo *Squeak*. Nesse estudo estudantes do Ensino Médio criaram desenhos e animações para representar conteúdos de Biologia. Segundo o autor as situações didáticas criadas ampliaram as interações em que as NTDs podem oferecer benefícios desde que sejam utilizadas em perspectiva colaborativa.

Pesquisa voltada para as disciplinas de Ciências e Matemática

Em sua tese Silva (2008) apresenta uma metodologia de apoio ao processo de aprendizagem nas áreas de Ciências e Matemática via a autoria de objetos de aprendizagem por alunos do Ensino Médio, sendo o *Squeak Etoys* um dos *softwares* utilizados. De acordo com o autor, tal proposta é uma extensão da metodologia de projetos de aprendizagem de Fagundes. Ao analisar os projetos desenvolvidos pelos estudantes, o pesquisador relata a utilização do comando *Teste* em diversos projetos; no entanto, apenas explica como foi utilizado, não fazendo referência sobre a aprendizagem das proposições lógicas e nem sinaliza alguma dificuldade nessa utilização. Além disso, Silva (2008) afirma que foi possível verificar que o aluno, com apoio técnico e através de um ambiente de autoria para o usuário final, é capaz de produzir parte (ou totalidade) de seus desejos de aprendizagem – representados por objetos de aprendizagem.

Pesquisa voltada para as disciplinas de História e Geografia

Nessa pesquisa Pires (2013) criou um webquest sobre história e geografia no *Scratch* e aplicou em seus alunos. Portanto se refere à apropriação dos professores das NTDs com o objetivo de utilizar em suas aulas.

Pesquisas voltadas para a disciplina de Artes Visuais

De acordo com Oliveira (2013) seu estudo tinha como objetivo perceber até que ponto a promoção de atividades pedagógicas baseadas na aprendizagem cooperativa poderia por um lado incrementar a motivação pela área da informática e, por outro, diminuir relacionamentos de competitividade negativa entre turmas de áreas divergentes, mas complementares. Em função disso, a estratégia implementada baseou-se na criação de um trabalho cooperativo entre tais turmas, o desenvolvimento de um jogo multimídia, em que cada turma assumiu papéis diferenciados, tendo em conta as áreas curriculares de cada uma. De acordo com este pressuposto, a turma de Artes Visuais assumiu a produção de todos os elementos gráficos, videográficos e sonoros enquanto a

turma e Ciências e Tecnologias programou o jogo.

Já a pesquisa de Moreno (2012) foi realizada com estudantes do curso de licenciatura em Artes Visuais e tinha como objetivo promover uma aproximação entre Arte/Tecnologia e o ensino/aprendizagem de Artes Visuais através da problematização de aspectos relevantes para tal área de conhecimento na relação crítica e criativa com as tecnologias. Nesse estudo a autora apresenta propostas de ensino usando o *Scratch* a partir de uma experiência prática realizada com tais alunos.

Pesquisas voltadas para as séries iniciais do Ensino Fundamental

Nesse estudo, Faria (2014) visa compreender de que forma podem diversos recursos digitais contribuir para o a aprendizagem e para o desenvolvimento integral das crianças do jardim de infância; ele analisa também os contributos da rede social para a interação com os gestores de educação, educadores, professores, investigadores e para o desenvolvimento profissional docente. O *Squeak Etoys* é utilizado no desenvolvimento de um projeto sobre os “peixes, nós e as redes”. Primeiro as crianças desenvolveram uma investigação sobre tudo o que estava relacionado com o rio, construíram uma rede de pescar em sala de aula. Logo em seguida utilizaram o *Squeak* para desenhar e animar um rio, com peixes, barcos etc.

Já a pesquisa de Dullius (2012) trata das possibilidades de utilização do *Squeak* como uma ferramenta de autoria com o objetivo de apoiar e estimular as múltiplas alfabetizações, o processo de criação de ideias e a fluência digital. Já Ferreira (2010) realizou um estudo no qual as crianças criavam os seus próprios projetos com o *Etoys* e nesse processo o autor procurou identificar uma série de características do *Squeak* que fomentem a construção de conhecimento, enquanto ferramenta de programação destinada a ajudar as crianças a aprender criando. As atividades abordam diferentes conteúdos como matemática, física, astronomia, biologia, etc.

Pesquisas voltadas para as disciplinas de Games e Ciência da Computação

Entre esses trabalhos destaca-se a pesquisa de Marques (2013) que desenvolveu uma proposta de ensino de programação através do *Scratch*. Para isso o autor trabalhou com um grupo de estudantes do 12º ano do ensino secundário de uma escola de Portugal. Segundo ele os estudantes já tinham experiência de programação, mas não conheciam o *Scratch*. Dessa maneira, seu estudo apresenta uma série de atividades com o objetivo de ajudá-los, nesse processo de criação. Dentre elas, se sobressaem as atividades que envolviam o uso de listas no *Scratch*, pois possibilitam a aprendizagem de uma série de conceitos matemáticos (sequências de números: pares, ímpares, primos, divisores de n, etc.) que, segundo Marques, são os conceitos mais complexos e abstratos para os estudantes e muito importantes na área da programação. Para avaliar a produção

final dos estudantes, o pesquisador se baseou num estudo de Maloney et al (2008) que define que um projeto será considerado de programação se este apresentar pelo menos um dos seguintes critérios: (a) Comunicação e sincronização, (b) Ciclos/Fases, (c) Declarações condicionais, (d) Interação com utilizador, (e) Lógica Booleana, (f) Variáveis e (g) Números aleatórios. Assim, Marques apenas apresenta o número de vezes que cada critério foi utilizado em cada jogo e, assim, os classifica em diferentes níveis de complexidade. Infelizmente para o interesse de nossa pesquisa, o pesquisador não investiga especificamente a aprendizagem da lógica proposicional, quando os estudantes desenvolvem algoritmos e nem descreve alguma dificuldade com as operações lógicas da conjunção, disjunção, negação ou implicação que seus estudantes poderiam ter apresentado.

Em sua tese Gallardo (2014) afirma que uma das dificuldades na adoção, por parte dos professores, da aprendizagem baseada em jogos (também conhecidos como *Game-Based Learning (GBL)* em contextos formais de ensino é que tais jogos precisam estar alinhados e adaptados ao currículo formal das disciplinas. Sendo assim, sua tese contribui com um modelo para a representação computacional de jogos de tabuleiros e uma série de guias para implementá-los nas diferentes áreas. Os jogos que compõe seu modelo foram desenvolvidos com um grupo de 35 professores e 253 estudantes. Em seu trabalho Gallardo cita o *Squeak* como uma alternativa para o professor construir tais jogos, porém, em sua pesquisa, não aborda o que estamos investigando.

Como se viu, não foram encontradas pesquisas que abordassem a aprendizagem de lógica de programação numa perspectiva genética piagetiana (representada pelo conjunto $A \cup B$ da figura 6 acima), tal fato caracteriza este estudo como inédito e com potencial para apresentar contribuições para tal área de conhecimento.

5.3 METODOLOGIA DA PESQUISA

Para a coleta de dados esta pesquisa, num primeiro momento, a intenção do pesquisador foi influenciada pela experiência-piloto e desejava-se investigar o raciocínio lógico dos estudantes a partir da programação e da utilização dos jogos lógicos de Zoltan Dienes, pois estes foram desenvolvidos durante o movimento da matemática moderna e exploram as operações lógicas (conjunção, disjunção e negação) que são amplamente utilizadas em programação. Tal escolha foi realizada, pois Piaget (1972) afirmava que as noções essenciais que caracterizavam a matemática moderna eram mais próximas das estruturas de pensamento “natural” do que os conceitos usados na matemática tradicional.

Assim, foi desenvolvida uma oficina de *Squeak Etoys* com uma turma do nono ano do ensino fundamental do Colégio de Aplicação da Universidade Federal do Rio Grande do Sul na disciplina de matemática. Nela foram realizados quinze encontros de uma hora e quarenta minutos. Nessa turma alguns estudantes já conheciam o *Etoys*, pois dois anos antes tiveram uma oficina. Alguns deles já estavam habituados a utilizá-lo e, no final de um dos encontros, para surpresa dos pesquisadores um estudante pediu ajuda para resolver um problema de um projeto que havia criado de forma independente. Tal projeto era uma simulação que fazia uso do comando *Teste*, o diálogo com o estudante (que será apresentada no capítulo 7) foi tão significativo que a metodologia da pesquisa foi modificada.

Durante a coleta de dados foi construído um experimento com três desafios de programação com complexidade crescente, cuja solução envolvia o uso da operação lógica da implicação. Com o objetivo de compreender a forma como os estudantes iriam se apropriando das operações lógicas envolvidas na sua ação e identificar as etapas desse desenvolvimento, foram realizadas entrevistas com os estudantes segundo o método clínico piagetiano.

As entrevistas foram realizadas com nove crianças, com idades entre 10 e 16 anos, que cursavam as séries finais do Ensino Fundamental de duas escolas públicas. Sendo quatro estudantes (com idades entre 14 e 16 anos) de uma turma do nono ano do Ensino Fundamental do Colégio de Aplicação da UFRGS e cinco crianças de 10 anos que faziam parte de um grupo de programação na escola estadual de Ensino Fundamental Dinah Péri Pereira que participava do Projeto UCA. Uma das entrevistas foi realizada por dois entrevistadores, o autor da pesquisa e o parceiro e pesquisador do LEC/UFRGS Bruno Fagundes Sperb. Já as demais foram realizadas por um ou por outro.

Logo, os dados, a análise e os resultados apresentados nesta tese versam sobre o uso do raciocínio condicional quando crianças e adolescentes resolvem desafios de programação e tem como base teórica os estudos de lógica de Piaget. Portanto, a partir de agora, ficam num segundo plano as discussões sobre os projetos de aprendizagem no *Etoys* e da inclusão da escola numa cultura digital.

6. A LÓGICA NA TEORIA PIAGETIANA

Como se pretendia realizar uma investigação sobre o desenvolvimento do raciocínio lógico-matemático dos estudantes ao utilizarem o comando *Teste* do *Etoys* a partir de uma perspectiva construtivista piagetiana, tornou-se importante apresentar a sua teoria e compreender seus estudos de lógica.

6.1 A TEORIA PIAGETIANA

Interessado em estudar como ocorre o processo de construção do conhecimento científico na criança, Piaget investigou tal processo desde o nascimento do bebê até a adolescência. Pode-se dizer que Piaget estava interessado em descobrir como uma criança (o bebê) se desenvolve ao ponto de ser capaz de criar modelos abstratos e sofisticados que podem ampliar os limites da ciência e mudar a forma como se interpreta a realidade que nos cerca. Dessa maneira, a teoria de conhecimento proposta pelo autor foi chamada como Epistemologia Genética.

Jean Piaget era biólogo de formação e se inspirou nessa área de conhecimento ao afirmar que a inteligência é uma *adaptação* e isso equivale a dizer que é, também, uma *organização*, cuja função consiste em estruturar o universo, tal como o organismo estrutura o meio imediato. Segundo ele, o organismo adapta-se construindo materialmente novas formas para inseri-las nas do universo, por outro lado, a inteligência a prolonga construindo, mentalmente, as estruturas suscetíveis de aplicação às do meio. Logo *adaptação* e *organização* são funções da inteligência.

De acordo com o autor, na biologia há adaptação quando o organismo se transforma em função do meio e tal variação traz ganhos tanto para o meio quanto para o organismo que nele se conserva. Piaget ressalta que é importante pensá-la como uma sequência de processos que se desenvolvem continuamente ao longo do tempo.

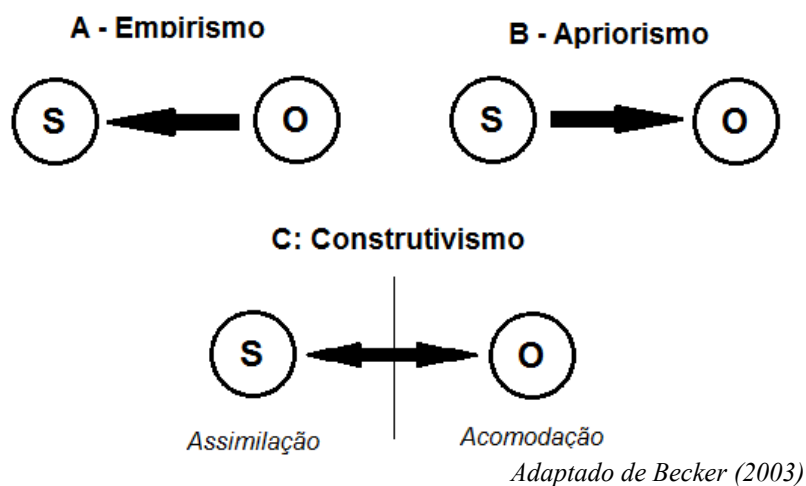
Assim, a adaptação intelectual se dá através do equilíbrio progressivo entre dois mecanismos: a *assimilação* e a *acomodação*. Segundo o autor, a inteligência é *assimilação* “na medida em que incorpora nos seus quadros (esquemas) todo e qualquer dado da experiência” (p.18, 1982) e é *acomodação*, pois “ao incorporar os novos elementos nos esquemas anteriores, a inteligência modifica incessantemente os últimos para ajustá-los aos novos dados” (p.18, 1982). Logo só haverá adaptação quando houver uma acomodação perfeita, ou seja, nada que venha da realidade modifica os esquemas do sujeito. No entanto, tal fato indicaria que haveria coerência entre os esquemas e os novos dados coletados da experiência, ou seja, assimilação. Assim, Piaget (1982) afirma que “as coisas nunca são conhecidas nelas mesmas”, pois a acomodação necessita da

assimilação e vice-versa. Dessa maneira, a adaptação só acontece quando atinge um sistema estável, ou seja, quando existe equilíbrio entre assimilação e acomodação.

A segunda função da inteligência é a *organização* que é um processo complementar à adaptação e é definida pelas relações entre as partes e o todo. Em que cada esquema está coordenado com os demais e constitui (ele próprio) uma totalidade de partes diferenciadas. Segundo Piaget “esses dois aspectos do pensamento são indissociáveis: é adaptando-se às coisas que o pensamento se organiza e é organizando-se que estrutura as coisas” (p.19, 1982).

Sendo assim, a teoria de conhecimento proposta por Piaget diverge das duas correntes filosóficas da época: o *behaviorismo* ou *empirismo*, que dizia que a origem do conhecimento ocorre em função do meio, anulando a intervenção do sujeito (como mostra a figura 7A abaixo, no sentido do objeto para o sujeito) e o *apriorismo* que afirma que tal origem está no sujeito (figura 7B), portanto é inata e depende exclusivamente de um processo de maturação, ocorrendo independente do meio.

Figura 7: Teorias de Conhecimento



Em sua teoria, Piaget ressalta que a origem do conhecimento ocorre na interação entre *sujeito*² e *objeto* e através da ação do primeiro (figura 7C). Essa ação assimilativa do sujeito tem como objetivo incorporar o objeto aos seus sistemas de significação. No entanto, o objeto lhe oferecerá uma resistência, então o sujeito precisa se modificar e reestruturar tais sistemas, ou seja, a partir de uma ação de acomodação. Tal processo ocorre de maneira contínua e progressiva, no qual o sujeito e objeto constroem-se à medida que se diferenciam. Logo, para Piaget a adaptação se dá através da ação.

Dessa maneira Piaget reconhece que o meio tem influência no processo de construção, mas

² Lembrando que o sujeito é aquele que busca o conhecimento e o objeto é tudo aquilo que se deseja conhecer e que não é o sujeito. Portanto o objeto pode ser qualquer coisa: um conceito de matemática, de sociologia, confeitar bolos, construir robôs, a própria epistemologia genética, etc.

não é algo extrínseco ao sujeito e a realidade objetiva é construída a partir de sua própria ação. Além disso, explica também que há um aspecto hereditário na origem do conhecimento, mas para Piaget são as estruturas biológicas inatas da espécie humana tais como nervos, músculos, reflexos, etc. No livro *O nascimento da inteligência na criança*, o pesquisador investiga minuciosamente o surgimento dos primeiros esquemas motores que evoluem de exercícios reflexos automáticos (sucção, preensão, visão, audição e fonação) e se transformam em ações intencionais dos bebês para conhecer o seu meio imediato. Para Piaget, esquema é tudo aquilo que se conserva na ação. Como Chiarottino explica:

[...] o esquema “pegar”; derivado do reflexo de preensão. “pega”. Pega algo. A Piaget, porém, não importa o “ser capaz de pegar este ou aquele objeto”; importa o próprio “pegar”, o “saber pegar” (seja lá o que for). Digamos que a Piaget não interessa o “conteúdo” de cada ato particular de pegar (algo), mas a forma do pegar” (1988, p. 11).

Com esse estudo o autor teve a oportunidade de observá-los isoladamente por um curto período de tempo, pois em pouco tempo os bebês começam a coordenar tais esquemas, formando sistemas de ações cada vez mais complexos e difíceis de observá-los isoladamente. Assim a criança age no mundo para organizá-lo e estruturá-lo, internamente, e assim constrói estruturas mentais que provêm dessas atividades motoras.

Nas entrevistas com as crianças, o olhar epistemológico de Piaget observou que existia uma lógica nas ações da criança e que era isomorfa à lógica matemática das classes e das relações.

Dois esquemas podem ser coordenados ou dissociados (reunião), um pode ser parcialmente incluído no outro (inclusão), ou somente ter uma parte em comum com o outro (intersecção); as partes de um esquema ou a coordenação de dois ou mais esquemas podem permitir uma ordem invariante de sucessões, ou certas permutações (tipos de ordem), assim como correspondência um-a-um, um-a-muitos ou muitos-a-um (bijeção, etc.) [...] Nós acreditamos, ao contrário, que existe, como uma função do desenvolvimento global da inteligência, uma espontânea e gradual construção das estruturas lógico-matemáticas elementares. (Piaget, 1972)

Como tal lógica era inconsciente às crianças e subjacente às suas ações, logo ele deduziu que, antes mesmo da linguagem, havia um tipo de estrutura lógica implícita ao comportamento humano e que provinha das ações do sujeito. Isso, segundo Piaget, indicava que a lógica matemática formal é da maneira que é, pois sua origem estava na lógica das ações. Para o autor essa capacidade de organizar e estruturar a experiência vivida vinha da própria atividade das estruturas mentais que funcionavam seriando, ordenando, classificando e estabelecendo implicações.

6.2 OS ESTÁDIOS DO DESENVOLVIMENTO E AS ESTRUTURAS LÓGICAS

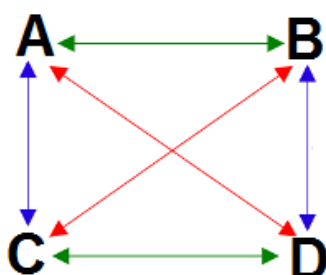
Sendo assim, ao longo de sua vida Piaget procurou reconhecer as relações e semelhanças

entre as formas de raciocínios utilizadas pelas crianças e os modelos formais da matemática. Por conseguinte, Piaget propôs que o desenvolvimento cognitivo ocorria através de quatro estádios. Vale ressaltar que tais estádios foram propostos por diferentes fatores, porém o que interessa para esta tese é indicar quais as estruturas lógicas que estão relacionadas a cada um deles.

6.2.1 O sensório-motor e o grupo de Deslocamentos

No estádio *Sensório-motor* (0 a 2 anos), a partir dos reflexos, há a construção dos primeiros esquemas e suas primeiras coordenações. Pode-se afirmar que nesse estádio há prevalência da percepção, portanto a inteligência é prática e ocorre em função do contato com o meio. Como vimos na secção anterior, segundo Piaget é possível observar uma lógica da ação desde os bebês que conduz à elaboração de certas estruturas. Tais observações levam-lhe à ideia de se inspirar no grupo de transformações de Henri Poincaré para criar um modelo lógico para o sensório-motor. Em seu estudo sobre a permanência do objeto o autor afirma que todos os bebês conseguem atualizar, sem dificuldades, as observações de Poincaré.

Figura 8: Grupo de Deslocamentos



Isso significa afirmar que o desenvolvimento progressivo das ações sobre os objetos e de seus deslocamentos obedeceria às seguintes quatro leis:

1º) *Lei da composição*: É a reunião ou dissociação de dois ou mais movimentos. Um bebê percebe que pode ir da varanda à cozinha realizando dois movimentos: primeiro indo da cozinha à sala e depois da sala à varanda.

2º) *Lei da inversão*: Para cada ação citada anteriormente, há a sua inversa. Essa criança pode ir engatinhando ou andando da cozinha para sala e retornar imediatamente.

3º) *Lei da transformação nula, identidade*: Voltar à posição inicial através de uma ação direta e outra inversa, ou seja, é uma *reversibilidade prática*. Exemplo: para voltar à cozinha realiza uma

composição de dois movimentos inversos (da varanda à sala e, depois, vai até a cozinha) anulando, assim, seu primeiro deslocamento.

4º) *Lei da associatividade*: Este bebê pode tomar diferentes caminhos para ir da cozinha à varanda, mesmo quando lhe são colocados uma série de obstáculos (banco, cercado, etc.), portanto o ponto de chegada é o mesmo.

Para Piaget isso seria, pois, de fato ações intencionais e não uma sucessão anárquica de movimentos, fazendo do bebê mestre de seus movimentos e das ações sobre o objeto. No entanto, é relevante ressaltar que tal ação ocorre para satisfazer uma necessidade prática e num curto intervalo de tempo, sem que o bebê tenha consciência de tal lógica. Em suma, na construção de um espaço contínuo, da permanência do objeto, se esconde na estrutura do grupo de transformações, isto é, a estrutura geral da inteligência dos bebês.

6.2.2 O pré-operatório: o período pré-lógico

No estágio Pré-operatório, segundo Chalon-Blanc (1997), Piaget não conseguiu encontrar, na lógica das ações da criança, um modelo formal para esse período. Por conseguinte, classificou tal estágio como pré-lógico, cuja função era encerrar o *sensório-motor* e preparar para o operatório.

Isso significa dizer que as crianças até realizam ações diretas nos objetos, porém tais ações não são reversíveis, portanto elas apresentam muitas dificuldades em compreender alguns conceitos. As provas de conservação de massa exemplificam tal fato, após transformar uma bola de massa de modelar na forma de num palito, as crianças afirmam que a quantidade de massa se modificou. Elas até conseguem memorizar o estado inicial de um objeto (“isto era uma bola”, “é fino, é longo”), porém não são capazes de, em pensamento, transformar a palito novamente em bola, logo não constroem um invariante e, assim, a conservação da massa não é alcançada. Isso indica que, sem a reversibilidade da ação as crianças não atingem uma forma de pensamento com equilíbrio estável que lhes permita compreender uma série de conceitos.

É importante ressaltar que uma das características fundamentais do estágio *Pré-operatório* é o surgimento da linguagem ou função simbólica. No entanto, ainda não há operação por falta de reversibilidade na ação.

6.2.3 Operatório concreto e a estrutura de agrupamento

No estágio *Operatório concreto* (7 a 12 anos) seu principal atributo é o surgimento da reversibilidade da ação, ou seja, da *operação*. Segundo Piaget, as operações são “ações

interiorizadas, quer dizer executadas não mais material, mas interior e simbolicamente, e ações que podem ser combinadas de todas as maneiras; em particular, que podem ser invertidas, que são reversíveis” (1982, p. 216). Segundo ele a estrutura lógica do operatório concreto é o agrupamento.

Chamamos de agrupamento todo conjunto de classes que é formado da seguinte maneira: Dada uma classe de elementos A, existe a classe B que é formada por elementos de A e não A, ou seja, $A + A' = B$. Por oposição, A' são os B que não são A, $B - A = A'$. Por sua vez há outra classe C que é formada pelos B e os não B, $B + B' = C$. Assim como, por inversão, B' são os C que não são B, $C - B = B'$.

A estrutura de agrupamento possui as mesmas quatro leis do sensório-motor, que são isomorfas ao grupo de transformações de Poincaré, porém não são mais ações de curto prazo para suprir necessidades práticas, mas ações internalizadas que atendam a necessidade de compreender o real. Portanto no operatório concreto têm-se as seguintes leis ou operações:

1º) *Lei da composição*: Na classificação são aquelas ações que tem por objetivo reunir ou dissociar os elementos num todo. Na composição de duas classes disjuntas as maçãs mais as laranjas formam uma classe que as inclui, as das frutas: $A + A' = B$. Já nas operações aritméticas do número de elementos de um conjunto: a composição de 4 mais 2 é obtido um terceiro número, o 6: $n(A) + n(A') = n(B)$.

2º) *Lei da inversão*: Para cada uma das ações realizadas anteriormente, comporta sua inversa, na reunião de $4 + 2$ temos o 6, então pode-se obter novamente o 4 fazendo $6 - 2$: $n(B) - n(A') = n(A)$. Assim como, da reunião de maçãs e laranjas, sempre pode-se obter as maçãs, dissociando das frutas as laranjas: $B - A' = A$. Dessa maneira há a reversibilidade

3º) *Lei da transformação nula, identidade*: Tal lei sempre assegura a reversibilidade a toda estrutura. O resultado da ação direta pela sua inversa é o estágio inicial, logo seu produto é nulo. Em função deste fato, nas provas de conservação de massa, torna-se possível para a criança verificar que a massa não se modifica, pois da mesma maneira que ela transforma uma bola em palito, pode-se fazer a ação inversa de transformar o palito em bola novamente, portanto a massa não se altera. Nos dados de pesquisas apresentados por Piaget a noção de conservação da massa é atingida por volta dos 7 a 8 anos, do peso por volta dos 9 a 10 anos, do volume entre os 11 e 12 anos. Logo, será a lei da identidade que o torna capaz de classificar, seriar, colocar em correspondência. Portanto, Becker (2010) afirma que, na teoria piagetiana, não haveria conjunto de operações ou sistemas operatórios sem a reversibilidade, pois a operação direta se apoia na inversa reciprocamente.

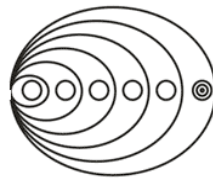
4º) *Lei da associatividade*: Nesse momento é importante distinguir o grupamento aritmético do lógico. No agrupamento aritmético a associatividade é completa no sistema. Por exemplo, existem

muitas maneiras de se obter o número 6. Note no exemplo abaixo que o valor da soma não se altera independente da disposição dos parênteses.

$$\begin{aligned}
 (1 + 1 + 1 + 1 + 1) + 1 &= 5 + 1 \\
 (1 + 1 + 1 + 1) + 1 + 1 &= 4 + 2 \\
 (1 + 1 + 1) + 1 + 1 + 1 &= 3 + 3 \\
 (1 + 1) + (1 + 1) + (1 + 1) &= 2 + 2 + 2 \\
 6 &= 6
 \end{aligned}$$

Tal fato ocorre, em função da inclusão hierárquica da estrutura de número.

Figura 9: Inclusão hierárquica



Já no agrupamento lógico, que engendra as classificações elementares, a associatividade está limitada pela tautologia. Maçãs mais maçãs, formam uma classe de maçãs. ($A + A = A$). Enquanto que $1 + 1$ faz 2 e não 1, por conseguinte, a tautologia suprime a iteração aditiva $+1$ e não permite a livre distribuição dos parênteses.

$$\begin{aligned}
 (A + A) - A &\neq A + (A - A) \\
 \emptyset &\neq A
 \end{aligned}$$

Sendo assim, tal tautologia impõem algumas armadilhas para as crianças nos problemas que envolvam a inclusão de classes. Após ser perguntada “se irmos acrescentando maçãs, no final teremos mais maçãs ou frutas?” Em crianças desse estágio, é comum afirmarem que há mais maçãs do que frutas (B), indicando que tal tautologia não está bem construída. Visto que, ao acrescentar maçãs ao conjunto das frutas; será, na melhor das hipóteses (nos casos em que $A' = \emptyset$), igual ao número de frutas, mas nunca maior.

$$A + A + A + A + A + A = A \leq B$$

Assim, esse tipo de resposta indica que a criança desse período não consegue considerar que adicionar qualquer maçã ao conjunto das maçãs está ao mesmo tempo adicionando uma fruta ao conjunto das frutas. Portanto, para Piaget, isso indica que falta uma dissociação nula pela adição, ou

seja, uma composição simultânea entre reunião e reversibilidade por inversão.

Logo o agrupamento explica uma primeira forma de equilíbrio estável que, apoiada nas estruturas de classificação e seriação, torna possível a reversibilidade operatória (por inversão ou reciprocidade), no entanto a criança só realiza tal reversibilidade apoiando seu pensamento numa primeira manipulação concreta do objeto, obtendo êxito a partir da segunda ou terceira tentativa, etc. Isso se deve ao fato de que, num agrupamento, as composições são constituídas pouco a pouco através da proximidade entre as classes, mas sem atingir uma combinatória entre elas. Tanto a inclusão de classes, quanto a necessidade de se recorrer ao real só será superada no estágio seguinte, quando a criança torna-se capaz de realizar tais operações a partir de hipóteses e/ou deduções, ou seja, de realizar uma operação sobre proposições lógicas que levam à compreensão da própria tautologia e da inclusão de classes.

6.2.4 O operatório formal – Da lógica das proposições ao INRC

No *operatório formal* (após os 12 anos), como vimos anteriormente, será caracterizado pela capacidade da criança, agora adolescente, de levantar hipóteses e realizar deduções, tal fato indica o surgimento do pensamento dedutivo. Segundo Piaget (1976) tais características estão ligadas a uma estruturação operatória inteiramente nova, fundada sobre a lógica das proposições e sobre um pensamento formal distinto do pensamento operatório concreto. Portanto, para compreender tal lógica, torna-se importante conhecer as seguintes definições apresentadas por Piaget no seu Ensaio de lógica operatória:

Definição 1: – Chamaremos de **proposições** p, q, r, etc. os enunciados categóricos, verdadeiros ou falsos e afirmativos (positivos) ou negativos. [...] *Definição 2:* – Será chamada de “**operação interproposicional**” toda composição que permite construir, por meio de proposições quaisquer, p, q, r, etc. das quais se conhecem apenas os valores de verdade ou de falsidade, outras proposições bem determinadas e caracterizadas respectivamente pelas diversas combinações possíveis destes únicos valores de verdade. [...] *Definição 3:* – Chamaremos de “**intraproposicionais**” as operações que permitem decompor uma proposição em elementos (esta decomposição podendo ser levada a graus variáveis), e construir novas proposições determinadas pelas transformações destes elementos; os valores verdadeiros e falsos das proposições assim engendradas resultam então das combinações entre os próprios elementos. Piaget, (1976, p.32).

De acordo com tais definições, os enunciados “ x_i é um pássaro” e “ x_i é um ser vertebrado” são proposições, pois ambas podem assumir valores verdadeiros ou falsos. Note que tais proposições foram representadas na linguagem comum, mas podem ser retratadas com os signos convencionais da lógica. p: “ x_i é um pássaro” é verdadeira, q: “ x_i é um ser vertebrado” é verdadeira e suas respectivas falsidades \bar{p} e \bar{q} .

Com isso pode-se criar um conjunto de operações formadas pela composição de uma com outra ou com ela mesma, com o objetivo de formar novas proposições com valores de verdade ou falsidade bem determinados. Tais operações advêm da combinação entre proposições e podem ser classificadas como *inter* ou *intraproposicionais*.

As operações intraproposicionais são aquelas que a análise da veracidade ou falsidade da nova proposição ocorre a partir dos *conteúdos* das primeiras. Ou seja, para analisar o enunciado “o canário é um pássaro invertebrado” se recorre à análise dos elementos da classe pássaros e invertebrados: é verdade que um canário “é pássaro”, mas é falso que “é invertebrado”, logo o enunciado “o canário é um pássaro invertebrado” é falso, pois ambas deveriam ser verdadeiras. Outro exemplo, “todo quadrado é um retângulo” é uma proposição verdadeira, pois ser retângulo significa possuir “quatro lados” e “quatro ângulos retos” e ambas proposições são verdadeiras para o quadrado. Ou seja, as operações intraproposicionais são constituídas pelas estruturas de classes e relações e são características do estágio operatório concreto.

Já as operações interproposicionais deixam de recorrer às classes para analisar a veracidade ou falsidade das proposições p , q , etc. Retomando os exemplos do parágrafo anterior:

Exemplo 1: Sendo p : *o canário é um pássaro* e q : *o canário é invertebrado*, então avaliar “*se o canário é um pássaro invertebrado*” é analisar a operação $(p \cdot q)$.

Pela linha (2) tabela verdade 2 (abaixo) tem-se que $(p \cdot q)$ é falso.

Tabela 2: Tabela verdade da operação $(p \cdot q)$

	p	q	p · q
(1)	Verdadeiro	Verdadeiro	Verdadeiro
(2)	Verdadeiro	Falso	Falso
(3)	Verdadeiro	Verdadeiro	Falso
(4)	Falso	Falso	Falso

Exemplo 2: Sendo p : *o quadrado possui quatro lados* e q : *o quadrado possui quatro ângulos retos*, então avaliar “*se o quadrado é um retângulo*” é analisar a operação $(p \cdot q)$. Pela linha (1) tabela verdade 2 (acima) tem-se que $(p \cdot q)$ é verdade.

Exemplo 3: Um número é divisível por seis quando é divisível pelos números primos dois e três. Sendo p : *x_i é divisível por dois* e q : *x_i é divisível por três*, então avaliar “*se 16 é divisível por seis*” é analisar a operação $(p \cdot q)$. Pela linha (4) da tabela verdade 2 (acima) tem-se que $(p \cdot q)$ é falso. Já pela linha (1) tem-se que “*dezoito é divisível por seis*”.

Nos exemplos vistos acima, foi utilizada a mesma tabela verdade para diferentes conteúdos. Dessa maneira, a lógica das proposições negligencia os *conteúdos* e passa a considerar apenas a

forma das combinações entre proposições; constituindo, assim, um sistema novo. Desse ponto de vista, segundo Piaget, a lógica proposicional é superior à das classes.

“Ora, como a lógica das classes e das relações mostra já uma estrutura formal independente do seu próprio conteúdo, a lógica das proposições pode ser considerada como referindo-se a uma forma cujo conteúdo é formado por formas de nível inferior. O cálculo das proposições representa, portanto uma forma de formas e consiste em operações de segunda potência ou operações que se exercem sobre operações de primeira potência” (1976, p.206)

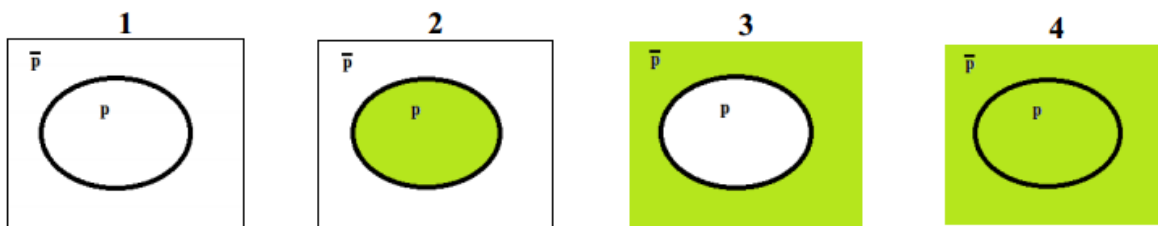
Dada a proposição simples p , que pode ser verdadeira (1) ou falsa (0), portanto toma seu valor do conjunto $\{1,0\}$. Assim todas as aplicações possíveis de tal proposição é dada por todas as combinações entre 0 e 1, ou melhor, por $f: \{1,0\} \rightarrow \{1,0\}$ e que são representadas na tabela verdade abaixo.

Tabela 3: Tabela verdade das combinações possíveis de p

p	1	2	3	4
1	0	1	0	1
0	0	0	1	1

Note que se p é representada pela aplicação 2 $\{1,0\}$, então 3 será a negação \bar{q} . Abaixo temos a representação em diagramas de Venn das operações possíveis.

Figura 10: As quatro combinações lógicas possíveis para a proposição p



6.2.4.1 As 16 possibilidades lógicas

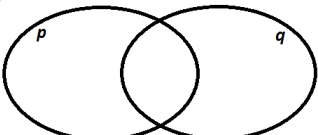
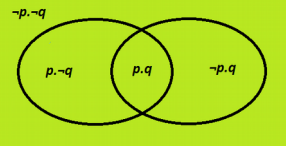
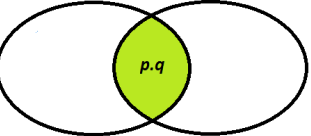
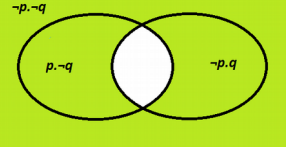
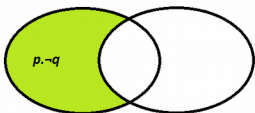
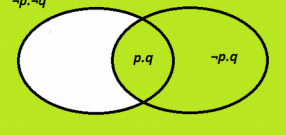
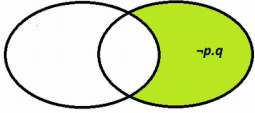
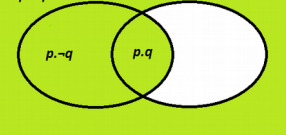
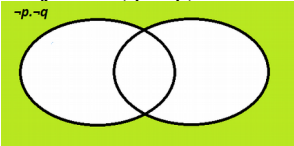
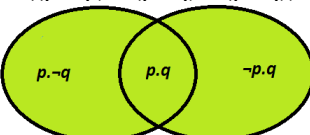
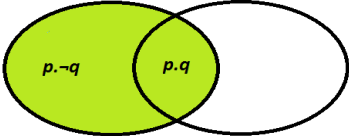
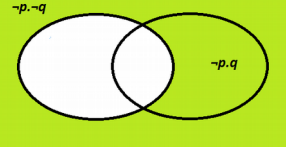
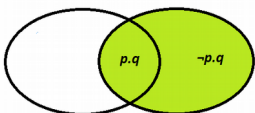
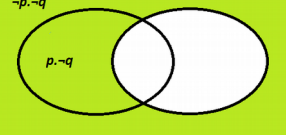
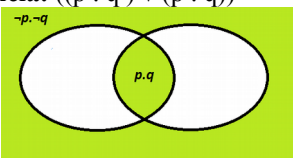
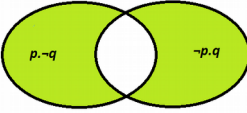
No final da secção anterior foram estudadas as possibilidades lógicas de uma proposição simples, agora vamos examinar as operações lógicas, ou seja, toda combinação de quaisquer proposições (tomadas duas a duas) que definem outras proposições bem determinadas. Dadas duas proposições simples p e q , independentes uma da outra, em que cada uma pode assumir um valor verdadeiro (1) ou falso (0). Assim o par ordenado (p, q) pode assumir os seguintes valores do conjunto $\{1,0\} \times \{1,0\} = \{(0,0); (1,0); (0,1); (1,1)\}$. Todas as possibilidades de combinações entre elas é $2^{2^2} = 16$ que é dada por $g: \{1,0\} \times \{1,0\} \rightarrow \{1,0\}$.

Tabela 4: Tabela verdade das possibilidades lógicas de p e q

(p, q)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(1, 1)	0	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1
(1, 0)	0	0	1	0	0	1	0	0	1	1	0	1	1	1	1	1
(0, 1)	0	0	0	1	0	0	1	0	1	0	1	1	0	0	1	1
(0, 0)	0	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1

Para facilitar tal constatação o quadro abaixo apresenta cada uma das operações sendo representadas de três maneiras distintas: a) pelos signos proposicionais p e q ; b) pelo diagrama de Venn (a região verde indica quando a proposição verdadeira) e c) pelo seu valor na tabela verdade (TV) para $\{p, q\} = \{(0,0); (1,0); (0,1); (1,1)\}$.

Tabela 5: Diferentes representações das 16 possibilidades lógicas

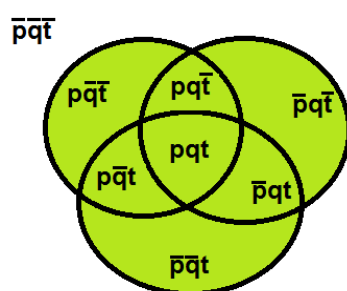
<p>1. A negação: $((\bar{p} \cdot \bar{q})) = (0)$</p> <p>TV: {0 0 0 0}</p> 	<p>16. A Tautologia: $((p \cdot q) \vee (p \cdot \bar{q}) \vee (\bar{p} \cdot q) \vee (\bar{p} \cdot \bar{q}))$</p> <p>TV: {1 1 1 1}</p> 
<p>2. A Conjunção: $(p \cdot q)$</p> <p>TV: {1 0 0 0}</p> 	<p>15. A Incompatibilidade: $((p \cdot \bar{q}) \vee (\bar{p} \cdot q) \vee (\bar{p} \cdot \bar{q}))$</p> <p>TV: {0 1 1 1}</p> 
<p>3. A Não Implicação: $(p \cdot \bar{q})$</p> <p>TV: {0 1 0 0}</p> 	<p>14. A implicação: $((p \cdot q) \vee (\bar{p} \cdot q) \vee (\bar{p} \cdot \bar{q}))$</p> <p>TV: {1 0 1 1}</p> 
<p>4. A Não Implicação Recíproca: $(\bar{p} \cdot q)$</p> <p>TV: {0 0 1 0}</p> 	<p>13. A implicação Recíproca: $((p \cdot q) \vee (p \cdot \bar{q}) \vee (\bar{p} \cdot \bar{q}))$</p> <p>TV: {1 1 0 1}</p> 
<p>5. A Negação Conjunta: $(\bar{p} \cdot \bar{q})$</p> <p>TV: {0 0 0 1}</p> 	<p>12. A Disjunção: $((p \cdot q) \vee (p \cdot \bar{q}) \vee (\bar{p} \cdot q))$</p> <p>TV: {1 1 1 0}</p> 
<p>6. A Afirmação de p: $((p \cdot q) \vee (p \cdot \bar{q}))$</p> <p>TV: {1 1 0 0}</p> 	<p>11. A Negação de p: $((\bar{p} \cdot q) \vee (\bar{p} \cdot \bar{q}))$</p> <p>TV: {0 0 1 1}</p> 
<p>7. A afirmação de q: $((p \cdot q) \vee (\bar{p} \cdot q))$</p> <p>TV: {1 0 1 0}</p> 	<p>10. A Negação de q: $((p \cdot \bar{q}) \vee (\bar{p} \cdot \bar{q}))$</p> <p>TV: {0 1 0 1}</p> 
<p>8. A Equivalência: $((p \cdot q) \vee (\bar{p} \cdot \bar{q}))$</p> <p>TV: {1 0 0 1}</p> 	<p>9. A Exclusão Recíproca: $((p \cdot \bar{q}) \vee (\bar{p} \cdot q))$</p> <p>TV: {0 1 1 0}</p> 

Note que cada uma das dezesseis operações, também são proposições.

O mesmo estudo pode ser feito para três proposições distintas e independentes, nesse caso

todas as possibilidades serão definidas por $2^{2^3} = 256$ em que a tautologia (ou afirmação completa) pode ser representada no diagrama abaixo. À medida que o número de proposições aumenta, a descrição de todas as possibilidades (embora possível) cresce exponencialmente, tornando tal trabalho extenso, cansativo³ e insuficiente, visto que nenhuma dedução consiste apenas em tal enumeração. Logo, de acordo com Piaget, é possível e necessário “substituir este ideal atomístico por outro ideal, que procurará o fechamento do sistema na estrutura de sua totalidade, e não mais na numeração, impossível de fato, de seus elementos” (1976, p.226).

Figura 11: Diagrama de Venn das três proposições lógicas



A solução para tal problema pode ser encontrada na discriminação das dezesseis possibilidades. Se observar com atenção a tabela verdade 3, pode-se constatar que as dezesseis possibilidades lógicas podem ser interpretadas como oito pares de inversas ou complementaridade: a Tabela 4 foi organizada para colocar em evidência tal fato, note que são complementares: a *Tautologia* (16) e a *Negação* (1); a *Conjunção* (2) e a *Incompatibilidade* (15); a *Não implicação* (3) e a *Implicação* (14); a *Não implicação recíproca* (4) e a *Implicação recíproca* (13); a *Negação conjunta* (5) e a *Disjunção* (12); a *Afirmação de p* (6) e a *Negação de p* (11); a *Afirmação de q* (7) e a *Negação de q* (10); a *Equivalência* (8) e a *Exclusão recíproca* (9).

Por conseguinte, segundo para o autor (1976) a lógica das proposições oferece uma economia ao pensamento, pois as operações interproposicionais, a partir de suas inversas, são dominadas pela existência de três conexões distintas de inversas, ou melhor, de reversibilidade: a *negação* (ou complementaridade simples); a *reciprocidade* (que constitui uma complementaridade em relação à equivalência); e o que chamaremos de *correlatividade* (permuta dos “ou” e dos “e”). A relação entre essas três espécies de conexões indicam que a reversibilidade domina toda a lógica das

3 No caso de quatro proposições todas as possibilidades seriam $2^{2^4} = 65.536$ e cinco, torna-se $2^{2^5} = 4.294.967.296$.

proposições, bem como a lógica das classes e das relações. Tais semelhanças lhe deram a ideia de se inspirar no grupo de Klein para modelar o estádio formal, concebendo o conjunto INRC.

6.2.4.2 Grupo INRC

Como foi constatado, Piaget (1976) verificou que era insuficiente utilizar somente os algoritmos da lógica das proposições na análise do pensamento formal adolescente. Assim seria necessário utilizar o grupo das quatro transformações (inversões e reciprocidades) chamado de grupo INRC.

O grupo de Klein é um grupo comutativo composto de quatro transformações (T1 a T4), tais que a composição de duas delas resulta na terceira e o conjunto formado por três, a identidade. O INRC é um caso particular do grupo de Klein, pois apresenta-se apenas nas estruturas de “conjuntos das partes”, no terreno das operações da lógica das proposições. Logo ele é composto pelos quatro tipos de operações inversas possíveis no grupo das dezesseis proposições.

Como exemplo, tomando a proposição $(p \cdot q)$ como a “afirmação completa”. Logo a Identidade (**I**), existe sempre a operação inversa $(\sim p \vee \sim q)$ ou negação (**N**); a operação recíproca (**R**) $(\sim p \cdot \sim q)$ que atua de forma equivalente à identidade e, por último, a sua correlativa (**C**). Composto, dessa forma, os três casos possíveis de inversa em relação à identidade”

De uma forma mais geral, Piaget (1976) define tais operações da seguinte maneira.

Definição 1: A **Identidade** é qualquer transformação que não altera a proposição inicial (elemento neutro), ou seja, mantém seu “estado inicial”. Sendo assim, as demais transformações são algum tipo de negação dessa proposição.

Definição 2: A **Negação** (ou Inversa): que é a negação dos valores de p e q e dos conectivos lógico da identidade.

Definição 3: A **Reciprocidade** é a negação dos valores de p e q , mas não de seu conectivo.

Definição 4: A **Correlativa** é a negação do conectivo, portanto conserva-se os valores de p e q .

Se o conjunto das quatro operações INRC é isomorfo ao grupo de Klein, tais transformações constituem um grupo comutativo relativamente à sua composição.

Tabela 6: Transformações no INRC

	I	N	R	C
I	I	N	R	C
N	N	I	C	R
R	R	C	N	I
C	C	R	N	I

Através da tabela é possível garantir que:

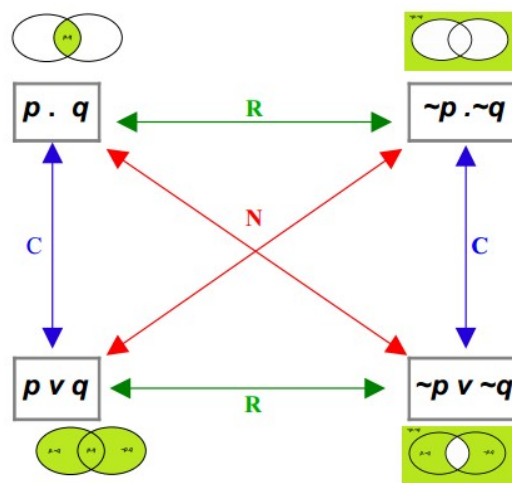
- (1) A composição de dois elementos do conjunto é ainda um elemento do conjunto.
- (2) A composição é associativa;
- (3) Cada elemento tem um inverso;
- (4) Existe um elemento neutro (que é I).
- (5) A composição é comutativa

Tomando como exemplo a operação de conjunção ($p \cdot q$), Então a transformação da identidade que não altera a proposição inicial será $I (p \cdot q)$. Sendo assim, a transformação da negação inverte o conectivo e seus valores lógicos, então $N (p \cdot q) = (\sim p \vee \sim q)$. Já a transformação recíproca inverte apenas os valores lógicos das proposições, dessa maneira,

$R (p \cdot q) = (\sim p \cdot \sim q)$. Por último, a transformação correlativa inverte apenas o conectivo lógico, $C (p \cdot q) = (p \vee q)$.

A figura 12 abaixo é uma representação das diferentes transformações do grupo INRC aplicadas à operação da conjunção.

Figura 12: INRC da operação de conjunção



Através da relação entre esses quaternos, pode-se observar uma das características mais interessantes do INRC, Ao tomar uma operação qualquer e aplicar-lhe as outras duas, tem-se como resultado a terceira transformação:

$$RC = N \text{ ou } CR = N;$$

$$RN = C \text{ ou } NR = C;$$

$$NC = R \text{ ou } CN = R.$$

Além disso, se aplicar as três transformações temos a identidade:

$$NRC = I$$

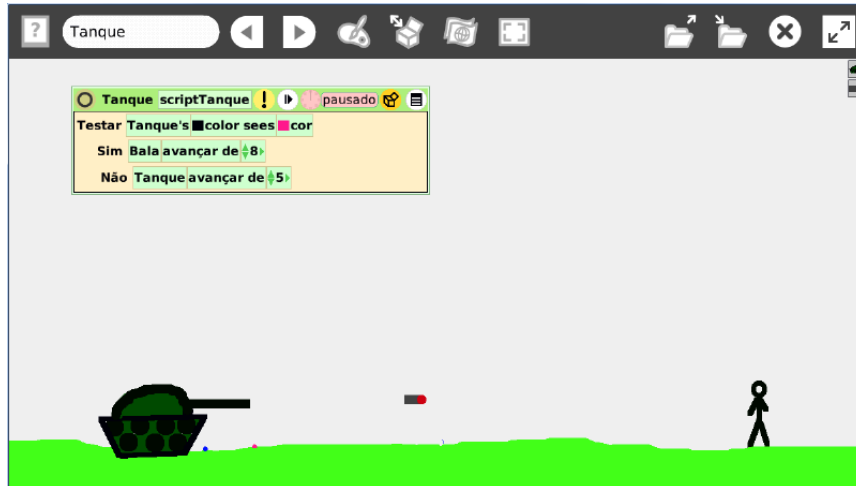
7. EXPERIMENTO LÓGICO NO *ETOYS*

Como já foi dito, foi desenvolvida uma oficina de *Squeak Etoys* com uma turma do nono ano do ensino fundamental do Colégio de Aplicação da Universidade Federal do Rio Grande do Sul na disciplina de matemática. De início, foi proposta a construção de polígonos ao estilo LOGO para realizar uma introdução à programação no *Etoys*. Pretendia-se com essas atividades discutir, com os estudantes, a importância da: sequência de comandos no *script*, utilização do comando repita, criação de um *script* dentro de outro e introduzir o uso de variáveis. Do ponto de vista da matemática, o objetivo era discutir as questões sobre os ângulos e suas relações com os polígonos.

Nessa turma alguns estudantes já conheciam o *Etoys*, pois dois anos antes tiveram uma oficina. Alguns deles já estavam habituados a utilizá-lo e, no final de um dos encontros, para surpresa do pesquisador um estudante pediu ajuda para resolver um problema de um projeto que havia criado de forma independente. Tal projeto era uma simulação em que sua intenção era fazer um tanque avançar na direção de uma pessoa e, ao chegar num determinado ponto, disparar um projétil contra ela.

É importante ressaltar que, embora a análise realizada seja voltada para a lógica utilizada na programação, a violência que tal cena representa é algo que preocupa o autor desta pesquisa, pois é comum o interesse de algumas crianças em reproduzir esse tipo de jogo, fato que faz parte de seu cotidiano como jogador de game. Ademais, o que mais preocupa, é que tais jogos exploram a banalização da violência e são comercializados em larga escala, logo representam um segmento lucrativo do mercado de games. Segundo Hannah Arendt, "muito da presente glorificação da violência é causada pela severa frustração da faculdade da ação no mundo moderno" (1994, 60). Dessa maneira, é importante que a escola aja e reaja a isso, instigando, periodicamente, o debate sobre a relação entre *violência x games*. Mas não com o objetivo de demonizar os jogos e, sim, para promover uma reflexão por parte dos estudantes acerca dessa cultura de violência. Dito isso, será apresentada a análise da animação proposta pelo estudante, que fazia uso do comando *Teste* na sua programação, como mostra a figura 13 abaixo.

Figura 13: Projeto do tanque disparando um projétil



Ao executar a simulação constatou-se que o tanque avançava na direção da pessoa e parava quando seu canhão encostava-se à bala (que “flutuava” parada e visível desde o início da simulação), só então o projétil começava o seu movimento na direção da pessoa. A dificuldade encontrada pelo estudante era fazer a bala ficar invisível até o momento do disparo. Como o tempo do encontro já havia acabado o pesquisador combinou de analisar o projeto em casa e eles conversariam no encontro seguinte.

Desde o início a complexidade da programação do estudante ultrapassava todos os projetos desenvolvidos na experiência-piloto:

- 1) Para os dois objetos se deslocarem horizontalmente, o estudante precisou modificar a direção da frente do objeto.
- 2) Uso da Negação: o tanque avança quando a condição não é verdadeira.
- 3) O *script* de um objeto dispara ações de outros objetos. Pode-se perceber no *script* que o objeto Bala é disparado quando a condição do *Teste* é confirmada. Sendo assim esse *script* coordena duas ações distintas de dois objetos distintos.
- 4) O uso de uma marcação para disparar o movimento da bala. Num primeiro momento se pensou que a bala era disparada quando o canhão (cor preta) tocasse a parte rosa da bala. Nesse caso, foi previsto o seguinte problema de programação: quando ele fizer a bala ficar invisível, a cor preta não verá a cor rosa, portanto a ação da bala não será desencadeada. No entanto, ao analisar o *script* posteriormente, a cor rosa do *Teste* não está na bala, mas num ponto próximo do chão. Observou-se, ainda que houvesse outro ponto azul, seria outra marcação feita pelo estudante? Para quê?

Dessa maneira, foi realizada uma entrevista com esse estudante (chamaremos de F) num outro dia, pois seria importante para esta pesquisa ouvir sua explicação de como ele havia concebido tal projeto.

Entrevistador: Eu quero pedir a você que me explique como funciona essa animação. Você pode rodá-la e depois me explicar como ela funciona?

Estudante F: Tá, primeiro eu pensei em fazer um projeto em que o objeto (a bala), após um seguimento de ações que ele poderia fazer, acertasse um alvo (indicando um boneco na tela do computador) e nisso ela (a bala) se escondesse. Eu usei o *Testar* para fazer tudo acontecer. No começo eu coloquei “a cor preta” (do Tanque) quando encostar nessa corzinha rosa (uma pequena marcação no desenho), ele começasse a ativar a bala para ela avançar até acertar o boneco.

Entrevistador: Tu pode executar o que tu fizeste até agora?

O tanque avança e – ao encostar-se à marca rosa – a bala começa a se movimentar horizontalmente. No entanto, tal movimento ocorre antes dele chegar até ela. O Estudante F começa a fazer modificações no objeto.

Entrevistador: O que você está mudando ali? (no *script*).

Estudante F: Eu coloquei a velocidade maior, para que ela vá mais rápido.

Entrevistador: Você ajustou a velocidade de quem para ir mais rápido?

Estudante F: Nesse caso, da bala. Para ficar até mais realístico.

Entrevistador: Mas no caso da animação que você fez, a bala saiu voando do... (Nada, completa o adolescente). Isso, do espaço, né?

Estudante F: Sim, daí no futuro, eu queria que a bola estivesse invisível, vamos dizer, escondida. E quando a cor preta do tanque encostasse na corzinha rosa que tem no chão, a bala aparecesse... para parecer mais realístico até.

Entrevistador: Uma dúvida que eu tenho: Por que você utilizou esse pontinho rosa? Por que você sentiu a necessidade de utilizar esse pontinho rosa?

Estudante F: Por que eu achei... Pelo meu nível de conhecimento, eu ainda não sei como (*fazer*) o tanque disparar do nada, daí o jeito mais fácil que achei era quando ele encostasse numa flor, a bala aparecia.

Entrevistador: Se o objetivo é fazer a bala andar, porque você não escolheu a cor da bala?

Estudante F: Não entendi.

Entrevistador: Por que você não escolheu uma dessas cores da bala em vez de fazer esse pontinho no chão?

Estudante F: Por que eu achei que um pontinho no chão ficaria bem discreto, seria um chão verde, vamos dizer, e ficaria mais discreto, por exemplo, seria uma florzinha. E aquele pontinho ali vai fazer toda parte de todo projeto. Ficaria mais escondido.

Entrevistador: Uma coisa que eu observei é que além do ponto rosa, tem um ponto azul, aquele ponto foi “sem querer” ou tem um propósito para ele?

Estudante F: Antigamente era de propósito, mesmo. Eu queria que quando (o tanque) encostasse no ponto azul, disparasse a bala e quando encostasse no ponto rosa, ele parasse. Mas eu vi que eu poderia fazer tudo isso num mesmo ponto.

Entrevistador: E quem para?

Estudante F: O tanque, daí aciona a bala.

Entrevistador: E tu já conseguiste fazer isso?

Estudante F: No atual? Sim.

Entrevistador: Tu podes-me dizer como funciona o *Teste*? (o estudante não entende a pergunta). Como é que está ali a programação?

Estudante F: Aqui tá como, vamos *testar* esse *script*. Por exemplo, quando “a cor tal”, encostar “na cor

tal” é para *testar* isso daqui (com o *mouse* circula os comandos do sim e do não).

Entrevistador: E o que está testando nesse caso?

Estudante F: Tá testando quando a bala encostar na cor rosa, ela avança 20 e o tanque não avance 8. Daí o tanque pararia e a bala dispararia.

Entrevistador: Tu podes repetir? (a explicação)

Estudante F: Oi? (não entendeu a pergunta)

Entrevistador: Tu podes repetir a tua explicação, quando o preto vê o rosa... a bala anda vinte. **Estudante**

F: Sim.

Entrevistador: E se isso não acontecer?

Estudante F: O propósito não estaria certo, não corresponderia com o que eu queria fazer.

Entrevistador: Pode executar o *script*. (O *Script* apresenta o efeito desejado)

Diante desta entrevista, o pesquisador ficou com algumas dúvidas: Como esse adolescente compreende e explica a lógica do comando *Teste*? Essa animação foi criada durante a aula ou é a continuação de um projeto que ele desenvolveu anteriormente? Pelo nível de consciência que tal programação exige, se supõe que seja a segunda opção. Além disso, durante a aula tal estudante pediu auxílio de seu professor para resolver o problema. Segundo o relato do professor, o estudante primeiramente apresentou o seguinte *script*.

Figura 14: Script do projeto do tanque



O estudante não estava identificando que o carro se movimentava independente da condição do *Teste* ser confirmada ou não, pois havia o comando avançar fora do *Teste*. Somente quando o professor interveio – excluindo o comando do *script*– ele foi compreender o que estava acontecendo. Esse relato deixou o investigador curioso a respeito da compreensão do estudante F em relação à lógica inerente ao comando *Teste*, pois tal fato poderia representar uma incompreensão, mesmo que o estudante tenha sucesso ocasional na sua utilização. Talvez, uma alternativa seria apresentar uma situação parecida, na qual ele precisasse realizar tal programação.

Dessa maneira, foram construídos três desafios que envolviam situações que, para serem resolvidos, envolviam o uso e a compreensão: 1) da lógica do comando *Teste*; 2) da operação lógica da disjunção e 3) da operação de conjunção. Assim, a partir das entrevistas, os pesquisadores teriam indícios de como os estudantes se apropriariam de tais operações lógicas.

É relevante ressaltar que a construção desses desafios ocorreu entre um encontro e outro, portanto não era uma atividade previamente planejada. Logo foi necessário realizar alguns ajustes

nesse instrumento de pesquisa ao longo das entrevistas, ou seja, durante a coleta de dados. Além disso, foi só durante a análise dos mesmos que o pesquisador compreendeu que seu experimento versava sobre a operação lógica da implicação e sobre o desenvolvimento do raciocínio condicional. No momento das entrevistas, pensava-se que os três desafios eram independentes um do outro, não observando que tratava-se da utilização da implicação em situações cada vez mais complexas. Ademais, a necessidade de compreender profundamente o desenvolvimento da lógica segundo a teoria Piagetiana adveio da análise dos dados. Portanto, tudo que envolve o INRC nesta tese foi compreendido e realizado após a coleta de dados.

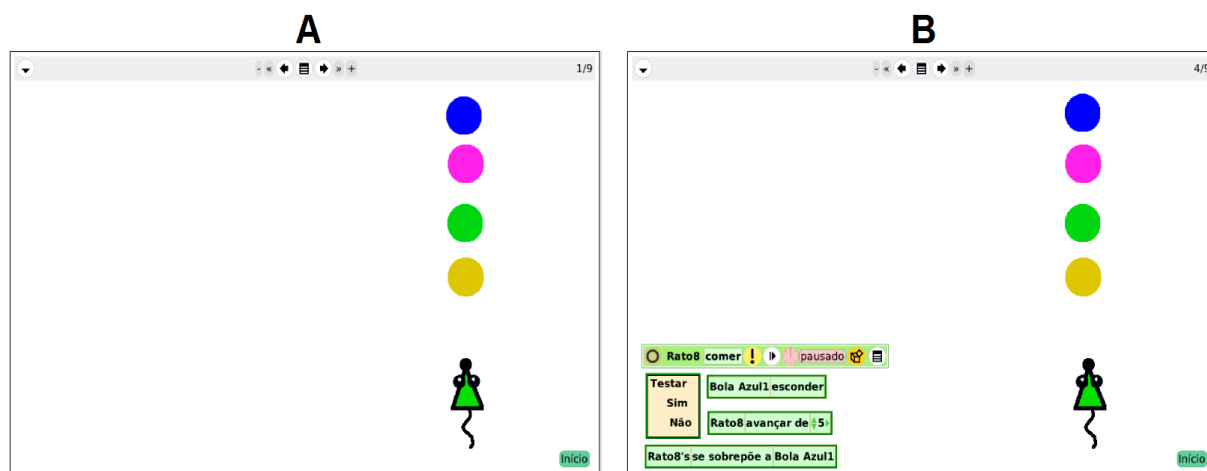
7.1 DESCRREVENDO O EXPERIMENTO

DESAFIO 1A: No primeiro desafio era apresentado ao aluno a tela abaixo (figura 15A) e lhe era solicitado que criasse um programa para fazer o rato “caminhar” e “comer a comida azul” (a bola azul) ao encontrá-la.

DESAFIO 1B: Quando o entrevistador identificava que o estudante (iniciante em programação) não conhecia os comandos necessários para resolver tal desafio, então lhe apresentava uma nova tela (figura 15B) com alguns comandos sugeridos para realização do desafio.

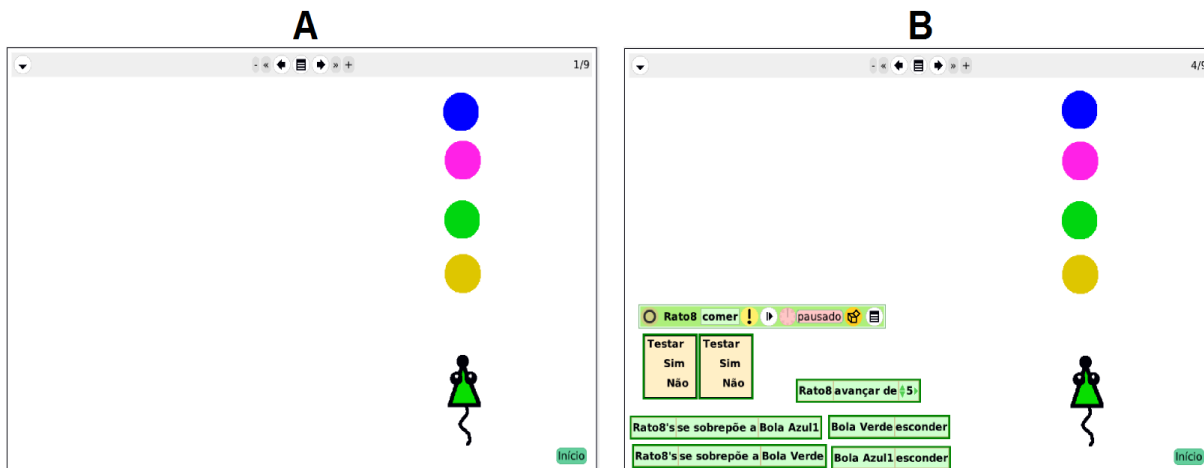
No desafio 1B, durante as entrevistas, uma das intervenções realizadas era pedir para o estudante ler em voz alta os comandos sugeridos. Além disso, com as crianças mais novas, o comando “se sobrepõe” foi substituído pelo comando de reconhecimento de cor, pois se avaliou que este era mais intuitivo para elas.

Figura 15: Experimento de pesquisa: Desafio 1 A e B



DESAFIO 2A: Quando o estudante tinha sucesso ao realizar o desafio 1, devia programar o rato para comer a comida de uma cor ou outra. (bolas de duas cores diferentes). Tendo sucesso, caso achasse relevante, o entrevistador poderia solicitar à criança que programasse o rato para comer um número de comidas maior. **DESAFIO 2B:** Caso ele não conseguisse encontrar os comandos necessários, novamente apresentava-lhe uma tela com comandos sugeridos.

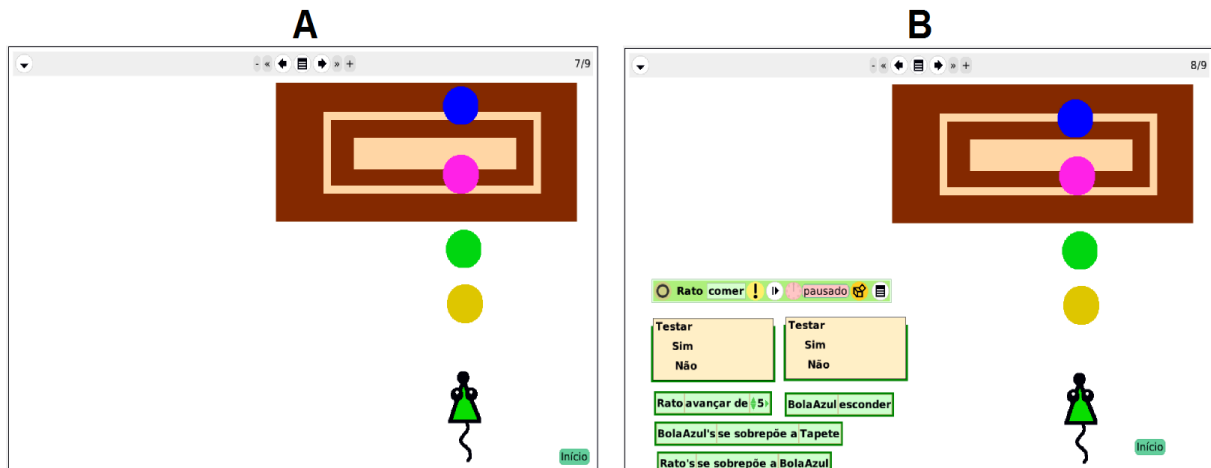
Figura 16: Experimento de pesquisa: Desafio 2 A e B



DESAFIO 3A: Nesse desafio os estudantes deviam programar o rato para ele comer a bola azul só quando ela estivesse sobre o tapete, ou seja, quando ele encontrasse a comida fora do tapete, não deveria comê-la. Tendo sucesso nesse desafio, o entrevistador poderia pedir para estudante programar o rato comer dois (ou mais) tipos de comida só quando estiver em cima do tapete.

Caso o estudante apresentasse alguma dificuldade em utilizar os comandos necessários, apresentava-se o Desafio 3B.

Figura 17: Experimento de pesquisa: Desafio 3 A e B



Nas duas primeiras entrevistas, o experimento era apresentado com comidas de apenas duas cores. No entanto, caso o estudante utilizasse como condicional o reconhecimento de cores, tal desafio propunha a utilização da operação da disjunção desde o início, aumentando o grau de dificuldade do desafio e dificultando a observação isolada da compreensão dos componentes do comando *Teste*. Logo, para resolver tal dificuldade, optou-se por criar comidas de cores diferentes.

Além desse, outro inconveniente foi “bola azul apagar” nos desafios do tipo B, pois cada vez que o “objeto comida” era apagado, não havia como recuperá-la, portanto era necessário criar um novo cada vez que isso ocorria. Como os estudantes estavam no início da apropriação deste *software* isso gerou grande dificuldade para eles e, assim, sua atenção e foco eram desviados, deixando a atividade mais cansativa. Como solução, foi sugerido o comando “bola azul esconder” e a criação do botão “início”, que colocava todos os objetos na posição inicial, caso fosse necessário recomeçar.

7.2 O INRC NOS DESAFIOS DO EXPERIMENTO DE PESQUISA

Com o intuito de compreender as ações e as possíveis dificuldades dos estudantes com o uso da lógica ao longo da resolução dos desafios, torna-se importante analisar cada um deles sob a ótica da teoria piagetiana. Dito de outra maneira, cada resolução deve ser analisada a partir da lógica das proposições presentes no grupo INRC.

7.2.1 Desafio 1: A implicação simples

Nesse primeiro desafio o estudante deve programar o rato para “caminhar” e “comer a comida azul” (a bola azul) ao encontrá-la. O comando que faz o rato “caminhar” é relevante para encontrar a comida, mas é irrelevante para a solução do problema. O que interessa é o uso do comando *Teste* para, através da condicional, identificar se o objeto rato está sobre o objeto comida (ou bola azul), se sim: ela deve ficar invisível. Logo, através da lógica proposicional, pode-se representar tal algoritmo da seguinte maneira. Tomando:

p : rato sobrepõe à bola Azul e q : Bola azul está invisível, temos:

$$p \rightarrow q$$

De acordo com o grupo INRC, chamaremos de **Identidade** qualquer transformação que não altera a proposição inicial $p \rightarrow q = (p \cdot q) \vee (\sim p \cdot q) \vee (\sim p \cdot \sim q)$. Logo tal implicação significa que o algoritmo é verdadeiro para toda situação em que: *o rato se sobrepõe à bola e a mesma está invisível* OU *o rato não a sobrepõe e a bola está invisível* OU *o rato não está sobre ela e a bola não está invisível*. Logo, a conclusão é que, se a proposição *o rato se sobrepõe à bola* for verdadeira é condição suficiente para que a comida fique invisível.

Segundo a definição 2, sua operação inversa completa, a **Negação** de $(p \rightarrow q)$ é $N(p \rightarrow q) = p \cdot \sim q$. Expressando de outra maneira, é quando a criança criar um *script* em que for verdadeira a situação em *o rato estar sobre a bola e esta permanecer visível*. Ainda é importante ressaltar que a operação direta com a sua inversa se anula:

$$(p \rightarrow q) \cdot (p \cdot \sim q) = 0.$$

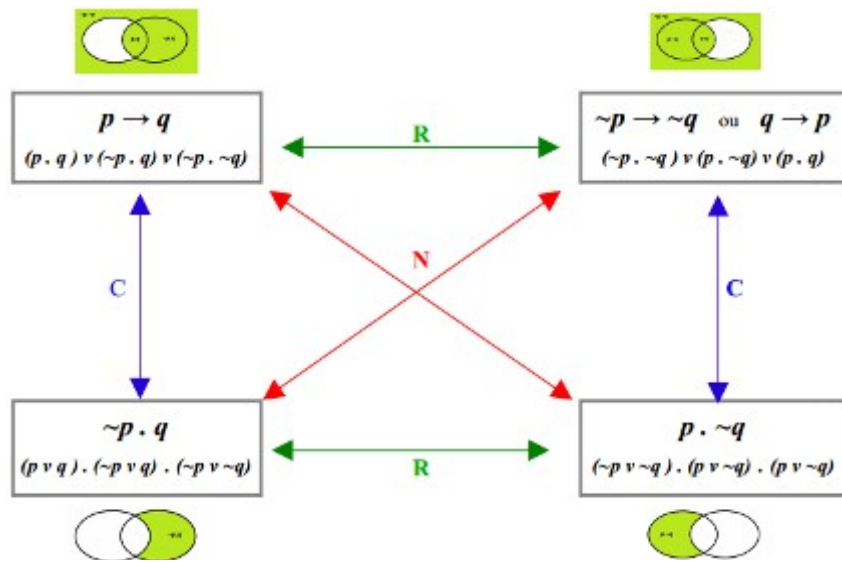
Obtém-se a operação **Recíproca** da identidade aplicando a definição 3: $R(p \rightarrow q) = (\sim p \rightarrow \sim q) = (\sim p \cdot \sim q) \vee (p \cdot \sim q) \vee (p \cdot q) = (q \rightarrow p)$ Ou melhor, criar um *script* que seja verdadeiro quando: *o rato não se sobrepõe à bola e ela não está invisível* OU *o rato está sobre a bola e a ela não está invisível* OU *o rato está sobre a bola e ela está invisível*. Isso significa afirmar que *se o rato não se sobrepõe à bola é condição suficiente para ela não estar invisível*. Além disso, é fundamental observar que a operação direta (Identidade) com sua recíproca dá uma equivalência.

$$(p \rightarrow p) \cdot (q \rightarrow q) = (p \cdot q) \leftrightarrow (p \cdot q)$$

Por último, a operação **Correlativa** será $C(p \rightarrow q) = (p \vee q) \cdot (\sim p \vee q) \cdot (\sim p \vee \sim q) = (\sim p \cdot q)$. Traduzindo para a linguagem do desafio, é quando a única situação verdadeira for quando *o rato não está sobre a bola e ela está*

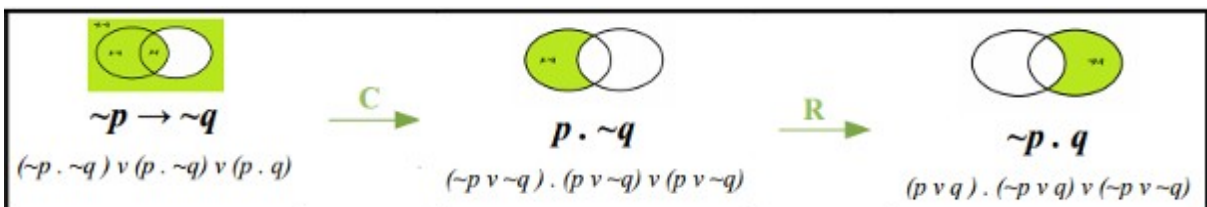
invisível. Veja no diagrama (figura 18) abaixo o grupo formado pelas operações.

Figura 18: INRC da solução do Desafio 1



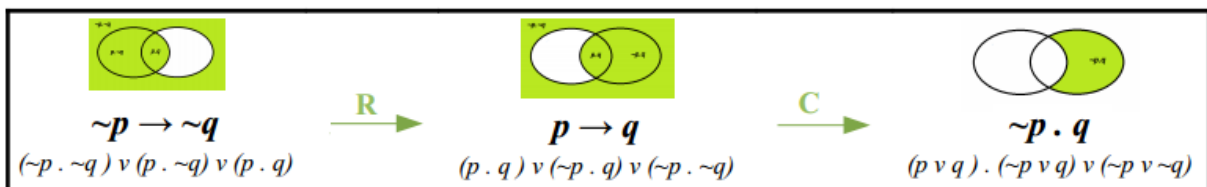
O diagrama acima expressa de forma resumida todas as transformações possíveis entre as quatro operações inversas, independente de qual for a identidade. Por exemplo: se assumirmos como ponto de partida a implicação $(\sim p \rightarrow \sim q)$ e aplicarmos sua correlativa seguida da recíproca têm-se sua negação. (CR=N, Correlativa de $(p \rightarrow q)$)

Figura 19: Aplicando as transformações correlativa e recíprocas



Agora, podemos constatar a comutatividade do conjunto se aplicando à identidade a recíproca seguida pela correlativa e o resultado será o mesmo (RC=N, Correlativa de $(p \rightarrow q)$).

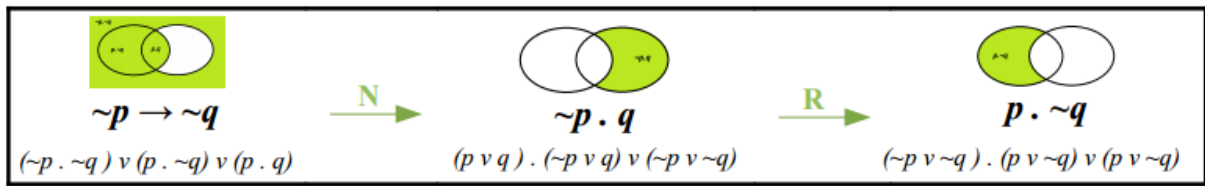
Figura 20: Aplicando as transformações: comutativa de R e C



Empregando a negação seguida da recíproca obtém-se a correlativa (NR=C, Negação de $(p$

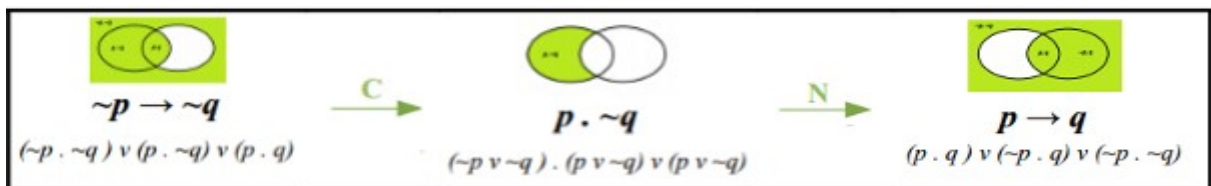
$\rightarrow q)$.

Figura 21: Aplicando as transformações: comutativa de N e R



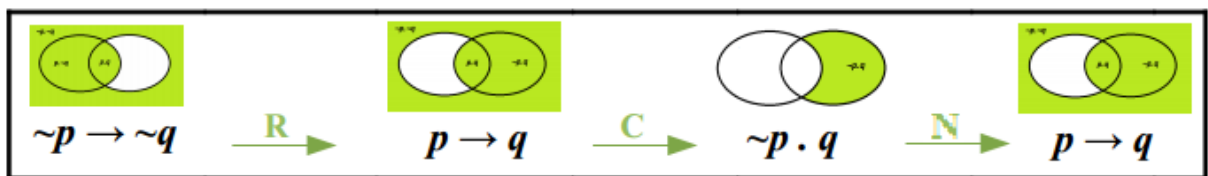
Analogamente, empregando a correlativa seguida da negação obtém-se a recíproca ($CN=R$, identidade de $(p \rightarrow q)$).

Figura 22: Aplicando as transformações C e N



Por último, é importante constatar que a aplicação de três transformações seguidas gera a quarta, assim, tem-se $NRC = I$

Figura 23: Aplicando três transformações do INRC seguidas



Assim avaliamos que grupo INRC, por ser um sistema de operações inversas, constitui-se numa ferramenta interessante para analisar a aprendizagem de programação.

7.2.2 Desafio 2: Implicação com disjunção

Tendo superado o primeiro desafio, o estudante deveria estender sua programação para que o rato “coma a comida azul ou verde”. Assim, através da lógica proposicional, pode-se representar tal algoritmo da seguinte maneira. Tomando:

p_1 – rato sobrepõe à bola azul e q_1 : Bola azul está invisível;

p_2 – rato sobrepõe à bola verde e q_2 : Bola verde está invisível,

Assim temos:

$$(p_1 \rightarrow q_1) \vee (p_2 \rightarrow q_2)$$

Assim, pelo INRC a **Identidade** será a expressão abaixo.

$$(p_1 \rightarrow q_1) \vee (p_2 \rightarrow q_2) = (p_1 \cdot q_1) \vee (\sim p_1 \cdot q_1) \vee (\sim p_1 \cdot \sim q_1) \vee (p_2 \cdot q_2) \vee (\sim p_2 \cdot q_2) \vee (\sim p_2 \cdot \sim q_2)$$

Logo tal implicação significa que o algoritmo é verdadeiro para toda situação em que: (o rato se sobrepõe à bola azul e ela fica invisível OU o rato não sobrepõe à bola azul e ela está invisível OU o rato não a sobrepõe e a bola azul não está invisível) OU (o rato se sobrepõe à bola verde e ela fica invisível OU o rato não sobrepõe à bola verde e ela está invisível OU o rato não à sobrepõe e a bola verde não está invisível). Então, a conclusão é que se a condição rato se sobrepõe à bola azul ou verde for verdadeira é condição suficiente para que a respectiva comida fique invisível.

Aplicando a definição 2, com o objetivo de se obter a **Negação** temos:

$$N((p_1 \rightarrow q_1) \vee (p_2 \rightarrow q_2)) = ((\sim p_1 \vee \sim q_1) \cdot (p_1 \vee \sim q_1) \cdot (p_1 \vee q_1)) \cdot ((\sim p_2 \vee \sim q_2) \cdot (p_2 \vee \sim q_2) \cdot (p_2 \vee q_2))$$

$$N((p_1 \rightarrow q_1) \vee (p_2 \rightarrow q_2)) = (p_1 \cdot (\sim q_1)) \cdot (p_2 \cdot (\sim q_2))$$

Pode-se concluir que tal proposição não possui inversa, visto que a intersecção dos conjuntos $p_1 \cdot \sim q_1$ e $p_2 \cdot \sim q_2$ é vazia, como mostra o diagrama de Venn (figura 24) abaixo.

Aplicando a definição 3, com o objetivo de se obter a **Recíproca** temos:

$$R(p \rightarrow q) = (\sim p \cdot \sim q) \vee (p \cdot \sim q) \vee (p \cdot q) = q \rightarrow p$$

$$R((p_1 \rightarrow q_1) \vee (p_2 \rightarrow q_2)) = (\sim p_1 \rightarrow \sim q_1) \vee (\sim p_2 \rightarrow \sim q_2)$$

$$R((p_1 \rightarrow q_1) \vee (p_2 \rightarrow q_2)) = (\sim p_1 \cdot \sim q_1) \vee (p_1 \cdot \sim q_1) \vee (p_1 \cdot q_1) \vee (\sim p_2 \cdot \sim q_2) \vee (p_2 \cdot \sim q_2) \vee (p_2 \cdot q_2)$$

$$R((p_1 \rightarrow q_1) \vee (p_2 \rightarrow q_2)) = (q_1 \rightarrow p_1) \vee (q_2 \rightarrow p_2)$$

Por conseguinte isso significa criar um *script* que seja verdadeiro quando: (o rato não se sobrepõe à bola azul e ela não está invisível OU o rato está sobre a mesma e a bola azul não está invisível OU o rato está sobre a bola azul e ela está invisível) OU (o rato não se sobrepõe à bola verde e ela não está invisível OU o rato está sobre a mesma e a bola verde não está invisível OU o rato está sobre a bola verde e ela está invisível). Isso significa afirmar que se o rato não se sobrepõe à bola é condição suficiente para ela não estar invisível. Então, a conclusão é que se a condição rato

não se sobrepõe à bola azul ou verde for verdadeira é condição suficiente para que a respectiva comida não fique invisível.

Aplicando a definição 4, com o objetivo de se obter a **Correlativa** temos:

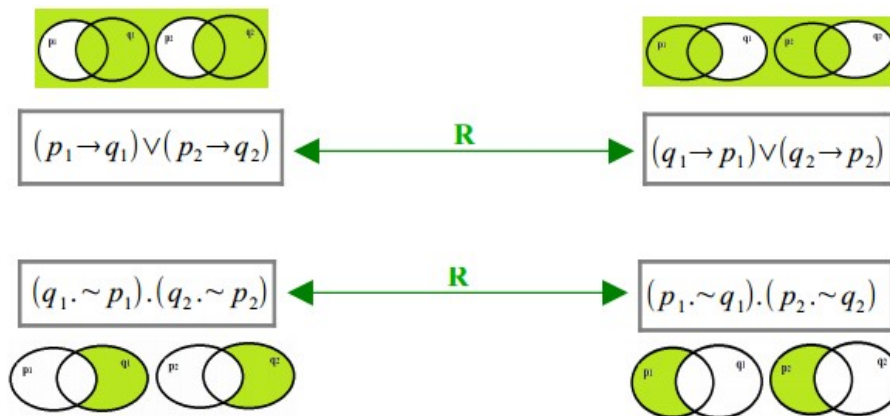
$$C((p_1 \rightarrow q_1) \vee (p_2 \rightarrow q_2)) = ((p_1 \vee q_1) \cdot (\sim p_1 \vee q_1) \cdot (\sim p_1 \vee \sim q_1)) \cdot ((p_2 \vee q_2) \cdot (\sim p_2 \vee q_2) \cdot (\sim p_2 \vee \sim q_2))$$

$$C((p_1 \rightarrow q_1) \vee (p_2 \rightarrow q_2)) = (q_1 \cdot (\sim p_1)) \cdot (q_2 \cdot (\sim p_2))$$

Analogamente à negação, tal proposição não possui correlativa, visto que a intersecção dos conjuntos $q_1 \cdot \sim p_1$ e $q_2 \cdot \sim p_2$ também é vazia, como mostra o diagrama de Venn (figura 24) abaixo.

No diagrama abaixo ressalta-se que os conjuntos formados pela aplicação da correlação e da negação, são entre si, operações recíprocas.

Figura 24: A solução do desafio 2 não comporta o INRC



Pode-se fazer a análise proposicional para os casos em que envolva os quatro tipos de comidas, tal que:

p_1 – rato sobrepõe à bola verde e q_1 : Bola azul está invisível;

p_2 – rato sobrepõe à bola verde e q_2 : Bola verde está invisível;

p_3 – rato sobrepõe à bola rosa e q_3 : Bola rosa está invisível;

p_4 – rato sobrepõe à bola amarela e q_4 : Bola amarela está invisível; temos:

$$(p_1 \rightarrow q_1) \vee (p_2 \rightarrow q_2) \vee (p_3 \rightarrow q_3) \vee (p_4 \rightarrow q_4)$$

Espera-se que o grau de complexidade aumente a medida que o número de variáveis cresça.

7.2.3 Desafio 3: Implicação com conjunção

No terceiro desafio o estudante deve programar o rato para que ele coma a bola azul só quando ela estiver sobre o tapete, ou seja: se ele encontrá-la fora do tapete, não deve comê-la. O que interessa nesse desafio é o uso do comando *Teste* para, através da condicional, identificar se a comida azul está sobre o tapete e se o objeto rato está sobre ela, se sim para as duas condições: a comida azul deve ficar invisível. Dessa maneira, pode-se representar tal algoritmo da seguinte maneira através da lógica proposicional. Tomando:

p : Bola azul sobrepõe ao tapete, q : rato sobrepõe à bola azul e t Bola azul esconder, temos:

$$(p \rightarrow q) \rightarrow t$$

Como se pode constatar, que o terceiro desafio envolve três proposições bem definidas e independentes, tal fato indica que o número de operações lógicas possíveis é igual a 256, portanto envolve relações ternárias mais complexas que as atuais. Sabe-se que Piaget escreveu um livro no qual descreve e analisa tais sistemas lógicos, infelizmente o pesquisador não teve acesso a tal obra. Por outro lado, o próprio Piaget (1976) realiza um estudo binário das relações ternárias, trabalhando com um sistema de três proposições unidas por duas relações. Logo se utilizou tal estratégia para analisar o terceiro desafio. Talvez essa não seja a melhor opção, mas tal estratégia forneceu bons subsídios para a investigação das ações dos adolescentes.

Tomando $(p \rightarrow q) \rightarrow t$ como ponto de partida, tal proposição significa afirmar que

$$(p \rightarrow q) \rightarrow t = ((p \rightarrow q).t) \vee (\sim(p \rightarrow q).t) \vee (\sim(p \rightarrow q).\sim t)$$

$$(p \rightarrow q) \rightarrow t = (((p.q) \vee (\sim p.q) \vee (\sim p.\sim q)).t) \vee ((p.\sim q).t) \vee ((p.\sim q).\sim t)$$

$$(p \rightarrow q) \rightarrow t = ((p.q.t) \vee (\sim p.q.t) \vee (\sim p.\sim q.t)) \vee (p.\sim q.t) \vee (p.\sim q.\sim t)$$

Logo tal implicação significa que o algoritmo é verdadeiro para toda situação em que: (“a Bola azul está sobre o tapete e o rato se sobrepõe a ela e A bola azul fica invisível” OU “a Bola azul não está sobre o tapete e o rato se sobrepõe a ela e A bola azul fica invisível” OU “a Bola azul não está sobre o tapete e o rato não se sobrepõe a ela e A bola azul fica invisível”) OU (“a Bola azul está sobre o tapete e o rato não se sobrepõe a ela e A bola azul não fica invisível” OU “a Bola azul está sobre o tapete e o rato não se a sobrepõe e A bola azul não fica invisível”). Então, a conclusão é que se a condição bola azul se sobrepõe ao tapete e o rato se sobrepõe à bola forem ambas verdadeiras é condição suficiente para que a comida fique invisível.

Sendo assim, a **Negação** de $(p \rightarrow q) \rightarrow t$ será:

$$N((p \rightarrow q) \rightarrow t) = (p \rightarrow q) \cdot \sim t$$

$$N((p \rightarrow q) \rightarrow t) = ((p \cdot q) \vee (\sim p \cdot q) \vee (\sim p \cdot \sim q)) \cdot \sim t$$

$$N((p \rightarrow q) \rightarrow t) = (p \cdot q \cdot \sim t) \vee (\sim p \cdot q \cdot \sim t) \vee (\sim p \cdot \sim q \cdot \sim t)$$

Traduzindo para a linguagem do desafio, o *script* é verdadeiro para toda situação em que: (“a Bola azul está sobre o tapete e o rato se sobrepõe a ela e A Bola azul não fica invisível”) OU (“a Bola azul não está sobre o tapete e o rato se sobrepõe a ela e A Bola azul não fica invisível”) OU (“a Bola azul não está sobre o tapete e o rato não se sobrepõe a ela e A bola azul não fica invisível”). Logo a negação representa a inversa em que a bola azul não fica invisível quando as condições Bola azul sobrepõe ao tapete e rato sobrepõe à Bola são verdadeiras.

Já a **Recíproca** de $(p \rightarrow q) \rightarrow t$ será definida por:

$$R((p \rightarrow q) \rightarrow t) = (\sim(p \rightarrow q) \cdot \sim t) \vee ((p \rightarrow q) \cdot \sim t) \vee ((p \rightarrow q) \cdot t) = t \rightarrow (p \rightarrow q)$$

$$= ((p \cdot \sim q) \cdot \sim t) \vee (((p \cdot q) \vee (\sim p \cdot q) \vee (\sim p \cdot \sim q)) \cdot \sim t) \vee (((p \cdot q) \vee (\sim p \cdot q) \vee (\sim p \cdot \sim q)) \cdot t)$$

$$= (p \cdot \sim q \cdot \sim t) \vee ((p \cdot q \cdot \sim t) \vee (\sim p \cdot q \cdot \sim t) \vee (\sim p \cdot \sim q \cdot \sim t)) \vee ((p \cdot q \cdot t) \vee (\sim p \cdot q \cdot t) \vee (\sim p \cdot \sim q \cdot t))$$

Por conseguinte tal implicação significa que o comando *Teste* é verdadeiro para toda situação em que: (“a Bola azul está sobre o tapete e o rato não se sobrepõe a ela e A bola azul não fica invisível”) OU (“a Bola azul está sobre o tapete e o rato se sobrepõe a ela e A bola azul não fica invisível”) OU (“a Bola azul não está sobre o tapete e o rato se sobrepõe a ela e A bola azul não fica invisível”) OU (“a Bola azul não está sobre o tapete e o rato não se sobrepõe a ela e A bola azul não fica invisível”) OU (“a Bola azul está sobre o tapete e o rato se a sobrepõe e A bola azul fica invisível”) OU (“a Bola azul não está sobre o tapete e o rato se a sobrepõe e A bola azul fica invisível”). Então a conclusão é que se a condição bola azul não se sobrepõe ao tapete e o rato não se sobrepõe à bola forem ambas verdadeiras é condição suficiente para que a comida não fique invisível. Ou ainda, que se a comida azul não ficar invisível é verdadeira, então ela não está sobre o tapete e o rato não se sobrepõe a ela.

Por último, sua operação **Correlativa** é definida por:

$$C((p \rightarrow q) \rightarrow t) = ((p \rightarrow q) \vee t) \cdot (\sim(p \rightarrow q) \vee t) \cdot (\sim(p \rightarrow q) \vee \sim t)$$

$$C((p \rightarrow q) \rightarrow t) = (((p \cdot q) \vee (\sim p \cdot q) \vee (\sim p \cdot \sim q)) \vee t) \cdot ((p \cdot \sim q) \vee t) \cdot ((p \cdot \sim q) \vee \sim t)$$

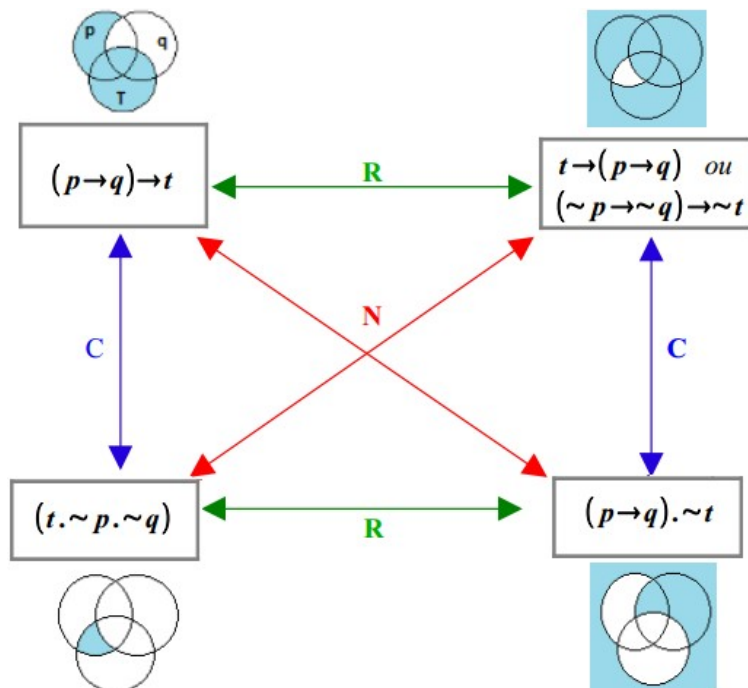
$$C((p \rightarrow q) \rightarrow t) = ((p \cdot q \vee t) \vee (\sim p \cdot q \vee t) \vee (\sim p \cdot \sim q \vee t)) \cdot ((p \cdot \sim q) \vee t) \cdot ((p \cdot \sim q) \vee \sim t)$$

$$C((p \rightarrow q) \rightarrow t) = (t \cdot \sim(p \rightarrow q))$$

$$C((p \rightarrow q) \rightarrow t) = (t \cdot \sim p \cdot \sim q)$$

O *script* correlativo à operação identidade será aquele em que for verdadeira apenas a situação em que “a Bola azul se sobrepõe ao tapete. A bola não se sobrepõe à bola e esta última não está invisível”

Figura 25: INRC da operação de implicação com conjunção



8. ANÁLISE DOS RESULTADOS

Como foi dito, os desafios do experimento foram criados durante o trabalho com uma turma do nono ano do Ensino Fundamental do CAP/UFRGS, a partir da conversa entre um estudante e seu professor. O objetivo do experimento era observar como eles conseguiam explicar a lógica presente no comando *Teste* do *Etoys*. Os quatro estudantes entrevistados (com idades entre 14 e 16 anos) realizaram os desafios apresentando comportamentos parecidos, embora demonstrando mais ou menos facilidade. Sendo assim, o pesquisador ficou instigado a observar se crianças mais novas conseguiriam (ou não) resolver ou explicar tal lógica dos desafios. Para isso foram entrevistadas cinco crianças de 10 anos que faziam parte de um grupo de programação na escola estadual de Ensino Fundamental Dinah Péri Pereira que participava do Projeto UCA. Nesse grupo elas estavam aprendendo a desenvolver games numa oficina de *Etoys* orientada pelo pesquisador do LEC/UFRGS Bruno Fagundes Sperb. Logo, já tinham experiência na utilização do comando *Teste* nos seus projetos.

Embora as crianças entrevistadas constituam um grupo pertinente aos interesses dessa pesquisa – uma vez que poderiam apresentar comportamentos característicos do operatório concreto – tal escolha ocorreu em função dessa parceria com o LEC/UFRGS. Visto que, para coleta de dados, teve-se dificuldade em encontrar grupos de crianças que trabalhassem com programação na escola e, ainda, com o *Etoys*. Durante o desenvolvimento da pesquisa observou-se que aqueles que trabalhavam com o *Etoys* acabaram migrando para o *Scratch*, pois este apresentava uma interface mais intuitiva e organizada para se aprender a programar. Além disso, o governo federal brasileiro parou de financiar o projeto UCA/RS, diminuindo, assim, as atividades desse tipo.

Dessa maneira foram entrevistados nove sujeitos ao todo – cinco com dez anos e quatro adolescentes – porém alguns arquivos de vídeo foram corrompidos, reduzindo tal número para oito. Apesar do conjunto de entrevistados não contemplar todas as idades entre dez e dezesseis anos, os dois grupos foram emblemáticos, visto que suas ações evidenciaram comportamentos que evidenciam as diferenças entre o operatório concreto e o formal. É necessário ressaltar que tal fato era desconhecido para o autor, portanto somente após a análise dos dados e do estudo da lógica piagetiana tomou-se consciência. A seguir apresentam-se os resultados obtidos em cada um dos desafios.

8.1 OBSERVAÇÕES INICIAIS SOBRE OS DESAFIOS

Alguns estudantes, para resolver o desafio 1, apresentaram como primeiro objetivo construir um *script* que fizesse o rato contornar a comida, pois não admitiam que ele “andasse sobre a comida”. Como tal algoritmo é bastante complexo, os entrevistadores falavam que o rato poderia passar por cima da comida já na apresentação do desafio. Quando o estudante demonstrava um forte interesse nessa programação, os entrevistadores o deixavam realizar o seu desejo, porém, após algumas tentativas tal ideia era abandonada pela própria criança ou sugerida pelo entrevistador quando (este) percebia um certo cansaço do sujeito.

As crianças de dez anos, ao resolverem os desafios 1 e 2, utilizaram o sistema de coordenadas cartesianas para fazer a “comida desaparecer”, cuja estratégia era atribuir valores de x e y suficientemente grandes para que não ficasse no campo visível da tela. Pensa-se que tal fato é influência das atividades desenvolvidas na oficina de Games, na qual tal estratégia era amplamente utilizada. No entanto, a maioria das crianças não havia compreendido tal sistema, pois era algo novo. Com isso, a complexidade do desafio ficou ainda maior e sua atenção era desviada em função das dificuldades que diziam respeito ao uso do plano cartesiano e não da lógica do comando *Teste*.

Para auxiliar o leitor, abaixo apresenta-se uma tabela que resume o desempenho de cada entrevistado nos desafios e, a partir da análise realizada, os níveis cognitivos identificados.

Tabela 7: Desempenho dos entrevistados nos desafios

	Nível I	Nível II		Nível III	
		A	B	A	B
Desafios	Insucesso em 01	Sucesso em 01	Sucesso em 01e 02	Sucesso total, com compreensão parcial da lógica	Sucesso total com compreensão
Estudante L	x				
Estudante M		x			
Estudante K			x		
Estudante N			x		
Estudante J			x		
Estudante G				x	
Estudante H				x	
Estudante F					x

8.2 NÍVEL I

Durante a entrevista uma estudante apresentou uma indiferenciação com relação aos elementos do comando *Teste*, não distinguindo o que era condicional ou comando que executa uma ação quando colocado na linha do *Sim* ou/e do *Não*. Então chamamos de Nível I esse período de indiferenciação, pois a criança não consegue traduzir tais ações num algoritmo, embora tenha compreendido o que era para ser feito. Nas entrevistas identificou-se dois estudantes de dez anos que apresentaram uma dificuldade inicial para resolver o desafio 1.

8.2.1 Entrevista com estudante L (10 anos).

O desafio 1A foi apresentado à estudante que levou um tempo para entender o que precisava ser feito, pois de imediato pensou que fosse um jogo.

Entrevistador: Tu vês um rato aí?
Estudante L: Agora percebi que é um rato.
Entrevistador: Aquilo é um rato, e o resto o que são?
Estudante L: bolas!
Entrevistador: Isso! Mas vamos usar a imaginação? Aquilo ali são as comidas para o rato. O desafio é o seguinte: esse rato tem que avançar até a bola, e a gente tem que o fazer comer a bola.
Estudante L: Esse é de mexer pela tecla?
Entrevistador: Pode fazer como tu quiseres, se quiseres programar as teclinhas?
Estudante L: Não, não, obrigado!
Entrevistador: O desafio é fazê-lo chegar até a comida azul.
Estudante L: Já pode começar?
Entrevistador: Claro! (tenta apertar o botão de início e constata que não há resposta)
Entrevistador: Ainda não tem nada programado, é tudo contigo!
Estudante L: Como eu posso fazer? (Passa para a próxima página do livro.)
Entrevistador: Esses são os outros passos do desafio.
Estudante L: Mas como se joga esse negócio, como se movimenta?
Entrevistador: Eu que te pergunto! Se eu te perguntar assim, como a gente cria um *script* para ele fazer um movimento... se eu quero que ele vá pra cima, automático, como faz?
Estudante L: Avançar?
(já fazendo o *script* e executando, para então deslocar os comandos disponíveis e ver o resultado)
Entrevistador: Ah, mas então tu sabes como fazer!
Estudante L: E agora
Entrevistador: Agora ele tem comer essa bola ai! (Bola azul)

A estudante abriu o scriptor da bola amarela, passou o cursor pelos *Testes*, mas não chegou a pegar nenhum comando, ficando em silêncio, sem saber o quê fazer. Dessa maneira o entrevistador começou a conversar.

Entrevistador: Estás pensando? Tu podes ir me dizendo bem como tu tá pensando... **Estudante L:** É que assim, eu me esqueci como é que se tipo, se... não sei se tem como fazer, eu acho que sim... se o rato tocar nessa bola, essa bola some...

Então, a estudante L teve a ideia de que “comer a bola” significava que ela deveria sumir ao ser encostada pelo rato.

Entrevistador: Tem sim, muito bem. Vamos fazer!
Estudante L: Mas como?
Entrevistador: Tu podes tentar fazer bem como vai surgindo na tua memória? Pode fuçar!
Estudante L: Esse aqui né! (comando se sobrepõe).
Entrevistador: Lembras que no *Etoys* sempre tem vários jeitos de fazer alguma coisa né?
Estudante L: É que eu esqueci.

A estudante procurou no menu, na categoria dos *Testes*, parecendo estar em dúvida sobre qual deles utilizar e checando com o *mouse* um a um.

Estudante L: Esse aqui! (em *Teste*). Hum... Não!
Entrevistador: Porque não? Tenta!
(Leva o *Teste* para o mundo, criando um *script* novo, mas continua em dúvida se utiliza ou não as condicionais da categoria.)
Entrevistador: Diz-me como tu estás pensando.
Estudante L: A mesma coisa.
Entrevistador: Tu lembras o que é o *Testar*?
Estudante L: É tipo, quando... Mas eu esqueci.
Entrevistador: Ó, ali estão todos os “quandos”.
(no momento que ela passa o *mouse* sobre um dos comandos o entrevistador começa a ler em voz alta) Ó: quando a bola amarela vê a cor... A outra...
Estudante L: O negócio some.
(Sendo assim, ela arrasta para a caixa de *Teste* a condição de cor).
Estudante L: E agora?
Entrevistador: Eu que te pergunto!
(Ela havia criado o *Teste* apenas com a condicional “bola amarela ___cor vê a ___cor”; como mostra a figura 26 abaixo.)

Figura 26: Script 1 da estudante L



A estudante L executou os dois *scripts* mesmo sem ter ações e sem trocar as cores da condicional a ser testada. A criança constatou o resultado, mas ficou parada em silêncio sem saber o que fazer. Assim, tornou-se claro que a estudante não conseguia resolver o desafio, então o entrevistador interveio retomando uma atividade da oficina de Games e lhe ensinou novamente a utilizar o comando *Teste*. Logo em seguida, eles voltam ao desafio 1B e, após muitas intervenções indutivas, L conseguiu resolver o desafio. Espontaneamente, replica para as demais comidas (como é exigido no desafio 2), porém suas dúvidas apareceram constantemente atrapalhando seu desempenho. Logo, o último desafio nem lhe foi apresentado.

Após este relato fica claro que, embora a estudante L tenha tido experiência com o uso dos *Testes*, ela ainda não compreende os elementos que o compõe – e consequentemente o raciocínio condicional – pois, mesmo quando o entrevistador explora um *script* já conhecido, ela não se refere à condição a ser testada nas suas explicações ou nas tentativas de resolver o problema.

8.3 NÍVEL II A

Como os desafios 1, 2 e 3 expressam uso da implicação em situações mais complexas, foram categorizados como nível II aqueles estudantes que tiveram sucesso pelo menos no primeiro.

8.3.1 Entrevista com o estudante M (10 anos)

No entanto, durante a análise dos dados foi necessário criar um subnível para aqueles estudantes que apresentaram sucesso no primeiro, porém insucesso no segundo desafio.

8.3.1.1 Desafio 01

Estudante M: O rato tem que avançar e comer as bolinhas?

Entrevistador: Primeiro pode ser só a bola azul!

Estudante M: E o que é esse início?

Entrevistador: É para o rato voltar para o lugar dele. (Cria prontamente o *script* de avançar)

Estudante M: Tem como fazer isso, mas...

Entrevistador: Mas?

Estudante M: quase que nem aquele jeito, aquele jogo que eu tenho, que se tu encostar naquela coisa ele sai e volta para o lugar, mas aqui ele não volta para o lugar, tem que fazer ele sumir... **Entrevistador:** E como dá pra fazê-lo sumir?

Estudante M: Dá pra fazer pelo x e pelo y né... (pelas coordenadas cartesianas)

Entrevistador: Sim, acho que dá.

Estudante M: E qual é o mais fácil?

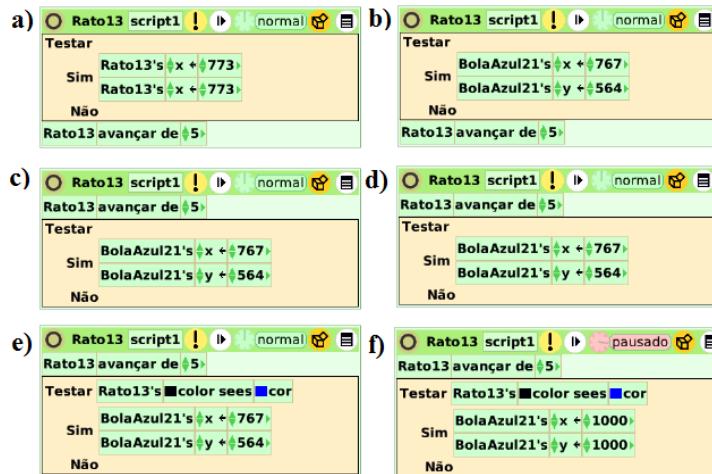
Entrevistador: Ah, isso depende de cada um, não sei qual o mais fácil.

Estudante M: Eu boto junto com o de avançar ou não?

Entrevistador: Isso tu é quem sabe!

Constrói o *script* e nele coloca ambos os comandos de avançar e um *Teste* para fazer a comida “desaparecer”. Prontamente faz o *Teste* colocando na linha do *Sim* as coordenadas x e y, do objeto “rato”, como mostra a figura 27A abaixo. Mas logo se dá conta de um problema e fala:

Figura 27: Scripts do desafio 01 do estudante M



Estudante M: Ah, mas eu to botando o x e y do rato, e não é, é o da comida!

Ele começou a fazer a mudança: criando outro *Teste*, acessado pelo *scriptor* da bolinha, mas pondo no *script* já criado. Criou, então, o *script* B apresentado na figura 27B acima e executando-o. Quando o rato não come a comida azul, o estudante fica surpreso e demonstrando insegurança nas suas ações.

Estudante M: Eu botei errado de novo (retira e recoloca o *Teste* no *script*).

Entrevistador: O que tu botaste errado?

Estudante M: Não, está certo.

Então ele mudou as cores na condicional adequadamente, logo a seguir passou a explorar o visualizador do objeto “rato” entrando na categoria “entrada”, depois, “speech bubbles”, etc. Sem entender o que a criança estava fazendo o entrevistador resolve perguntar.

Entrevistador: O que tu estás pensando em fazer agora?

Estudante M: É pra fazê-lo sumir.

Entrevistador: Quem?

Estudante M: A comida! (fica analisando os comandos da categoria “balões de fala”)

Estudante M: Aqui é pra ele falar alguma coisa quando encostar, então não é aqui. (Fica procurando no visualizador do rato).

Entrevistador: Quem tu queres fazer sumir?

Estudante M: A bola, a bola azul... Então não é aqui, né!

(Abre o visualizador da bola azul e continua sua procura, indo até a categoria dos balões de fala.)

Ele modificou a ordem dos comandos no *script* e o executou novamente para ver se acontecia algo de diferente, no entanto o resultado foi o mesmo.

Estudante M: O rato está encostando na comida?

O estudante respondeu a pergunta já trazendo para o *Teste* a condição “rato vê a cor”, no entanto ele ficou com dúvida sobre a composição de comandos no *script*.

Estudante M: Ah, mas isso aqui pode ser junto ou tem que ser separado?
Entrevistador: Isso tanto faz, tu podes fazer dos dois jeitos!

Enquanto criava o *script* E, figura 27E acima, M vai explicando seu *Teste*.

Estudante M: Sim, porque aqui (linha do *Sim*) não pode falar do rato, tem que falar das bolas, não pode falar do rato, porque senão o rato é que vai sumir!

Tal ação indicou que, para M, as coordenadas x e y do *script* E não eram suficientes, portanto havia a necessidade de um comando específico para fazer a bola sumir. Diante disso, surgiu a necessidade de saber qual era a função que as coordenadas x e y tinham em tal *script*, sendo assim o entrevistador procurou investigar.

Entrevistador: Como tu pode me explicar agora esse *script*(figura 27E) que tu criasses?
Estudante M: Agora precisa... Com isso daqui (os comandos dentro do *script*) não vai ajudar em nada, porque precisa de um *script* que vá fazer a bola sair!

Para elucidar, M executou o *script* para o entrevistador constatar que a bola não sumiria. Porém, o que o estudante M não percebeu que a comida não sumia, pois as coordenadas x e y do *script* eram as de sua posição inicial, portanto ficaria visível na tela. Em frente a tal indiferenciação, o entrevistador resolveu perguntar sobre as coordenadas escolhidas.

Entrevistador: Aquele y e aquele x o que é?
Estudante M: É pra bola sair!
Entrevistador: E porque não está saindo?
Estudante M: Porque tem que ter mais um *script*!
Entrevistador: Mas o que são aqueles x e y ali, o que eles tão dizendo?
Estudante M: Pra ela mudar de lugar!
Entrevistador: E porque ela não tá mudando?
Estudante M: Porque não é pra mudar de lugar, é pra ela sair!
Entrevistador: Mas ela pode desaparecer mudando de lugar. Que lugar é aquele ali?
Estudante M: É onde ela está!
Entrevistador: Então tu a está mandando ir pra onde ela está?
Estudante M: Ah, então se eu botar no 0.
(Modifica as coordenadas para (0,0) e executa novamente)

Estudante M: Ela sumiu! Bah! foi pra cá! (prontamente retira o *script* de lugar, procurando pela bola azul).
Entrevistador: Foi pro $x=0$ e $y=0$.
Estudante M: Bah! Mas ela foi lá pra baixo! (canto inferior esquerdo da tela.)

Diante da surpresa que a criança demonstrou ao descobrir a localização da comida azul, pode-se inferir que o sistema de coordenadas ainda não havia sido compreendido pelo estudante. Além disso, pelo diálogo, observou-se que o estudante não tinha construído noções das coordenadas que indicassem possíveis posições dos objetos na tela do computador. Ele utilizou a origem do plano cartesiano, sem conseguir prever a que região da tela se referia.

Por outro lado, como programar o rato para fazê-lo comer a comida já está superado, após a descoberta da origem do plano cartesiano, M tem a ideia de utilizar valores altos para as coordenadas, criando o *script* da figura 27F (acima).

8.3.1.2 Desafio 02

O estudante M (10 anos) foi o único que não conseguiu resolver o desafio 2. Quando tal desafio lhe é apresentado e explorado inicialmente, o estudante parece confuso e começa a afirmar que é muito difícil. Em nenhum momento ele demonstrou a necessidade de utilizar duas condicionais ou mesmo replicar o *script* anterior.

8.4 NÍVEL II B

Já os estudantes que tiveram sucesso nos dois primeiros desafios, porém não sabiam explicar a lógica do comando *Teste* foram categorizados no nível IIB. Ao analisar tal fato do ponto de vista da abstração reflexionante, tal sucesso foi alcançado via *reflexionamento*⁴ do *script* do desafio anterior, pois eles não sabiam explicar a lógica presente em tais algoritmos. Revelando que tal abstração foi realizada a partir de uma reprodução estática da sequência dos comandos do *script* do desafio anterior, mas sem a compreensão da implicação envolvida.

A título de organização, as entrevistas serão apresentadas a partir dos desafios.

8.4.1. Desafio 01

8.4.1.1 Estudante K (10 anos)

⁴ Segundo Piaget (1995) a *Abstração Reflexionante* é composta por dois componentes complementares entre si: o *Reflexionamento* – que é transpor (ou projetar) num plano superior o que se colhe num patamar precedente – e a *Reflexão* – que é reconstruir o que foi trazido do plano precedente com aqueles já situados no novo plano, gerando algo novo.

Num primeiro momento K pensou que o rato já estava programado e procurou tal *script* para executá-lo. Após ser retomada a ordem do desafio, K arrastou o rato “manualmente” até a comida azul, acreditando que essa era a solução. Assim, ele selecionou a bola azul e arrastou os comandos “*Teste*” e “bola azul está sobre a cor”, porém o entrevistador o lembrou de que ele precisava fazer duas coisas: primeiro fazer o rato avançar até a bola azul e depois comê-la. Então a criança disse que precisava utilizar o comando girar para contornar todas as outras comidas e chegar à bola azul. Desse modo, o entrevistador disse que essa era uma maneira, mas que não precisava fazê-la, pois o rato poderia passar por cima das demais comidas e só deveria comer a última. Essa intervenção desestabilizou o estudante, que logo em seguida afirmou que precisava utilizar um *script* da bola e depois outro para o rato.

Após refletir um pouco, K falou que poderia utilizar uma “perseguição” (atividade realizada na oficina de construção de Games), mas sem a bola azul se movimentar. Não satisfeito com ideia, a criança afirmou que o desafio era muito difícil e que não sabia como resolvê-lo. Nesse momento, o entrevistador perguntou se ele lembrava como se fazia um objeto avançar e a criança respondeu que sim e (imediatamente) criou um *script* com o comando avançar “*É só isso? Ah tá! Agora eu tenho que fazer (a bola) desaparecer*”.

O estudante K ficou pensando sem sugerir uma solução, dessa maneira o desafio 1B lhe foi apresentado. Diante dos comandos o estudante, de imediato, utilizou o comando *Teste* no *script* e procurou inserir o comando avançar, porém teve dúvidas quanto ao local onde colocá-lo (fora do *Teste*, no Sim ou no Não), como mostra a figura 28 abaixo.

Figura 28: Estudante K procurando onde colocar o comando avançar



Estudante K: Eu preciso de um *Teste* e de um avançar.

Ele demonstrou uma hesitação nas suas ações, mas largou o avançar no *Sim*. Depois retirou o comando, ameaçando colocar fora, antes e após o *Teste*; logo em seguida, sugeriu (com seus

movimentos) que iria colocá-lo no *Não*, mas acaba largando novamente no *Sim* e falando alto.

Estudante K: Assim ele não vai avançar!

Entrevistador: Por quê?

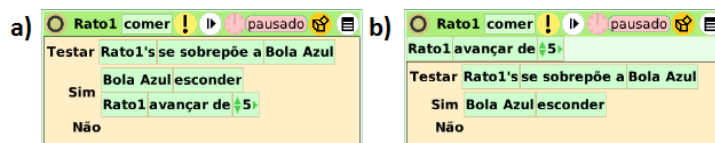
Estudante K: Ah, não, vai avançar, sim. (mudando de opinião), porque o *Sim* está dizendo: “*Sim, avance*” (No entanto, ainda não há condicional no *Teste*)

Entrevistador: Tá dizendo “*Sim, avance*” se o quê?

Estudante K: Para a bola azul, porque ele (o rato) vai seguir reto.

Enquanto conversavam, K colocou os demais comandos formando o *Script A* da figura 29 abaixo e afirma: “Acho que é assim!”.

Figura 29: Estudante K com dúvidas sobre lugar do comando avançar



Ele executou o *script*, mas como o rato ficou parado, K retirou o comando avançar da linha *Sim* e o colocou antes do *Teste* e, assim, o rato começou a andar imediatamente. Antes do rato chegar na bola azul o entrevistador lhe questionou.

Entrevistador: E quando ele chegar à Bola azul, o que vai acontecer?

Estudante K: Acho que a bola azul vai desaparecer. Viu, desapareceu. Se não me engano ela foi para de baixo do livro (Suporte do jogo).

É possível identificar que, de início, o estudante K tenha apresentado dificuldade em compreender o desafio, só conseguindo quando relacionou com uma experiência anterior (o jogo “perseguição”). Mesmo assim, demonstrou hesitação para começar a programação, só não desistiu, pois o entrevistador sentiu a necessidade de lhe apresentar o desafio 1B. Com a sugestão dos comandos, K apresentou a solução em poucos instantes. Porém, pode-se constatar que (para o estudante) a primeira decisão importante é onde colocar o “avançar”. Na sua fala foi possível observar sua crença de que a linha *Sim* confirma o comando avançar ou (desaparecer) e não a condição inserida na linha *Testar*. Da mesma forma que o *Não* negaria tais comandos.

Embora tenha conseguido realizar uma previsão correta, sua explicação não leva em consideração todas as possibilidades, indicando uma concepção limitada da lógica envolvida no comando *Teste* e, conseqüentemente, da operação de implicação.

8.4.1.2 Estudante J (10 anos)

Entrevistador: O quê você fez aí?
Estudante J: Peguei o avançar e o fiz andar acima, até ficar em cima da comida.
Entrevistador: E ele comeu a bola azul?
Estudante J: Não, só passou por cima.
Entrevistador: O que nós podemos fazer para ele comer a comida azul?
Estudante J: Tu quer que ele (o rato) encoste na bola azul e ela suma de repente?
Entrevistador: É uma alternativa. Essa é a ideia.

De imediato o estudante J inseriu o comando *Teste* e a condicional de reconhecimento de cores (*script* da figura 30A abaixo).

Entrevistador: Antes de você rodar o *script*, o que você programou ali?
Estudante J: Quando o rato encosta na bola azul e vai para $x = 400$ e o $y = 400$. Mas eu não sei onde é que vai dar.
(Inicia a animação e quando o rato encosta na comida azul, o objeto vai para a posição indicada)
Estudante J: Ah não! O rato que foi, tem que pedir que a bola vá.

O estudante realizou as modificações criando o *Script B* abaixo (figura 30B) e executou apenas o primeiro *script*, portanto o rato se movimentou para cima, sem realizar o *Teste*.

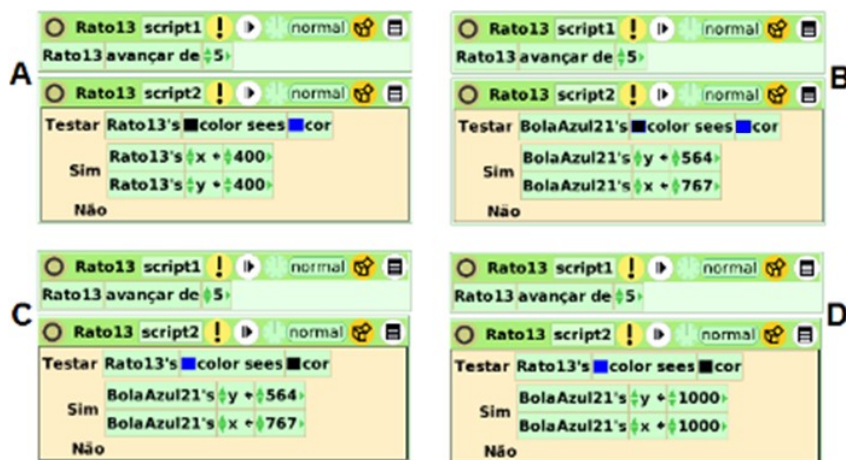
Estudante J: O que aconteceu ali?
Entrevistador: Ele não comeu... ah é por que eu não liguei esse daqui (*script* do *Teste*).

Ao executar os dois *scripts* o rato avança, mas quando encosta na comida a bola não desaparece da tela, pois as coordenadas no *Teste* são as de seu estado inicial.

Entrevistador: Você pode me explicar o que você programou nesse segundo *script*?
Estudante J: Quando o preto da cor do rato, encostar no azul, que é essa bola azul. Ela vai para um lugar, não sei, pelo x e o y . Deixa tentar uma outra coisa, ao contrário.

Assim o estudante trocou as cores da condicional, criando o *Script C* da figura 30.

Figura 30: Scripts para o desafio 01 do Estudante J



Entrevistador: Está com “o preto vê a cor azul”, ali (na condicional)?

Estudante J: Não, ele ...

O estudante inverte as cores voltando ao *script* da figura 30B anterior e executa novamente sem sucesso.

Entrevistador: A ideia é que você mude?

Estudante J: Sim, quando essa cor (preta) vê a cor azul, mas não está dando.

Entrevistador: E esses valores (x, y) aqui, para onde estão mandando a bola?

Estudante J: Não sei. Deixa eu botar mil neles, tipo: para dar uma sumida (na comida), se eu quiser voltar eu coloco zero nos dois, que dá para ver.

Por último, criou o *Script D* da figura 30 acima.

Nota-se que o uso do comando *Teste* é fácil para o estudante J, pois, para resolver o desafio 01, não apresentou dúvidas sobre qual comando vai na condicional *Testar* ou no *Sim*. Ele percebeu imediatamente qual foi seu primeiro equívoco, pois a mudança de posição do rato era perceptível, no entanto o *bug* do *script B* não foi. Neste caso, sua primeira tentativa foi modificar a seleção das cores, vendo que tal ação não surtiu efeito e com a intervenção do entrevistador, colocou valores para as coordenadas suficientemente grande para a comida desaparecer. É importante constatar que o fato de criar um *script* separado para fazer o rato avançar diminui as variáveis envolvidas no problema.

8.4.1.2 Estudante N (10 anos)

O estudante N é a criança (dentre as entrevistadas) com maior conhecimento e destreza na programação com *Etoys*. Pois, de imediato ele colocou o *Teste*, inseriu a condicional e os comandos de coordenadas x e y, executou o *script*, identificou e corrigiu os problemas com coordenadas ou com os objetos utilizados. (como mostra a figura 31 abaixo.).

Figura 31: Solução do desafio 1A do estudante N



Diante disso o entrevistador resolveu apresentar o desafio 1B, pois nele há um comando que

N desconhecia: o “rato se sobrepõe à bola azul”.

Entrevistador: Agora vamos passar pra próxima etapa?

Estudante N: sim!

Entrevistador: Agora o desafio é tu fazer com esses comandos que tem aí. Com eles tu podes fazer o rato comer aquela comida, tudo que tem aí é o necessário para tu fazeres.

Estudante N: Dá pra aumentar a velocidade?

Entrevistador: Podes fazer como tu quiseres.

Assim, construiu o *script* da figura 32A abaixo, e prontamente o executou; o que fez a bola sumir. Diz “opa” e, para recuperá-la, rapidamente buscou o comando “mostrar” e executou apenas uma vez.

Logo em seguida criou o *script B* da figura 32 abaixo, onde ele utilizou a condicional de reconhecimento de cor e ignorou a existência do comando rato sobrepõe. Durante este processo de construção o estudante chegou a inserir o valor da coordenada x, porém se dá conta que não é necessário, assim, excluiu e executou tal *script*. Como o avançar estava na linha do *Sim*, o rato não se moveu na tela, então o entrevistador perguntou.

Entrevistador: Porque não avançou?

Estudante N: Eu só tava testando uma coisa, só pra ver se a bola... se some sem isso aqui (as coordenadas).

Ou seja, N queria testar se a bola sumiria sem a utilização das coordenadas, como teve insucesso na sua tentativa, ele recolocou as coordenadas novamente na linha *Sim* do *Teste*, construindo o *Script C* da figura 32 abaixo. Executou o *script* e observou que nada acontecia, pois o rato não andava. Assim, retirou o avançar do *Teste* e o colocou num novo *script*, tendo dois *scripts*. Nessas duas ações pode-se observar que N, diante do fracasso de sua tentativa, começa a utilizar os comandos que conhecia previamente e, por último, retirou o avançar do *Teste* na tentativa de minimizar os problemas.

Entrevistador: Porque não estava avançando ali antes?

Estudante N: Acho que porque eu botei junto com esses aqui... Daí ele parou de se mexer.

Então ele trouxe a ferramenta “todos os *scripts*” e clicou no botão *Go*, o estudante já havia criado dois *scripts* diferentes. Fazendo com que o rato fosse e voltasse continuamente. Isso ocorreu, pois havia muitos outros *scripts* rodando no mesmo projeto, inclusive alguns não visíveis.

Entrevistador: Quando tu dizes pra rodar todos, ele vai rodar todos os *scripts*, até da outra página!

Prontamente ele eliminou o botão *Go* (que executa todos os *scripts*) e assim, executou somente os dois que apareciam na tela.

Entrevistador: Que vai acontecer agora quando o rato tocar na bola?
Estudante N: A bola vai se esconder como se o rato tivesse comido ela
Entrevistador: Tem certeza?
Estudante N: Sim!
Entrevistador: E vai fazer mais alguma coisa?
Estudante N: Vamos ver! (Executa os *scripts* novamente e observa)
Estudante N: Não. Está certo!

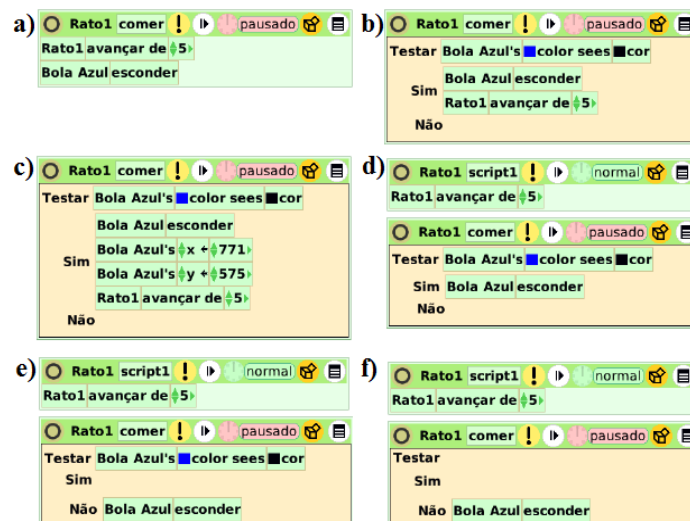
Para recuperar a bola escondida, repetiu o procedimento de criação, execução e eliminação do *script* “mostrar”.

Entrevistador: E esse x e y, que eles estão fazendo ali? **Estudante N:** É para ... Ah não sei, né... porque acho que me confundi numa coisa.

Então ele eliminou as coordenadas e executou, com sucesso, o *Script D*.

Estudante N: É que eu coloquei umas coisas estranhas ali junto, não sei explicar direito. Mas mesmo sem elas (coordenadas x e y) funciona mesmo assim, olha!

Figura 32: Scripts de solução do desafio 1B do estudante N



Entrevistador: Como tu pode explicar esse *Teste* aqui, como ele funciona?
Estudante N: Bom, ele tá funcionando com “bola azul esconder” (comando na linha *Sim*) e a bola azul, quando preto tocar nela, ela se esconde. Se eu tivesse colocado ali (no *Não*) ela não escondia né.
Entrevistador: E se tu colocar o esconder ali no *Não*, ela não vai se esconder, é isso que tu achas?
Estudante N: Sim!

O estudante criou o *script E*, executou-o e observou a bola azul se esconder assim que o rato começou a andar, ficando, assim, surpreso.

Estudante N: Se eu colocar aqui (no *Não*) ela vai automaticamente, sem precisar de nada... que legal (a criança parece confusa).

Entrevistador: Porque será que quando tu botaste (o comando esconder) no *Não* e ela desapareceu?

Estudante N: Não sei! Acho que quando tu botar no *Não*, isso daqui (indica a condicional) não vai precisar!

Sendo assim ele criou o *script F* da figura 32 acima, depois o executou e observou que nada havia acontecido.

Estudante N: Bah! Que coisa estranha, não dá pra entender muito bem isso não.

Ao constatar que N sabia como fazer o *script* corretamente, porém não conseguia explicar o seu funcionamento, então o entrevistador resolveu explorar algumas situações (figura 33 abaixo) para ensiná-lo tal lógica.

Entrevistador: Um *Teste* funciona assim: se a cor azul da bola azul vê a cor preta do rato. Quando tu põe isso no *Teste* tu está afirmando, tu está dizendo que uma cor vê a outra. Se isso for verdade: Tudo que a gente quer que aconteça se põe na linha do sim. Pode colocar muitas coisas.

Agora, tudo que a gente quer que aconteça quando aquilo (a condicional) não tiver acontecendo...

Estudante N: A gente coloca no *Não*.

Entrevistador: Isso. Coloca ali no *sim*, vamos ver o que acontece na seguinte situação 1, figura 33 abaixo.

O estudante N executou o *script* sem o *loop*, utilizando o recurso “passo a passo”, que executa cada comando na ordem apresentada dentro do *script*.

Entrevistador: A bola azul tá vendo a cor preta?

Estudante N: Não!

Entrevistador: Só vai esconder se for verdade aquilo ali (a condicional), não é?

Entrevistador: Por que não está escondendo? Porque isso (indicando com o dedo o comando esconder) só acontece, quando aquilo (indicando a condicional) for verdade.

Estudante N: Sim.

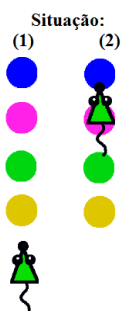
Entrevistador: Agora nessa situação 2 (figura 33 abaixo) Isso (indicando a condicional) é verdade?

Estudante N: É!

Entrevistador: Então agora ela se esconde.

Estudante N: Hum.

Figura 33: Situações propostas ao estudante N



Entrevistador: Vamos experimentar esse (*Script E* da figura 32 acima) na situação 1 da figura 33 acima.
Entrevistador: Nós temos Bola azul, a cor azul vê a cor preta, agora, se isso for verdadeiro, que vai acontecer?

O entrevistador ficou indicando com o dedo a linha do *Sim*, que está sem comandos.

Estudante N: Hum, Quando... Se verdadeiro... Hum... Ah, como se explica isso? Ah, não sei explicar isso direito! É bem difícil!
Entrevistador: Quando não for verdadeiro, aí...
Estudante N: Ela se esconde!
Entrevistador: Agora tá vendo a cor azul?
Estudante N: não!
Entrevistador: Então ela vai se esconder, não é?
Estudante N: É...
Entrevistador: Roda pra gente ver? (após executá-lo).
Estudante N: É, verdade!

Dessa maneira, o entrevistador identificou que N não estava entendendo muito bem, mas conseguia responder corretamente, pois já havia testado as duas situações. Sendo assim, o entrevistador resolveu aplicar o *Script E* na situação 2.

Entrevistador: O que acontece se tu fizeres um ver o outro?
Estudante N: Como assim?
Entrevistador: Se o preto do rato vê a cor azul (situação 2)? Se você rodar o *script*, vai esconder ou não vai?

O estudante não consegue antecipar e parece não ter ideia do que vai acontecer, sendo assim ele executa o *Script E*.

Estudante N: Não! Ah, porque tá dizendo não pra esse daqui, e só esse daqui tá funcionando, agora entendi! Hum, Isso aí.

Ao analisar a entrevista, parece que o estudante N compreendeu o significado do *Não* no *Teste* pela primeira vez e abandonando, assim, a interpretação de que a negação se refere aos comandos listados nessa linha. Vale ressaltar que essa criança possuía grande destreza na programação com *Etoys* e conseguiu resolver o desafio com agilidade, mas não conseguiu explicar como funcionava.

Após a análise das entrevistas é nítido o fato de que as crianças classificadas nesse nível apresentam dificuldade em compreender a lógica do comando *Teste*, mesmo quando sabem como utilizá-lo, não conseguem explicá-lo.

8.4.2. Desafio 02

Quanto ao segundo desafio, as crianças deste nível o resolveram sem nenhuma dificuldade, apenas replicando instantaneamente o *Script* para uma comida de cores diferentes. Tal resultado não era esperado, visto que na experiência-piloto a operação da disjunção representava uma dificuldade pertinente e específica. Refletindo sobre tais resultados, encontram-se duas justificativas: a) a situação era bastante parecida, portanto não era necessário fazer modificações de um *Teste* para outro e b) tais crianças participavam de uma oficina de construção de Games, no qual é comum utilizar o *Teste* para programar um objeto para ser controlado pelo teclado.

8.4.2.1 Estudante K (10 anos)

O entrevistador apresentou o Desafio 2A, porém ao terminar sua explicação, o estudante K, foi para a página seguinte com o objetivo de acessar os comandos sugeridos (Desafio 2B). Refletindo sobre tal ação, pensa-se que é consequência de um comportamento desenvolvido a partir da experiência de programar. Visto que, pode-se observar, no início da entrevista, ao conhecer um novo projeto em *Etoys*, essa criança estava acostumada a explorar os *scripts* escondidos para tentar compreender e aprender os algoritmos utilizados.

Estudante K: Ah, precisa de dois *Testes*.

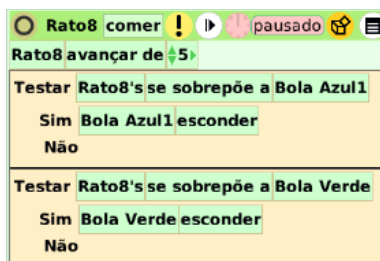
Imediatamente o estudante K colocou os dois *Testes* em paralelo e inseriu o comando avançar antes do *script Teste*, então o entrevistador resolveu questioná-lo sobre o desafio anterior.

Entrevistador: Por que quando você coloca o avançar no *Sim* ele não anda?

Estudante K: Ah, não sei explicar.

Enquanto conversavam, o estudante K continuava criando o algoritmo que resolveria o desafio, apresentado na figura 34 abaixo.

Figura 34: Script da solução do desafio 2 do estudante K



Assim que o rato “comeu” as bolas azul e depois a verde, K foi para o Desafio 3A imediatamente, avançando de fase através das páginas do livro, antes mesmo do pedido do entrevistador.

8.4.2.2 Estudante J (10 anos)

O desafio 2A foi apresentado ao estudante J, que imediatamente cria o *script* abaixo.

Figura 35: Scripts criados pelo estudante J para o desafio 02



Entrevistador: Antes de executar, você pode me explicar o tu fez?

Estudante J: Eu fiz a mesma com os números, mas aí eu fiz diferente, fiz a mesma coisa para a amarela. Quando a bola amarela encontrar com o preto vai para $y = 1000$ e $x = 1000$. E a mesma coisa com a azul.

Entrevistador: Se você tivesse que programar para o rato comer todas as comidas, o que você faria?

Estudante J: Faria a mesma coisa que eu fiz nesse, faria um para o amarelo, para o azul, para o verde e para a rosa. Daí eu mandaria todos para o 1000. Quando o rato encostar nelas eles iriam pra... não sei, coloca 1000 no y e 1000 no x.

Logo em seguida ele executou o *script* e obteve sucesso.

Entrevistador: Onde é que é o 1000 x e y?

Estudante J: Aqui (aponta para os comandos inseridos no *script*). Não sei. Sei que o y é para cima e o x é para os lados. Eu não entendo muito desses 1000 e dos setecentos e poucos. Não entendo bem.

Entrevistador: Não imagina onde a bola está?

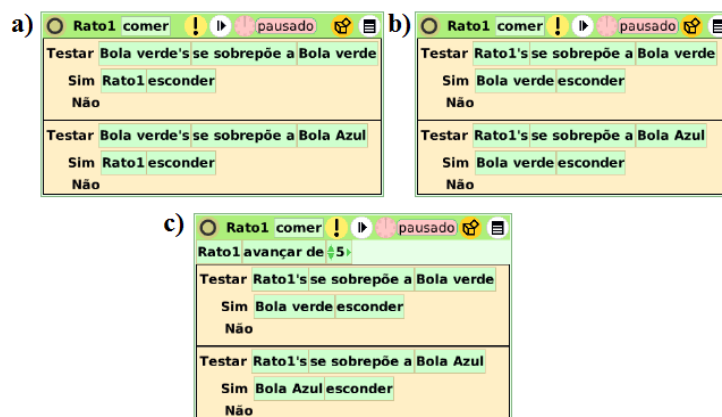
Estudante J: Não, não sei.

8.4.2.3 Estudante N (10 anos)

O desafio 2A foi apresentado ao estudante que criou dois *Testes* no mesmo *script*, sem nenhuma indicação do entrevistador, e utilizando o comando “se sobrepõe”. Porém tal *script* (da figura 36A abaixo) apresentava muitos problemas. Dessa maneira, o entrevistador pediu que ele explicasse tal *script*. Durante sua explicação, N identificou o erro e exclamou: “Ah não, eu sempre faço isso!”.

Ou seja, prontamente após a pergunta, se deu conta de que havia utilizado comandos referentes às bolas, sem a interação com o rato. Dessa maneira, modificou o *script*, mas com comandos relacionados aos objetos corretos, logo em seguida o executou e observou seu sucesso nesse desafio.

Figura 36: *Scripts* criados pelo estudante N para o desafio 02



Entrevistador: Você pode me explicar como funciona o *Script B*? (figura 36B acima).

Estudante N: Eu fiz assim: quando o rato se sobrepõe à bola verde, ela vai se esconder. E quando se sobrepõe à bola azul, a bola azul vai se esconder.

Entrevistador: Porque fizeste o avançar separado?

Estudante N: Ah, sim, dá para colocar junto aqui ó! (cria o *Script C* da figura 36 acima) Só que aí eu não fico me confundindo, né.

Entrevistador: Sim, tu te organizas melhor.

8.4.3 Desafio 03

Pode-se observar que as crianças deste nível não obtém êxito neste desafio. Tais estudantes numa primeira tentativa reproduzem o *reflexionamento* citado no desafio anterior, porém diante do insucesso, não conseguem realizar a *reflexão* necessária para estabelecerem a abstração reflexionante e compreender todas as classes e as subclasses envolvidas. Confirmando, assim, que tal abstração é realizada a partir de uma reprodução estática da sequência de comandos do *script* do desafio anterior, mas sem compreensão da implicação envolvida. Como consequência deste fato, observa-se que eles apresentam dificuldade de compreender a ordem deste desafio.

Além disso, pensa-se que tal ação pode ter sido gerada em função da forma como o desafio 2 foi apresentado aos entrevistados, pois foi solicitado aos estudantes que programassem o rato para “comer duas comidas” em vez de “comer uma ou outra”, portanto surge a crença de que dois *Testes* em paralelo correspondiam a uma conjunção de condicionais.

8.4.3.1 Estudante K (10 anos)

Como o estudante J não havia conseguido compreender o Desafio 3, o entrevistador apresentou este desafio, mostrando exemplos “manualmente”.

<p>Entrevistador: Ele (o rato) tem que comer a bola azul, mas só quando ela estiver em cima do tapete. Se ele encontrar ela aqui (arrasta a bola para fora do tapete e, depois, o rato até ela) a bola azul não pode desaparecer. Se ele encontrar ela aqui (arrasta a bola para outra parte do tapete e, depois, o rato até ela) a bola azul tem que desaparecer. Tu entendeste?</p> <p>Estudante K: Mais ou menos.</p> <p>Entrevistador: Imagina que isso é um jogo e essa bola estivesse fugindo.</p> <p>Estudante K: Que nem uma perseguição?</p> <p>Entrevistador: Isso, como uma perseguição. A bola está fugindo (simula manualmente a situação) e o rato encontrar ela fora do tapete, não some. Só come se ela estiver em cima desse tapete mal assombrado.</p> <p>Estudante K: Um tapete mágico.</p> <p>Entrevistador: Isso, só come se eles se encontrarem E estiverem no tapete mágico, se eles se encontrarem fora, não come. Entendeu?</p> <p>Estudante K: Sim.</p> <p>Entrevistador: Tem como fazer?</p> <p>Estudante K: Sim.</p>
--

Novamente o estudante K, foi para a página seguinte, onde estavam os comandos sugeridos (desafio 3B) e começou a realizá-lo. A impressão que se tem é que ele, de imediato, procurou reproduzir o *script* do desafio anterior, como mostra o *Script A* da figura 37 abaixo. No entanto, ao descobrir o comando “bola azul se sobrepõe ao tapete” falou: “Não precisa de dois *Testes*”, assim,

seleciona tal comando e tenta construir o *Script B* da figura 37 abaixo.

Estudante K: Um *Teste* em cima de outro *Teste*. (constrói o *Script A*), Acho que não preciso de um. (Exclui o segundo *Teste* e tenta criar o *Script B*)
Entrevistador: O quê?
Estudante K: Esse aqui não vai precisar.
Entrevistador: Não vai precisar de dois *Testes*?
Estudante K: Sim, não vai.

No entanto, ao tentar coloca a condicional na linha *Não*, percebeu que não era possível criá-lo.

Estudante K: Não, vai!

Assim, o estudante inseriu o segundo *Teste* no *script* modificando-o novamente.

Estudante K: Então esse aqui (condicional “bola azul sobrepõe ao tapete”) vai aqui (na condicional do segundo *Teste*, como mostra o *Script C* da figura 37 abaixo).
Entrevistador: Antes de executá-lo me explica o que vai acontecer?
Estudante K: Eu coloquei a “bola azul se sobrepõe ao tapete” (indica com o *mouse*), aí ele (o rato) deveria comer... a bola azul esconder (indica com o *mouse* no segundo *Teste*). Aí quando ele chegar no tapete, mas não vai encostar, vai ser quando ele chegar perto. Sendo assim, quando ele comer, quando (o rato) estiver sobre a bola azul (indica com o *mouse* a condicional do segundo *Teste*).

O entrevistador não entendeu a explicação dada, então o deixa executar o *script* e quando o rato se sobrepôs ao tapete, perguntou:

Entrevistador: Olha, ele se sobrepôs ao tapete e não aconteceu nada, por quê?

O estudante não entendeu o porquê da pergunta e ficou em silêncio. Sendo assim, continuaram com a execução do *script* até o rato encontrar e comer a comida que está sobre o tapete.

Entrevistador: Deu certo?
Estudante K: Acho que sim.
Entrevistador: E se o rato encontrar a bola fora do tapete ele vai comê-la?
Estudante K: Não, por causa da programação dele. Ele está programado para comer no tapete. Se ele tivesse programado para comer fora estaria assim, olha;

O estudante tira a condicional e mostra como se ela estivesse na linha *Não*, como mostra o *Script D* da figura 37 abaixo.

Estudante K: Só que não tem como.

Podem-se observar dois fatos nessa fala: a) a hipótese de que o *Não* nega o objetivo geral do desafio, em vez de negar a condicional apresentada na linha *Testar*. b) como consequência dessa indiferenciação, parece que uma das primeiras crenças apresentadas pelas crianças é de que, para ter sucesso no desafio, é suficiente um *Teste* com uma condicional e/ou comando que envolva o objeto tapete. Note que a linha *Sim* do primeiro *Teste* está vazia.

Após o diálogo anterior, K se esquece de colocar a condicional “bola azul se sobrepõe ao tapete” no *script* e executa o *Script E*, da figura 37 abaixo.

Entrevistador: A bola azul vai se esconder? (ela está fora do tapete)

Estudante K: Não. (No entanto, ele observa ocorrer o contrário)

Entrevistador: Isso não era para acontecer, né?

Estudante K: Não.

Entrevistador: Lembra que ele só pode comer quando estiver em cima do tapete?

O estudante percebeu que o *Script E* estava errado e recriou o *Script C*. Logo em seguida, o executou para a situação em que a comida está fora do tapete e se surpreende quando o rato a come mesmo assim. K fica parado observando em silêncio e pensativo.

Entrevistador: E todos os comandos necessários para fazer o desafio estão ali.

Ele continuou sem saber o que fazer.

Entrevistador: Você sabia que se pode colocar um *Teste* dentro de outro *Teste*?

Estudante K: Aham! Dá para colocar assim.

Imediatamente o estudante colocou o segundo *Teste* na linha *Não* do segundo, mas o retirou e colocando na linha *Sim*, criando o *Script F* da figura 37 abaixo.

Estudante K: Mas só dá para colocar na linha *Não*. (contrariando sua recente ação)

Entrevistador: Na verdade você pode colocar em qualquer lugar no *Sim* ou no *Não*.

A criança ficou em silêncio e, logo após, executou o *Script F*, mas o rato continuou comendo a bola quando essa estava fora do tapete. Novamente ele ficou em silêncio, pensativo e sem saber o que fazer.

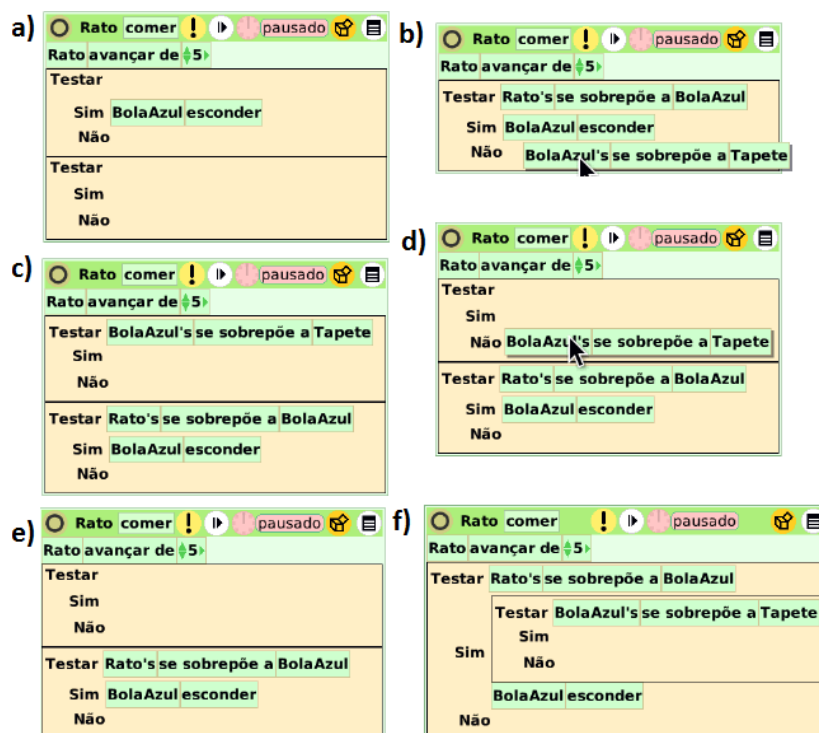
Entrevistador: São duas coisas que devem acontecer para ele comer a bola, né! Quais são as duas coisas?

Estudante K: Ele não poderia comer a bola (ela se encontra fora do tapete) ele tem comê-la lá no tapete.

Entrevistador: Isso. São duas coisas: a primeira é ele (o rato) ver a bola azul e a outra é a bola azul estar sobre o tapete.

Mesmo com o entrevistador indicando as duas condicionais nos *Testes* o estudante K não soube o que fazer, ficando paralisado.

Figura 37: Scripts criados pelo estudante K para o desafio 03



8.4.3.2 Estudante J (10 anos)

Sua primeira ação foi a de reproduzir o *script* do desafio 1A para a comida rosa.

Entrevistador: Você pode me explicar o que programou?

Estudante J: Ele vai esconder a bola rosa quando ela estiver em cima do tapete e agora ela está em cima do tapete.

Entrevistador: Agora se eu colocar a comida aqui (fora do tapete), quando o rato encontrá-la ele vai comer ou não?

Estudante J: Não vai, porque ela não está em cima do tapete.

Entrevistador: Vamos executar o *script*.

Estudante J: Espera aí, eu sei o que fazer com a bola rosa... Posso fazê-la avançar?

Entrevistador: Pode.

O estudante J criou um *script* para a comida azul avançar, como o rato tinha uma velocidade maior, ao executar o *script* eles se encontraram sobre o tapete.

Estudante J: Eu a levei para cima do tapete e o rato foi atrás dela.

Entrevistador: Muito bem, uma excelente ideia, e se eles estivessem se encontrado fora do tapete, o rato vai comê-la?

Estudante J: Não.

O entrevistador não entendeu a resposta do estudante e seguiu seu raciocínio como se a resposta tivesse sido sim.

Entrevistador: Tem como fazer uma programação para que isso também aconteça: quando ela (a comida) estiver em cima do tapete o rato come e quando estiver fora, o rato passe e não a coma? **Estudante J:** Então você quer que eu faça o rato passar por ela e não coma?

Entrevistador: Quando a comida estiver fora do tapete, mas se ela estiver em cima do tapete e o rato passar por ela, ele vai comê-la.

Estudante J: Sim.

Entrevistador: Tem como fazer isso?

Estudante J: Sem mover ela (a comida) com o avançar?

Entrevistador: Sem mover a bola.

Estudante J: Daí vai ser complicado.

A criança começou a procurar outros comandos, inclusive alguns que ele não entendia como funcionavam.

Entrevistador: Você está procurando algum comando em específico?

Estudante J: É, eu não aprendi a fazer isso, ainda. Não aprendi.

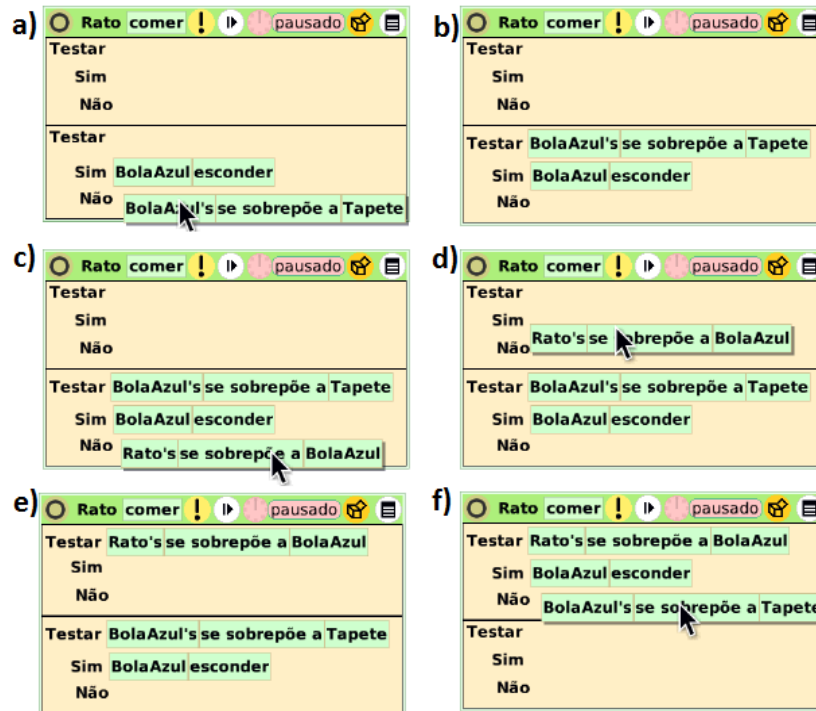
Entrevistador: Já que não tem um comando que você não aprendeu, vamos para a próxima fase.

Sendo assim o desafio 3B lhe foi apresentado.

Estudante J: Ah é esse comando que eu quero achar “sobrepõe e esconder”.

A primeira ação do estudante J foi construir dois *Testes* paralelos e começar a distribuir os comandos dentro deles. Primeiro ele colocou o comando “bola azul esconder” na linha *Sim* do segundo *Teste* e o “bola azul se sobrepõe ao tapete” na linha *Não* (figura 38A, abaixo), que não foi aceito, pois se tratava de uma condicional. Nessa tentativa, a criança descobriu que a condicional era aceita na linha *Testar* do *Teste* e formou o *Script B* da figura 38 abaixo. Depois o estudante J tentou inserir – várias vezes sem sucesso – a condicional “rato se sobrepõe ao tapete” na linha *Não* do segundo *Teste*, como indica o *Script C* da figura 38 abaixo. Logo em seguida, fez a mesma tentativa para as linhas *Não* e *Sim* do primeiro *Teste* como é mostrado no *Script D* da figura 38. Não obtendo sucesso, ele inseriu a condicional na linha *Testar*, formando o *Script E*, abaixo. Tal ação não foi intencional, ela ocorreu porque a linha verde apareceu, tornando possível tal inserção.

Figura 38: Scripts criados pelo estudante J para o desafio 03



Quando a criança foi executar o *script* o entrevistador perguntou:

Entrevistador: Espera um momento, me explica o que você programou no *script*?

Estudante J: Eu estou tentando fazer a bola se esconder quando estiver dentro do tapete, mas não vai.

Enquanto ele explicava começou a modificar o *Script E*, inserindo o comando “bola azul esconder” no *Sim* do primeiro *Teste* e tentou várias vezes inserir a condicional “bola azul se sobrepõe ao tapete” na linha do *Não*, como indica o *script F* da figura 38 acima. Após o insucesso dessa ação, falou:

Estudante J: Eu não estou entendendo esses programas (se refere aos comandos) eu não sei onde é que eu tenho que botá-los.

O estudante tira um dos *Testes* do *script* e fez a seguinte pergunta.

Estudante J: Tudo bem se eu deletar esse daqui (um *Teste*)?

Entrevistador: Ok, mas você pode me explicar o que quer fazer?

Estudante J: Eu quero fazer a bola azul ir para o tapete (no momento ela está fora) e a fazer esconder, mas eu quero tirar esses comandos (aponta para o *script*), pois eles não tem *script*. **Entrevistador:** Tudo bem, então faça.

Vendo que o estudante não estava compreendendo o que era para ser feito, um segundo

entrevistador que filmava a entrevista, procurou retomar a ordem do desafio.

Entrevistador 2: J, você se lembra o que era parar fazer nesse desafio?

Estudante J: Que eu fizesse o rato comer as bolas.

Entrevistador 2: Quando o rato passar sobre a bola e ela não estiver em cima do tapete não vai acontecer nada. Se ela tiver em cima do tapete, ele come ela. Tu entendes?

Estudante J: Sim.

O estudante respondeu de forma mecânica, sem parar de procurar os comandos.

Entrevistador 2: O que eu quero te mostrar é uma coisa que eu nunca te mostrei antes. Diz-me se eu estou certo, se é isso mesmo que você quer!

O entrevistador 2 buscou o primeiro *Teste* e inseriu no *script*, mas quando pegou o segundo a criança se manifestou.

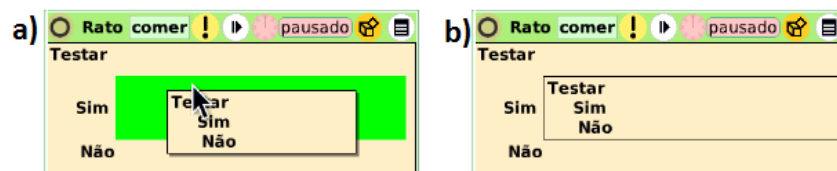
Estudante J: Não, não. Eu não sei para que serve esses *Testar* aí.

Entrevistador: O que eu quero te mostrar é isso, tu sabias que você pode fazer isso daqui?

O entrevistador 2 arrastou o segundo *Teste* até aparecer a linha verde na linha *Sim*. Figura 39A abaixo

Estudante J: O que é isso? Larga aí! (figura 39B). Não entendi. Como assim?

Figura 39: Entrevistador mostrando que é possível inserir um *Teste* dentro do outro



Entrevistador 2: É que isso daqui nós não havíamos visto antes, que é fazer um *Teste* dentro de outro *Teste* (o estudante fica em silêncio). Você achou difícil?

Estudante J: Não.

Entrevistador 2: Você quer tentar fazer?

Estudante J: Sim.

Sendo assim, o estudante J refez a construção de um *Teste* dentro do outro e ficou parado olhando sem saber o que fazer.

Entrevistador: O que você quer testar primeiro?

Estudante J: Não sei.

Como o estudante J ficou sem saber o que fazer a entrevista foi encerrada.

8.4.3.3 Estudante N (10 anos)

Entrevistador: Tu tens que fazer um programa para quando o rato tocar na bola azul ele deve comê-la, mas isso só pode acontecer quando a comida azul estiver sobre o tapete.

Estudante N: Como assim? Ah, entendi. Só pode comer quando tiver em cima do tapete... E aí?

Entrevistador: Será que dá pra tentar?

Estudante N: Vamos tentar, mas eu nunca fiz isso aqui na minha vida!

Estudante N: Pode ser do mesmo jeito também, né!

Entrevistador: Experimenta, tu sabes que não tem certo nem errado no *Etoys*!

Note que antes de construir um *script*, o estudante resolveu perguntar se poderia fazer a mesma coisa dos anteriores. Sendo assim, *Script A* (figura 40 abaixo) foi construído e entrevistador interveio antes de o estudante o executá-lo.

Entrevistador: Antes de rodá-lo, tu podes me dizer o que tu fez ali? Como é que tu estás pensando isso?

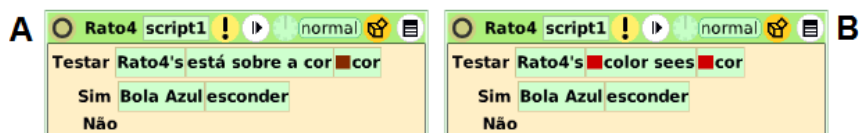
Estudante N: Aqui ó: “Rato está sobre a cor marrom a bola azul se esconde” (com o *mouse* circula o tapete). Então quando o rato estiver em cima do tapete, essa parte aqui (circula com o *mouse* o tapete), Bah! (...) mas se eu fizer isso ela vai se esconder até quando eu por ali (na borda do tapete e antes de chegar na comida).

(Executa tal *script* e confirma sua antecipação)

Estudante N: Ah! Bah! Mas é difícil esse hein, nossa!

Dessa maneira, ele construiu o *Script B* da figura 40 abaixo, onde tentou *criar um Teste* com o rato na condicional, mas dessa vez utilizando o reconhecimento de cores “cor__ vê cor__”. No entanto, nem selecionou as cores, pois pareceu identificar que não adiantaria.

Figura 40: Scripts criados pelo estudante N para o desafio 03



Estudante N: Bah! Mas é difícil.

Infelizmente o tempo do encontro acabou antes de explorar mais situações, no entanto já está claro que, em função da incompreensão da lógica envolvida no comando *Teste*, o desafio 03 é, por enquanto, inalcançável às crianças deste nível. Porém, observou-se que tal situação será superada no nível seguinte, justamente pela influência do pensamento formal.

8.5 NÍVEL III A

Durante as entrevistas ficou claro que os adolescentes utilizavam estratégias diferentes das

crianças e que lhes permitiam compreender o terceiro desafio, contrapondo os sujeitos de 10 anos que nem compreendiam a ordem do problema, mesmo após várias explicações. No entanto, o sucesso (parcial ou total) dos adolescentes também não foi obtido de forma espontânea, sendo alcançado mediante a intervenção do entrevistador.

Justamente o que amplia as possibilidades de compreensão e resolução é que, nas estratégias utilizadas pelos adolescentes, impera uma ação combinatória. Tal combinatória é realizada a partir de operações inversas, no entanto num primeiro nível (III A), tais operações não formam um sistema com suas recíprocas reunidas em sua totalidade. Com isso os sujeitos deste nível, apresentam uma compreensão parcial do algoritmo.

8.5.1 Desafio 01

Apresenta-se, a seguir, os diálogos com o estudante G, que foi um dos primeiros entrevistados, portanto trabalhou com a primeira versão do experimento no qual havia duas comidas para cada cor (azul e amarela), como mostra a figura 41 abaixo.

Figura 41: Primeira versão do experimento com comidas de apenas duas cores



8.5.1.1 Estudante G (16 anos)

O estudante G tinha pouca experiência com o *Etoys*, portanto conhecia poucos comandos, sendo assim nesta entrevista foi o seu primeiro contato com o comando *Teste*. Dessa maneira, de início, o entrevistador foi mais instrutivo, para conseguir observar como seria a apropriação do comando *Teste* pelo adolescente, sua interpretação em relação ao seu funcionamento e se conseguiria resolver os desafios propostos.

O desafio 1A é apresentado e o estudante G montou dois *scripts* distintos: um para fazer o rato avançar e outro para a comida se esconder. Como ele ficava sem saber o que fazer o desafio 1B lhe foi apresentado e o adolescente começou a programar imediatamente.

Entrevistador: Quais comandos você está utilizando?

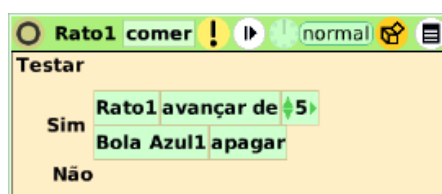
Estudante G: Esses daqui.

Entrevistador: E por quê?

Estudante G: Por que estão aqui. **Entrevistador:** Você pode ler quais são todos os comandos disponíveis.

Após ler os comandos, o estudante começou a introduzir e retirar aleatoriamente uma série de comandos dentro do *script*. Logo em seguida ele inseriu o comando *Teste* no *script* e começou a colocar vários comandos dentro dele, insistindo várias vezes, sem êxito, na inclusão da condicional “rato sobrepõe à bola azul” no *Sim* e no *Não*. Mas não tentou colocar tal condicional na linha *Testar*, mesmo que a linha verde tenha aparecido várias vezes. Sendo assim, quando o *script* abaixo (figura 42) é construído, o entrevistador decide intervir, pois o estudante parece não compreender que a linha verde indica onde o comando é aceito dentro do *Teste*.

Figura 42: Primeira tentativa do estudante G



Entrevistador: Essa caixa se chama *Teste*. Você viu que os comandos não são aceitos em qualquer lugar? Você já descobriu quais são os lugares onde se pode colocar o “rato avançar” e o “Bola azul”. Só que o “o rato sobrepõe à bola azul” só cabe em uma dessas sentenças. Qual é?

Então o estudante G retirou o *Teste* de dentro do *script* e tentou colocar a condicional nele, porém ela não foi aceita.

Estudante G: É difícil, não sei.

Entrevistador: Esse comando só é válido quando ele está dentro do *script*.

Como o estudante colocou o *Teste* dentro do *script*, mas ficou parado sem saber o que fazer, então o entrevistador continuou sua intervenção.

Entrevistador: Você viu que o *Teste* tem três partes? O *Testar*, o *Sim* e o *Não*? Em qual delas o comando “o rato sobrepõe à bola azul” é aceito?

(O estudante experimenta e descobre).

Estudante G: No *Testar*.

Entrevistador: Você pode ler quais são todos os comandos disponíveis.

Sendo assim, o estudante G construiu o *Script A*, como mostra a figura 43 abaixo. Pode-se observar que ele coloca um comando em cada item do *Teste* e, parece que essa é uma das primeiras hipóteses de uso de tal comando, visto que esse tipo de construção já foi observado no projeto-piloto desta pesquisa e em diversas oficinas.

Identificando que o rato não iria se mover ao executar o *Script A* (figura 43 abaixo), então o entrevistador solicita ao estudante que o execute.

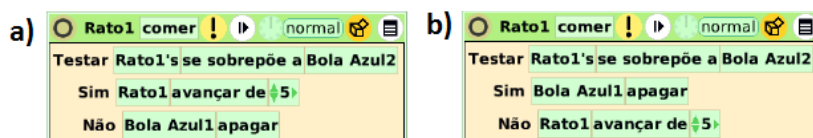
Entrevistador: O que aconteceu?
Estudante G: Ele não se mexeu.

Assim, imediatamente o adolescente inverteu os comandos do *Sim* e do *Não*, formando o *Script B* da figura 43 abaixo e o executa.

Entrevistador: O que você está fazendo?
Estudante G: Estou virando para ver se funciona.
Entrevistador: O que aconteceu?
Estudante G: Ele bateu na bola2 azul.
Entrevistador: E por que o rato parou?
Estudante G: Por que eu errei o *script*. Eu tenho que pôr a bola 2.

Neste momento é importante ressaltar que a construção do *Script B* ocorre através de uma ação combinatória, em que a posição dos comandos é invertida imediatamente, pois configura sua combinação inversa possível. Pode-se constatar que além do estudante não ter compreendido que no *Script A* o rato não avançaria – porque, segundo a condicional, só ocorreria quando o rato se sobrepôs à bola azul 2 – ao executar o *Teste* a bola azul 1 sumiria imediatamente. No entanto, tal fato não foi observado, pois abaixo da bola azul 1 havia uma “irmã gêmea” (objeto idêntico), assim tal ação não foi perceptível a ambos. Tal incompreensão conduziu à conclusão de que nada havia ocorrido.

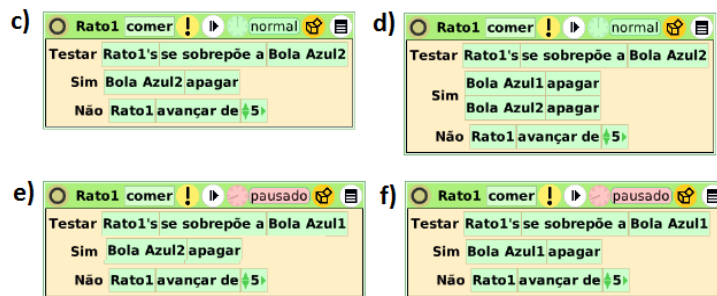
Figura 43: *Scripts A e B* que o estudante G criou para o desafio 01



Já no *Script B*, o rato avançou, pois ele não estava sobre a bola azul 2, porém pararia de avançar quando houvesse a sobreposição. Além disso, a bola azul 1 apagou, mas não foi perceptível. Entretanto, tal experiência contribuiu para que o Estudante G concluísse que “o rato

bateu na bola azul” e, ao analisar o seu *script*, percebeu a incoerência entre a bola utilizada no *Testar* e no *Sim*. Dessa maneira o adolescente realizou a modificação necessária e construiu o *Script C* da figura 44 abaixo.

Figura 44: Demais *Scripts* criados pelo Estudante G para o desafio 01



Ao executar o *Script C* o estudante observou o resultado desejado e já saiu explicando como se fazia para que a segunda bola azul desaparecesse também.

Estudante G: Daí coloca “o apagar a bola azul 1” (insere no *Sim*) e o “sobrepõe... (tenta inserir duas condicionais no *Testar*) ah.. só dá para colocar uma. (cria o *Script D*) Daí troca tudo (por fim, cria o *Script E*).

Dessa maneira, o estudante executou o *Script E* e o rato “bate” e para na bola azul 1.

Entrevistador: Por que ele (o rato) parou ali?

Estudante G: Por que eu inverti.

O estudante inseriu o comando “Bola azul1 apagar” e obteve sucesso no desafio. Como o objetivo era identificar a compreensão do estudante em relação aos componentes do comando *Teste* o entrevistador continuou a entrevista.

Entrevistador: Por que, ao executar o *Script F*, o rato não foi apagando tudo? (o entrevistado não compreende a pergunta) Nesse caso, ele (o rato) apagou a comida azul 1. Por que ele a apagou e parou de caminhar?

Estudante G: Por que clicando aqui ele anda. Clicando de novo, ele para. Sempre que clicar aqui ele anda. Você ainda pode escolher o tempo, o fazendo andar mais rápido.

O estudante desviou da pergunta e explicou como se fazia para executar um *script* em *loop*, clicando no temporizador.

Entrevistador: Tá. Então eu vou criar uma comida azul e vamos executar o *Script F*. A partir do que está escrito ali, o que tu acha que vai acontecer?

Estudante G: Ela vai sumir, a bola azul vai apagar.

Entrevistador: E o rato vai continuar a andar sempre ou vai parar?

Estudante G: Vai parar. Quando ele o encontra, para.

Entrevistador: Por quê? Saberá me dizer?

O estudante tenta executar o *script*, mas o entrevistador pede que responda antes de executar.

Estudante G: Não, isso eu não sei. Que eu saiba ele para!

Ao rodar o *Script F*, o rato comeu a comida azul e continuou avançando, para a surpresa do estudante F, que ficou sem conseguir explicar.

Toda essa sequência de *scripts* foi realizada em torno de um minuto, o que indica uma evolução rápida na composição dos *scripts*, pensa-se que são os primeiros indícios do pensamento formal, em que as composições possíveis são regidas por uma ação proposicional. Por exemplo, ao criar o *Script A* o estudante seguiu a hipótese de se colocar um comando em cada linha do *Teste*, ao observar que sua programação deu errado, construiu imediatamente o *Script B* invertendo os comandos da linha *Sim* e *Não*. Ora, tal *script* é a negação completa de *A*, logo representam operações diretamente inversas. Tal ação confere uma agilidade no pensamento que não foi observado nas ações das crianças de 10 anos.

Mesmo sem que se tenha solicitado, o estudante G decidiu criar uma programação para o rato comer as duas comidas azuis, ampliando o desafio 1 no dois. Sua primeira ideia foi colocar as duas condicionais no *Testar* e um comando específico para cada comida no *Sim* (*Script D*). Embora a invenção de utilizar dois *Testes* paralelos não tenha aparecido até então, pode-se observar que a necessidade de dois *Testes* tenha surgido espontaneamente. Além disso, o adolescente procurou criar um *script* que realizasse os dois simultaneamente, porém seu conhecimento da linguagem não lhe possibilitou a construção de dois *Testes* paralelos num mesmo *script*. Consequentemente, ele cria dois *scripts* distintos (*Script B* e *Script F*) e assíncronos. Tal fato já indica que a operação lógica da disjunção, já está presente nas suas intenções, só não sabia como fazê-la naquele momento.

8.5.1.2 Estudante H (15 anos)

Já no primeiro desafio foi fácil perceber que a estudante H estava num nível bem iniciante no *Etoys*, por exemplo, ela não se lembrava de que para fazer um objeto girar para esquerda ou andar para trás era necessário utilizar números negativos. Assim como outros estudantes, H procurou programar o rato para desviar e contornar as demais comidas até chegar na comida azul, mas desistiu depois de uma conversa com o entrevistador. Sendo assim, quando o desafio 1B foi proposto a estudante H se surpreendeu que fosse possível resolver o problema com poucos

comandos.

Sendo assim, ela inseriu o comando *Teste* no *script* e por tentativa e erro incluiu os três comandos em diferentes ordens. Nessa exploração aleatória ele descobriu que o comando “rato sobrepõe à bola azul” não era aceito na linha *Não* do *Teste*. Após uma breve manipulação ela construiu o *script* abaixo.

Figura 45: Primeiro script criado pela estudante H



Entrevistador: Quando você executar esse *script* o que espera que vai acontecer?

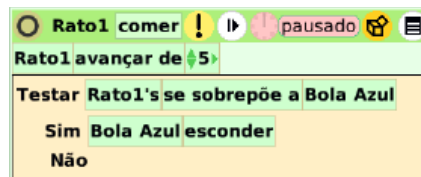
Estudante H: Primeiro ele tem que avançar, mas necessariamente eu tenho que por (os comandos) na ordem?

Entrevistador: Não sei. Que ordem você colocou?

Estudante H: Eu coloquei numa ordem aleatória, mas nesse caso ele tinha que estar na ordem, pois por mais que seja em segundos, ele (o *script*) vai executar um comando, depois outro, depois outro.

A adolescente continuou modificando o *script*, novamente ela tentou colocar a condicional no *Sim* e perguntou por que não conseguia pô-lo ali. O entrevistador não escutou e pediu para ela ler os comandos, pois tinha dúvidas se a estudante havia compreendido quais eram os comandos disponíveis. Após ler cada comando e modificá-los ininterruptamente, a estudante H monta o *script* abaixo (figura 46) e o executou, porém antes do rato chegar à primeira comida ela interrompeu a execução e disse “não vai dar”. Como era a solução do desafio e por ela ter mudado de ideia, então o entrevistador a questiona.

Figura 46: Segundo script criado pela estudante H



Entrevistador: Porque você mudou?

Estudante H: O quê?

Entrevistador: O *script*, você estava executando e depois parou.

Estudante H: Por que ele ia comer a amarela. Quer ver? Não custa nada tentar.

Ao executar o *script* o rato passa por cima da comida amarela.

Estudante H: Viu? Comeu.

Entrevistador: Passar por cima não é comer.

Enquanto isso o rato continuou andando e quando encostou na comida azul ela desapareceu.

Estudante H: Ah! (com expressão de surpresa)

Entrevistador: O que aconteceu ali?

Estudante H: Ele (o rato) comeu... Quer dizer: Ele se sobrepôs à bola azul e escondeu ela. (enquanto explicava indicava com o *mouse* os comandos no *Teste*).

Entrevistador: Esse *script*, como vimos, é a solução do desafio, mas antes de executá-lo, tu pensavas que o rato comeria a comida amarela. Então você pode me explicar como funciona aquele *script*?

Estudante H: Na verdade a minha tentativa era colocar o avançar dentro (do comando *Teste*), mas como eu não consegui eu coloquei fora. Daí eu tinha que colocar os comandos na ordem que tinha que acontecer: primeiro avançar, depois o outro.

Entrevistador: Primeiro ele (o rato) ia avançar e depois?

Estudante H: É, e depois ele ia se sobrepor sobre a bola azul e depois ela (a bola) ia se esconder.

Entrevistador: Por que a comida amarela, verde e rosa não sumiram, não foram comidas? **Estudante H:** Por que no comando (condicional) não está escrito sobrepor à bola rosa, está escrito bola azul.

Durante toda explicação a estudante indicava com o *mouse* o que ela fala.

Observa-se que a adolescente H realizou a mesma estratégia que o estudante G, conseguindo resolver o desafio ao criar a negação completa do primeiro *script* criado. Ambos apresentaram agilidade na construção deste par de inversas, em que as composições possíveis são regidas por uma ação proposicional. A partir dessas entrevistas observa-se um comportamento que diferencia a ação exploratória entre os adolescentes e as crianças de dez anos, os mais novos não exploram as diferentes possibilidades de se colocar os *Testes* e os comandos nos *scripts*.

8.5.2 Desafio 02

Da mesma maneira que as crianças, os adolescentes apresentam facilidade para resolver o desafio 02, pois basta replicar o *script* do desafio anterior. Como é a primeira vez que fazem uso do comando *Teste*, a utilização de dois *Testes* não surge imediatamente, mas é possível identificar quando irrompe a necessidade de utilizar duas condicionais. Como não sabem como fazê-lo o desafio 02B lhes é apresentado.

8.5.2.1 Estudante H (15 anos)

Dessa maneira foi apresentado à estudante H o desafio 2A, que foi compreendido rapidamente. Como sua primeira ação, ela criou o primeiro *Teste* e enquanto procurava os comandos fazendo a seguinte pergunta.

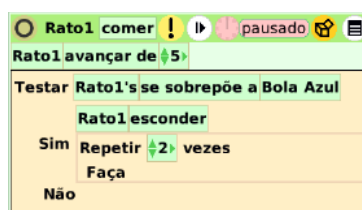
Estudante H: Eu vou ter que por dois (*Testes*) para que ele possa comer as duas?

Entrevistador: Não sei. Tu achas que precisa de dois? Por quê?

Estudante H: Tem uma forma que eu duplique o que ele faça? Entendeu?

Durante a manipulação ela encontrou o comando repetir e construiu o *script* abaixo. Nele pode-se observar que havia dois problemas: a) o rato se esconderia ao invés da comida e b) o comando repetir gera um *loop* numa sequência de comandos, mas não verifica as condicionais.

Figura 47: Primeiro script criado pela estudante H para resolver o desafio 02



Estudante H: Tá... Ele vai repetir e vai passar duas vezes pela bola azul, mas não pela outra como eu queria no caso?

Entrevistador: Sim, entendi. E como você pode resolver isso?

Estudante H: Não sei. (modifica a ordem em que coloca o comando repetir).

Nessa fala foi possível identificar que a estudante H sentia a necessidade de repetir a condicional, isto posto, sua primeira ideia era utilizar o comando repetir, porém não sabia como fazê-lo. Ademais, ela não diferenciava a ordem de inclusão entre os comandos *Teste* e repetir, pois o *Repetir* está dentro do *Teste*, contrastando com sua própria fala, uma vez que se expressava como se o comando *Teste* estivesse dentro do comando *Repetir*.

Não sabendo como intervir, o entrevistador lhe apresentou o desafio 2B. Após olhar a sugestão de comandos a estudante fala:

Estudante H: Mas se eu descobrisse uma forma que ele repetisse, mas que não repetisse a mesma bola eu ganharia bem menos tempo.

Entrevistador: Como assim?

Estudante H: Não sei como explicar.

Entrevistador: Você poderia ler quais os comandos estão apresentados? E com esses, você consegue resolver?

Estudante H: Sim, posso.

Após a leitura a estudante iniciou a construção obtendo o *Script A* da figura 48A, porém o modificou e concluiu sua construção com o *script* da figura 48B.

Figura 48: Scripts criados pela estudante H para o desafio 02B



Entrevistador: Você havia colocado o comando que tem escrito “bola azul” no primeiro *Teste* e por que, depois, você trocou?

Estudante H: Por que ele teria que voltar. Não, necessariamente, mas se ele está vindo nessa direção (vertical em direção às comidas), aproveita que já está aqui e come primeiro a verde e depois come a azul. Mas se eu colocasse primeiro a azul, eu não sei se ele comeria primeiro a azul. Entendeu?

Entrevistador: Sim, antes de executar ele (o *Script B*) você acha que ele vai fazer o que está pensando?

Estudante H: Eu acho que sim, depois que eu fiz aquele (experimento 1B), acho que sim.

Após executar o *script*, a estudante ficou surpresa quando o rato encostou na comida verde e sumiu a azul. Ela esperou o rato chegar ao local onde a bola azul estava e afirmou:

Estudante H: “Ele também não comeu a outra” (comida verde).

Entrevistador: O que aconteceu? (a adolescente fica em silêncio, sem entender) Você pode me dizer os comandos que tu colocaste ali (no *script*)?
(Durante a leitura ela percebe o *bug*.)

Estudante H: Eu coloquei o avançar, depois o sobrepõe e... Ah! (arruma o *script*) Ele fez o que eu disse, mas eu fui tola, mesmo!

A estudante H criou o *Script C* da figura 48 acima, o executou e observou o que era desejado. Sendo assim o entrevistador pergunta o que teria que ser feito para que o rato comesse todas as comidas.

Estudante H: Eu teria que fazer a mesma coisa com todas.

Entrevistador: A mesma coisa o quê?

Estudante H: Colocar mais *Testar*; e dentro colocar “sobreponha à bola rosa” e “bola rosa e assim para a amarela.

Após o *script* ficar pronto, o entrevistador retomou a questão da ordem dos comandos *Testes* no *script* e perguntou se invertesse a ordem dos *Testes* no *Script C* da figura 48 acima alteraria o resultado.

Estudante H: Acho que sim, por que... Não, necessariamente, foi só uma lógica, pois querendo ou não ele vai executar todos (com a mão aponta os *Testes*). Não sei.

Entrevistador: Vamos executar, então.

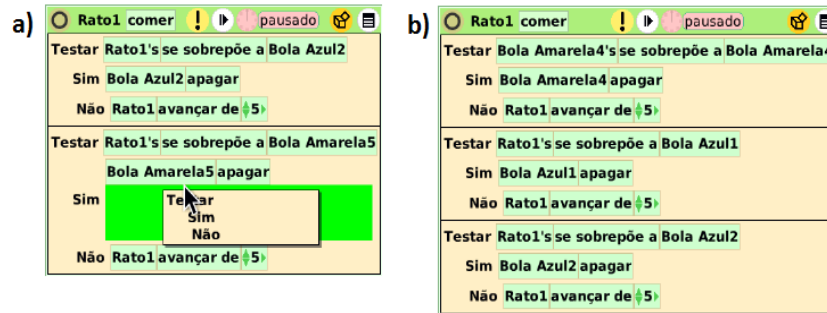
Estudante H: É... Não fez diferença.

8.5.2.2 Estudante G (16 anos)

Apesar do estudante G apresentar uma dificuldade maior na utilização dos comandos do objeto correto (por exemplo, construiu sem perceber um *script* com comandos de diferentes objetos), seus resultados foram parecidos com os da adolescente H. No entanto, ficou claro o momento em que G, espontaneamente, descobriu a possibilidade de utilizar um *Teste* dentro do outro. Isso ocorre em função de uma ação exploratória mais sistemática das diferentes possibilidades de distribuição dos comandos num *script*.

O desafio 2B foi apresentado e o estudante G colocou os dois *Testes* dentro do *script* e reproduziu a programação para o rato comer as bolas azuis 1 e 2. Por conseguinte, o entrevistador o estimulou a criar um *script* para comer também as amarelas. Como todos os comandos sugeridos já haviam sido utilizados, o estudante começa a procurá-los um a um. Quando foi inserir um novo *Teste* uma descoberta foi feita: o estudante passa o *Teste* em cada uma das linhas *Sim* e *Não* de ambos os *Testes* do *script*, até aparecer a linha verde como mostra a figura 49A abaixo. Ficou parado por um instante observando, porém o largou em paralelo com os demais e construiu o *Script B* da figura 49 abaixo. Logo em seguida executou o *script* e ficou surpreso com o resultado.

Figura 49: Primeiros scripts criados pelo estudante G para o desafio 02B



Entrevistador: O que foi que aconteceu?

Estudante G: Ele apagou a amarela antes de chegar nela.

Entrevistador: Por que ele exclui antes?

Estudante G: Por causa... Não sei.

O estudante começou a trocar a ordem dos *Testes* dentro do *script*. Neste processo colocou o terceiro *Teste* na linha *Não* do segundo. Parou e ficou observando, logo em seguida retirou um de dentro do outro e o colocou como primeiro *Teste* do *Script B*.

Entrevistador: Você pode fazer um para apagar a outra bola amarela?

O estudante inseriu um *Teste* análogo gerando o *Script C* (figura C 50 abaixo) e o executou.

Estudante G: Apagou também.

Entrevistador: Porque está apagando antes de encostar?

(Novamente o estudante não sabe o que fazer)

Entrevistador: O que está escrito lá no *Testar*?

Estudante G: “bola amarela 4 se sobrepõe à bola amarela4”.

Entrevistador: Você viu que para a bola azul está dando certo, o que tem de diferente num *Teste* e o outro.

Estudante G: Ah tá, aqui está escrito o “rato se sobrepõe” e aqui “Bola amarela se sobrepõe”. Ah tá, eu errei. Não é esse.

O estudante modificou o algoritmo até chegar ao *Script D* da figura 50 abaixo.

Figura 50: Scripts C e D criados pelo estudante G para o desafio 02B



Entrevistador: Você acabou de criar esse *script* que está aí, você pode me explicar como você o montou?

Estudante G: Eu programei, para primeiro sumir a bola amarela (aponta para o último *Teste* dentro do *script*), a primeira. Cada vez que ele anda, vai encostando em cada uma e vai sumindo. Daí eu separei para sobrepor: à bola azul 2, que é a primeira (de cima para baixo), depois à bola amarela4, que é a segunda. E daí de novo para sobrepor: à bola azul 1, que é a terceira... E por aí vai. Mas eu não sei se vai executar nessa ordem.

Entrevistador: Vamos ver se dá certo.

Ao executar o *script* D da figura 50 acima, de imediato as comidas azul e amarela sumiram (de baixo para cima) e a última começou a andar.

Estudante G: Não, deu tudo errado.

Entrevistador: O que aconteceu?

Estudante G: A bola azul 2 começou a andar. Ah errei aqui! (identifica que tem um comando “bola azul 2 avançar no *Não* e modifica)

Entrevistador: Você viu que algumas comidas sumiram, né? **Estudante G:** Sim. **Entrevistador:** Por que elas sumiram assim que começou a rodar? **Estudante G:** Eu errei o *script* esse daqui (aponta para o segundo *Teste*.)

Entrevistador: Quais bolas sumiram?

Estudante G: as duas últimas.

Entrevistador: Quais?

Estudante G: a azul 1 e a amarela 5.

Entrevistador: Olhando par o *Script*, por que elas sumiram?

Estudante G: Olhando para cá. Não sei... Ah está faltando um rato.

Neste momento o adolescente reconheceu seu o equívoco. Diante deste fato, o entrevistador sugeriu ao estudante construir o *script* por partes, primeiro colocando os comandos necessários e por último inserindo as condicionais no *Teste* adequado. Em seguida, o estudante cria o *Script* E (figura 51 abaixo).

Figura 51: Scripts E e F criados pelo estudante G para o desafio 2B



Entrevistador: A tua dúvida é que se agora você pega o comando do rato ou da bola, né?

Estudante G: Sim.

Entrevistador: Pergunto-te: Quem é que tem que reconhecer a bola?

Estudante G: O rato, então eu vou no rato e pego o comando (sobrepõe);

A partir disso o adolescente construiu todo o *Script F* (figura 51 acima).

Entrevistador: No final, quais foram as últimas modificações que você fez?

Estudante G: Eu peguei o “sobrepõe à bola” do rato, que antes eu estava pegando da bola e se tu pegar da bola ela vai sumir (quando executar o *script*).

Entrevistador: Vamos executar o *script*?

Estudante G: Sim.

Logo em seguida o *script* foi executado e o desafio, resolvido. Pode-se observar que, assim como a estudante H, a principal dificuldade do adolescente G é na utilização dos comandos do objeto correto dentro de um *script*, por exemplo: *Scripts D e E* (estudante G) e *Script B* (estudante H). Pensa-se que tal ocorrência se deve à pouca experiência com *Etoys*, pois as crianças de 10 anos não apresentavam a mesma confusão. Além disso, diante de tantos comandos, surgiu a crença de que a ordem dos *Testes* no *script* influenciaria no resultado final, porém durante a interação tal suposição foi abandonada.

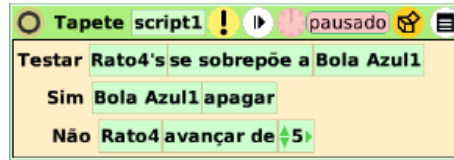
Do ponto de vista lógico, tal desafio também não apresenta alguma novidade, porém a utilização de dois *Testes* não tenha ocorrido de forma espontânea, sendo necessária a sugestão de comandos.

8.5.3 Desafio 03

8.5.3.1 Estudante G (16 anos)

Sua primeira ação foi reproduzir a programação do *Script E* da seção anterior.

Figura 52: Primeiro script criado pelo estudante G para o desafio 03



Entrevistador: A ideia do desafio é que ele coma a comida azul só quando ela estiver sobre o tapete, sendo assim, do jeito que você programou eu pergunto o seguinte: se eu colocar a comida fora do tapete o rato não vai comê-la?

Estudante G: Sim.

Entrevistador: Vai comê-la?

Estudante G: Sim, vai comê-la.

Entrevistador: Então, segundo aquele *script*, o rato vai comê-la independente dela estar ou não sobre o tapete?

O estudante ficou em silêncio por um tempo e concluiu com dúvida.

Estudante G: Se eu pegar o *Teste* do tapete ele vai funcionar.

Pôde-se observar, nesse instante, que o adolescente sentia a necessidade de realizar a programação levando em consideração o tapete, portanto selecionou um comando *Teste* do menu do objeto Tapete, formando o mesmo *script*(acima). Ao executá-lo o estudante se surpreendeu quando o rato comeu a bola fora do tapete. Ele não identificou que era uma condicional que deveria se referir a tal objeto, ao invés da caixa alaranjada do *Teste*.

Entrevistador: O que aconteceu?

Estudante G: Esse problema eu não sei.

Entrevistador: Você pode ler os comandos que estão no *script*.

Estudante G: “Rato4 se sobrepõe à bola azul1”, “bola azul1 apagar” e “Rato4 avançar de 5”.

Entrevistador: Então esse *script* se refere ao tapete?

Estudante G: Sim.

Entrevistador: Por que é em relação ao tapete?

Estudante G: Por que se a bola estivesse fora do tapete ela não comeria, agora se estivesse dentro ele iria comer.

Entrevistador: E como é que a comida azul identificaria se está sobre o tapete?

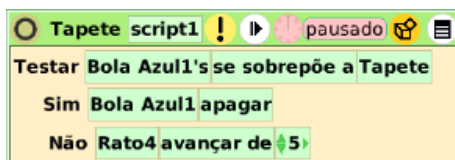
Estudante G: Pelos comandos.

Entrevistador: E você tem todos os comandos necessários ali?

Estudante G: Sim, eu tenho. É... Eu preciso de mais que se refere ao tapete, que eu não sei qual é.

Note que o estudante interpretou o que desejava programar e não o algoritmo construído. Não sabendo como intervir, o entrevistador apresentou o desafio 3B e solicitou ao o estudante a leitura de cada um dos comandos sugeridos. Após a ler, G construiu o *script* abaixo. Assim, se observou que a condicional que se referia ao tapete ganhou importância central, deixando a outra condição desnecessária.

Figura 53: Segundo script construído pelo estudante G para o desafio 03



Estudante G: Eu acho que é assim.

Entrevistador: Tu acha que é assim?

Estudante G: Sim e sobrou um comando.

Note que ele colocou a condicional “rato está sobre a bola azul” no canto inferior da tela.

Entrevistador: Então, ao rodar o *script* e a bola azul não estiver em cima do tapete ela não vai apagar?

Estudante G: Não vai.

O *script* foi executado e o rato não comeu a comida fora do tapete, assim o estudante começou a explicar.

Estudante G: Aqui está “bola azul se sobrepõe ao tapete” (aponta com o *mouse* a condicional), portanto só vai apagar quando a bola azul estiver fora do tapete.

De imediato ele paralisou o *script*, colocou a comida azul sobre o tapete e rodou novamente o *script*. Quando a bola azul sumiu ele ficou perplexo.

Estudante G: Ué!?

Entrevistador: O que aconteceu?

Estudante G: Apagou em cima do tapete.

Entrevistador: E onde estava o rato?

Estudante G: Aqui.

Entrevistador: Ela desapareceu antes do rato...

Estudante G: Chegar.

Entrevistador: Então o que tu precisa?

Resumindo o que aconteceu até então, a primeira hipótese de G foi reproduzir o *script* do desafio 01, porém, após ser questionado pelo entrevistador, sentiu a necessidade de utilizar um comando do objeto Tapete, então buscou em seu menu o comando *Teste*, mas não se preocupou com

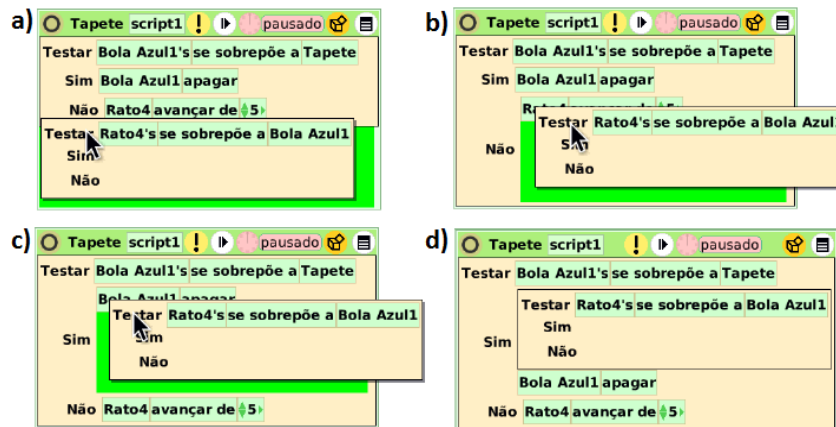
a condicional. Feito isso, ao ser solicitado que analisasse seu *script*, o estudante interpretou o que desejava programar e não o algoritmo construído.

Logo em seguida, no desafio 3B, G criou um *script* colocando um comando para cada linha do comando *Teste*, assim, recriando uma das primeiras hipóteses observadas desde a experiência-piloto a respeito do uso de *Teste*. Nessa construção o estudante G desprezou um *Teste* e a condicional “rato sobrepõe à Bola Azul”, pois considerou suficiente que seu *script* tivesse apenas um comando ou uma condicional que fizesse referência ao objeto Tapete. Porém o estudante não percebeu que neste *script*(figura 53 acima) a condicional era “a Bola Azul sobrepõe ao Tapete”, então quando ela se encontrasse nessa posição, ao executá-lo, sumiria instantaneamente, independente do rato.

Para comprovar a eficácia de seu algoritmo, G primeiro experimentou a situação em que a comida azul estava fora do tapete, como o rato não a comeu, explicou que isso se devia à condicional escolhida para o *Teste*, no entanto o estudante se surpreendeu quando essa desapareceu e não procurou revisar se sua condicional está correta. Assim ele começou a explorar novas possibilidades de arranjos entre os comandos.

Dessa maneira, o estudante tentou inserir a outra condicional junto do mesmo *Teste*, vendo que não era possível ele pegou o outro *Teste* e inseriu os dois no *script*, formando o *Script D* abaixo.

Figura 54: Estudante G explora diferentes posições para o segundo Teste



Ao procurar inserir a segunda condicional, o estudante G chegou a colocar os dois *Testes* em paralelo – reproduzindo a organização do desafio 2 – imediatamente, sem executar o *script*, percebeu tal impossibilidade e começou a explorar outras alternativas através de uma ação combinatória.

Agindo dessa maneira os adolescentes podem construir 256 arranjos distintos – desconsiderando o comando avançar – visto que há duas possibilidades para colocar o primeiro

Teste (T_1); quatro para o segundo (T_2) (levando em consideração que se pode colocá-los em paralelo); duas para a primeira condicional (C_1) uma para a segunda (C_2); quatro para o comando sobrepor (S) e quatro para o apagar (A):

$$\frac{2}{T_1} \cdot \frac{4}{T_2} \cdot \frac{2}{C_1} \cdot \frac{1}{C_2} \cdot \frac{4}{S} \cdot \frac{4}{A} = 256$$

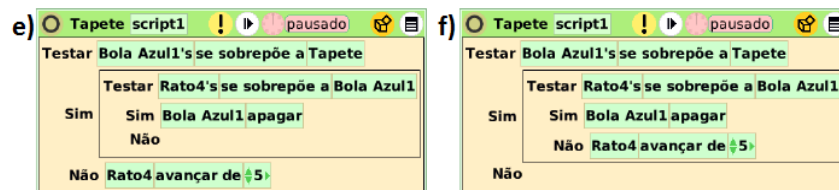
Como foi visto, G primeiro tentou colocar o *Teste* na linha *Não* do primeiro *Teste*, depois o trocou e o inseriu no *Sim*, formando o *Script D* da figura 54 acima. Após refletir por um tempo perguntou:

Estudante G: Assim não dá? (se refere ao *Script D* da figura 54 anterior).

Entrevistador: Qual a tua ideia (com o *script*)?

Estudante G: Que ele apague em cima do tapete (indica com o *mouse* a condicional que se refere ao tapete). Já sei! (constrói o *Script E* da figura 53 abaixo).

Figura 55: *Scripts E e F criados pelo estudante G para o desafio 03*



Entrevistador: Vamos rodar o *script*.

Estudante G: Sim. Na verdade eu acho que não vai funcionar.

Entrevistador: Por quê?

Estudante G: Por que esse daqui (comando avançar) tem que estar aqui.

O estudante se referiu à linha *Não* do outro *Teste* e construiu o *Script F* na figura 55 acima.

Estudante G: Só que esse aqui (condicional “bola sobrepõe ao tapete”) não devia estar dentro dessa aqui (condicional “rato sobrepõe à bola azul”). Por que eu tentei colocar os dois juntos. **Entrevistador:** Então vamos executá-lo.

Ao executar o *Script F* o rato começou a se movimentar em direção ao tapete.

Entrevistador: Pode parar. O rato já está se movimentando e essa era uma dúvida que tu tinhas né?

Estudante G: sim.

Entrevistador: Mas como vimos o rato deve comer a bola azul só quando ela estiver dentro do tapete, então eu te pergunto: Se a bola estiver fora do tapete o quê vai acontecer?

Estudante G: Ela vai sumir porque a bola vai se sobrepor à bola azul.

Ao executar o *script* para a segunda situação – bola fora do tapete – o rato ficou parado e o estudante ficou surpreso.

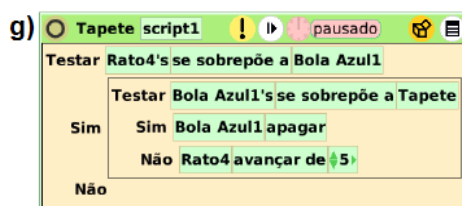
Estudante G: Não... Ela não tá andando.
Entrevistador: Por que ela não está andando?
Estudante G: Não sei.
Entrevistador: E se você colocá-la no tapete, o que acontece?

O estudante colocou a bola no tapete e o rato começou a avançar; então ele retirou o rato o que o faz parar. Sendo assim, o estudante ficou brincando com essa ação várias vezes, enquanto refletia.

Entrevistador: Vamos arrumar primeiro esse problema. O que se pode modificar? (ele fica em silêncio)
Quando é que o rato anda?
Estudante G: Quando ele está fora do tapete, então se trocar os dois (as condicionais)... Estou perdido.
Entrevistador: Então vamos testar isso que você falou.

Dessa maneira, o estudante inverteu as condicionais e executou o *Script G* da figura 56 abaixo.

Figura 56: Script G criado pelo estudante G para o desafio 03



Estudante G: Ele não anda.

O estudante modificou o *script* retornando ao *Script F* anterior (figura 56).

Entrevistador: Tem mais alguma alteração que tu possa fazer?
Estudante G: Tirando (o segundo *Teste*) do *Sim* e colocando no *Não* (*Script H* figura 56 abaixo).
Estudante G: Não andou. Mas se eu trocar de lado (inverte as condicionais)... Vamos ver. A Bola sumiu!
Entrevistador: Por que a bola sumiu?
Estudante G: Não sei. Por que eu botei no *Não* e troquei a coisa (as condicionais) pra ela sobrepor e quando o rato andar ela apaga.
Entrevistador: Como é que é?
Estudante G: Não sei.
Entrevistador: Eu não escutei, pode repetir a tua explicação?
Estudante G: Aqui (aponta para a primeira condicional) quando o rato encostar na bola, a bola some. Só que daí, quando o rato começa a andar ela some. E isso eu não estou entendendo. A não ser que eu troque de novo.

Logo em seguida, construiu o *Script I*, mas não se interessou em executá-lo. Parou, pensou e resolveu voltar a construir o *script F* da figura 56 acima.

Entrevistador: Quer tentar fazer alguma coisa com o *Script*?

Estudante G: Não. Eu só sei que se ela (a comida) estiver dentro do tapete ela vai comer e se tiver fora, não vai comer... O resto eu não sei.

Figura 57: *Scripts H e I* construídos pelo estudante G para o desafio 03



Como se pode observar o Estudante G não sabia como resolver o problema do rato não andar quando a bola estava fora do tapete. Porém a operação da conjunção das ações de comer a bola quando ela estivesse sobre o tapete foi claramente resolvido. Mesmo que o adolescente não sabia explicar como isso ocorria em função do uso de *Teste*, ele possuía essa certeza a partir de uma crescente combinatória entre as possíveis localizações dos comandos dentro dos *Testes* e no *script*.

Nessa última parte da entrevista, pode-se notar que, ao se desconsiderar o comando avançar, G apresentou a solução correta do desafio 3 no *Script E*. No entanto, o estudante G pensava que o comando avançar deveria estar dentro de um dos *Testes*, tal fato o levou ao insucesso. Isso ocorre, pois o nível de dificuldade do problema aumenta quando se leva o comando avançar em consideração, pois os arranjos possíveis quintuplicam (um fora do *Teste* e quatro dentro deles), totalizando 1280 possibilidades.

Porém pode-se identificar que o estudante G realizou suas ações a partir de operações inversas. Por exemplo, ao construir o *Script E* e identificar que o rato não avançaria quando a comida estava fora do tapete, o adolescente inseriu o comando avançar na linha do *Não* do outro *Teste* (*Script F*), criando uma ação recíproca em relação ao comando avançar. Percebendo que tal modificação não daria resultado, inverteu as condicionais (*Script G*), mas esta inversão não alterou o resultado esperado, pois é um *script* equivalente ao *Script F*. Logo em seguida, trocou o segundo *Teste* para a linha *Não*, criando o *Script H* que é o correlativo ao E, não solucionando o problema, procurou a posição recíproca do comando avançar. Entretanto, tais operações inversas não constituíram entre si um sistema de operações inversas recíprocas reunidas em sua totalidade, logo houve um insucesso. É provável que se, desde o início, G tivesse colocado o comando avançar fora do *Teste* teria tido sucesso no desafio.

8.5.3.2 Estudante H (15 anos)

Quando o experimento 3A foi proposto e a estudante apresentou o seguinte questionamento:

Estudante H: Então eu tenho que fazer a bola azul se movimentar?

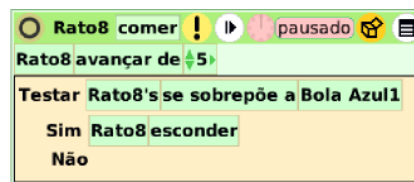
Entrevistador: Não precisa, mas se tu quiseres que ela se movimente, pode sim.

Estudante H: Mas não faz sentido. Deu para entender o que eu quis dizer? Tu disse que o rato tem que comer a bola azul, mas só quando ela estiver no quadrado (tapete). Isso quer dizer que ela vai se movimentar.

Entrevistador: Tudo bem, tu pode fazer. Mas a ideia é essa: você faz o *script* para que isso aconteça, então nós vamos testar diferentes situações e ver se isso ocorre.

Estudante H: Tá. (a estudante constrói o *script* abaixo, figura 58).

Figura 58: Primeiro Script criado pela estudante H para o desafio 03



Entrevistador: O que vai acontecer?

Estudante H: Ele vai passar por cima de todas as bolas e quando ele chegar na bola azul ele vai comer ela.

Entrevistador: E se eu tirar a bola azul do tapete ele vai comê-la ou não?

Estudante H: Vai comer igual.

Entrevistador: Ok, então você está resolvendo o desafio?

Estudante H: Não. Mas como vou fazer isso?

Entrevistador: Vamos executar o *script* para ver o que acontece. Mas vamos tirar a bola azul do tapete.

Estudante H: Mas ele vai comer igual.

Entrevistador: Mas vamos fazer eles se encontrarem.

A estudante demonstrou surpresa quando o rato sumiu ao encontrar a bola azul fora do tapete.

Estudante H: Ué? Pra onde que foi o bicho?

Entrevistador: O que aconteceu?

Estudante H: No caso, a bola azul o escondeu (o rato) hehe.

Entrevistador: Quais são os comandos que você utilizou no *script*?

Estudante H: Rato avançar, Rato se sobrepõe à bola azul e Rato... Ah! Eu devo trocar o rato para bola azul esconder.

A estudante modificou o *script* e o executa novamente para testar a situação em que a comida estivesse fora do tapete.

Entrevistador: O desafio foi resolvido?

Estudante H: Foi. Ou não?

Entrevistador: A ideia é que ele só coma...

Estudante H: Ah, só dentro do tapete, então não. Mas é só colocar ela lá dentro (risos).

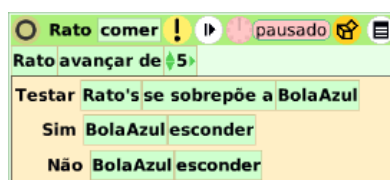
Entrevistador: O que pode ser feito?

Estudante H: Onde tem o *Testar*, Eu tenho que colocar um comando no *Não*, para ele não fazer.
Entrevistador: E que comando seria esse?
Estudante H: Eu... É que se eu colocar uma bola azul se esconder ali (linha do *Não*) ele não esconderia, mas também ele não esconderia lá (aponta para o tapete).
Entrevistador: Vamos fazer?

Nessa última fala pode-se identificar a crença da estudante que a linha *Não* negaria a ação do comando colocado em vez de negar a condicional a ser testada. Neste caso, negaria a “bola azul esconder”, portanto a comida não desapareceria. Isso gerou um conflito na estudante, que foi expresso através da fala “*mas, também, ele não se esconderia lá*”. Afinal, o mesmo *script* mandaria ela se esconder e não se esconder.

Sendo assim, o entrevistador pediu que fizesse tal modificação, criando o *script* da figura 59, abaixo. Logo em seguida, pediu-lhe que o executasse para ver o que aconteceria. Ao fazer isso, ela se surpreendeu ao constatar que a comida azul se escondia quando o *script* era rodado.

Figura 59: Segundo script construído pela estudante H

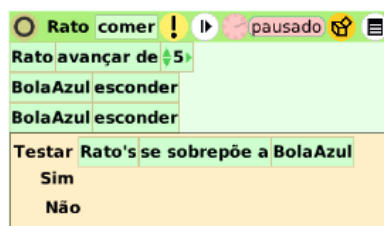


Após um tempo pensando ela compreendeu o equívoco de sua crença que o *Sim* permitia a realização da ação e o *Não*, o impedia.

Estudante H: Já sei o que foi.
Entrevistador: Então você pode me dizer como é que funciona o *script*?
Estudante H: Não é que o *Sim* faça e o *Não* “não faça”. Que eles mudem alguma coisa, tipo: ângulo, velocidade, alguma coisa do gênero. Que nem o (girar) 360°. (quer dizer qualquer comando)

Após tal fala, fica claro que sua constatação foi a negação de sua crença em relação ao *Sim* e ao *Não*, pois sua explicação não foi além disso, nem fez referência à presença da condição “se sobrepõe à bola azul”. Para investigar sua hipótese, o entrevistador propôs um novo algoritmo (figura 60 abaixo) e perguntou se isso modificava alguma coisa.

Figura 60: Script proposto pelo entrevistador à estudante H



A primeira atitude da estudante foi tentar executar o *script*, porém o entrevistador interveio.

Entrevistador: Sem executar o *script*. Muda alguma coisa se os comandos estão dentro ou não do *Testar*?

Estudante H: Acho que sim, né.

Entrevistador: ok.

Estudante H: Posso executar.

Entrevistador: Sim

Estudante H: Ué! Cadê a bola?... Eu empaquei nesse.

Não sabendo como intervir, o entrevistador propôs o desafio 3B. Como nos outros desafios o entrevistador pediu para a estudante ler os comandos apresentados. Assim, a estudante H construiu o *script* à medida que ia lendo cada comando.

Estudante H: Tem dois *Testes* (ela os coloca um paralelo ao outro), tem o (o comando) avançar, o (o comando) bola azul, tem o sobrepõe ao tapete.

Entrevistador: Quem sobrepõe ao tapete?

Estudante H: A bola azul.

Nota-se que, assim como G, a estudante H confundiu-se frequentemente em utilizar os comandos dos objetos a serem programados, que a levaram a repetir ou modificar várias vezes o algoritmo. Além disso, pode-se identificar uma primeira crença da estudante que a linha *Não* negava a ação do comando colocado em vez de negar a condicional a ser testada, porém tal crença foi abandonada nas interações seguintes, no entanto observa-se que a função da condicional não está bem clara nas falas de H.

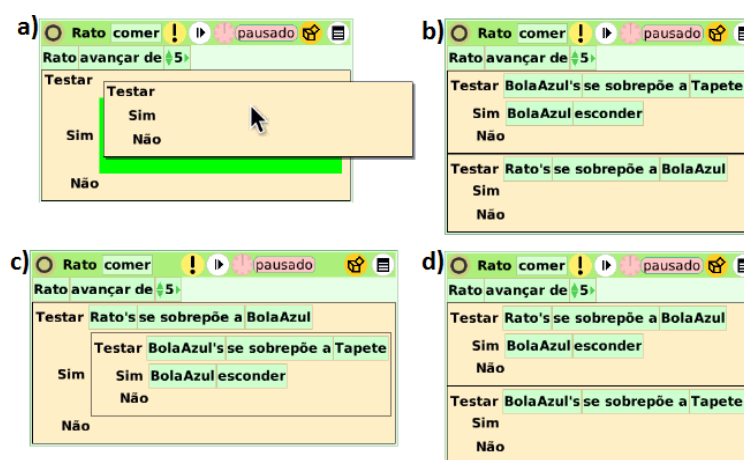
O desafio 3B lhe foi apresentado, durante as primeiras manipulações com os dois *Testes* a estudante parou de movimentá-los quando apareceu a linha verde no *Sim* do comando *Teste* (figura 61A). A adolescente ficou em silêncio olhando, mas o colocou em paralelo novamente. Pensa-se que essa ação poderia indicar o momento em que ela descobriu que é possível colocar um *Teste* dentro do outro.

Logo em seguida a estudante formou o *Script B* da figura 61, porém ela sentiu a necessidade de modificar a ordem dos comandos e construiu o *Script D* da figura 61 abaixo. Nessas

modificações pôde-se constatar que, para a estudante, a ordem dos *Testes* importava. Bem como a ação “bola azul esconde” deveria estar junto da condição “rato sobrepõe à bola azul”. Logo, pode-se afirmar que ela estava reproduzindo o algoritmo que havia dado certo para o desafio (2b) anterior, no entanto o *Teste* que apresentava a condicional “bola azul sobrepõe ao tapete” tornou-se desnecessário no algoritmo, pois não possuía comandos.

Enquanto ela realizava a modificação do *Script B* para o *D*, a estudante colocou o *Teste* na linha do sim e exclama: “Mas que droga!” e logo em seguida colocou os dois *Testes* em paralelo. Essa foi a primeira vez que a adolescente colocou um *Teste* dentro do outro, mas foi de forma acidental, sem intenção.

Figura 61: A estudante H explora os comandos sugeridos no desafio 03B



Após construir o *Script D* a estudante pediu para executá-lo e o entrevistador pergunta o que ela esperava que fosse acontecer.

Estudante H: O rato primeiro vai avançar, né. E quando ele chegar, ele sobrepõe à bola azul e ele vai esconder. No outro *Testar* sobrepõe ao tapete.

Entrevistador: Então ele está pronto? **Estudante H:** Eu acho, né. Isso quer dizer que não? (ela se refere ao fato de entrevistador ter feito tal pergunta).

Entrevistador: Lembrando o objetivo do desafio.

Estudante H: Ele come a bola azul só quando ela estiver fora do tapete. **Entrevistador:** Quando ela está em cima do tapete.

A estudante ficou pensativa e o modificou como mostra o *Script E* (figura 62).

Estudante H: Por que antes (no desafio 2) agente trocou a ordem e não fez diferença, ele executou os comandos como deveria ter executado. Mas acho que aqui vai fazer diferença se eu colocar o “Sobrepõe ao tapete” primeiro.

O *script* foi executado para as duas situações em que a comida azul estava fora e dentro do tapete e a estudante. Ela se admirou quando o rato comeu a comida fora do tapete.

Estudante H: Eu teria que colocar um comando que não coma fora.

Vendo que a estudante não sabia como fazer, então o entrevistador resolveu dizer que esses são os comandos necessários para resolver o desafio.

Entrevistador: Esses são os comandos necessários para...

Estudante H: Concluir todo desafio? Jesus! (ela fica pensativa e em silêncio). Ele não pode comer fora do tapete.

Sendo assim ela pegou o *Teste* que tem a condicional “bola azul sobrepõe ao tapete” e resolveu modificar a ordem, no entanto ela colocou um *Teste* dentro do outro, pois durante essa manipulação a linha verde apareceu no *Não* do *Teste* que tinha a condicional “o rato sobrepõe à bola azul”, como indica o *Script F* na figura 62 abaixo. Após isso ela ficou parada, olhando para o *script* e pensando em silêncio.

Entrevistador: No que você está pensando?

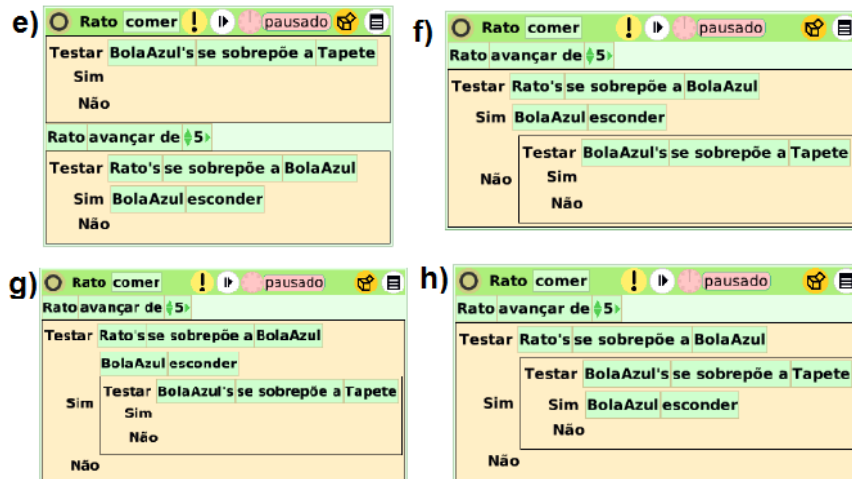
Estudante H: Em que lugar eu teria que colocar para que ele não coma fora.

Entrevistador: Que lugar tu tem que colocar o quê?

Estudante H: O outro *Testar* com o comando “sobreponha ao tapete” para quando ela (a bola) estiver fora e ele (o rato) não coma a bola. Só que eu não sei. Vou testar esse *script*(*Script F*).

Ela executou o algoritmo e o rato comeu a bola, então a estudante tirou um *Teste* de dentro do outro e colocou fora do *script*. Então a estudante refletiu mais um pouco e colocou novamente um *Teste* dentro do outro, só que dessa vez na linha *Sim* como mostra o *Script G* da figura 62 abaixo.

Figura 62: Estudante H estuda a possibilidade de colocar um Teste dentro do outro



Logo em seguida ela mesma *arrastou* a comida para fora do tapete para fazer um *Teste*, porém o entrevistador interveio.

Entrevistador: Qual foi o problema
Estudante H: Ele comeu!
Entrevistador: Não executa. O que você espera que vai acontecer com esse *script* aí?
Estudante H: Eu espero que ele coma a bola no tapete, mas não fora.

Assim sendo, a adolescente executou o *Script G*, mas o rato comeu a bola fora do tapete.

Estudante H: Aí, Jesus!
Entrevistador: Você pode me explicar como os comandos estão dispostos dentro do *script*?
Estudante H: Avançar fora do *Testar*, dentro do *Testar* tem “sobrepõe à bola azul” no *Sim* tem outro *Testar* que dentro tem um “sobrepõe ao tapete”.
Entrevistador: Tu podes me explicar como funciona o primeiro *Testar*?
Estudante H: Como assim?
Entrevistador: Você está vendo que tem dois *Testar*, né. No primeiro *Testar* tu disse assim: “rato sobrepõe à bola azul, se *Sim*...”
Estudante H: Se *Sim* tem o “bola azul” fora e dentro tem outro *Testar*
Entrevistador: Mas o que está sendo testado nesse primeiro *Teste*?
Estudante H: O rato sobrepõe à bola azul.
Entrevistador: E se *Sim* o que ele tem que fazer?
Estudante H: Esconder a bola azul.
Entrevistador: E se *Não*?
Estudante H: Não tem nada.
Entrevistador: E no segundo *Teste*, o que está sendo testado?
Estudante H: A bola azul sobrepõe o tapete.
Entrevistador: E se isso acontecer, o que tem ou não que acontecer?
Estudante H: Como assim?
Entrevistador: No segundo *Teste*, tem algo a ser testado e se *Sim* e se *Não*. No primeiro *Teste* o que está sendo testado?
Estudante H: Esconder a bola azul.
Entrevistador: No segundo *Teste*, o que está sendo testado?
Estudante H: Sobrepõe o tapete... Depois ele tem que esconder, no segundo. Ou não?
Entrevistador: E nesse segundo *Teste* tem algum comando dito, caso ocorra ou não?

Estudante H: Não. Por isso não funciona?

Entrevistador: Ele está funcionando?

Estudante H: Não

Entrevistador: E como você pode modificar?

Estudante H: Eu posso colocar um comando dentro do *Testar* para que ele possa executar. Só que qual (o comando) eu não sei.

Dessa maneira ela refletiu um pouco e arrastou o comando “bola azul esconder” para o *Sim* do segundo *Teste* como mostra o *Script H* da figura 62 acima e afirmando “vou testar”. Portanto, executou o *script* para a comida fora e dentro do tapete e comemorou seu sucesso em cada situação.

Observa-se na evolução dos *scripts* da estudante H, que a partir de uma ação combinatória ela descobriu que poderia inserir um *Teste* dentro do outro (*Script A e C*), no entanto, de início os colocou em paralelo formando o (*Script E*), ela acreditava que a ordem fazia diferença. Como ela não obtém sucesso, ao constatar que tal hipótese não era verdadeira, admitiu colocar um *Teste* dentro do outro como uma nova possibilidade a ser experimentada. Primeiro incluiu o segundo *Teste* na linha *Não* do primeiro, pois desejava que a comida azul não se escondesse quando estivesse sobre o tapete, formou (assim) o *Script F*. Note que com essa ação ela procurou negar a segunda condicional, tal crença lhe impôs uma dificuldade para tomar consciência que a ausência de comandos no segundo *Teste* o tornava ineficaz e equivalente ao *script* do desafio 1.

Com o fracasso dessa tentativa, H realizou uma espécie de operação correlativa ao *Script F* (em relação ao comando *Teste*), colocando o segundo *Teste* na linha do *Sim*, (*Script G*). Porém, como o segundo *Teste* continuava vazio, o tornando equivalente ao *Script F*. Somente ao ser questionada sobre o quê estava sendo testado no segundo *Teste* e o que aconteceria se fosse verdadeiro ou falsa tal condicional, é que a estudante H percebeu que era necessário inserir um comando no segundo *Teste*. Por conseguinte, ela solucionou tal desafio ao construir o *Script H* colocando o comando “Bola azul esconder” na linha *Sim* do segundo *Teste*.

Ao comparar os *scripts* produzidos pelos dois adolescentes, supõe-se que o fato da estudante H colocar o comando avançar fora do *Teste* facilitou a resolução do problema, em função disso ela obteve a solução sem utilizar muitas operações inversas. Já o adolescente G realizou mais operações inversas, mas não conseguiu perceber que havia “resolvido” o problema se desconsiderasse o comando avançar. De qualquer maneira, ambos não conseguiram formar um sistema de operações inversas e recíprocas, indicando o motivo pelo qual eles não conseguiram resolver os desafios de forma espontânea, apenas com o auxílio das intervenções do entrevistador.

Além disso, pode-se constatar várias semelhanças entre as duas entrevistas, a primeira ação foi a de reproduzir o *script* do desafio 01. Quando apresentados ao desafio 3B, com as sugestões de

utilizar dois *Testes*, ambos cogitaram colocá-los em paralelo. Aumentando, assim, a suspeita do autor da pesquisa de que os estudantes resolveram o desafio 2 com a crença de que dois *Testes* em paralelo correspondiam a uma conjunção de condicionais. Ademais, apresentaram como primeira hipótese para a linha *Sim* e *Não* como a confirmação (ou não) dos comandos presentes nelas, ao invés da condicional em si, porém ambos a abandonaram durante a entrevista.

Outro fato que se observa é que ambos não conseguiram resolver os desafios de forma espontânea, apenas com o auxílio das intervenções do entrevistador. No entanto, constata-se que os adolescentes, quando não sabem o que fazer, começam a experimentar diferentes combinações possíveis entre os comandos e entre os *Testes* e, assim, chegam a experimentar diferentes e novas possibilidades de arranjos e conseguindo apresentar a solução do desafio (mesmo sendo inconsciente no caso de G). Foi este comportamento presente nas ações dos adolescentes que conduziram esta pesquisa a investigar se as crianças mais novas também agiriam dessa forma. Nesse momento da pesquisa, não se sabia que, segundo a teoria piagetiana, esta ação combinatória é justamente o que diferenciava o comportamento dos adolescentes e o das crianças. Além do mais, é exatamente tal ação combinatória que permite aos estudantes descobrirem espontaneamente que é possível colocar um *Teste* dentro do outro, enquanto que as crianças não conseguiram.

Por último, a maioria das crianças de 10 anos entrevistadas não se confundem com os objetos envolvidos na programação, no entanto tal confusão é presente nos adolescentes, pensa-se que tal fato ocorre, pois os primeiros têm mais experiência com o *Etoys* que os segundos.

8.6 NÍVEL III B

Pelas entrevistas realizadas pode-se constituir uma segunda, e última, fase do nível III, na qual os adolescentes conseguem resolver e explicar a lógica inerente à condicional, presumindo-se que tal fato ocorre em função da constituição contínua de um sistema de operações inversas e recíprocas reunidas em sua totalidade.

8.6.1 Entrevista com o Estudante F (14 anos)

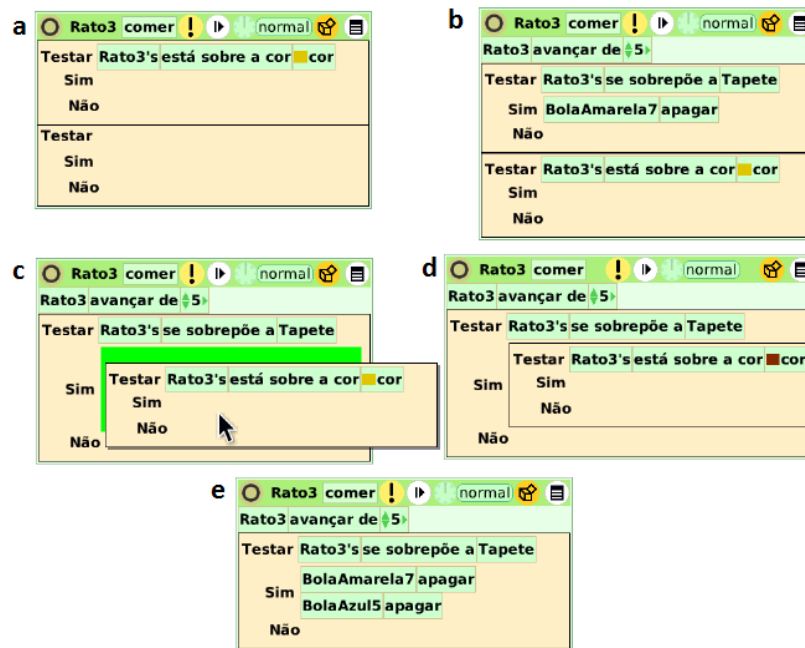
Estudante F: O rato tem comer só as bolas que estão em cima do tapete?

Entrevistador: Sim. (parece que o entrevistador não prestou atenção na pergunta feita).

Estudante F: Quando o rato está sobre a cor... (monta o *Script A* da figura 63) não. Quando o rato está sobre o tapete a bola amarela apaga (*Script B*).

Nos dois primeiros *scripts* ficou visível que a primeira ação do estudante F é assimiladora, ou seja, procurou aplicar uma sequência de comandos que já haviam sido utilizados com sucesso anteriormente, portanto colocou os *Testes* em paralelo. Assim, novamente, aparece a crença de que dois *Testes* em paralelo correspondiam a uma conjunção de condicionais. No entanto, F percebeu que a informação relevante é “as comidas que estão no tapete”, então modificou a ordem da condicional numa tentativa de alterar o resultado.

Figura 63: Primeira sequência de script criada pelo estudante F para o desafio 03



Durante o manuseio do comando *Teste*, o adolescente começou a utilizar uma ação combinatória e vislumbra a possibilidade de colocar um *Teste* dentro do outro (*Script C*). Com a fluência que o estudante executava as ações, o entrevistador ficou na dúvida se aquela fora uma descoberta espontânea e aleatória ou se ele já conhecia tal possibilidade e ainda não havia pensado em utilizá-la. De qualquer maneira, o estudante F explorou essa possibilidade criando o *Script D* modificando a cor da comida para a do tapete. No entanto, após uma breve reflexão, desistiu e construiu o *Script E* o executando para ver se “deva certo”.

Ao “rodar” a animação as comidas somem quando o rato “encosta” no tapete.

Estudante F: Assim!?

Entrevistador: O rato chegou na comida?

Estudante F: Não. Então eu devo usar o comando... Quando o rato vê a cor...

Ao ficar sabendo que poderia utilizar outros comandos, o estudante F cria o *Script F*

mostrado na figura 64 abaixo e o executou, porém observa o rato comer só a comida azul.

Estudante F: Deu certo, mas ele comeu só a comida azul.
Entrevistador: Por quê?
Estudante F: Ah! Já vi meu erro (e constrói o *Script G*).

Logo em seguida, o estudante executou o *script* e, ao sumir as comidas azul e amarela que estão em cima do tapete, se deu por satisfeito.

Figura 64: *Scripts F, G e i* criados pelo estudante G para o desafio 03



Porém, ele afirma que não conseguiu utilizar o comando “o rato está sobre o tapete” e perguntou se tinha que utilizá-lo. Nesse momento ele refletiu por um momento e falou:

Estudante F: Mas era pra eu utilizar ele... Tá: O rato está sobre o tapete daí ele comeria a bola que estaria dentro do tapete ou algo do tipo. Deu certo, mas ele comeu só a comida azul.
Entrevistador: Nesse caso (o *Script G* da figura 64 acima), se o rato tivesse encontrado a comida fora do tapete, o que aconteceria?
Estudante F: Ela ia sumir.
Entrevistador: O tapete (a condição) não foi utilizado.
Estudante F: Ah, agora entendi. Criaria um *Testar*, quando o rato estiver sobre o tapete e daí colocaria o *Testar* dentro dele, do próprio *Testar*. Acho que assim funcionaria (faz o *Script I*).
Entrevistador: Tu poderia me explicar aquela programação, como funciona aquele *script*?
Estudante F: Eu coloquei que quando o rato está sobre o tapete, ele vai fazer – o que é o *Sim* – ele vai fazer esses dois *Testes*, que quando o rato estiver sobre a bola amarela ela vai apagar e quando estiver sobre a bola azul, ela vai apagar. Se ele não estiver sobre o tapete, não aconteceria isso. **Entrevistador:** E tu já tinhas feito alguma programação onde se pode colocar um *Teste* dentro do outro?

Estudante G: Já, mas eu não me lembrava. Mas já fiz.

Por desatenção do entrevistador, o estudante F entendeu de forma equivocada a ordem do desafio 3, tal incidente prejudicou momentaneamente a observação dos objetivos desta pesquisa, pois ele pensou que devia programar o rato para comer apenas as comidas que estão sobre o tapete. Com isso a crença de que a condicional que fazia referência ao tapete é mais importante e, assim, determinou a construção de uma sequência de *scripts*, nos quais era possível observar tal intenção: (a) na troca da ordem dos *Testes* no *Script A* para *B*; (b) na mudança de cor amarela da comida para a marrom do tapete no *Script C* para *D* e (c) na construção do *Script E*, elegendo tal condicional como a única necessária.

Por outro lado, o adolescente percebeu seu erro ao executar tal algoritmo, como consequência reproduziu a solução do desafio 2 (*Script G* que não faz uso da condicional do tapete) e, ao executá-lo, pareceu satisfeito com o resultado. Somente ao ser questionado se o rato comeria a comida fora do tapete é que ele compreendeu a ordem do desafio 3 e, de imediato, falou a programação correta e construiu o *Script I*. Em seguida explica corretamente a lógica da condicional.

Logo, é importante ressaltar, que não é o fato de ele ter construído o algoritmo corretamente que garante a compreensão da lógica, mas a destreza ao explicar a lógica subjacente que garante. Afinal, foram apresentados alguns exemplos em que os sujeitos construíam o algoritmo correto, mas não conseguiam explicar como o computador iria “ler” tais procedimentos ou davam uma explicação deturpada. Logo, para compreender como um computador funciona é importante que tal atividade de programação seja amplamente difundida nas escolas.

A partir de tais fatos parece inevitável concluir que as ações do estudante F foram regidas por uma estrutura formal, cujas operações inversas já compõem um sistema reversível na sua totalidade, o INRC. Embora, na sua entrevista, não se observou diretamente diferentes *scripts* com operações inversas entre si, quando se analisa os dados de todos os sujeitos envolvidos na pesquisa sob uma perspectiva piagetiana do desenvolvimento cognitivo, pode-se chegar a tal afirmação. Uma vez que as observações das ações dos sujeitos do nível anterior eram compostas por operações inversas, mas que não compunham um sistema que reunisse os inversos e recíprocos e, conseqüentemente, esses sujeitos não conseguiram explicar satisfatoriamente a lógica do *Teste*, apresentando maior dificuldade na produção de *scripts* para o desafio 3.

Além disso, o que também corrobora para a conclusão de que F está num estágio mais avançado de desenvolvimento cognitivo (em relação à compreensão da lógica) é que ele resolveu o desafio construindo um *script* mais complexo que o exigido (para comer dois tipos de comida).

Mesmo assim, na sua explicação todos os elementos do algoritmo são relatados: “Eu coloquei que: quando o rato está sobre o tapete, ele vai fazer – o que é o *Sim* – ele vai fazer esses dois *Testes*, quando o rato estiver sobre a bola amarela, ela vai apagar e quando estiver sobre a bola azul, ela vai apagar. Se ele não estiver sobre o tapete, não aconteceria isso”. Logo, ele sabe calcular a verdade ou a falsidade dos conjuntos a partir de p , q e t , pois pode considerar todas duplas possíveis entre p , $\sim p$, q , $\sim q$, t e $\sim t$.

Diante de tais constatações, julga-se que há informações suficientes para abordar e discutir o problema de pesquisa apresentada na introdução desta tese.

8.7 O DESENVOLVIMENTO DO RACIOCÍNIO CONDICIONAL QUANDO SE APRENDE A PROGRAMAR

Retomando o problema de pesquisa proposto nesta tese: *Como ocorre o desenvolvimento do raciocínio condicional dos estudantes quando se aprende a programar através do uso de Teste do Squeak Etoys numa perspectiva construtivista de aprendizagem?*

De início retomam-se algumas questões apresentadas na introdução deste trabalho. Foi visto que há uma carência de pesquisas que analise o desenvolvimento da lógica nas crianças quando estão aprendendo a programar. Geralmente, as pesquisas voltam-se para a análise dos projetos desenvolvidos pelos estudantes sem explorar a aprendizagem da lógica (Silva, 2008; Marques, 2013; Gallardo, 2014; etc.) ou dos conceitos de diferentes disciplinas (Dalla Vecchia, 2012; Capela, 2013; Quintanilla Córdor, 2012; Silva, 2009; Pinto, 2010; etc.), tal como foi realizado na experiência-piloto desta pesquisa.

Uma das primeiras questões apresentadas foi se todas as crianças que participaram da pesquisa conseguiam utilizar as operações lógicas corretamente e se elas compreendiam-nas da mesma forma. Os dados coletados e analisados evidenciam notoriamente que não. Ou seja, a capacidade de utilizar as operações lógicas e a forma como elas compreendem se modifica ao longo do desenvolvimento cognitivo. A partir desse ponto, esta pesquisa procurou compreender como ocorre a aprendizagem das operações lógicas numa perspectiva genética piagetiana. Dessa maneira, os estudos de lógica piagetiana foram determinantes para a compreensão dos dados que dizem respeito sobre a utilização do raciocínio condicional para resolver os desafios.

Como se viu, Piaget indicava que havia uma lógica nas ações que permitia às crianças organizar e estruturar a experiência vivida. Tal capacidade emerge da atividade de estruturas mentais que são isomorfas às estruturas da álgebra matemática. Tendo como referência os sujeitos que participaram deste estudo, as estruturas mentais investigadas dizem respeito ao pensamento

operatório concreto e formal.

A primeira delas, a estrutura de agrupamento, que torna a reversibilidade operatória possível. No entanto, a criança só realiza tal reversibilidade apoiando seu pensamento numa primeira manipulação concreta do objeto. Isso se deve ao fato de que num agrupamento as composições são constituídas pouco a pouco através da proximidade entre as classes, mas sem atingir uma combinatória entre elas. A dificuldade com a inclusão de classes é, na verdade, uma dificuldade em compreender a implicação $p \rightarrow q$, ou melhor, as subclasses que compõem tal operação.

Sendo assim, a partir das entrevistas definiu-se como **Nível I** àquelas crianças que apresentaram dificuldade em constituir classes em relação às proposições apresentadas. No contexto, do desafio isso significava apresentar uma indiferenciação com relação aos elementos do comando *Teste*, quais comandos são condicionais ou ações que servem para ser colocadas no *Sim* e no *Não*. Como se pode observar na tabela 06, a estudante L apresentava uma necessidade grande de recorrer ao real nessa situação.

Por conseguinte, o Nível II foi caracterizado pelas crianças que resolveram pelo menos o desafio 01, pois tal fato indicaria a constituição das classes formadas por relações intraproposicionais. Além disso, foi necessário criar um subestádio A, quando houve insucesso no desafio 02, visto que tal fato indica que a composição de diferentes classes (obtendo-se assim uma disjunção) é um obstáculo intransponível. No entanto, durante a análise dos dados foi necessário criar um subnível para aqueles estudantes que apresentaram sucesso no primeiro, porém insucesso no segundo desafio. Assim, o estudante M foi o sujeito que representa o **Nível II A**, pois não conseguiu realizar a operação de disjunção através do *Etoys* no desafio 02.

Já os estudantes (K, N e J) que tiveram sucesso nos dois primeiros desafios, mas não conseguiram explicar a lógica do *Teste* foram classificados num **Nível II B**. Ao analisar tal feito, do ponto de vista da abstração reflexionante, indica que tal sucesso é alcançado via *reflexionamento* do *script* do desafio anterior para o desafio 02, pois não sabiam explicar a lógica presente em tais algoritmos. Em função desse fato, seus *scripts* revelam que tal abstração é realizada através de uma reprodução quase que estática da sequência de comandos do *script* do desafio anterior, mas sem compreensão da implicação envolvida.

Tal fato ação evidente no desafio 03, numa primeira tentativa, os estudantes criam *scripts* com Testes paralelos, o que indica que ele têm a crença de que tal programação corresponde a uma conjunção de condicionais. Assim, novamente, realizam o *reflexionamento* e que foi citado no parágrafo anterior. Porém diante do insucesso, não conseguiram realizar a *reflexão* necessária para estabelecerem a abstração reflexionante e compreender todas as classes e subclasses envolvidas na disjunção e como se diferencia da conjunção.

Analisando tal questão do ponto de vista proposicional, a implicação presente no desafio 01, o obstáculo cognitivo encontrado pelas crianças do estágio operatório concreto é identificar que a implicação $p \rightarrow q$ é composta pelas subclasses (p, q) ou $(p, \sim q)$ ou $(\sim p, \sim q)$. No entanto, pensa-se que tal construção é alcançada à medida que suas negações também são compreendidas: ou seja, $A' = (\sim p, q)$ ou $(\sim p, \sim q)$ e $B' = (p, \sim q)$ ou $(\sim p, \sim q)$. Viu-se que, segundo Piaget, tanto a implicação (raciocínio condicional), quanto a necessidade de se recorrer ao real só será superada no estágio seguinte, quando a criança torna-se capaz de realizar operações sobre proposições lógicas que levam à compreensão da própria tautologia.

Sendo assim, denominou-se como Nível III os sujeitos, cuja ação combinatória conduziu ao estabelecimento de relações proposicionais durante a resolução do desafio 03. Tal fato pode ser observado em todos os adolescentes entrevistados e é caracterizado a partir da atividade de estruturas formais do pensamento que são representadas pelo INRC. Num primeiro momento, foi identificado um subestádio A composto pelos sujeitos que obtiveram a solução do desafio 03, mas explicar parcialmente (ou não) a lógica presente do raciocínio condicional. São os casos de H e G, quando eles não sabiam o que fazer, ambos começaram a experimentar diferentes combinações possíveis entre os comandos e entre os *Testes* e, assim, chegaram a experimentar diferentes e novas possibilidades de arranjos ao ponto de apresentar a solução do desafio. Além do mais, é exatamente tal ação combinatória que permitiu aos estudantes descobrirem espontaneamente que era possível colocar um *Teste* dentro do outro e, assim, superar a crença de Testes paralelos corresponde a uma conjunção de condicionais. Enquanto que as crianças mais novas não conseguiram.

Além disso, durante a construção dos *scripts*, foi possível constatar que tal ação era regida por operações inversas, porém eles não conseguiram reunir num mesmo sistema os inverso e os recíprocos. Logo, não eram capazes de explicar completamente a lógica presente no raciocínio condicional, pois não formaram todas as duplas possíveis entre $p, \sim p, q, \sim q, t$ e $\sim t$. Com isso os sujeitos desse nível, apresentaram uma compreensão parcial do algoritmo. Dessa forma, a crença de que a linha *Sim* ou *Não* confirmava o comando inserido (ou o objetivo geral do desafio) e não a condicional persistiu desde o operatório concreto até o nível III A.

Por último, no **Nível III B** foi constituído pelo sujeito que conseguiu resolver e explicar a lógica inerente à condicional nos três desafios. Logo, esse adolescente pode reunir num mesmo sistema os inversos e os recíprocos, sabendo calcular a verdade ou a falsidade dos conjuntos a partir de p, q e t . Podendo, assim, considerar todas duplas possíveis entre $p, \sim p, q, \sim q, t$ e $\sim t$.

Para finalizar, segundo os dados e as análises presentes nesta tese, o desenvolvimento do raciocínio condicional a partir do uso do comando *Teste* no *Etoys*, ratificam a afirmação de Piaget

(1976) de que a implicação entre proposições caracteriza todos os casos do pensamento formal, como consequência, o raciocínio hipotético-dedutivo do adolescente. Dessa maneira, a presente tese contribui para a afirmação de que a teoria piagetiana é atual e necessária para a compreensão da aprendizagem da lógica no desenvolvimento do pensamento computacional.

8.8 REPENSANDO O EXPERIMENTO

Algumas preocupações estiveram presentes durante a aplicação do experimento e na análise dos dados. A primeira a destacar é que, em função da escassez de grupos ou escolas que trabalhassem com *Etoys*, não se conseguiu entrevistar sujeitos que contemplassem todas as idades entre 10 e 16 anos. Porém, apesar do conjunto de entrevistados não contemplar todas as idades, o grupo constituído foi emblemático, visto que suas ações evidenciaram as diferenças entre o pensamento da criança e do adolescente. Por outro lado, temos consciência que, se tivéssemos entrevistados crianças com onze, doze e treze anos de idades, (talvez) teríamos dados que indicassem melhor como é a transição entre o operatório concreto para o formal. Logo, torna-se relevante que este estudo continue a ser realizado.

A segunda preocupação diz respeito ao método de entrevista clínica piagetiana, sendo mais específico, à intervenção dos entrevistadores. Em relação às entrevistas com as crianças de dez anos, durante a aplicação do experimento ficava claro que, ao chegar num determinado ponto, elas não conseguiam mais resolver o desafio (como é o caso da Estudante M no Desafio 02), sendo assim o entrevistador interveio com o objetivo de observar se tal informação seria relevante (ou não) para elas. Já em outras entrevistas, ficava explícito que as crianças haviam chegado ao seu limite e não teria como avançar (como é o caso da Estudante L no Desafio 01 e o exemplo da atividade intitulada perseguição, dos estudantes J e N no desafio 03), assim o entrevistador aproveitava a oportunidade numa perspectiva pedagógica, ou seja, com o objetivo de promover a aprendizagem de tal ferramenta. Dessa maneira, pode-se concluir que houve indução; sendo assim, foi questionado se os dados deveriam ser desconsiderados. Pensa-se que, não, pois após a análise das entrevistas e dessas intervenções indutivas, fica nítido o fato de que as crianças dessa idade apresentavam dificuldades em compreender e explicar a lógica do comando *Teste* e que o último desafio estava além de suas possibilidades.

Já com os adolescentes, também foi observado que espontaneamente nenhum deles resolveria o problema, no entanto as intervenções “você pode explicar como funciona o *script*” ou “você pode ler os comandos utilizados no *Teste*”, foram importantes para que os adolescentes

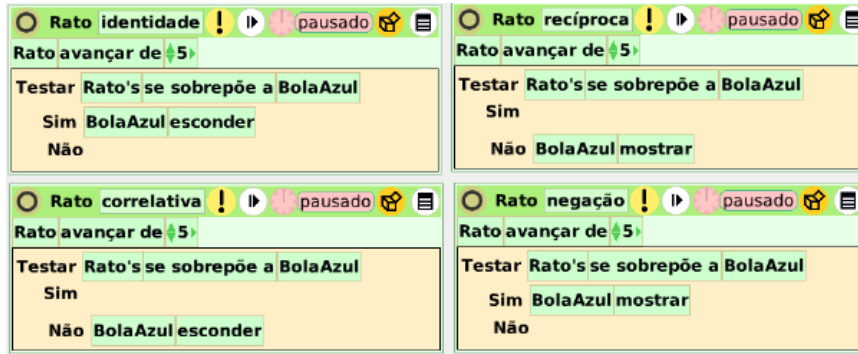
refletissem sobre seus próprios algoritmos, enquanto eles iam lendo e explicando. No entanto, teve-se a impressão de que os três adolescentes já supunham algum erro ou que havia algum problema na programação, quando tal pergunta era feita. Dessa maneira, tal fato abre margem para a afirmação de que houve indução, pois não foi uma ação espontânea. Porém, é importante ressaltar, que no método clínico piagetiano o entrevistador também tem suas hipóteses e, nesse sentido, pode intervir com o intuito de comprová-la ou refutá-la. Tais intervenções foram realizadas com tal objetivo, de fazer o sujeito refletir sobre suas ações, mas não para indicar a solução. Sendo assim, pensa-se que os *scripts*, a ação combinatória e as operações inversas foram construções genuínas dos sujeitos, não sendo influenciadas pelos entrevistadores, mas fruto do seu pensar sobre o próprio pensar.

A terceira dúvida versava sobre as fases B's dos desafios: será que apresentar (apenas) os comandos da solução não induziriam a uma ação combinatória de comandos sem reflexão por parte dos entrevistados? Logo, tal comportamento seria uma consequência de uma proposta equivocada de experimento? Porém, nas entrevistas com as crianças mais novas havia indícios que isso não era verdade; no entanto, foi somente após o estudo da lógica piagetiana que tal hipótese confirmou-se falsa. Afinal, estava-se constatando justamente a ação combinatória que expressa a diferença entre o pensamento da criança e do adolescente.

Por outro lado, durante a análise dos dados, apesar de observar as operações inversas e correlativas a partir de uma proposta de *script*, não foram constatados *scripts* que representassem as quatro operações que compõem o INRC em que a solução do desafio fosse a operação de identidade. Com isso procurou-se desenvolver *scripts* que traduzissem tais operações (figura 65 abaixo).

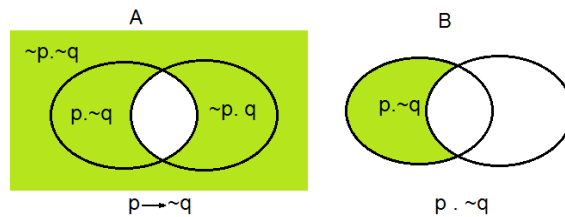
Pode-se criar a recíproca através das implicações $\sim p \rightarrow \sim q$ ou $q \rightarrow p$, como no *Etoys q: Bola Azul esconder*, não pode ser condicional, então optou-se por utilizar $\sim p \rightarrow \sim q$. Para tal, basta negar a condicional “Rato se sobrepõe a Bola Azul” inserindo o comando da negação de q ($\sim q$): “Bola azul mostrar” na linha *Não* do *script* recíproca (figura 65 abaixo). Dessa maneira torna-se visível que faltou a sugestão do comando “Bola Azul mostrar” que é a negação do comando esconder no desafio 1B, visto que a negação da condicional “sobrepõe a Bola Azul” já estava contemplada através das linhas *Sim* e *Não* do *Teste*. A ausência do comando “mostrar” inviabilizou nas entrevistas a produção do *script* que exemplificaria a negação completa e a recíproca, explicando o porquê de que as crianças e adolescentes não o utilizaram.

Figura 65: Scripts que representam o INRC no desafio 01



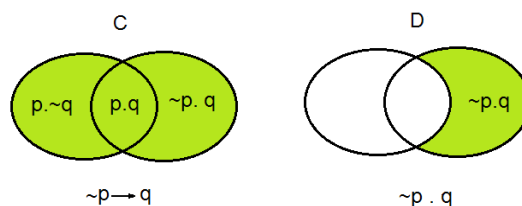
Além disso, viu-se que a negação ($p \rightarrow q$) é $p \cdot \sim q$; no entanto em programação a operação da conjunção não ocorre na sua forma “pura”, mas através de uma condicional, ou seja, via implicação $p \rightarrow \sim q$ (figura 65). Logo o *script* não representa exatamente a negação da identidade como mostra a figura 66A abaixo, mas uma operação de incompatibilidade. Porém, do ponto de vista prático do desafio, a incompatibilidade manterá o que é essencial na negação de $p \rightarrow q$, no qual $p \cdot \sim q$ é verdadeira nessa implicação, ou seja, q é falsa, quando p é verdadeira.

Figura 66: Diagramas de Venn da Negação e da Incompatibilidade gerada pelo script



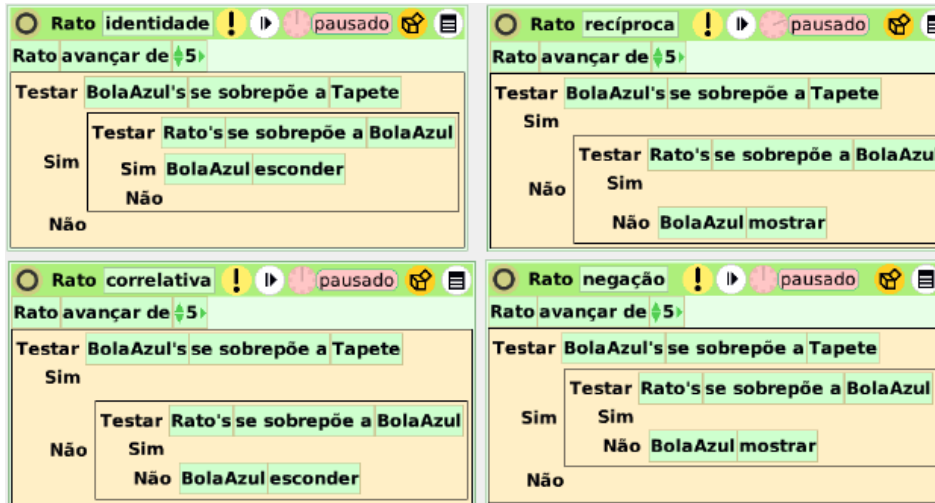
Analogamente o mesmo ocorre para a correlativa $\sim p \cdot q$ (figura 67 D abaixo) que poderia ser representada pela implicação $q \rightarrow \sim p$, gerando a operação de *incompatibilidade* indicada na figura 66 A acima. Porém, como q : “bola azul esconder” não pode ser condicional no Etoys, cria-se o *script* correlativo (figura 66 acima) através da negação de p ao colocar na linha *Não* o comando q : “Bola Azul esconder”, portanto a implicação $\sim p \rightarrow q$ (figura 67 c abaixo).

Figura 67: Diagramas de Venn da Correlação e da disjunção gerada pelo script



Seguindo a mesma ideia, procurou-se traduzir o INRC para o terceiro desafio. Embora as operações de negação ($(p \rightarrow q). \sim t$) e correlativa ($(\sim p . \sim q.t)$) não existam na sua forma “pura”, pensa-se que as implicações abaixo conservam as respectivas inversas do ponto de vista prático.

Figura 68: Scripts que traduzem o INRC no desafio 03



Independente deste fato parece que a relação inversa entre as quatro operações pode ser observada neste quarteto de *scripts* que representam o INRC.

9. CONSIDERAÇÕES FINAIS

Para auxiliar na compreensão da proposta e das conclusões obtidas até o momento, torna-se relevante retomar a trajetória desta pesquisa. De início este estudo foi desenvolvido com o propósito de investigar como se dá a aprendizagem de matemática no contexto da cultura digital. Assim, o interesse pela aprendizagem da *Matemática como um meio para se aprender* conduziu à investigação com o *Etoys* e, conseqüentemente, ao aprender a programar. Nessas circunstâncias, após uma experiência-piloto que explorou diferentes possibilidades de desenvolver a matemática, foi inevitável concentrar o foco da pesquisa especificamente na aprendizagem da lógica. Isso ocorreu, pois se observou que antes dos estudantes se debruçarem nos conceitos próprios da matemática, necessitavam compreender conceitos básicos de lógica, quando inseridos num ambiente fecundo à programação.

Frente a intenção de investigar a aprendizagem de lógica das crianças da escola básica – além do interesse em produzir estudos que contribuíssem para a Educação Matemática e para a inclusão da escola numa Cultura Digital – esta pesquisa visou contribuir também para a área da Ciência da Computação, ou melhor, para sua inserção no meio escolar. Nesse cenário, o conceito de *pensamento computacional* destaca-se. Como se viu Cuny, Snyder e Wing; 2010 e Wing; 2007 o definiram como a capacidade de promover o processo de formulação de problemas do mundo real e solucioná-los. Além disso, tais capacidades podem ser transpostas para qualquer área de conhecimento; isso não ocorre ao acaso, afinal Papert (2008) cunhou o termo *pensamento computacional* inspirado pelos princípios de resolução de problemas desenvolvidos pelo matemático húngaro Pólya. Sendo assim, quando se assume o compromisso de se desenvolver práticas pedagógicas pautadas na resolução de problemas e na utilização e na compreensão da lógica, pensa-se que é um território fecundo para a atividade do professor de matemática. Seja para possibilitar a aprendizagem de conceitos de matemática ou para o favorecimento do desenvolvimento do raciocínio lógico das crianças. Nessa perspectiva, entende-se que a Educação Matemática continua sendo o principal foco de tal estudo.

Como se viu, não foram encontradas pesquisas atuais que abordassem a aprendizagem da lógica na perspectiva da epistemologia genética. Logo, identificou-se que seria inédito e relevante produzir estudos que ajudassem os professores a compreenderem como ocorre tal processo de desenvolvimento quando as crianças aprendem a programar a partir de uma linguagem visual de programação como o *Etoys*.

Nesta tese foi investigado como ocorre o desenvolvimento do raciocínio condicional a partir do uso do comando *Teste* do *Etoys*. Os dados mostram que a atividade de programação propicia um

ambiente em que os sujeitos se veem provocados a mobilizar o raciocínio condicional e a refletir sobre ele a partir dos resultados e das suas ações realizadas durante a programação. Ainda, a análise revela a importância do pensamento combinatório, que permite aos sujeitos que estão num estágio cognitivo mais avançado testarem, sistematicamente, todas as possibilidades de ordenamento e inclusão dos comandos sugeridos, e a obterem as conclusões lógicas adequadas. Entretanto, os procedimentos apresentados não estão orientados para o acompanhamento deste processo de desenvolvimento do raciocínio em cada um dos sujeitos, constituindo, assim, uma fotografia de tal evolução. Dessa forma, neste trabalho há um início de construção de teoria, carecendo de um confronto com maior volume de dados e de explicações sobre os processos de transição entre as etapas. Pensa-se que tal estudo ajudará professores e pesquisadores a compreenderem o processo de aprendizagem da lógica de seus estudantes quando estes aprendem a programar.

Além do mais, tem-se consciência que essa não é a única maneira de explorar a aprendizagem da implicação; porém supõe-se que os resultados encontrados aqui aparecerão em diferentes estudos, pois se relacionam com o pensamento formal descrito nos estudos de lógica de Piaget. Nesse sentido, pensa-se que esta pesquisa deve ser continuada e ampliada para outras operações lógicas não investigadas. Por conseguinte, o experimento proposto nesta tese pode (e deve) ser feito em diferentes contextos: com crianças mais novas, com estudantes de idades entre onze e treze anos, do ensino médio e, também, com os graduandos dos cursos de ciências da computação. Todas essas pesquisas ajudariam na compreensão da aprendizagem da lógica que se estende ao nível de ensino superior.

Ademais, o experimento proposto nesta tese pode ser utilizado como atividade pedagógica ou como instrumento de investigação do desenvolvimento do raciocínio lógico dos estudantes. Outra possibilidade interessante, é utilizá-lo em diferentes tipos linguagem de programação visual ou não, para crianças ou adultos. Atualmente vê-se que o *Scratch* tem alcançado o maior número de usuários pelo mundo e avalia-se que tal número crescerá, pois há uma versão que possibilita edição online de projetos. Tal fato amplia as possibilidades de interação e aprendizagens num contexto em rede e, conseqüentemente, fomenta estudos e pesquisas em tal área. Sendo assim, o autor desta pesquisa pretende concentrar seus esforços – como professor de matemática e pesquisador – para fomentar o desenvolvimento de políticas públicas para a formação de clubes de programação nas escolas da rede pública. Mantendo o objetivo de investigar a aprendizagem de matemática e o desenvolvimento do raciocínio lógico quando se aprende a programar numa perspectiva piagetiana. Procurando, assim, contribuir para a inserção da escola pública numa cultura digital.

REFERÊNCIAS

- ALLEN-CONN, B.J.; ROSE, K.: **Powerful ideas in the classroom using *Squeak* to enhance math and science learning**. Tradução Cláudio Gilberto Cesar e Susana Seidel. [Glendale: Viewpoints Research Institute, 2003]. Disponível em: <<https://www.ufrgs.br/soft-livre-edu/arquivos/Squeak-ideias-poderosas.pdf>>. Acesso em: maio de 2016.
- ARENDT, H. (1994). ***Sobre a violência***. Rio de Janeiro: Relume – Dumará.
- ARRUDA, E.: **Ciberprofessor: Novas Tecnologias, Ensino e Trabalho Docente**. Belo Horizonte: Autêntica, 2004.
- BATTRO, A. M.: **Dicionário terminológico de Jean Piaget**. Tradução: Lino de Macedo. São Paulo: Pioneira, 1978.
- BATTRO, A. M.; PERCIVAL, J. D.: **Hacia una inteligencia digital**. Buenos Aires: Academia Nacional de Educación, 2007. Tradução nossa.
- BARANAUSKAS, M.C.C.: **Procedimento, função, objeto ou lógica? linguagens de programação vistas pelos seus paradigmas**. In VALENTE, J. A. Computadores e Conhecimento: repensando a educação. Campinas: Gráfica Central da UNICAMP, 1993. Disponível em:<http://pan.nied.unicamp.br/publicacoes/publicacao_detalhes.php?id=19>. Acesso em: maio de 2016.
- BECKER, F.: **O Caminho da aprendizagem em Jean Piaget e Paulo Freire: da ação à operação**. Petrópolis, RJ, Vozes, 2010.
- BECKER, F. **A origem do conhecimento e a aprendizagem escolar**. Porto Alegre (RS): Artmed; 2003.
- BEUG, A.: **Teaching Introductory Programming Concepts: A Comparison of *Scratch* and *Arduino***. Dissertação de mestrado, California Polytechnic State University. San Luis Obispo, 2012. Tradução nossa.
- BONA, A. S. D.: **Portfólio de Matemática: um instrumento de análise do processo de aprendizagem**. Dissertação (Mestrado). Programa de Pós-Graduação em Ensino de Matemática. Porto Alegre: UFRGS, 2010.

BRASIL. Ministério da Educação. Secretaria de Educação Fundamental. **Parâmetros Curriculares Nacionais: Matemática (1º e 2º ciclos do ensino fundamental)**. v. 3. Brasília: MEC, 1997.

CAPELA, H.M.M.S.: **Utilização de recursos digitais na aula de matemática**. Dissertação de mestrado, Universidade Católica Portuguesa, Lisboa, 2013.

CASTELLS, M.: **A sociedade em rede**. São Paulo: Paz e Terra, 1999.

CHIAROTTINO, Z.R.: **Psicologia e Epistemologia Genética de Jean Piaget**. São Paulo, EPU, 1988.

COSTA, I. T.; FAGUNDES, L. C.; NEVADO, R. A.: **Projeto TecLec: Modelo de uma Nova Tecnologia em EaD incorporando os Recursos da Telemática**” . Em: *Informática na Educação: Teoria & Sociedade*, v. 63, p. 105 -111, 1998.

CUNY, J., SNYDER, L., & WING, J.: **Demystifying Computational Thinking for Non Computer Scientists**. work in progress, 2010. Disponível em :<<http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf> >. Acessado em maio de 2016.

DALLA VECCHIA, R.: **A modelagem matemática e a realidade do mundo cibernético**. Tese (Doutorado em Educação Matemática) Universidade Estadual Paulista Júlio de Mesquita. Orientação: Marcus Vinicius Maltempo. Rio Claro, 2012.

D'AMBRÓSIO, U.: **Educação para uma sociedade em transição**. Campinas, SP: Papirus, 1999.

D'AMBRÓSIO, U.: Prefácio In: BAGATINI, F. M., e ALMEIDA, M.E. [et al]: **O computador portátil na escola: mudanças e desafios nos processos de ensino e aprendizagem**. Avercamp, São Paulo, 2011.

DEVAL, J.: **Introdução à prática do método clínico: descobrindo o pensamento da criança**. Tradução Fátima Murad. Porto Alegre. Artmed, 2002.

DIENES, Z. P.: **A matemática moderna no ensino primário**. Rio de Janeiro, Fundo de Cultura, 1967.

DIENES, Z. P.: **As seis etapas do processo de aprendizagem em matemática**. São Paulo, Herder, 1972.

DULLIUS, S.R.: **Ambientes de autoria como possibilidade para múltiplas alfabetizações.** Monografia de especialização. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

DUTRA, Í. M.: **Mapas conceituais no acompanhamento dos processos de conceituação.** Porto Alegre: UFRGS, 2006. Tese (Doutorado em Informática na Educação), Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, 2006.

FAGUNDES, L. ; SATO L. S.; MAÇADA, D. L.: **Aprendizes do futuro: as inovações começaram.** Brasília: MEC, 1999. Coleção Informática para a Mudança em Educação/Mec/Seed/Proinfo.

FAGUNDES, L.C.; NEVADO, R. A.; BASSO, M. V. A., BITTENCOURT, J. V.; MENEZES, C. S., MONTEIRO, V. C.: **Projetos de aprendizagem: uma experiência mediada por ambientes telemáticos.** Revista brasileira de informática na educação. Vol. 14, n. 1 (jan./abr. 2006), p. 29-39.

MARQUES, M.C.P.O.: **O ensino da programação no desenvolvimento de jogos através ambiente Scratch.** Relatório de estágio de mestrado, Universidade do Minho, Braga, 2013.

FARIA, A.F.L.: **Tecnologias digitais no jardim de infância: comunicação, aprendizagem e desenvolvimento profissional docente,** Tese de doutorado. Universidade do Minho, Braga, 2014.

FERREIRA, P. A.O.R.: **Desenvolvimento de projectos em Squeak e construção de conhecimento em alunos do 1º Ciclo,** Dissertação de mestrado. Universidade do Minho, Braga, 2010.

FERREIRA, S.M.O.: **Ouso de software educativo em ambientes de aprendizagem. Um estudo de caso com alunos do 1º Ciclo do Ensino Básico.** Dissertação de mestrado, Universidade do Minho, Braga, 2009.

FIorentini, D.; Lorenzato, S.: **Investigação em educação matemática: percursos teóricos e metodológicos.** Campinas: Autores Associados, 2007.

FREIRE, P.: **Pedagogia da autonomia: saberes necessários à prática educativa.** 31. ed. São Paulo: Paz e Terra, 2005.

GALLARDO, J.M.: **Design and implementation techniques for location-based learning games.** Tese de doutorado, Universitat Pompeu Fabra, Barcelona, 2014.

GONÇALVES, J.P.C.: **Análise e desenvolvimento de extensões de controlo para ambiente**

Scratch. Dissertação de mestrado, Universidade do Minho, Braga, 2012.

INGALLS, D., KAEHLER, T., MALONEY, J., WALLACE, S., & KAY, A.: **Back to the future: the story of Squeak, a practical Smalltalk written in itself**. Report of Viewpoints Research Institute. Glendale, CA, 1997. Disponível em: <http://www.vpri.org/pdf/tr1997001_backto.pdf >. Acesso: maio de 2016.

KAY, A.: **The Early History of Smalltalk**. Association for Computing Machinery . Massassuchets, 1993. Disponível em: <http://www.smalltalk.org/smalltalk/TheEarlyHistoryOfSmalltalk_Abstract.html >. Acessado: maio de 2016.

KAY, A.: **Background on How children learn**. VPRI Research. Note RN-2003-002 disponível em <<http://www.vpri.org>>. Acesso em maio de 2016.

KAY, A.: **The Dynabook Revisited - A Conversation with Alan Kay**. In *The Book and the Computer*, 2002. Disponível em: <<http://www.Squeakland.org/content/articles>>. Acessado em maio de 2016.

LEMOS, A.: **Cibercultura: tecnologia e vida social na cultura contemporânea**. Porto Alegre: Sulina, 2002; p 116 e 123.

LÉVY, P.: **Cibercultura**. São Paulo, 1999.

LIMA FILHO, C.B.: **Aprendizagem colaborativa mediada pelo Squeak**. Dissertação mestrado. Orientação: Prof.Dr. José Aires de Castro Filho. Universidade Federal do Ceará. Fortaleza, 2013.

LINDNER, E.L.: **Uma arquitetura pedagógica apoiada em tecnologias da informação e comunicação: processos de aprendizagem em Química no ensino médio**. Tese de doutorado. Universidade do Rio Grande do Sul, Porto Alegre, 2009.

LOPES, D. Q.: **Brincando com Robôs: desenhando problemas e inventando porquês**. Santa Cruz do Sul; EDUNISC, 2010.

MENEZES, C. S.: **Desenvolvimento de Jogos Digitais como Estratégia de Aprendizagem**. (Organizador): Informática Educativa II – Linguagens para a Representação do Conhecimento. Vitória: UFES, 2003.

MITRA, S.; RANA, V.: **Children and the Internet: Experiments with minimally invasive**

education in India. The British Journal of Educational Technology, volume 32, issue 2, pp 221-232. (2001). Disponível em: <<http://sugatam.wikispaces.com/>>. Acesso em maio de 2016.

MORAIS, A. D.; MATTOS, E. B.V.; FAGUNDES, L. C.: **A Matemática do Squeak Etoys e a Educação Matemática: uma perspectiva de Projetos de Aprendizagem.** In: XVIII Contresso Internacional de Informática Educativa, TISE 2013, Porto Alegre. Nuevas Ideas en Informática Educativa, 2013a. v. 9. p. 375-383.

MORAIS, A. D.; FAGUNDES, L. C. ; BASSO., A. V.: **Squeak Etoys & Lógica: refletindo sobre o uso do comando Teste em sala de aula.** In: XVIII Contresso Internacional de Informática Educativa, TISE 2013, Porto Alegre. Nuevas Ideas en Informática Educativa, 2013b. v. 9. p. 542-545.

MORAIS, A. D.; FAGUNDES, L. C.; BASSO., A. V.: **Reflexões sobre o raciocínio lógico ao aprender a programar no Squeak Etoys.** RENOTE. Revista Novas Tecnologias na Educação, v. 11, p. 1-10, 2013c.

MORAIS, A. D.; FAGUNDES, L. C.: **A Inclusão Digital da Escola ou a Inclusão da Escola na Cultura Digital?** Diálogo (Canoas), v. 19, p. 97-113, 2011.

MORAIS, A. D.: **Fórmula (-1): Desenvolvendo objetos digitais de aprendizagem para as operações com números positivos e negativos.** Dissertação (Mestrado). Programa de Pós-Graduação em Ensino de Matemática. Porto Alegre: UFRGS, 2010.

MORIN, E.: **A cabeça bem-feita: repensar a reforma, reformar o pensamento.** Rio de Janeiro: Bertrand Brasil, 2000.

MORENO, S.R.: **Interfaces abertas: dispositivos programáveis no ensino de artes visuais.** Dissertação de mestrado, Universidade Federal de Minas Gerais, Belo Horizonte, 2012.

NEGROPONTE, N.: **A Vida digital.** Tradução Sérgio Tellaroli; supervisão técnica Ricardo Rangel. São Paulo. Companhia das Letras, 1995.

NEVADO, R.; CARVALHO, M.J.S.; MENEZES, C.S.: **Aprendizagem em rede na educação a distância: estudos e recursos para a formação de professores.** 1. ed. Porto Alegre: Ricardo Lenz, 2007.

NIQUINI, D. P.: **Informática na educação: implicações didático pedagógicas e construção do conhecimento.** Brasília : Universa, 1996.

OLIVEIRA, M.F.B.: **Aprendizagem cooperativa no ensino da multimédia: uma estratégia pedagógica para a promoção da motivação e da competitividade positiva**. Relatório de estágio de mestrado em Ensino de Informática, Universidade do Minho, Braga, 2013.

PAPERT, S.A.: **A máquina das crianças:repensando a escola na era da informática**. Tradução de Sandra Costa. Porto Alegre: Artes Médicas, 2008.

PAPERT, S. A.: **Logo: Computadores e educação**. São Paulo: Brasiliense, 1986.

PARENTE, A.: **Tramas da rede: novas dimensões filosóficas estéticas e políticas da comunicação**. Organizador: André Parente. Porto Alegre: Sulina, 2004.

PEREIRA, P.J.T.: **Produção de objetos de aprendizagem com Scratch: um experiência no ensino de informática com alunos de um Curso de Educação e Formação no Agrupamento de Escolas de Cabeceiras de Basto**. Relatório de estágio de mestrado em ensino de informática, Universidade do Minho, Braga, 2013.

PIAGET, J; GRÉCO, P.: **Aprendizagem e conhecimento**. 1ª ed. São Paulo: Freitas Bastos, 1974.

PIAGET, J.: **Comentários sobre a Educação Matemática**. Tradução: Eduardo Britto Velho de Mattos. Comments on mathematical education.In: Developments in mathematical education : proceedings of the 2nd International congress on mathematical education, Exeter, August 29th September 2nd, 1972 .

PIAGET, J.; INHELDER, B.: **Gênese das estruturas lógicas elementares**. Tradução: Álvaro Cabral. 2ª ed., Rio de Janeiro, Zahar, 1975 .

PIAGET, J.; INHELDER, B.: **Da lógica da criança à lógica do adolescente: ensaio sobre a construção das estruturas operatórias formais**. Tradução de Dante Moreira Leite. São Paulo, Pioneira, 1976a.

PIAGET, J.: **Ensaio de lógica operatória**; segunda edição de Tratado de lógica, ensaio de logística operatória. 1949 estabelecida por Jean-Blaise Grize; Tradução Maria Ângela Vinagre de Almeida. Porto Alegre, Globo; São Paulo. 1976b.

PIAGET, J.: **A psicologia da inteligência**. Tradução Álvaro Cabral. Rio de Janeiro, Zahar Editores, 1977.

PIAGET, J.: **O nascimento da inteligência na criança**. Tradução Álvaro Cabral. Rio de Janeiro, Zahar Editores, 1982.

PIAGET, J.: **Abstração Reflexionante: relações lógico-aritméticas e ordem das relações espaciais**. Tradução Fernando Becker e Petronilhas Beatriz Gonçalves da Silva. Porto Alegre: Artes Médicas, 1995.

PINTO, A.S.: **Scratch na aprendizagem da Matemática no 1º Ciclo do Ensino Básico: estudo de caso na resolução de problemas**. Dissertação de mestrado, Universidade do Minho, Braga, 2010.

PIRES, R.S.P.: **O uso da WebQuest nas aulas de História e de Geografia para a promoção de competências nos alunos: um estudo com alunos do 9º ano de escolaridade**. Relatório de estágio de mestrado em História e Geografia, Universidade do Minho, Braga, 2013.

PIXEL. **Projeto Pixel**. Colégio de Aplicação da Universidade Federal do Rio Grande do Sul. 2011. Disponível em: <<http://www.ufrgs.br/projetopixel>>. Acesso em maio de 2016.

QUINTANILLA CÓNDROR, C.N.: **La mediación de Squeak Etoys em el desarrollo del concepto de fracción: una experiencia construcionista em una escuela de Galicia**. Tese de doutorado, Universidade de Santiago de Compostela, Santiago de Compostela, 2012.

RIEBER, L.P.: **Seriously considering play: designing interactive learning environments based on the blending of microworlds, simulations, and games**. USA, 1996. Disponível em <<http://lrieber.coe.uga.edu/play.html>>. Acesso em: 04 de junho de 2013.

RESNICK, M.: **Reviving Papert's Dream**. Educational Technology: The magazine for managers of changes in education. Volume 52, number 4, July-August 2012. . Disponível em: <<http://web.media.mit.edu/~mres/papers/educational-technology-2012.pdf>> Acesso em: maio de 2016.. Tradução Nossa.

RESNICK, M.; OCKO, S.: **LEGO/Logo: Learning Through and About Design**. Epistemology and Learning Group, MIT: Media Laboratory. Cambridge. MA. 1991. Disponível em: <<http://ilk.media.mit.edu/papers/ll.html>> Acesso em: maio de 2016. Tradução Nossa.

RESNICK, M.: **Learn to Code, Code to Learn**. Epistemology and Learning Group, MIT: Media Laboratory. Cambridge. MA, 2013. Disponível em: <<http://ilk.media.mit.edu/papers/ll.html>> Acesso em: maio de 2016. Tradução Nossa.

SILVA, J.T. : **Metodologia de apoio ao processo de aprendizagem via autoria de objetos de aprendizagem por alunos**. Tese de doutorado. Universidade do Rio Grande do Sul, Porto Alegre, 2008.

SILVA, L.P.P.: *Squeak e aprofundamento de competências numéricas em crianças do 1º ano de escolaridade*, Dissertação de mestrado. Universidade do Minho, Braga, 2009.

SILVA, R.F.V.R.: *Implementação de grafos no ambiente de programação visual Scratch*. Dissertação de mestrado, Universidade do Minho, Braga, 2012.

SOUSA, R.M.; LENCASTRE, J. A.: *Scratch: uma opção válida para desenvolver o pensamento computacional e competências de resolução de problemas*. CIEC – Textos em atas. Universidade do Minho, 2012.

TEIXEIRA, A. L. V.S.: *Integração das TIC na educação: o caso do Squeak Etoys*. Tese de doutorado, Universidade do Minho, Braga, 2011.

VALENTE, J. A. Prefácio In: PAPERT, S. A. *Logo: computadores e educação*. São Paulo: Brasiliense, 1986, p.7-10.

VEEN, W. & VRAKKING, B.: *Homo Zappiens: educando na era digital*. Porto Alegre: Artmed, 2009.

WING, J. M.: *Computational Thinking*. (2007). Disponível em: <http://www.cs.cmu.edu/afs/cs/usr/wing/www/Computational_Thinking.pdf>. Acessado em maio de 2016.

ANEXOS

ANEXO A

1. ETOYS E MATEMÁTICA

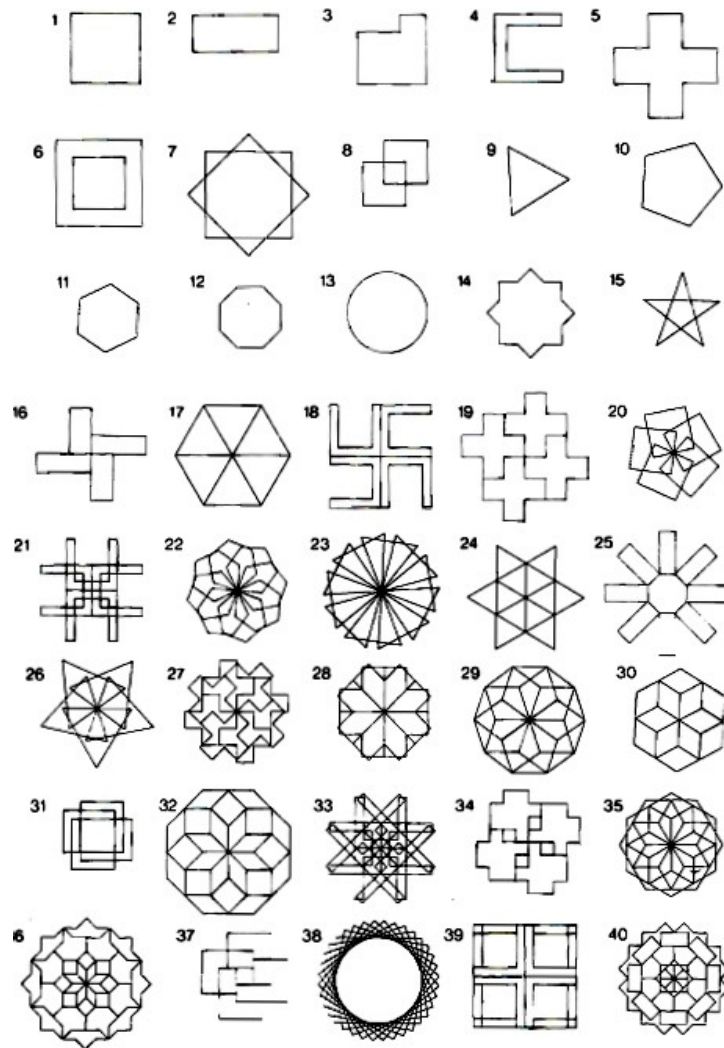
Alan Kay ressalta o fato de não se precisar de tecnologia para se aprender ciências e matemática. Mas, assim como na música, ela é um instrumento que amplifica as ideias dos estudantes. Porém, Kay (2002) alerta que o que um professor pode fazer é criar as melhores condições para cada criança aprender. Uma vez que se tem isso, então o computador pode ajudar imensamente. Além disso, os princípios pedagógicos que fundamentam o *Etoys* confluem com os Parâmetros Curriculares Nacionais (1997) que indicam como objetivos gerais do Ensino Fundamental que os alunos sejam capazes de: a) Utilizar as diferentes linguagens (verbal, matemática, gráfica, plástica e corporal) como meio para produzir, expressar e comunicar ideias, interpretar e usufruir das produções culturais, em contextos públicos e privados, atendendo a diferentes intenções e situações de comunicação. b) Saber utilizar diferentes fontes de informação e recursos tecnológicos para adquirir e construir conhecimentos; c) Questionar a realidade formulando problemas e tratando de resolvê-los, utilizando para isso o pensamento lógico, a criatividade, a intuição, a capacidade de análise crítica, selecionando procedimentos e verificando sua adequação.

Portanto, neste momento emerge a necessidade de se refletir sobre quais tipos de aprendizagem de matemática são possibilitadas pelo uso do *Etoys*.

1.1 Tartarugas no *Etoys*

Como o *Etoys* é um *software* que se inspira na filosofia Logo, ele mantém todas as possibilidades de desenvolver trabalhos bem ao estilo da tartaruga. Dessa maneira, pode-se dizer que o *Etoys* herda todas as atividades de geometria que são próprias do Logo e que foram desenvolvidas em diversos estudos de pesquisas no LEC/UFRGS. Na figura 69 abaixo apresenta-se uma série de figuras e mosaicos do livro *A Confusão da Tartaruga* de Newell (1988) que podem ser reproduzidas no *Etoys*.

Figura 69: Polígonos e desenhos geométricos ao estilo LOGO

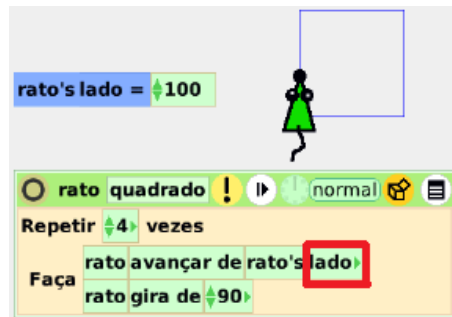


Apenas com esse tipo de figuras é possível contemplar um trabalho rico que aborde os conteúdos de polígonos irregulares, regulares, estrelados, diagonais, ângulos internos e externos, etc.

1.2 Introdução à Álgebra

Com o *Etoys* ainda é possível realizar uma introdução à álgebra através da introdução de um trabalho com variáveis. Por exemplo, após os estudantes construírem os polígonos regulares através do *Etoys*, pode-se explorar a criação de algoritmos genéricos, basta introduzir o uso de variáveis como é apresentado na figura 70 abaixo. Nela a medida do lado do quadrado é definida através da variável “lado”, sendo assim o usuário tem o controle de escolher o tamanho do quadrado que o rato desenhará.

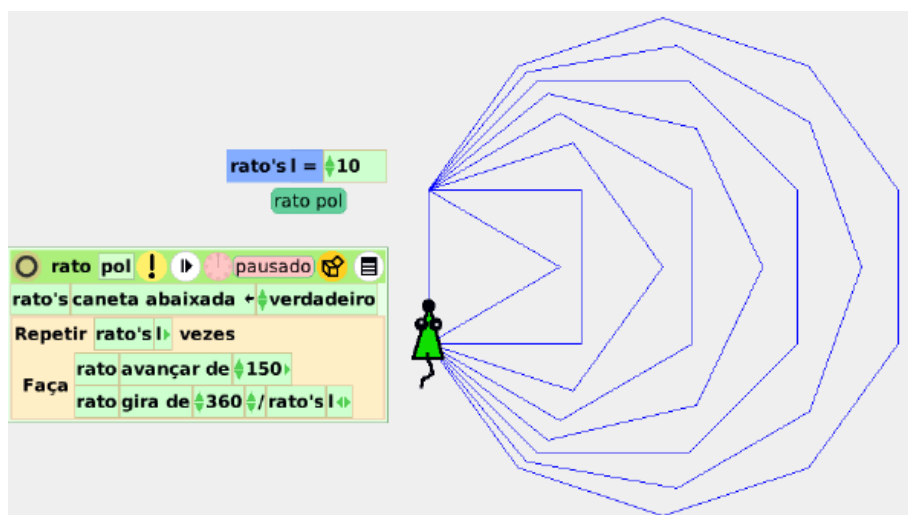
Figura 70: Script de um quadrado



Dessa maneira tem-se uma situação, em que a ideia de utilizar uma palavra (um símbolo) para representar um número, possui um resultado prático nesse mundo virtual. Como é possível atribuir diferentes valores para o símbolo “lado”, tem-se o desenvolvimento da noção de variável, que muitas vezes só é explorado pela primeira vez no início do ensino médio.

A partir disso é possível desenvolver cálculos e operações com essas variáveis, elevando, assim, a complexidade da programação. Por exemplo, na figura 71 abaixo se apresenta um construtor de polígonos genéricos, no qual o usuário tem o controle de escolher qual polígono regular que o rato desenhará. Basta determinar o número de lados que é definido através da variável “l”.

Figura 71: Construtor de um polígono genérico.



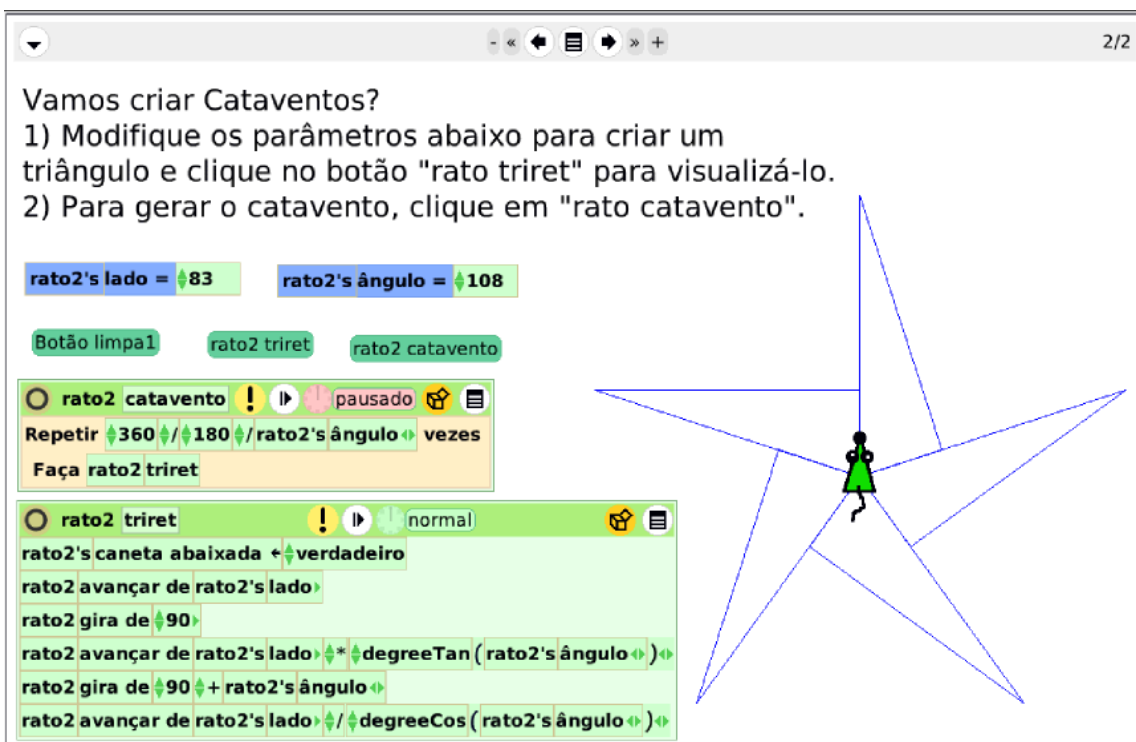
Note que a construção do *script pol* exige o desenvolvimento da relação matemática entre o ângulo de giro do rato e número de lados do polígono. Tendo o trabalho com a linguagem LOGO como referência, o autor da pesquisa já constatou que esse tipo de atividade propicia a descoberta de tal relação por estudantes a partir de 10 anos. Ou seja, que desde cedo, neste contexto, é possível que as crianças comecem a realizar suas primeiras operações com variáveis.

1.3 Triângulos e trigonometria

A partir do trabalho realizado com polígonos é comum explorar o desenho de triângulos não equiláteros. Esse tipo de desenho se torna mais complexo que os polígonos regulares, visto que é preciso o conhecimento de trigonometria. Sem ele, os estudantes realizam tal tarefa num exaustivo processo de tentativa e erro. Porém pensa-se que este é o momento propício para a introdução da trigonometria para alunos das séries finais do ensino fundamental.

Para exemplificar essa possibilidade, na figura 72 abaixo se mostra um construtor de cataventos criado como atividade de exploração numa das oficinas de *Etoys* no LEC/UFRGS. Pensa-se que tal atividade é adequada para estudantes das séries finais do ensino fundamental e médio.

Figura 72: Construtor de cataventos

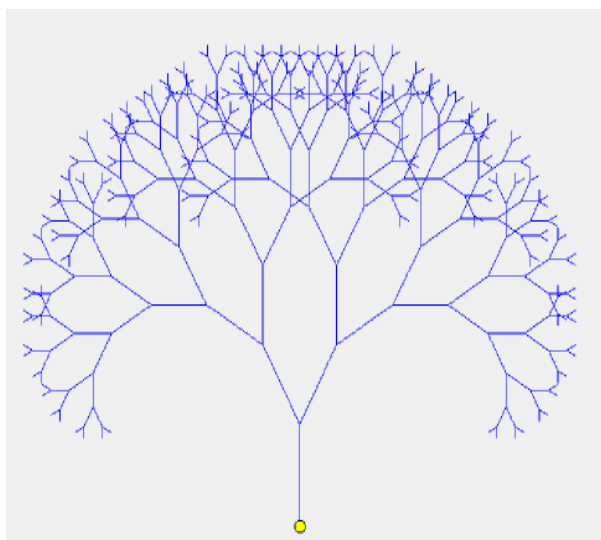


1.4 Fractais e algoritmos recursivos

Num nível avançado é possível utilizar os algoritmos recursivos no *Etoys* e, assim, abordar a construção de fractais, na figura 73 abaixo se apresenta um fractal desenvolvido por Pitt (s/d) e que

está publicado no site do *The EtoysIllinois Project*⁵ cuja construção está disponível passo a passo na sua página.

Figura 73: Algoritmos recursivos: árvore fractal



Pode-se dizer que até o momento foram apresentadas atividades que o *Etoys* herdou do Logo, agora serão apresentadas propostas que trazem as novidades relacionadas às novas possibilidades produzidas pelo desenvolvimento do *Squeak Etoys*.

1.5 Matemática dos movimentos

Uma das primeiras atividades desenvolvidas no *Etoys* é a construção de um carro que se movimenta de forma autônoma. Tal ação é possível através do botão relógio que fica executando qualquer *script* em *loop*. Só a existência desse botão dá novos significados e possibilidades para as atividades ao estilo LOGO. No livro *Ideias poderosas para a sala de aula*, Allen-Conn e Rose (2003) compartilham doze projetos desenvolvidos por crianças no *Etoys*, que permitem investigar um número de ideias poderosas de matemática e ciências e que ilustram conceitos chave a partir de uma perspectiva nova. Dessa maneira, as autoras indicam quais conteúdos de matemática que foram trabalhados nesse processo.

A primeira série de projetos tem como ponto de partida a criação do seu próprio carro. Pode-se ver na figura 74A que para dirigir o seu carro, a criança atribuirá valores numéricos (positivos, negativos e o zero) aos comandos “avançar de”, “girar de”, “carro \rightarrow x”, “carro \rightarrow y” e “carro \rightarrow direção”.

5 *The EtoysIllinois Project*: <https://Etoysillinois.org/library?sl=1611>

No comando “avançar de” esses valores se referem à distância (em pixels) que o carro irá se deslocar. Sendo assim, os valores positivos fazem o carro “andar para frente”, os negativos, “para trás” e o zero faz o carro “ficar parado”. Dessa maneira inicia-se uma exploração intuitiva do conceito de velocidade. Já os valores numéricos do comando “girar de” indicam medida de ângulos de rotação em graus, no qual os números positivos indicam um giro para direita e os negativos, para esquerda. Sendo assim, a medida que o “carro anda pelo mundo” (se movimenta na tela do computador) a criança pode observar que os valores numéricos dos outros três comandos vão se modificando.

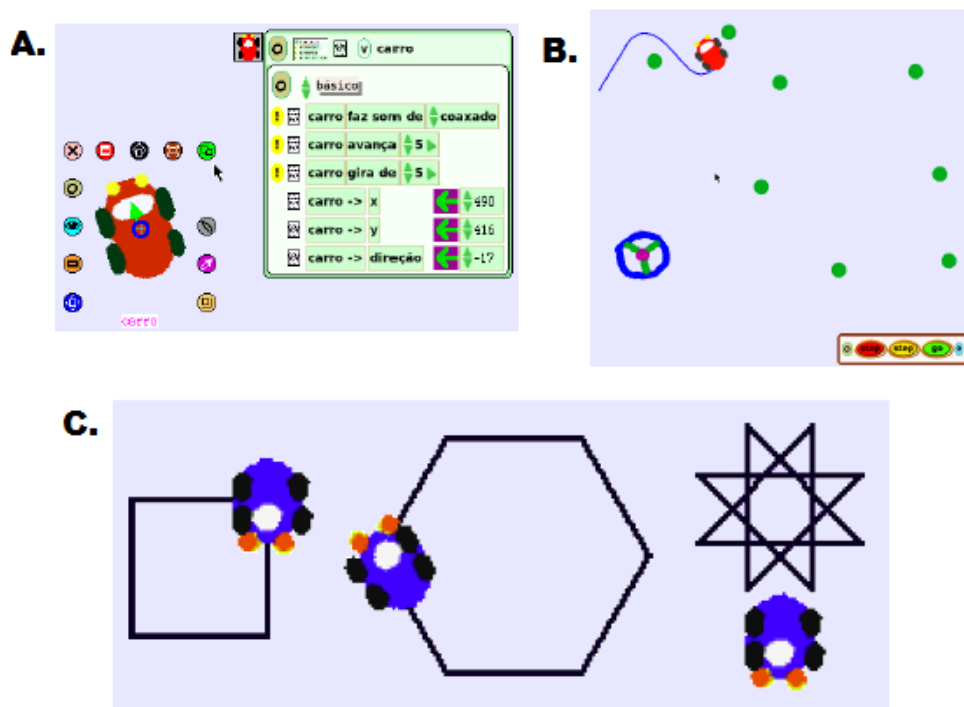
Dessa maneira, tal tipo de projeto promove a aprendizagem de ideias poderosas da matemática como o conceito de zero, de números negativos e positivos. Avalia-se que, nesse sentido, o *Etoys* se aproxima das propostas de atividades do jogo *Fórmula (-1)*, presentes na dissertação de mestrado de Moraes (2010). Já que, ele promove o desenvolvimento do raciocínio aditivo e multiplicativo, pois é um micromundo em que as operações com números positivos e negativos são possibilitadas através de uma experiência virtual de um modelo físico de movimento.

Já os valores numéricos atribuídos aos comandos “carro \rightarrow x”, “carro \rightarrow y” e “carro \rightarrow direção”, podem ser explorados quando se deseja dirigir o carro a partir de um volante virtual, como mostra a figura 74B abaixo. Dessa maneira os valores numéricos atribuídos aos dois primeiros comandos dizem respeito ao sistema de coordenadas cartesianas do mundo, cuja origem é o vértice inferior esquerdo da tela. Portanto promove a compreensão das coordenadas x e y do plano cartesiano.

O número atribuído ao comando “carro \rightarrow direção” se refere à medida em graus da direção. É importante ressaltar que tais comandos utilizam sistemas de referências distintos: a) para o “avançar de”, “girar de”, o sistema de referência é o próprio objeto, enquanto “carro \rightarrow x”, “carro \rightarrow y” e “carro \rightarrow direção”, utilizam a tela do computador como referência. É fundamental que a criança realize tal diferenciação.

Além disso, a partir desses dois comandos, há a possibilidade de promover a compreensão de formas geométricas simples e ângulos (ao estilo LOGO), que podem ser vistas na figura 74C abaixo.

Figura 74: A matemática ao brincar com carros virtuais



Porém uma das atividades mais interessantes produzidas com *Etoys* é a criação de carros inteligentes que reconhecem e andam numa estrada, figura 75A.

Figura 75: A matemática dos carros inteligentes

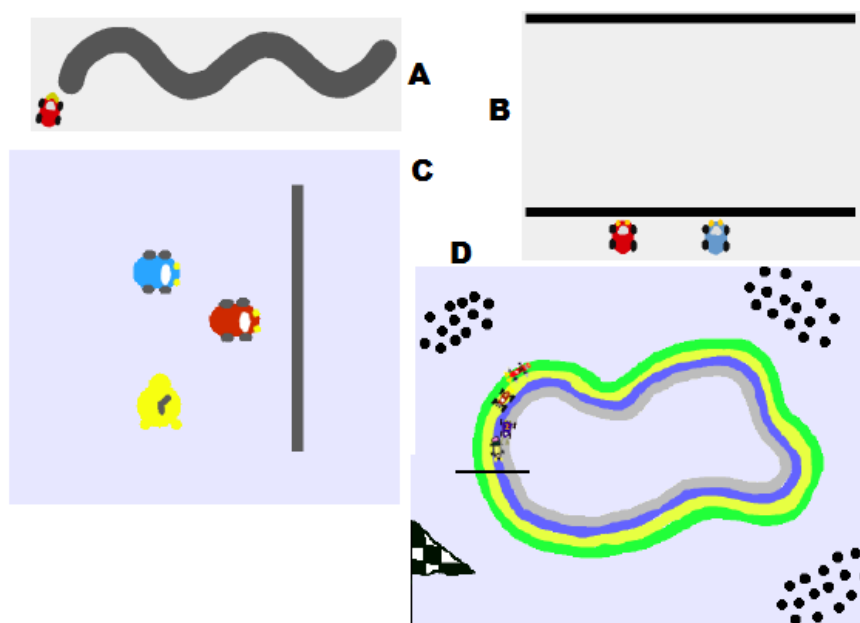


Figura 76: Janela do comando Teste

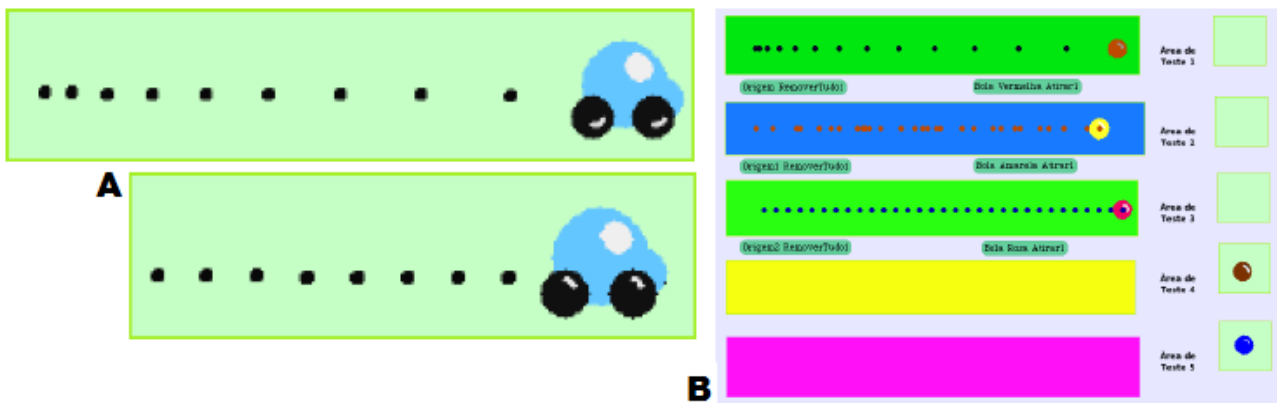


Tal reconhecimento é realizado através de sentenças condicionais simples, presente no comando *Teste*, como se vê na figura 76 ao lado. Sendo assim, a parte dianteira do carro (região amarela) reconhece a estrada cinza em que ele deve andar.

Já na atividade (figura 75B, C e D acima) em que dois carros devem competir entre si é fundamental que eles tenham velocidades variáveis e diferentes. Para isso se explora o comando “incrementar por” associado a um número. Dessa forma pode-se fazer o carro andar cada vez mais rápido (atribuindo um incremento positivo), ou parando (incremento negativo) ou com velocidade constante (incremento zero) obtém a simulação de uma aceleração. Com isso surge a oportunidade de se entender os conceitos de variáveis e de números que são constantes ou incrementados (e também decrementados). Além disso, pode-se atribuir incrementos aleatórios, oportunizando o entendimento o conceito de aleatoriedade quando se aplica à número.

Para exemplificar tais conceitos, as autoras desenvolveram com as crianças simulações para ilustrar tais movimentos dos carros, através da plotagem de pontos (ver figura 77A abaixo). Segundo elas, tal plotagem é possível através de comandos que utilizam os conceitos de “média”, “moda” e “mediana” da estatística. Pode-se observar na figura 77B que um dos projetos servia para plotar os movimentos de bolinhas em vez de carros.

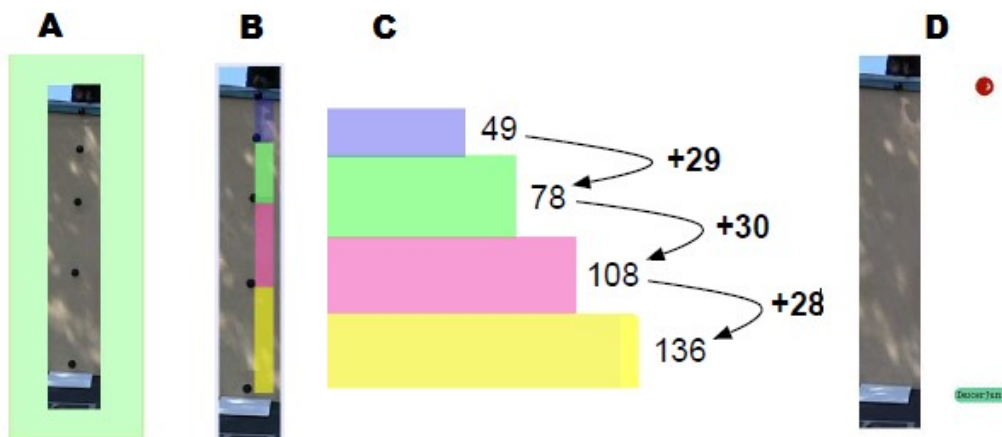
Figura 77: Aceleração e velocidade no Etoys



Tendo feito isso, Allen-Conn e Rose (2003) promoveram algumas experiências sobre gravidade com as crianças. Sendo assim, partiram da intuição das crianças que a faz acreditar que uma bola de 5 quilos chega ao chão antes de uma bola de espuma. Depois de experimentar e filmar uma série de atividades ao ar livre, incluindo a observação da queda de várias bolas e mesmo medir o tempo das quedas. Constataram que algumas crianças entenderam que a sua intuição inicial estava equivocada, enquanto outros mantiveram suas crenças iniciais. Sendo assim elas, elas propuseram

realizar simulações computacionais de tal experimento no *Etoys*.

Figura 78: Modelo computacional no *Etoys* da experiência de gravidade



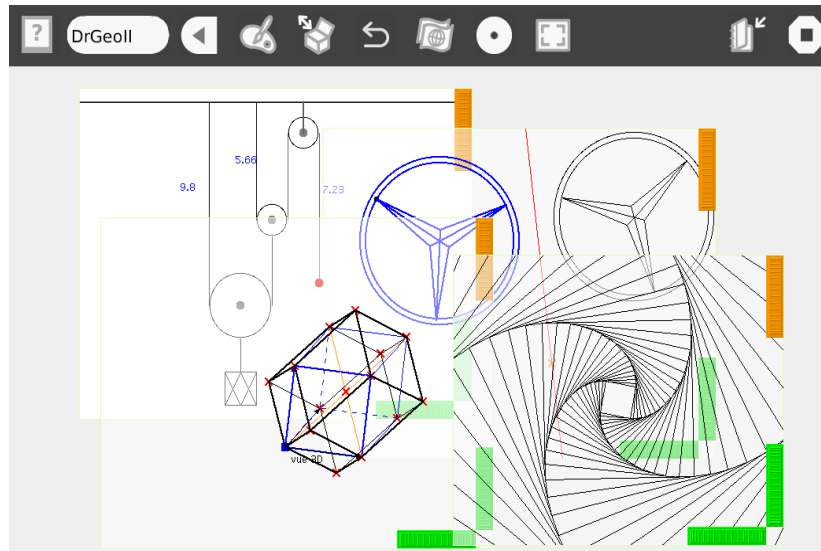
Na figura 78A acima, vemos a execução do vídeo que as crianças filmaram da experiência, logo em seguida (na figura 78B) eles desenharam no *Etoys* retângulos para medir a distância entre as bolas. Ao medir a largura dos retângulos, perceberam que a variação tinha um incremento entorno de 30 pixels. Logo em seguida, as crianças conseguiram simular sincronicamente o movimento de uma bola virtual, junto com a do vídeo. Segundo as pesquisadoras, as crianças puderam resolver os conflitos de suas intuições com a ajuda de modelos computacionais e simulações criadas por elas mesmas.

Allan Kay costuma explorar tal exemplo em suas palestras, para exemplificar que com o uso dos computadores, crianças da escola básica podem compreender raciocínios matemáticos que só são estudados nas disciplinas de cálculo diferencial de cursos de graduação. Sendo assim, Allen-Conn e Rose afirmam que “*Squeak* é muito mais que um processador de textos – é um processador de ideias. É uma linguagem, uma ferramenta e um ambiente de criação de meios.” (2003, p. v)

1.6 Geometria dinâmica

No *Etoys* é possível fazer construções com régua e compasso virtual, pois nele está incorporado o *software* de geometria dinâmica, chamado Dr. Geo II. Abaixo, na figura 79 apresenta-se algumas construções geométricas encontradas no site: <http://blog.offset.org/hilaire/index.php?tag/Etoys/page/2>.

Figura 79: Construções geométricas feitas no Dr. Geo II do Etoys



Até este momento essas são as atividades que possibilitam a aprendizagem de matemática no *Etoys*. Ao refletir sobre elas, pensa-se que tais atividades contemplam a aprendizagem de todos os conceitos chave da matemática do ensino básico. Além disso, avalia-se que os estudantes deste nível de ensino podem explorar melhor aquelas atividades que envolvam o sistema de coordenadas cartesianas, bem como o uso da lógica proposicional presente nos *Testes*.

Desde o início da formulação desta tese, identificam-se duas maneiras de promover a aprendizagem da matemática pelos estudantes através do uso do *Etoys*. A primeira maneira seria a partir de atividades que exploram diretamente objetos matemáticos, cujo tema é a própria matemática. Temos como exemplos atividades tais: desenhar triângulos, polígonos, mosaicos, criar figuras simétricas, medianas no Dr. Geo.

Já a segunda maneira, seria explorar a matemática que se aprende ao se programar no *Etoys*, no qual o objeto criado pelo estudante não seja necessariamente matemático. Sendo assim, a aprendizagem de matemática se dá como um meio para se aprender. Já que o objetivo do estudante não seria aprender matemática, mas sim produzir algo no *Etoys* e “de brinde” aprende-se outras coisas tais como matemática, física, biologia e etc. Os projetos desenvolvidos e apresentados por Allen-Conn e Rose são exemplos deste tipo de aprendizagem de matemática.

No contexto deste estudo, utilizaremos *Matemática como Objeto* de aprendizagem para designar a primeira e *Matemática como um meio para se aprender*, a segunda. Para o autor desta pesquisa, o estudo da aprendizagem da matemática como meio para aprender se configura como um desafio, uma vez que gera uma série de dúvidas e incertezas, por exemplo: a) Será que é possível aprender matemática em qualquer produção no *Squeak Etoys*? Num primeiro momento, supõe-se

que não, e neste caso, torna-se interessante investigar (b) quais seriam as produções que promovam tal aprendizagem? Ainda há aquelas dúvidas relacionadas às questões práticas da atividade profissional de professor: (c) Será que tais aprendizagens contemplarão os conteúdos curriculares? (d) Será que elas promoverão a aprendizagem de conteúdos que estão além daqueles previstos no currículo?

ANEXO B

1. PILOTO: OFICINA DE *Etoys* NO CAP

Numa tentativa de delinear novas hipóteses e encaminhamentos para as questões apresentadas no capítulo anterior, no início de segundo semestre de 2012 teve-se a oportunidade de realizar uma experiência de utilização do *Squeak Etoys* numa turma do nono ano do ensino fundamental do Colégio de Aplicação da Universidade Federal do Rio Grande do Sul (CAP/UFRGS). O professor Eduardo B.V. De Mattos, regente da turma, aceitou desenvolver tal trabalho ao tomar conhecimento da intenção desta pesquisa que seria promover a aprendizagem de matemática através da utilização do *Squeak Etoys* numa perspectiva de Projetos de Aprendizagem, visto que ele próprio já havia desenvolvido uma pesquisa de dissertação intitulada “Construção de conceitos de matemática via projetos de aprendizagem” (2010).

Dessa maneira foram realizados encontros semanais com duração de duas horas-aula e no horário da disciplina de matemática. A equipe desta experiência era formada por dois professores de matemática, e pelo psicólogo e pesquisador do LEC/UFRGS Bruno F. Sperb, que está desenvolvendo uma pesquisa sobre a *Construção de jogos digitais em Etoys na modalidade 1:1*.

Desde o início, a intenção foi realizar uma investigação bem aberta e que ajudasse a equipe a refletir sobre suas próprias pesquisas. A ideia era criar para os estudantes um espaço de criação livre em que os pesquisadores pudessem observar, formular e testar suas hipóteses. Particularmente para esta tese, a intenção deste piloto era promover os dois tipos de aprendizagem de matemática (como objeto de estudo e como meio para se aprender), sendo a primeira realizada durante os encontros de programação e a segunda, em produções dos alunos voltadas para seus trabalhos desenvolvidos no Projeto PIXEL. O Pixel é um projeto curricular das séries finais do Ensino Fundamental do CAP/UFRGS. Segundo o site do Colégio de Aplicação, com o Pixel⁶ “pretende-se colocar em prática atividades multidisciplinares que permitam ir além do conhecimento específico de cada disciplina, não explorando apenas os pontos comuns entre elas, mas possibilitando experiências reais de interação com o mundo” (PIXEL, 2011). Tal projeto procura realizar uma iniciação científica dos estudantes e suas atividades são inspiradas nos projetos de aprendizagem desenvolvidos no Projeto Amora.

Realmente tal experiência foi significativa e intrigante, pois promoveu uma série de

6 Site do projeto PIXEL: <http://www.ufrgs.br/projetopixel>

questionamentos e reflexões que, certamente, redirecionaram o rumo desta pesquisa. Neste sentido, reforça-se a convicção do autor deste trabalho de que uma pesquisa voltada para a educação precisa ser oriunda da prática de sala de aula. Assim como quando Piaget foi até as crianças para entender o desenvolvimento cognitivo, precisa-se ir até elas para ouvir e interagir. Abaixo se apresenta em cinco tópicos as descobertas propiciadas pela experiência-piloto.

1.1 ALFABETIZAÇÃO BÁSICA DE *ETOYS*

Foram necessários dezoito encontros (36 horas-aula) para que os estudantes começassem a ultrapassar as dificuldades que envolviam a aprendizagem do uso do próprio *software*, sendo que se realizou um total de vinte encontros. De certa maneira, isso representou uma frustração, pois no final a equipe ficou com a sensação de que os estudantes estavam prontos para alçarem voos cada vez mais altos com o *Etoys* somente a partir do último encontro.

Mas qual seria o problema neste fato? Por exemplo, havia uma estudante que estava interessada em desenhar e animar estrelas com o auxílio do *Etoys*. Nesse tipo de projeto o foco do desenvolvimento está na matemática. Porém seu foco era desviado, pois a utilização da interface do *software* ainda exigia a maior parte de sua concentração. Dessa forma, as questões de matemática ficavam em segundo plano. A partir deste fato, concluiu-se que é importante, para potencializar o desenvolvimento da proposta, que nos primeiros encontros sejam apresentados pequenas atividades, com objetivos bem específicos para que os estudantes aprendam a diferenciar e utilizar as ferramentas básicas do *Squeak Etoys*.

Além disso, parece relevante a realização de uma pesquisa para investigar esse processo de alfabetização do próprio *Etoys*, já que aprender a utilizá-lo já constitui, por si só, um processo análogo ao próprio desenvolvimento da inteligência humana. Visto que em tal processo há um estágio inicial de indiferenciação em que os estudantes partem de um modelo real (de acordo com o seu sistema de significação). E, a partir de sua ação, vão surgindo novas diferenciações em relação ao projeto inicial que o levam a novas dúvidas, problemas e explicações. Supõe-se que tal processo desencadeia novas possibilidades lógicas e, assim, modifica e amplia o sistema de significação do estudante.

Dessa forma, acompanha-se uma evolução da representação que o estudante tem em relação ao se projeto no *Etoys*. Também suspeita-se que seja possível identificar a evolução das abstrações dos estudantes, de um estágio inicial simples é ultrapassado para outro mais complexo através de

sucessivas tomadas de consciência que o levam a realizar reflexões e reflexionamentos, segundo o processo de Abstração Reflexionante de Piaget (1995).

1.2 A AUTORIA NA CULTURA DIGITAL: GAMES E ANIMAÇÕES

1.2.1 Os Games e a Cultura Digital

Durante a execução deste piloto pôde-se observar que a maioria dos estudantes se interessou em produzir e desenvolver jogos eletrônicos. Na maioria das vezes vislumbravam a possibilidade de recriar jogos clássicos como Tetris, Pacman e diversos tipos de labirintos. Interpreta-se que tal interesse surja como uma necessidade, por parte dos estudantes, de entender a própria cultura digital em que estão imersos. No entanto, o desenvolvimento de jogos impôs uma dificuldade que não foi prevista pela equipe pedagógica: jogos simples como Tetris e Pacman são jogos que já exigem um nível complexo de programação. Somente após a experiência é que foi percebido que iniciar o trabalho com jogos, sem antes promover certo domínio do *software*, não foi a estratégia mais interessante. Por outro lado, pode-se analisar tal fato como um exemplo da complexidade envolvida num projeto que é proposto pelo próprio estudante.

De outra maneira, precisa-se pensar em alternativas para o desenvolvimento de jogos pelos alunos, já que esse parece ser um tema de interesse. Talvez a solução para tal problema seja propor atividades com jogos mais simples que aqueles sugeridos pelos próprios estudantes. Ou seja, apresentar aos estudantes uma série de pequenos desafios de programação no *Etoys*, cuja solução envolva alguma ferramenta importante para o desenvolvimento de jogos. Esses desafios podem ser desenvolvidos nos vinte encontros iniciais, que devem ser voltados para a apropriação dos comandos e dos projetos básicos do *Etoys*.

Durante as reflexões sobre tal experiência uma questão relevante se impôs: a frustração dos alunos nos projetos de games lhes gerou insatisfação em relação ao uso do *Etoys* em sala de aula? No desenvolvimento do projeto-piloto, foi possível perceber que os estudantes ficaram satisfeitos com as aulas de *Etoys*. Sobre esse aspecto identificam-se duas justificativas para tal condução:

A primeira justificativa é que, após o reconhecimento da complexidade que envolvia a programação de jogos, alguns estudantes utilizaram espontaneamente (outros mediante intervenção do professor) a seguinte estratégia de estudo: a programação por partes. Por exemplo: Fazer o

Pacman ser controlado pelas teclas do computador ou fazer as peças do Tetris “caírem” na tela do computador ou fazê-las pararem na base da tela. Apesar de não conseguirem concluir o jogo, esses pequenos desafios serviram como uma forma de estudar e aprender a utilizar o *software*. Sendo assim, observa-se que os estudantes se empenhavam e ficavam felizes quando conseguiam superá-los. Além disso, alguns dos alunos manifestavam certo crescimento na própria autoestima por ter proposto algo que era difícil. Por outro lado, também havia aqueles que abandonavam seus projetos iniciais e se dedicavam a outros, algo comum na pedagogia de projetos.

A segunda justificativa é que mesmo diante da complexidade imposta “pelos jogos” os estudantes estavam conscientes de que era consequência da sua própria escolha, portanto também se sentiam responsáveis pelo problema. Avalia-se que tal feito é decorrente e é um mérito da proposta, pois deu visibilidade ao interesse desses jovens que estão inseridos numa cultura digital. De certa forma, fazer e recriar jogos eletrônicos são maneiras de compreender a sua própria realidade, a cultura digital. Neste contexto, cabe aos docentes e pesquisadores entender a tal fenômeno e torná-los capazes de promover mais aprendizagens dos estudantes.

1.2.2 Autoria e as Animações Quadro a Quadro.

Uma das preocupações ao iniciar o trabalho com o *Squeak Etoys* era propor atividades que animassem os estudantes a programar. Em função disso, desde o início, procurou-se apresentar atividades que pudessem empolgá-los, em vez de explorar atividades que exigissem muitos cálculos e discussão de conceitos específicos da matemática. Da experiência com o LOGO, sabe-se que alguns estudantes desenvolvem certo receio quando isso ocorre.

Uma dessas atividades era ensinar os estudantes a fazerem animações quadro a quadro. Essa proposta foi uma espécie de aposta, pois tal programação é longa e complexa para um nível inicial. Sendo assim, os estudantes aprenderiam a fazê-la de forma mecânica, seguindo um tutorial passo a passo e sem entender o porquê da utilização de alguns dos comandos. Por outro lado, supunha-se que fazer esse tipo de animação qualificaria o desenvolvimento do design dos seus futuros projetos.

Foi no quarto encontro que os estudantes aprenderam a realizar tais animações. Ao contrário do que se esperava muitos deles não gostaram da atividade e não prestaram atenção ao que era ensinado. Naquele momento, avaliou-se que isso ocorria pelos seguintes motivos: a) Alguns estavam interessados nos projetos das aulas anteriores, que era fazer uma corrida com os carros criados por eles; b) Já outros desistiam de fazer tal animação, pois o passo-a-passo desse tipo de

animação é longo e, assim, iam fazer outra coisa no computador. Porém alguns estudantes – poucos no grupo – se interessaram desde o início pela animação quadro a quadro.

Ao avaliar esse encontro, pensou-se que tal aposta fora um fracasso, já que não havia causado o impacto esperado. Além disso, nos encontros seguintes alguns estudantes assumiram tal animação como seu projeto principal. Diante desse contexto, o autor desta pesquisa começou a se perguntar: como se pode promover a aprendizagem de matemática quando o interesse manifestado pelos estudantes foram jogos eletrônicos e animações do tipo quadro a quadro? Em relação à construção de jogos uma possibilidade foi vislumbrada e será apresentada mais adiante. Quanto às animações do tipo quadro a quadro, procurar a matemática envolvida nesses projetos de alunos ainda representa um desafio.

Por um tempo, o cenário dos encontros da experiência-piloto foi o seguinte: alguns estudantes dedicavam-se a desenvolver cada vez mais os projetos que envolviam a corrida de carros, os games, as animações quadro a quadro, os polígonos ao estilo LOGO e, ainda, havia aqueles que não estavam interessados. Além disso, não se conseguia ver algum avanço significativo nas animações propostas pelos estudantes, visto que a cada encontro retomávamos o que havia sido discutido nos encontros anteriores. E tinha-se a impressão que a maioria não estava levando a sério as atividades.

Porém, em uma das aulas um estudante chamou um dos professores para mostrar o que ele havia feito. A execução do projeto estudante causou espanto, pois ele havia criado uma animação com impressionantes 38 quadros, levando em consideração que o editor de imagem do *Etoys* é similar ao *Paint* do *Windows*. O sentimento de felicidade tomou conta, afinal na avaliação dos professores aquele estudante era um daqueles que não pareciam interessados.

Naquele momento não se sabia se ele estava aprendendo matemática, no entanto o estudante havia criado uma trabalhosa animação que lhe exigiu bastante tempo e trabalho. Além disso, seu desenho era autoral, ele criou um personagem para a animação, que parecia ser inspirada em jogos de videogame, pois era uma tartaruga que recebia um raio e explodia.

Nessa altura, começou a acontecer um processo interessante, a cada semana um estudante criava uma animação quadro a quadro que fazia sucesso naquele grupo. Muitos estudantes, naquele encontro começavam a tentar reproduzi-la e, na semana seguinte, muitos tinham a sua versão na sua máquina. No décimo sétimo encontro, as animações quadro a quadro estavam amplamente disseminadas. Todos sabiam criá-las e as apresentavam com orgulho, mesmo quando era a reprodução de outra animação idealizada por um colega. Na nossa avaliação o orgulho estava em se

sentir capaz de criar ou reproduzir uma animação.

Porém, aqueles estudantes que se especializaram em tal animação, e a desenvolveram num nível suficiente para elucidar muitos projetos em ciências. Por exemplo, um dos estudantes desenvolveu uma animação sobre o crescimento das árvores. Tal animação poderia desencadear um processo de aprendizagem sobre conceitos de botânica. Outros desenvolveram animações no estilo HQs.

Como já foi comentado, se sentir capaz de produzir pode transformar a forma como alguém se relaciona com a tecnologia e com as suas pretensões futuras. Abaixo se lê um diálogo entre um professor e a estudante R, que demonstrou um crescimento significativo durante os encontros. Tal conversa ocorreu no último encontro e tinha como intenção saber o que os estudantes estavam pensando a respeito das aulas de matemática com *Etoys*.

A: *Você já tinha alguma experiência com esse tipo de programa? Já tinha criado alguma animação, vídeo ou outra coisa?*

R: *Não, nunca tinha feito animações.*

A: *O que você acha das aulas de Etoys?*

R: *Não sabia usar (o Etoys) há quatro aulas atrás.*

A.: *Como assim?*

R.: *Há quatro aulas atrás não sabia usar o meu tipo de trabalho, que é o desenho animado, o resto eu sabia.*

A.: *É mesmo, então tu te encontrou nessas aulas?*

R.: *“Sim, eu vou ser uma veterinária que vou fazer as minhas próprias propagandas, então não vou gastar dinheiro com isso!”.*

Sua

fala é significativa, pois manifesta: primeiro o sentimento de empoderamento e apropriação da tecnologia e, por último, um novo significado para a presença da tecnologia na sua vida. Essa última ainda é reforçada, pois a aluna disse que fez algumas animações para um trabalho de sua mãe. Infelizmente ela nunca enviou essas animações para os professores.

Embora não se tenha identificado qual matemática pode ser aprendida com a animação quadro a quadro, tal experiência possibilitou àquele grupo de estudantes o sentimento de satisfação e o sentido de apropriação da tecnologia. Como exemplo pode-se citar a conversa com o estudante M. Quando questionado se ele havia gostado das aulas ele respondeu:

M.:*“Tem umas animações no YouTube e eu vi que dá para fazer no Etoys”.*

A.: *Você tentou fazer essas animações?*

M.: *“Não, é que depois que eu comecei a usar o Etoys, dá para entender como são (feitas) essas animações. Tem um monte lá que dá para fazer”.*

De certa maneira, esse fato confirma uma das hipóteses desse trabalho, que, para se sentirem incluídos numa cultura digital, os sujeitos devem ser produtores de tecnologia e não apenas consumidores. Ao aprender a produzir tal animação o olhar desse aluno se modificou, ao ponto de analisar as animações que até então ele estava acostumado a consumir no site YouTube.

1.3 CONCEITOS DE MATEMÁTICA NO *ETOYS*

Durante tal experiência foi possível presenciar uma série de projetos que possibilitavam a aprendizagem de conceitos de matemática. No entanto, não houve tempo para desenvolver (a contento) um trabalho que possibilitasse uma sistematização de tais conceitos e os conteúdos envolvidos. Ao analisar tal situação, avalia-se que um dos possíveis motivos que ocasionaram tal demora foi o foco destinado à apropriação das ferramentas do próprio *Etoys*, afinal de contas, isso foi o tema da maior parte dos encontros. No entanto, supõe-se que tal sistematização evoluiria se o trabalho fosse desenvolvido por mais tempo.

Abaixo serão apresentados alguns projetos realizados pelos estudantes e os conteúdos de matemática que eles podem promover.

1.3.1 Avião na Pista

O estudante A desenvolveu uma simulação de um avião decolando no aeroporto. Esse estudante se enquadra naquele perfil de pessoas que se interessa por problemas de matemática e ciências. Enquanto seus colegas ficavam se dedicando a diversos projetos, o estudante elegeu esse como seu principal foco de estudo, desde os primeiros encontros. Sendo assim, ele incrementava seu projeto a cada nova descoberta, em função disso seu projeto foi o mais completo e complexo. Por exemplo, o estudante utilizou animações pré-construídas no *Etoys* para fazer as luzes ALSF (Approach Lighting System with Sequenced Flashing Lights) que é um conjunto de luzes estroboscópicas que ficam localizadas antes da pista. Nesse projeto em específico estão no final da pista.

Porém as demais animações de seu projeto são simulações de movimentos a partir da variação de sua posição na tela do computador. Tal simulação pode ser dividida em três momentos:

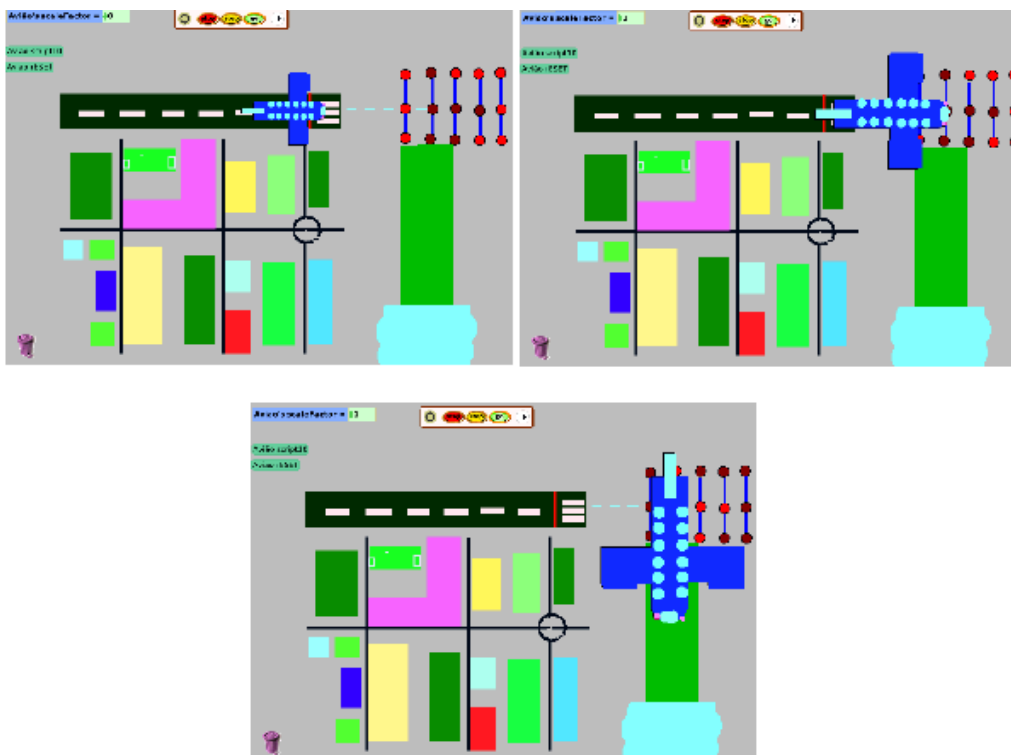
1)O avião movimenta-se horizontalmente durante o comprimento da pista. Para isso ele desenvolveu o “*script4*”.

2)Depois do final da pista o avião aumenta sua velocidade e suas dimensões aumentam segundo um

fator de escala, assim simula o ganho de altitude. A programação envolvida está representada nos “script4” e “script7”.

3) Para finalizar, a aeronave faz uma curva e segue voo (movimentando-se verticalmente) até sumir na nuvem; cujo algoritmo se encontra nos “script6” e “script9”.

Figura 80: Três imagens em sequência da animação Avião na Pista



Sendo assim, a animação é fruto de uma composição de *scripts*, cujas ações são coordenadas entre si. Claro que a composição dessa simulação em três etapas através dos *scripts* necessários para compô-la, não era consciente para o estudante desde o início. Elas foram consequência de um processo de construção contínuo e que foram desenvolvidas e organizadas a partir das diversas ações que o estudante foi realizando ao longo do projeto.

Sendo assim, pode-se afirmar que o primeiro desafio que o estudante A enfrentou após idealizar tal projeto foi a criação do efeito da decolagem. Com a intervenção de um dos professores, o estudante passou a explorar os comandos que estavam nos menus “geometria” e “mais geometria” do *Etoys*. A partir disso, o estudante realizou uma série de experimentações interessantes para obter tal efeito. Primeiro começou a explorar incrementos nos comandos que modificam o comprimento e a largura do objeto. Porém ficou surpreso ao perceber que tais modificações deformavam a forma do objeto. Logo em seguida, percebeu que tanto o comprimento, quanto a largura precisavam

crescer simultaneamente a partir de uma relação que mantivesse a proporção entre tais dimensões. Nesse momento, com a intervenção do professor, o estudante passou a utilizar o comando chamado “Fator de Escala” para obter sucesso com tal efeito. Vale ressaltar que o estudante precisou diferenciar os tipos de transformações possibilitadas pelo fator de escala, pode se utilizar tal comando para: (a) definir um valor fixo de escala, (b) incrementar/decrementar a escala segundo um valor constante e (c) multiplicar a escala do objeto a partir de um valor constante. O segundo promove um crescimento linear de suas dimensões, já o terceiro, um crescimento exponencial.

Cogita-se que, a partir dessa descoberta, seria interessante pedir ao estudante para produzir um material que explicasse como funcionava o comando “fator de escala” e qual era a matemática que estava por trás. Com isso o estudante poderia ampliar seus conhecimentos descobrindo a aplicação de tal conteúdo em outros contextos.

Após conseguir o efeito de decolagem, a dificuldade que o estudante enfrentou foi limitar o crescimento do avião e fazê-lo voltar ao seu tamanho original. Sendo assim, um dos professores lhe ensinou a criar um botão no *Etoys* que teria a função de *Reset* para redefinir as dimensões originais do objeto. Porém a programação dos comandos ficou a cargo do estudante, que levou bastante tempo para conseguir limitar o crescimento do objeto (o avião). Até porque esse problema era um desafio para o estudante e, também, para os professores.

Figura 81: Scripts da animação Avião na Pista



Podemos ver na figura 81 acima que o botão *Reset* ficou mais “recheado” de comandos que definem o estado inicial do avião na simulação. Esses comandos determinam: sua direção inicial, sua posição inicial (através das coordenadas x e y), o fator de escala inicial e o comando “mostrar” faz o objeto ficar visível novamente (já que ele desaparece nas nuvens no fim da simulação).

Isso indica que o estudante A, no final desta experiência, estava começando a utilizar o

sistema de coordenadas cartesianas na sua programação. Por outro lado, a análise de seus *scripts*, indica que esse foi um fato isolado, já que ele não utiliza o sistema de coordenadas cartesianas na programação dos demais movimentos do avião. Diante disso, questionou-se se o motivo disso foi o tempo insuficiente do projeto, ou o interesse do estudante ou um descuido da proposta realizada?

De qualquer maneira, tal projeto indica a possibilidade de desenvolver os conceitos de matemática relativos a:

- Proporcionalidade de figuras planas;
- Escalas e fator de escalas;
- Sistema de Coordenadas cartesianas;

1.3.2 Dança das Mimosas.

Outro exemplo interessante de projeto que propiciou a aprendizagem de conceitos de matemática foi proposto pela estudante B que, motivada em fazer animações quadro a quadro, fez uma divertida animação intitulada “Dança das Mimosas”. Nela havia a animação de um relógio analógico. Tendo essa animação como referência, sugeriu-se que ela criasse uma simulação de um relógio, com os movimentos dos ponteiros sincronizados com o relógio de seu *notebook*.

Ao aceitar tal desafio, a estudante passou as duas semanas seguintes pensando em como realizar tal simulação. Sua primeira dificuldade foi ultrapassar sua visão estática entre os três ponteiros do relógio, visto que na animação quadro a quadro ela desenha os três numa posição estática que era variada ao longo dos quadros. Com o objetivo de agilizar a produção da estudante, os professores poderiam ter ensinado a aluna a criar cada ponteiro como um objeto independente e como animá-los. No entanto, optou-se em não direcionar suas ações, mas deixar que ela resolvesse tal problema. Essa intervenção, não foi por acaso, pois os professores entendiam que essa descoberta deveria ser uma conquista da aluna, o que de fato se confirmou. Logo em seguida, a estudante B começou a trabalhar no movimento individual dos ponteiros das horas e dos minutos e, num segundo momento, na relação entre os movimentos entre eles.

Todavia, entende-se que deste ponto em diante o problema deixa de ser de programação e passa a ser exclusivamente de matemática, pois a relação entre os movimentos dos ponteiros pode ser vista como matemática. Infelizmente, não houve tempo hábil para a aluna desenvolver e concluir a animação do relógio. No entanto, este projeto indicou uma possibilidade de desenvolver os seguintes conceitos comuns à matemática e à física.

- Ângulos;

- Movimento Circular;
- Proporção nos movimentos circulares: (período, frequência);
- Velocidade angular;

É válido evidenciar que o movimento circular, no currículo tradicional, é um conteúdo do segundo ano do Ensino Médio, específico da Física, portanto é ensinado como se fosse desvinculado da matemática. Parece-nos que as razões matemáticas envolvidas nesse tipo de problema são uma excelente oportunidade para se ampliar o conceito de proporção estudada tradicionalmente na disciplina de matemática do Ensino fundamental. Ademais, nesse contexto, entende-se que os estudantes das séries finais do Ensino Fundamental têm condições de criar tal simulação com êxito e, assim, desenvolver abstrações que favoreçam a compreensão de tal conteúdo. Além disso, para fazer tal animação, os alunos precisarão realizar cálculos que envolvem uma proporção de ângulos, que é algo mais complexo que a proporção simples e direta estudada no ensino fundamental.

Dessa forma, esta simulação é uma ideia poderosa da matemática, pois tal conhecimento permite a simulação de uma série de fenômenos naturais, tais como: sistema solar, modelos atômicos, movimentos circulares. Experiências que possibilitam o desenvolvimento de um trabalho de funções trigonométricas.

Figura 82: Duas animações de relógios diferentes



Segundo Moraes, Fagundes e Basso (2013a) Os dados analisados neste estudo piloto mostram efetivamente que muitos conceitos de matemática tornam-se necessários aos estudantes durante as suas construções e programações. Nele é relevante destacar que as necessidades são apontadas pelos estudantes e não pelo professor de matemática, como destacado nas análises dos projetos “Avião na Pista” e “Dança das Mimosas”. Segundo os autores essa mudança, que evidencia uma transformação no papel do aluno e na função do professor, é estruturante da escola da cultura

digital. Avalia-se que essa ação do estudante transforma o seu sistema de significação de modo a construir novas relações entre os conceitos, novos conhecimentos e novas ou melhores estruturas cognitivas. Por fim, o estudo piloto indica que a estratégia pedagógica proposta é relevante para a Educação matemática, contribuindo para que os estudantes se mobilizem cognitivamente para aprender matemática.

1.4 DESENVOLVIMENTO DO RACIOCÍNIO LÓGICO-MATEMÁTICO ATRAVÉS DO USO DE TESTES NO *ETOYS*

Ao relembrar a pergunta que norteia esta pesquisa – como se pode aprender matemática a partir dos projetos desenvolvidos no *Squeak Etoys?* – um tipo específico de matemática se destaca: o raciocínio lógico-matemático. Avalia-se que isso ocorra em função de sua onipresença no ato de programar. Embora o estudo da lógica não faça parte dos currículos educacionais faz algum tempo, tal conhecimento é fundamental para o desenvolvimento da própria matemática demonstrativa, das ciências e da informática. Além disso, num contexto em que a escola esteja incluída na cultura digital, foram observados indícios de que tal conhecimento se torna necessário, pois serve de base para que os estudantes consigam entender e produzir tecnologia.

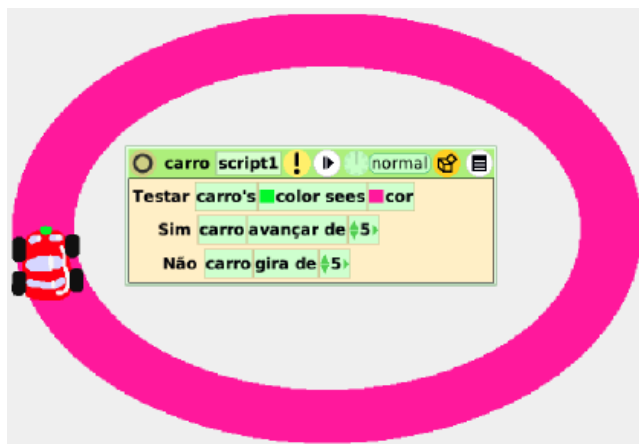
Neste contexto, questiona-se como a criança se depara com a lógica no contexto do *Etoys?* Com a finalidade de compreender as experiências vividas, serão descritos alguns dos relatórios feitos durante a execução do projeto-piloto.

Relato do 3º encontro: Utilizando *Teste* de cor na programação. **Data:** 27/08/2012

Como atividade os estudantes deveriam criar um objeto que se movimenta dentro de uma estrada ou pista. Para isso é necessário criar um algoritmo para reconhecer quando esse objeto está sobre a pista, ou não. Uma das formas de conseguir tal efeito é criar um sensor virtual, cuja programação faz uso do comando *Teste*, por exemplo: Para andar na pista rosa cria-se no carro um ponto verde que será o sensor. Logo em seguida, programa-se o *Teste* da seguinte forma: na linha *Testar* insere-se o comando de reconhecimento de cor. Isso pode ser traduzido através da seguinte pergunta: *a cor verde vê a cor rosa?* Se a resposta for *Sim*, o carro deve avançar 5, mas se for *Não* o

carro deve girar 5 graus.

Figura 83: Script para o carro reconhecer a pista rosa



A principal intenção era que os estudantes continuassem motivados com suas produções no *Etoys* ao programar um carro para “andar” numa pista, pois esse é um comportamento que as crianças avaliavam como sendo interessante. Além disso, julgava-se que o uso do comando *Teste* contribui para o desenvolvimento do raciocínio lógico-matemático dos próprios estudantes, já que seu emprego supõe uma implicação condicional do tipo “se, então”, que segundo Inhelder e Piaget (1976) é um raciocínio característico do pensamento operatório formal, pois já indica uma natureza hipotética-dedutiva. Além disso, na perspectiva do aprender a programar o uso de *Testes* é fundamental, pois a partir da experiência dos professores nas oficinas de *Etoys* realizadas pelo LEC/UFRGS tinha-se observado que tal recurso lógico possibilita o desenvolvimento de jogos, simulações e animações cada vez mais complexas e interessantes.

De outra forma, supunha-se que a compreensão dos *Testes* se relaciona com o estágio do desenvolvimento cognitivo em que a criança se encontra. Uma evidência de tal fato é que as crianças das séries iniciais têm mais dificuldades de resolver os problemas que surgem durante desenvolvimento de seu projeto quando necessitam da utilização dos *Testes*. Mesmo quando se explica a lógica envolvida, tal dificuldade persiste.

Como era esperado, tal atividade gerou interesse, pois alguns já haviam manifestado curiosidade por essa animação no primeiro encontro. Para ensiná-los a utilizar o *Teste* desenvolveu-se um exemplo passo a passo que foi projetado no quadro-negro enquanto os estudantes o reproduziam nos seus próprios *laptops*.

De início foi solicitado que eles criassem uma pista e dois objetos para que em seguida, fizessem o carro andar e girar ao mesmo tempo. Nesse momento pôde-se observar que eles haviam esquecido alguns princípios básicos do *Etoys*, tais como:

1) Desenhar a pista e os dois carros de uma única vez no editor de imagem, criando assim um único

objeto. É interessante perceber que isso só se configura um problema quando eles executam o *script* que tem como objetivo fazer o carro andar. Quando a pista também se movimenta ocorre um desequilíbrio, desencadeando um processo de diferenciação sobre a ação de criar um objeto através do editor de imagem. Supõe-se que este é um dos primeiros processos de diferenciação que os estudantes precisam tomar consciência.

2) Como editar um objeto: Os estudantes haviam esquecido como se fazia para criar um *script* (clicando no botão visualizador), modificar sua aparência (clicando no botão lápis), que poderiam duplicá-lo etc.

Após superar tais dificuldades mostrou-se como utilizar o comando *Teste*, como mostra a figura 84 abaixo.

Figura 84: Script para reconhecer a pista rosa



É importante ressaltar que o campo *Não* permaneceu vazio, com o objetivo de gerar um desequilíbrio no grupo, procurando assim desencadear um processo de aprendizagem sobre a lógica inerente ao comando *Teste*.

Enquanto alguns festejavam o seu sucesso, outros apresentaram os seguintes “bugs”:

Cores de mesma tonalidade



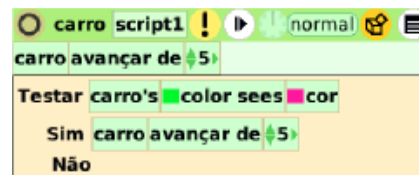
Ao construir o sensor, alguns alunos o criaram na cor vermelha, mas numa outra tonalidade daquele que o programa coloca como exemplo, no comando de reconhecimento de cores. Sendo assim, para o sensor, elas são de cores diferentes, portanto não havia o reconhecimento da cor e o carro continuava parado. Para resolver tal problema é necessário que o estudante utilize a pipeta para selecionar a cor desejada, o que foi ensinado.

Indiferenciação da sequência de comandos no Script

Uma estudante solicitou ajuda, pois “o carro estava saindo da pista e não parava de andar”. Ao analisar o seu *script* (que está representado na figura 85 abaixo) pode-se perceber que a estudante não havia entendido que o objeto executa o *script* de forma sequencial, ou seja, na ordem em que os

comandos são apresentados no interior do *script*.

Figura 85: Ordem dos comandos no *script*



Pela figura 85 acima pode-se perceber que os procedimentos realizados serão: a) avançar 5 e b) executar a condição apresentada no *Teste*. Nesse caso o carro vai andar independente do resultado do *Teste*.

Quando o problema foi compreendido, argumentou-se com o estudante C:

Professor 1 (P1): – Tu sabes que o computador lê e executa o *script* de cima para baixo, sendo assim quando ele ler o teu *script* o que ele fará primeiro?

Estudante C: – O *Teste*.

P1: – Tu viste que têm um comando antes do *Teste*?

Estudante C: – Bah é mesmo! Vô tirar, então. Isso mesmo

P1: – Vamos ver se dá certo?

Estudante C: – Agora deu certo, mas ele tá parando e não está girando.

P1: – Um problema de cada vez. Tu percebeste que ele estava andando independente da condição aplicada pelo *Teste*, por isso que ele não parava.

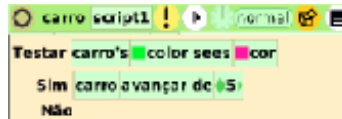
Foi só no encontro seguinte que se percebeu que a intervenção poderia ter sido melhor. Poderia ter solicitado para ela analisar o *script*, refazendo comando por comando com papel e caneta. Talvez ela tivesse percebido o problema por ela mesma e sugerido até outra solução. E seria uma conquista intelectual dela, já que ela própria descobriria uma maneira de contornar tal obstáculo, ao invés da tentativa (do professor) de tornar tal processo mais eficiente e não do professor, assim como sugere Papert (1994, p. 112).

Proposição Condicional

Embora a explicação da lógica envolvida no comando *Teste* tivesse sido feita previamente, não havia a crença de que os estudantes iriam entendê-lo de imediato. Desde o início tinha-se consciência de que o desenvolvimento de tal lógica se daria à medida que os estudantes utilizassem os *Testes* nos seus projetos, ou seja, seria agindo sobre os objetos de seus projetos que eles compreenderiam a lógica do *Se e então*.

Continuando com o diálogo entre um dos professores e a estudante C, abaixo apresenta-se como ficou o *script* após sua modificação.

Figura 86: Script estudante B



Estudante C: – Agora deu certo, mas ele tá parando e não está girando.

P1: – Vamos analisar o que está escrito comando Teste. Em Testar é como se o objeto perguntasse: O verde vê a cor rosa? Se a resposta para essa pergunta for sim, o que vai acontecer?

Estudante C: – Ele vai andar.

P1: – É o que está escrito na condição “Sim”. Mas se a resposta for Não?

Estudante C: – Ele tem que girar, mas não tá!

P1: – E onde está dito isso para o objeto?

Estudante C: – Hã?

P1: – No script, onde está escrito que ele deve girar?

Estudante C: – Ah, entendi.

Sendo assim, a estudante colocou o comando *girar 5* no *Não* do *Teste* e o carro passou a andar como era o esperado.

Diferentes Scripts e interpretações para o comando Teste

Quarto encontro: Projeto com *Teste*

Data: 03/09/2012

Como atividade os estudantes deveriam criar uma animação que fizesse uso dos *Testes*. Tinha-se como objetivo promover a tomada de consciência da lógica condicional a partir da aplicação dos *Testes* em diferentes situações. Como pesquisadores, pretendia-se identificar o que os estudantes haviam entendido em relação à lógica inerente ao uso de *Testes*, quais eram suas hipóteses.

Sendo assim, a aula iniciou com um dos professores (diferente do primeiro encontro) conversando com o grupo, no qual ele perguntou o que haviam feito na aula passada e um aluno respondeu:

Estudante D: – Fizemos um carro andar numa pista.

Professor 2 (P2): – Mas como fizeram isso?

Estudante E: – Usando um negócio lá que o professor mostrou.

P2: – Um negócio? Mas como era?

Estudante E: – Um negócio que usava umas cores?

P2: – Um negócio que usava umas cores... Nós temos uma certa dificuldade em utilizar as palavras pra explicar o que fizemos, né? Mas isso acontece porque, quando aprendemos a programar, estamos aprendendo uma linguagem, assim como outras.

Embora essa fala do professor fosse importante, observou-se que os estudantes não deram muita importância, estavam dispersivos, porém entenderam que o comentário era referente ao seu

vocabulário.

Estudante F: – Um sensor! Eu sei, nós criamos um sensor para ele ver se o carro estava na pista.

P2: – Mas como ele reconhecia a pista?

Estudante E: – Há, se o sensor estava em cima da pista ele andava, se ele não estava ele parava. (“não, ele girava” disse outro colega). Isso ele girava e não saía da pista.

P2: – Mas o que o faz reconhecer a pista?

Estudante E: – O sensor.

P2: – Então vocês criaram o sensor e depois?

(os estudantes ficaram sem entender a pergunta)

P2: – Mas como o sensor funciona? O que se tem que fazer?

Estudante D: – Com o negócio das cores, quando ele vê a cor da pista ele anda, quando não vê, gira.

Nesse momento, percebeu-se que os estudantes tinham como hipótese que era o sensor que realiza a distinção das cores e não a aplicação do *Teste*. Ou seja, não é necessário criar um “sensor” (um pontinho colorido) para o reconhecimento de cores, mas sim utilizar um *Teste*.

Voltando ao relato, uma evidência dessa indiferenciação dos estudantes, foi quando uma colega disse em voz alta: - *Não precisa de sensor, eu não criei um e o meu está funcionando!* Era verdade, na aula anterior, a menina não havia construído um sensor, ela criou um bicho estilo “Pacman” todo vermelho e utilizou a cor do próprio corpo do objeto para o *Teste*. Infelizmente sua fala passou despercebida tanto pelos estudantes, quanto por dois dos professores. Só depois do ocorrido percebeu-se que essa seria uma importante oportunidade para discutir a função do sensor e do *Teste* na atividade.

Logo em seguida, o professor que conduzia a atividade teve uma excelente ideia para ajudá-los na discussão, ele projetou no quadro um *script* (figura 87 abaixo) como exemplo e pediu que os estudantes explicassem o que eles haviam feito:

Figura 87: Teste apresentado para os estudantes



F: – Assim, quando o azul vê a cor branca ele anda, se ele não vê, ele gira.

Nesse momento percebeu-se que o julgamento das crianças estava preso à experiência vivenciada na aula passada (ao passo-a-passo) e não ao que estava sendo analisado naquele momento. Além de não saberem diferenciar as três condições que compõem o comando *Teste* (*Testar*, *Sim* e *Não*). Procurando a compreensão por parte dos estudantes, o professor fez a seguinte

intervenção.

P2: – Olha só, isso aqui envolve um raciocínio condicional. Qual é a condição nesse caso? Se a cor azul vê a cor branca. Se isso se confirma, então o carro deve fazer alguma coisa. Por este programa o que ele vai fazer?

Estudante D: – Ele vai andar.

P2: – É isso que está escrito aqui no Sim? (silêncio e incompreensão) e o que acontece quando a condição não é atendida? Ele gira 160° e alguém sabe o que significa random, os meninos já viram em algum lugar?

Estudante D: – Sim, sorteio aleatório.

P2: – Isso mesmo, então quando o azul não vê o branco ele gira aleatoriamente. Nesse script o carro sempre anda, pois o avançar não depende do Teste, e ele gira quando não está no branco.

Alguns estudantes apresentaram uma situação que, segundo eles, “o carro ficava louco”. (figura 88 abaixo). Sendo assim o professor 2 perguntou:

Figura 88: Bug apresentado pelos estudantes



P2: – Por que o carro apresenta esse comportamento estranho? Será que tem erro na programação? Uma estudante responde:

Estudante D: – Não, está certo. É que ele só vai andar no branco, como ele está no rosa (a nuvem) ele só fica girando, sem andar. Ele está preso.

No entanto, avalia-se que essa análise só foi possível após a discussão iniciada pelos professores e não pela compreensão dos comandos apresentados no *script* por parte dos estudantes.

Uma breve análise:

Tomando como exemplo o “bug” proposto pelos estudantes, far-se-á uma breve análise sobre os diferentes tipos de algoritmos utilizados pelos professores ou estudantes. Além da indiferenciação dos componentes do comando *Teste* e de estarem reproduzindo o exemplo da aula passada, acredita-se que a dificuldade apresentada pelos estudantes se deve ao nível de dificuldade distinta nos dos dois exemplos de *scripts* explorados pelos professores.

SCRIPT 1:

Figura 89: Script 1 proposto pelo primeiro professor



Neste primeiro *script* percebe-se uma aplicação direta do raciocínio condicional, pois o carro só anda se a condição for afirmativa e ele gira se a condição é negada. Afinal de contas a intenção é fazê-lo andar em cima da cor do mundo. Agora, caso isso não ocorra (quando encontra a nuvem rosa), há a negação da condição inicial, o carro girará.

SCRIPT 2:

Já no segundo exemplo, o *script 2* explora apenas a negação da condição: se “a cor azul não vê a cor branca”, o carro gira; ou seja o carro anda independente da condição do *Teste*.

Figura 90: Script 2 proposto pelo segundo professor



Logo não é uma aplicação direta da condição inicial, mas sua negação, que segundo Piaget (1995) a utilização de tal raciocínio impõem maior dificuldade às crianças já que a condição “não ver a cor branca” explora o atributo não branco das nuvens coloridas (local onde o carro não deve se movimentar). Sendo assim, tal programação é logicamente mais complexa para os estudantes que aquela do primeiro *script*.

Neste contexto, analisa-se que o resultado poderia ter sido melhor, se, desde o início, estivesse claro para os professores quais eram os raciocínios lógicos envolvidos nos *Testes* produzidos.

Além disso, também foi constatado que os dois professores utilizaram interpretações diferentes para o uso do comando. Este fato foi evidenciado nas apresentações feitas pelos professores:

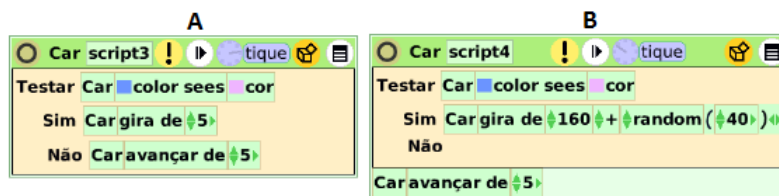
Professor 1: *Em Testar é como se o objeto perguntasse: O verde vê a cor rosa? Se a resposta para essa pergunta for Sim... se a resposta for Não...*

Professor 2: *Isso aqui envolve um raciocínio condicional. Qual é a condição nesse caso? Se a cor azul vê a cor branca. Se isso (condição Sim) se confirma, então se a condição Não se confirma,*

então...

Será que essa diferença de interpretação atrapalhou os estudantes? Por outro lado, essa experiência indica que a forma como o estudante interpreta a ação a ser realizada pelo objeto influencia na forma que os comandos serão apresentados no *script*. Tal afirmação é feita, pois foi possível observar que os estudantes que produziam *scripts* seguindo uma lógica diferente da apresentada pelos professores. Para eles o carro deveria reconhecer a nuvem e não o mundo. Dessa maneira os algoritmos produzidos por eles eram diferentes dos anteriores, como se pode ver na figura 91A abaixo.

Figura 91: Scripts equivalentes aos anteriores



Segundo esse *Script* o carro gira quando ele “vê” a cor rosa e avança quando não a “vê”. Logo o resultado do *Script 3* é idêntico àquele produzido pelo *Script 1*, em função disso pode-se afirmar que são equivalentes, no entanto o raciocínio lógico empregado é distinto, ou seja, são recíprocos. Diante dessa constatação surgiu a curiosidade de se produzir o algoritmo análogo ao *script 2*, apresentado na figura 91B. Nele o carro avança independente do *Teste* e gira quando a condição é confirmada, ou seja, quando o carro “vê” a cor rosa. É pertinente ressaltar que durante a realização do projeto-piloto não foi observada uma construção desse tipo pelos estudantes, portanto o *Script 4* foi criado para esta análise.

Apesar de ter-se observado a equivalência entre os algoritmos, não se teve evidências para constatar se os estudantes tomaram consciência da equivalência de algoritmos produzidos por eles. Geralmente eles produziam um *script* até conseguir o comportamento desejado, mas não se pôde observar se eles exploravam a produção de diferentes algoritmos que reproduzissem o mesmo comportamento. Esse tipo de exercício pode ser promovido nas atividades desenvolvidas na coleta de dados para a pesquisa.

Conforme Morais, Fagundes e Basso (2013c) tomar consciência das variáveis que estão envolvidas nesse processo é algo importante e útil para qualquer professor ou pesquisador interessado. Como foi visto, a forma como professores e estudantes interpretam a ação a ser realizada por um objeto no *Etoys* é distinta e isso influencia na forma como os comandos serão

apresentados no *script*. Dessa maneira, a prática de analisar os *scripts* equivalentes nas produções dos estudantes parece algo importante a ser estimulado ao longo das aulas, visto que isso pode contribuir para que os estudantes compreendam a lógica inerente aos algoritmos.

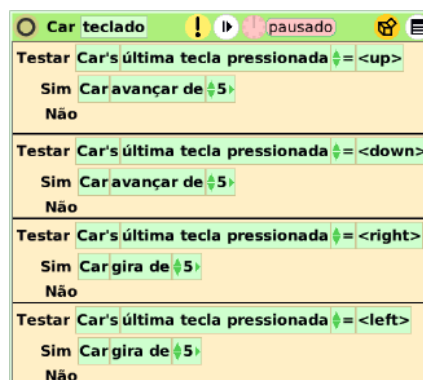
Quinto encontro: Projeto com *Teste* continuação

Data: 10/09/2010

Neste encontro os estudantes continuaram a desenvolver uma animação que necessitasse de *Testes*. O objetivo era promover a compreensão do *Teste* por parte dos estudantes e observar a forma como eles utilizariam/adaptariam o modelo de uso do *Teste* em diferentes situações demandadas pelo seu projeto. Pretendia-se, assim, identificar ações e hipóteses sobre o raciocínio condicional.

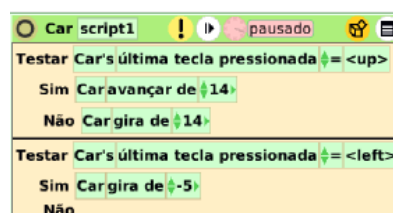
Logo no início da aula um estudante solicitou ajuda para resolver um problema. Seu projeto era fazer um carro de corrida controlável pelos teclados do computador. Para esse tipo de programação é preciso criar um *Teste* para cada tecla que se deseja utilizar, como vemos na figura 92 abaixo.

Figura 92: Script para controlar o carro com o teclado



Nessa experiência-piloto isso não surgiu de forma espontânea, foi preciso indicar aos estudantes que é possível inserir num único *script* quantos *Testes* se deseja. Na figura 93 abaixo se pode ver a programação realizada pelo estudante.

Figura 93: Script para controlar o carro com o teclado



Pelo *script* é possível analisar que há dois *Testes*: um para a tecla direcional *Up* e outro para a tecla direcional *Left*. O primeiro *Teste* (para a tecla *Up*) estava programado da seguinte maneira:

se a última tecla pressionada fosse a tecla *Up*, o carro avançaria 14 pixels, se não, giraria 14 graus. Já a programação do segundo *Teste* (para a tecla *Left*): se a última tecla pressionada fosse a tecla *Left*, o carro giraria – 5 graus.

Supõe-se que o estudante pensava que bastava inserir quatro comandos no *script* (um para cada tecla) para fazer tal programação. Sendo assim, apenas dois *Testes* seriam suficientes. Porém o estudante não havia percebido que segundo o *script* da figura 93 acima, ao pressionar a tecla *Left* não estaria pressionando a tecla *Up*, portanto o computador executaria duas ações simultâneas: girar –5 e girar 14. Isso era equivalente a girar: $-5 + 14 = 9$. Portanto o movimento resultante era um giro de 9 graus para a direita. No entanto, o professor não tomou consciência desses fatos de início, eles só foram percebidos durante a conversa com o estudante e que será descrita a seguir.

Estudante E: – Professor, dá uma olhada no meu *script*, ele tá com um problema.

Professor I: – Mas me explica o que tu quer que o carro faça.

Estudante E: – Eu quero controlar o carro com o teclado para estacionar ele na garagem.

PI: – E como funciona esse *Teste* que você criou?

Estudante E: – Quando apertar essa tecla (indica com seu dedo a tecla direcional *Up*) carro tem que andar pra frente, quando apertar essa tecla (indicando a tecla *Right*), o carro tem que girar para a direita e quando apertar a tecla da esquerda ele tem que girar para o outro lado. Essas duas estão funcionando (*Up* e *Right*), mas eu não entendo o porquê que ele gira mais devagar e para o lado errado (para a direita). Eu olhei, procurei, mas não consegui entender o porquê disso.

Num primeiro momento dois fatos chamaram a atenção na sua fala:

a) Se a sua preocupação era “girar mais devagar”, então o estudante não diferenciava (no seu *script*) que ele havia atribuído medidas de giro diferentes: 14° graus para direita e –5° graus à esquerda. Avaliou-se que uma pequena conversa resolveria sua principal dúvida.

b) Ele havia dito que as teclas *Up* e *Right* estavam funcionando bem, no entanto, ao analisar o *script*, isso não deveria ter acontecido, pois ele não havia programado a tecla *Right* do teclado, apenas as teclas *Up* e *Left*. Em função disso, foi intrigante o fato de o estudante não perceber que tinha um problema no seu *script*. Será que tal erro não aparecia no comportamento do carro? Sendo assim, foi realizada a seguinte intervenção:

PI: Vamos analisar o teu *script* tecla por tecla. Me diz qual a condição a ser testada e o que está escrito na condição Sim e Não?

Estudante E: Se apertar a tecla para cima, o carro vai andar para frente 14, mas essa está funcionando. No Sim diz para ir pra frente e no Não diz para girar 14 para direita.

PI: Mas você tinha dito que ele giraria para direita quando apertasse a tecla *Right*?

Estudante E: Pois é (risos), mas o importante é que tá funcionando.

PI: Ok, mas você entendeu porque isso está acontecendo?

Estudante E: Não.

PI: Da forma que está programado você pode fazer o carro girar para direita pressionando qualquer tecla do teclado com exceção da tecla *Up*.

Dessa maneira o estudante testou o que o professor havia dito com várias teclas. Porém disse que não havia

entendido o porquê daquilo. Sendo assim o professor lhe explicou o motivo e sugeriu que ele criasse um *Teste* para cada tecla. Ficando isso claro, conversou-se sobre o segundo *Teste* do *script*.

PI: – E quanto ao problema do giro, o que faz um carro girar mais rápido ou devagar?

Estudante E: – O número que se coloca, quanto mais alto o número mais rápido ele vai girar e quanto mais baixo mais devagar ele gira.

PI: – E seu eu diminuir esse valor ao ponto de ele ser negativo?

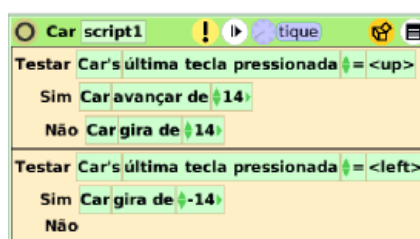
Estudante E: – Ah, daí ele gira para o outro lado.

PI: – Pois é, e se você quer que ele gire para lados contrários, mas na mesma velocidade o que você precisa fazer?

Estudante E: – Ah, colocar o mesmo valor. Mas já fiz isso e não adianta.

A: – Tu podes modificar para vermos o que acontece? Abaixo se apresenta a alteração feita pelo estudante.

Figura 94: Modificação no *script* feita pelo estudante



Estudante E: – Tá aí, ele para.

Ao estranhar tal resultado o professor analisou com mais calma o *script* produzido pelo estudante e, assim, tomou consciência da composição de ações e percebeu o que estava acontecendo. O carro não girava, pois ao pressionar a tecla *Left*, duas ações eram executadas: girar -14 e girar 14 , o que é equivalente a girar: $-14 + 14 = 0$. Após compreender isso, foi explicado ao estudante o que estava acontecendo e ele conseguiu realizar a programação desejada.

A partir desse exemplo pode-se identificar que a necessidade de construir um *Teste* para cada tecla não surgiu de forma espontânea na programação dos estudantes, foi necessário a intervenção de um dos professores. Supõe-se que isso ocorra, pois tal programação exige a realização de uma operação lógica até então não utilizada nos *scripts* produzidos pelos estudantes, a disjunção. Pode-se analisar logicamente o *script* da figura 94 da seguinte maneira: o carro irá se movimentar se a última tecla pressionada for a tecla *Up* OU a tecla *Down* OU a tecla *Right* OU a tecla *Left*. Basta um desses atributos/ações ser verdadeiro que o objeto realizará ação programada.

Quando um *script* possui apenas um *Teste* o objeto executará uma ação se a condição for verdadeira ou falsa, ou seja, envolve apenas a confirmação de um único atributo. Já quando há mais de um *Teste* a programação envolve a operação entre conjuntos, portanto sua complexidade será maior. No caso do *script* produzido pelo estudante (figura 94) a disjunção utilizada promoveu a composição de ações realizadas pelo objeto e isso dificultou a compreensão tanto do estudante,

quanto, momentaneamente, do professor. Além disso, para a compreensão do *script* era necessário a compreensão da adição com números positivos e negativos. Nesse caso, segundo Morais, Fagundes e Basso (2013b) a operação lógica da disjunção promoveu uma situação interessante para o estudante compreender as operações matemáticas com números inteiros. De certa maneira isso confirma que o *Etoys* é um micromundo da matemática.

Após esse encontro os professores ficaram na expectativa se nos projetos desenvolvidos pelos estudantes surgiria a necessidade de se realizar as outras operações lógicas, em que a conjunção era a mais esperada. No entanto, não foi observada a criação desse tipo de operação lógica, tal fato foi surpresa, pois a conjunção é bem comum na programação. Segundo Morais, Fagundes e Basso (2013b) pode-se concluir que o grau de complexidade desse tipo de *script* também é maior, o que poderia justificar o fato de tal tipo algoritmo não ter surgido nos projetos dos estudantes.

Vale ressaltar que na experiência-piloto não foi desenvolvida uma atividade que promovesse tal surgimento, visto que um de seus objetivos era identificar o que surge de forma espontânea ou quando é necessária a intervenção do professor ou de alguém com maior domínio da linguagem de programação. O interesse de controlar o carro pelo teclado foi proposto pelos estudantes, em função disso foi ensinado como se fazia tal tipo de programação.

Mas como ficaria a programação em *Etoys* que envolva a conjunção? Na figura 95 a baixo apresenta-se uma situação na qual tal programação é necessária. A situação é a seguinte: o rato que está passeando pelo mundo deve comer as comidas da cor azul que estão na parte cinza do mundo. Para produzir tal efeito ao invés de programar o rato para identificar a comida e apagá-la, programa-se a comida. Tal estratégia foi utilizada em diferentes projetos com as crianças, que num primeiro momento, apresentaram dificuldade em identificar tal inversão, mas que foi superada ao longo dos encontros. Essa foi a única maneira que os professores conseguiram desenvolver um *script* para produzir tal efeito.

Figura 95: Script com a operação lógica de conjunção

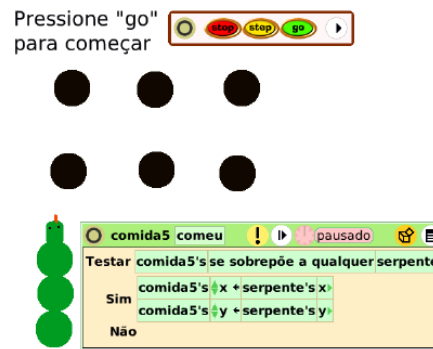


Sendo assim, o algoritmo do *script* “apagar” pode ser descrito da seguinte maneira: se o objeto *comida* estiver sobre a cor cinza E o objeto rato, então o objeto comida é apagado. Dessa maneira ao executar os *scripts* de todas as “comidas” o *Teste* será executado, porém quando o rato se sobrepuser às peças azuis que estão na região cinza (transformando as duas condições verdadeiras) tal comida sumirá dando a impressão que foi o rato a “comeu” ao encontrá-la.

Embora o grau de complexidade desse tipo de *script* seja maior – o que poderia justificar o fato de tal tipo algoritmo não ter surgido nos projetos dos estudantes – a falta da necessidade de se utilizar uma conjunção na programação dos projetos foi uma surpresa. Diante disso, surge a necessidade de investigar com mais cuidado como esse processo de construção se dá nas crianças que aprendem a programar no *Squeak Etoys*.

Além disso, outro aspecto que surpreendeu os professores foi em relação à negação, pois foi observado que, com exceção do algoritmo da figura 93, os *Testes* desenvolvidos pelos estudantes não exploravam a negação. Dito de outra maneira, ao desenvolver um algoritmo fazendo uso do comando *Teste* do *Etoys* os estudantes não utilizavam a linha *Não*, apenas a linha *Sim*, ou seja, as ações dos objetos só ocorriam quando a condição do *Teste* é verdadeira e nunca falsa. Como exemplo pode-se analisar todos os *Testes* utilizados na animação *Aviação na Pista* (Figura 96), nele a linha *Não* está vazia em todos os *Scripts*.

Figura 96: Script do Jogo da Cobra



Na figura 96 acima temos outro exemplo de *script* que não faz uso da negação, nele a estudante procurou desenvolver uma versão do clássico jogo da cobra. Nesse jogo cada vez que a cobra “come” uma comida seu comprimento aumenta. Para reproduzir tal efeito a estudante desenvolveu o seguinte *script*: quando o objeto *cobra* sobrepõe ao objeto *comida* esse assume as coordenadas da *cobra*. Dessa maneira ela passa a movimentar-se junto com a cobra. É importante ressaltar que esse foi o desafio que a estudante encarou durante os encontros. Em função da dificuldade encontrada pela estudante, o jogo da cobra não foi concluído; no entanto esse é mais um exemplo em que a estudante ficou satisfeita com sua produção mesmo diante das dificuldades impostas pela natureza do projeto.

Claro que nem em todos os algoritmos a utilização da negação é necessária, no entanto torna-se relevante pesquisar quais são as situações em que poderia ter sido utilizado, mas não foi e o porquê disso. Evidencia-se aqui a necessidade de investigar as crenças e hipóteses dos estudantes em relação às operações lógicas.