

Pontifícia Universidade Católica de São Paulo
Programa de Pós-graduação em Tecnologias da Inteligência e
Design Digital

Sandra Muniz Bozolan

O Pensamento Computacional:
Ensino e Aprendizagem através do
Software Processing

Dissertação de Mestrado em
Tecnologias da Inteligência e Design Digital

São Paulo

2016

Sandra Muniz Bozolan

O Pensamento Computacional:
Ensino e Aprendizagem através do
Software Processing

Dissertação apresentada à Banca Examinadora, exigência parcial para obtenção do título de Mestre em Tecnologias da Inteligência e Design Digital – área de concentração em Processos Cognitivos e Ambientes Digitais; Linha de Pesquisa Design digital e inteligência coletiva – pela Pontifícia Universidade Católica de São Paulo, sob orientação do Prof. Dr. Hermes Renato Hildebrand.

São Paulo

2016

Banca examinadora

AGRADECIMENTOS

Gostaria de agradecer primeiramente ao meu professor e orientador Dr. Hermes Renato Hildebrand, por apostar no meu projeto desde o nosso primeiro contato no Programa de Estudos Pós-Graduados em Tecnologias da Inteligência e Design Digital da PUC-SP. Estendo meus agradecimentos à PUC-SP pela oportunidade de estudar design, complexidade, arte e tecnologia em apenas um lugar. E também à CAPES pelo apoio que deu aos meus estudos com a bolsa que recebi.

As inspirações para este trabalho foram muitas e a maioria delas se tornou evidente ao longo das discussões propostas nesta pesquisa. No entanto, minha maior inspiração e a quem devo minha mais profunda gratidão é, sem dúvida, meu esposo Alexandre. Companheiro e cúmplice que sempre me incentivou a percorrer esta longa trajetória, me dando força quando demonstrava momentos de desânimo, por abrir minha mente ao novo e ao complexo e por todo amor.

A meus pais, por plantado em mim, o ideal de nunca desistir.

Ao meu filho, minha fonte de inspiração.

Ao Prof. Dr. José Armando Valente, por sua indicação de um rumo a seguir.

Aos professores e aos colegas do programa de Pós-Graduação em Tecnologias da Inteligência e Design Digital.

À Edna Conti, secretária do programa de mestrado e doutorado, que sempre me auxiliou com muita dedicação.

A todas as pessoas que cooperaram de alguma maneira para o desenvolvimento e conclusão desta pesquisa.

RESUMO

O presente estudo apresenta um estudo de campo que utiliza as Tecnologias Emergentes, aliadas ao Pensamento Computacional e pode auxiliar no processo de Ensino e Aprendizagem. Segundo José Armando Valente (1998), a utilização de computadores para a Educação possui quatro ingredientes importantes que fazem parte do processo de ensino e aprendizagem: o computador, o software educativo, o professor alfabetizado e capacitado para utilizar os computadores no meio educacional e o aluno. Assim, tenta-se compreender como a apropriação das tecnologias no espaço acadêmico, pode enriquecer este campo fértil de possibilidades e recursos, onde as categorias mais comuns são: tutorias, exercícios e práticas, jogos e simulação que possuem a modalidade diversas.

Este estudo de campo investigou, acompanhou e analisou os recursos e metodologias utilizadas pelos professores durante as aulas. Efetivamente, utilizou-se o software *Processing* como ferramenta para o ensino e aprendizagem e para desenvolver o conteúdo das disciplinas “Elementos Fundamentais de Matemática”, em 2015 e “Introdução ao Pensamento Computacional”, em 2016 com os alunos do Curso de Graduação em Comunicação Social – Midialogia, da UNICAMP. Os dados dos alunos utilizados para esta pesquisa foram compilados a partir de acompanhamento das aulas, respostas a questionários e relatórios realizados pelos alunos durante o curso. Esses dados foram categorizados, compilados e submetidos a análise, utilizando o método estatístico multidimensional que se baseia no software CHIC – Classificação Hierárquica, Implicativa e Coersitiva - que gerou as árvores de similaridade e os números de ocorrência das categorias. Com base nos dados e no resultado das análises, identificou-se alguns padrões que são responsáveis por este processo de ensino e aprendizagem. Eles são divididos em 7 fases: Emocional, Técnico-operacional, Imitação, Relação-comunicação, Relação-informação, Relação-expressão reflexiva e Autoformação. O entendimento das fases são elementos fundamentais para entender como o uso das Tecnologias Emergentes para o ensino e aprendizagem foram alcançados nas duas disciplinas do curso de Midialogia.

Palavras-chave: Tecnologias Emergentes, Ensino/Aprendizagem, Processing, Estudo de campo.

ABSTRACT

This study presents a field study that uses Emerging Technologies, coupled with the Computational Thinking and can assist in the teaching and learning process. According to Valente (1998), the introduction of computers in education has obligatorily four equally important ingredients in the process of teaching and learning: computer, educational software, literate teacher able to use the computer as educational medium and the student. So, try to understand how the appropriation of technologies in the academic space, can enrich this fertile field of possibilities and resources, where the most common categories are: tutorials, exercises and practices, games and simulation that have a different mode.

This field study investigated, monitored and analyzed the resources and methodologies used by the teacher during class. Effectively used the Processing software as a tool for teaching and learning and to develop the content of the subjects "Mathematics Fundamental Elements" in 2015 and "Introduction to Computational Thinking" in 2016 with the students of Communication Undergraduate Course Social - Medialogy, UNICAMP. The data extracted from this research through class monitoring and response questionnaires for the course students were compiled. These data were analyzed, categorized, compiled and submitted to analysis using the statistical method multidimensional through software CHIC – Coercion Implication and Hierarchic Classification - that generated the trees of similarity and the occurrence of numbers categories and from them, we used Excel, from which was generated trees similarity. Based on the data and analysis results, we identified some patterns that are responsible for this process of teaching and learning. They are divided into 7 phases: Emotional, Technical-Operational, Imitation, Relation-communication, Relation-information, Relation-reflexive expression and Self-formation. The understanding of the phases are fundamental elements to understand how the use of Emerging Technologies for teaching and learning were achieved in the two disciplines of the Midialogy course.

Keywords: Emerging Technologies, Teaching / Learning, Processing, Field Study.

LISTA DE FIGURAS

Figura 01 – Gráfico de similaridade obtido após a análise dos dados pelo software CHIC.....	18
Figura 02 – Les demoiselles de 1907.	54
Figura 3 – O robô “tartaruga” – Linguagem Logo.....	64
Figura 4 – Como editar e codificar mapas sem codificação.....	65
Figura 5 – Ambiente de desenvolvimento Python.....	66
Figura 6 – Aplicativo com figura geométrica.....	67
Figura 7 – Ambiente de desenvolvimento do Scratch	68
Figura 8 – Ambiente de desenvolvimento do App Inventor2.....	69
Figura 9 – Ambiente de desenvolvimento Alice.....	70
Figura 10 – Placa Arduino Uno.....	71
Figura 11 – Designer By Number.....	72
Figura 12 – Programa em Processing que captura o movimento do mouse na tela.....	73
Figura 13 – Programa em Processing denominado Hello, World!.....	74
Figura 14 – Código em Java para desenhar um retângulo e imagem gerada pelo código.....	74
Figura 15 – Código em Processing para desenhar um retângulo e imagem gerada pelo código.....	74
Figura 16 – Exemplo de árvore de similaridades do software CHIC.....	81
Figura 17 – Figura com transparência e movimento com codificação.....	84
Figura 18 – Sobre o gostar da matéria matemática.....	86
Figura 19 – Sobre dificuldade com matemática.....	87
Figura 20 – Sobre Já fez algum curso de informática.....	87
Figura 21– Sobre conhece alguma linguagem de programação.....	87
Figura 22 – Já criou algum aplicativo.....	88
Figura 23 – Sobre usou o próprio computador para elaborar os exercícios.....	88
Figura 24 – Que tipo de instituição concluiu o ensino médio.....	89
Figura 25– Possui computador em casa.....	89
Figura 26 – Gostam de matemática	89
Figura 27 – Dificuldade com a matéria matemática.....	90
Figura 28 – Fizeram algum curso de informática.....	90

Figura 29 – Já teve a matéria lógica.....	91
Figura 30 – Conhece alguma linguagem de programação.....	91
Figura 31 – Conhece alguma linguagem de programação.....	91
Figura 32 – Conhece alguma linguagem de programação.....	92
Figura 33 – Já criou algum aplicativo.....	92
Figura 34 – Conhece o Processing.....	92
Figura 35 – Conhecem alguma linguagem de código aberto.....	93
Figura 36 – Quando você encontra alguma dificuldade (em relação a matemática), o que faz?.....	93
Figura 37 – Redes sociais mais usadas.....	94
Figura 38 – Árvore de Similaridade Gerada pelo CHIC	101
Figura 39 – Nível 1 de Similaridade	103
Figura 40 – Nível 2 de Similaridade	103
Figura 41 – Nível 3 de Similaridade	104
Figura 42 – Nível 4 de Similaridade	104
Figura 43 – Nível 5 de Similaridade	105
Figura 44 – Nível 8 de Similaridade	105
Figura 45 – Nível 11 de Similaridade	106
Figura 46 – Nível 14 de Similaridade	106
Figura 47 – Nível 17 de Similaridade	106
Figura 48 – Nível 20 de Similaridade	108

LISTA DE TABELAS

Tabela 01 – Planilha Excel com dados a ser compilados	17
Tabela 02 – Sistema Dicotômicos	57
Tabela 03 – Sinopses das várias lógicas	60
Tabela 04 - Compilação de dados. Exemplo de uma planilha	79

LISTA DE QUADROS

Quadro 01 – Das Perspectivas Teórico-Epistemológicas.....	35
Quadro 02 – Classificação dos níveis com similaridade 1.....	102
Quadro 03 – Classificação dos níveis com similaridade <1 e ≥ 0.00129985 ..	102

Sumário

INTRODUÇÃO	12
O projeto de pesquisa	14
O software CHIC de análise dos dados	15
Estado da arte	19
Justificativas para a pesquisa	20
Objetivo geral	20
Objetivos específicos	21
Hipóteses	21
Fundamentações teóricas	22
Metodologia da pesquisa	23
CAPÍTULO 1 – BASE TEÓRICA	25
1.1 A cognição em Charles Sanders Peirce	25
1.2 O Processo de Ensino e Aprendizagem em Vigotski	29
1.3 A Mediação no Desenvolvimento Humano	32
1.4 A Zona de Desenvolvimento Proximal de Vigotski	35
1.5 O pensamento da Vigotski na Educação	36
1.6 Princípios de Morin	39
1.7 Pensamento Computacional	43
1.8 Pressupostos teóricos do processo de apropriação das Tecnologias Emergentes.	47
CAPÍTULO 2 - APRENDIZADO DA DISCIPLINA	52
2.1 A Linguagem Matemática	52
2.2 Picasso, Einstein e a Quarta Dimensão	53
2.3 Lógica & Matemática	56
2.4 Desenvolvimento de um aplicativo	61
2.4.1 Etapa 01 – Estratégia	61
2.4.2 Etapa 02 – Algoritmo	62
2.4.3 Etapa 03 – Implementação	62
2.5 Paradigmas de programação	62
2.5.1 Logo	63
2.5.2 Matlab	64
2.5.3 Python	66
2.5.4 Scratch	67
2.5.5 App Inventor	68

2.5.6 Alice	69
2.5.7 Arduino	70
2.5.8. Processing	71
3. DESENVOLVIMENTO DA PESQUISA.....	76
3.1 Objetivos e Ementa da Disciplina	76
3.1.1 Objetivos Gerais da Disciplina:	76
3.1.2 Objetivos Específicos das Disciplinas.....	77
3.1.3 Conteúdo Programático	77
3.1.4 Metodologia	78
3.2 Análise dos dados no software CHIC	78
3.3 O estudo da Disciplina Elementos Fundamentais de Matemática e Introdução ao Pensamento Computacional	82
3.3.1 O tempo da investigação	82
3.3.2 O Local	83
3.3.3 Sujeitos da Pesquisa	83
3.3.4 Coleta de dados.....	83
CAPÍTULO 4 - Análise dos Dados.....	86
4.1 Análise do Público Alvo	86
4.2 Análises de similaridade e implicativa.....	95
4.2.1 Grupos e categorias.....	95
4.2.1.1 Grupo conhecimento em linguagem de programação (CLP).....	95
4.2.1.2 Alfabetização em linguagem de programação	96
4.2.1.3. Alfabetização em HTML (AHTM)	96
4.2.1.4 Já criou programa para computador (JCPC)	96
4.2.1.5 Já criou algum aplicativo (JCAPP).....	97
4.2.1.6 Conhece o software Processing (CSP).....	97
4.2.2 Percepção sobre as matérias matemática e lógica.....	97
4.2.3 Percepção sobre o pensamento computacional	98
4.2.4 Percepção sobre: O que faz quando encontra dificuldade com a matéria matemática (DMM)	98
4.2.5 Percepção durante o processo de aprendizagem durante a criação do projeto final	99
4.2.5.1 Percepção sobre o uso do Processing (UP)	99
4.5.3 Análise de similaridade	101
CONSIDERAÇÕES FINAIS.....	110
Referências.....	115

APENDICE A	119
APENDICE B	120
APENDICE C	121
APENDICE D	124
APENDICE E	125
APENDICE F	126
APENDICE G	128
APENDICE H	130
APENDICE I	132
APENDICE J	134
APENDICE L	135
APENDICE M	137

INTRODUÇÃO

O presente trabalho tem como objetivo realizar um estudo de campo a fim de avaliar o processo de ensino e aprendizagem dos alunos do curso de comunicação quando eles utilizam linguagem de programação aplicada às tecnologias atuais, em particular, pretendemos observar como se dá a apropriação de conhecimento lógico e de linguagem computacional de pessoas que desenvolvem programas e aplicativos para computadores utilizando o software *Processing*¹ no ambiente acadêmico. Nosso estudo tem como foco o ensino e a aprendizagem nesta área, no Curso de Graduação em Comunicação Social com ênfase em Midialogia, na UNICAMP – Universidade Estadual de Campinas.

Hoje, podemos encontrar uma grande quantidade de dispositivos, aplicativos e programas computacionais que podem ser utilizados em sala de aula para se ensinar programação com computadores. Assim, com esta pesquisa, pretendemos realizar um estudo de campo sobre o processo de ensino e aprendizagem de linguagem lógica de programação em duas disciplinas ministradas para alunos do curso de comunicação com ênfase em Midialogia.

Diante dos recursos e aplicativos que podem ser utilizados como ferramentas de suporte ao ensino e a aprendizagem, os professores escolheram a linguagem de programação do Ambiente *Processing* que foi desenvolvida pelo MIT – *Massachusetts Institute of Technology* e que utiliza o código binário para criar imagens e programas computacionais de diversas naturezas, em especial para produções computacionais artísticas. A escolha deste estudo de campo tem como objetivo avaliar a metodologia de ensino utilizada e os resultados obtidos em relação a aprendizagem dos estudantes deste curso de graduação em comunicação.

Os dados e atividades desenvolvidas que foram analisadas nesta pesquisa foram produzidas, no 1º semestre de 2015 e 1º semestre de 2016

¹ O *Processing* é um ambiente e uma linguagem de programação *open-source* para pessoas que querem criar imagens, animações e interações. Inicialmente foi desenvolvido para ensinar fundamentos de programação de computadores em um contexto visual, particularmente para artistas. O *Processing* também é uma poderosa ferramenta para criar trabalhos profissionais.

quando os alunos das disciplinas Introdução aos Elementos Fundamentais de Matemática e Introdução ao Pensamento Computacional utilizaram o ambiente e o software *Processing*. A metodologia utilizada para este estudo de campo foi o acompanhamento das aulas e a aplicação de questionários *online*, disponíveis em formulários na web na plataforma Google. Para a análise dos dados produzidos utilizamos o software CHIC - *Classification Hiérarchique Implicative et Cohésitive*, (Classificação Hierárquica, Implicativa e Coesitiva) que permitiu realizar avaliações qualitativas e quantitativas a partir das atividades desenvolvidas nas aulas e com a aplicação e análise dos questionários.

A primeira versão do programa computacional CHIC foi desenvolvida por Saddo Ag Almouloud (1992) e consiste em uma ferramenta para a aplicação do método estatístico de análise implicativa desenvolvida por Régis Grás (2002) e da análise de similaridade de Israel César Learman. Já a linguagem de programação *Processing* é uma ferramenta utilizada por estudantes e profissionais de artes em algumas partes do mundo. A linguagem de programação foi criada pelo *Media Lab do MIT - Massachusetts Institute of Technology*, por Casey Reas e Ben Fry, em 2011. Ela é uma linguagem que permite elaborar programas de computador para as artes visuais, no entanto, como é uma linguagem de fácil uso, também vem sendo utilizada para a produção de outros tipos de aplicativos, como, por exemplo, programas que operam com códigos musicais e sonoros, modelagem, desenvolvimento de jogos digitais, etc. permitindo, assim, a realização de sistemas computacionais mais simples.

O *Processing* apresenta funções nativas e uma estrutura de processamento interno otimizadas para a produção de imagens. Ademais, tornou-se uma linguagem amplamente usada em escolas de arte de todo o mundo no ensino de programação para artistas e designers. Além disso, o *Processing* foi a inspiração de um outro projeto adotado em diversos *hackerspaces*, *makerspaces*, *fablabs* e *medialabs* espalhados pelo mundo: o Arduino (OAKIM, 2015, p.109).

Como já afirmamos, a linguagem de programação do *Processing* foi criada para ser utilizada por pessoas que não possuem muita habilidade para programar. Hoje, na Internet, encontramos muitos códigos de programas disponíveis, no website do *Processing*² que é um ambiente de programação de

² Disponível em < <http://www.openprocessing.org/> >. Acessado em 15 de julho de 2016.

código aberto e, portanto, de uso gratuito. Neste ambiente podemos encontrar um link para as páginas do *OpenProcessing* que é um local na Internet que disponibiliza muitos programas prontos e seus códigos e que podem ser acessados e modificados.

O projeto de pesquisa

Esta pesquisa é um estudo de campo sobre o processo de ensino e aprendizagem dos alunos que participaram da Disciplina Elementos Fundamentais de Matemática, no primeiro semestre de 2015 e Introdução ao Pensamento Computacional no primeiro semestre de 2016³, do Curso de Graduação de Comunicação Social com ênfase em Midialogia, na UNICAMP – Universidade Estadual de Campinas. As aulas teóricas e práticas de linguagem matemática e de programação foram ministradas neste curso de comunicação.

O uso das TDIC (Tecnologias Digitais de informação e Comunicação) que a partir de agora denominaremos de Tecnologias Emergentes de Comunicação e Informação, ou simplesmente de Tecnologias Emergentes⁴ (TM), favorecem o processo de construção do conhecimento por meio do Pensamento Computacional. De fato, o uso do *Processing* que é um ambiente de desenvolvimento integrado (IDE), criado pelo MIT para se realizar aplicativos em artes eletrônicas e projetos visuais, tem como objetivo ensinar noções básicas de programação computacional para contextos visuais.

Um dos objetivos do *Processing* é servir como ferramenta para usuários não-programadores que iniciam o processo de aprendizagem com base no

³ A partir de 2016 a disciplina passou a ser denominada de Introdução ao Pensamento Computacional. No entanto, trata-se do mesmo conteúdo programático.

⁴ Hoje, as **Tecnologias Emergentes** são aquelas que nascem a partir dos meios digitais de comunicação e informação no mundo contemporâneo e que também podem ser denominadas de Tecnologias Digitais da Informação e Comunicação (TDIC). A curto prazo consideram-se Tecnologias Emergentes aquelas que são utilizadas para produção e distribuição de conteúdo nos ambientes colaborativos, participativos e sociais e que utilizam as mídias digitais atuais, GPS e RFID ; a médio prazo são as que trabalham com os conteúdos e as plataformas abertas e os dispositivos móveis e a longo prazo são aplicações desenvolvidas para a Internet das coisas, realidade virtual, mista e aumentada, computação ubíqua, etc. O foco desta pesquisa concentra-se em desafios a curto e médio prazo, em particular, as tecnologias aplicadas ao desenvolvimento do pensamento computacional e que são utilizadas nos dispositivos móveis: mídias locativas. De modo abrangente, consideram-se Tecnologias Emergentes as aplicações e produções em *mashup*, *cloud computing*, nanotecnologia, biotecnologia, tecnologia da informação e comunicação, ciência cognitiva, robótica, computação quântica e inteligência artificial.

Pensamento Computacional, isto é, o uso da lógica de programação em sistemas computacionais para adquirir conhecimento baseado em sistemas digitais e nos computadores, que também denominamos de Letramento Digital. Com o uso destas metodologias os iniciantes em programação conseguem obter satisfação imediata quando usam esta linguagem com retornos visuais instantâneo. Ela possui capacidade gráfica de linguagem de programação Java que se torna simplificada por meio do uso do software *Processing* que facilita a aprendizagem e o desenvolvimento de aplicativos.

Ao acompanhar essas duas disciplinas e ao analisar a metodologia empregada para se ensinar programação no curso de graduação em comunicação, observamos a evolução dos alunos que seguiram o seguinte conteúdo programático:

- Matemática, artes e as mídias;
- As artes e a matemática como linguagem;
- Introdução ao *Processing*;
- Conceitos básicos sobre programação;
- Conceitos de programação sequencial;
- Conceitos de programação condicional;
- Conceitos de programação de controle de repetição;
- Conceitos de programação modular;
- Entrada e saída de dados em sistemas computacionais.

O software CHIC de análise dos dados

Depois de ministrada as aulas teóricas e práticas da disciplina, elaboramos dois questionários (Apêndices I e II) para observar a evolução do aprendizado. Com base em dados qualitativos resultantes da análise-interpretativa que foram associados às categorias criadas para o software CHIC – Classificação Hierárquica Implicativa e Coercitiva, geramos gráficos de similaridade.

O software CHIC permite aplicar um método estatístico multidimensional que é utilizado em estudos qualitativos de regras de associação e que atendem a tipos distintos de variáveis, quantifica a significação dos valores qualitativos,

as constâncias das regras associadas, ordena classes de regra e a tipicidade e contribuição de sujeitos ou categorias de sujeitos à constituição de certas regras (ALMOULOU, 2005).

O software CHIC representa as relações entre os dados, por meio de gráficos, caminho de regras e uma hierarquia de regras sobre regras, a partir de intervalos determinados. Assim, mediante essa ferramenta estatística multidimensional, os resultados foram sintetizados e estruturados com o intuito de contribuir para a análise do processo de construção do conhecimento e dos níveis de letramentos. As ocorrências e não ocorrências de determinadas categorias foram fixadas em variáveis e tratadas pelo software para gerar regras de associação, que podem ser visualizadas em gráficos denominados como árvore de similaridades.

Para gerar as árvores de similaridades foi necessário criar planilhas no Excel. As variáveis, em formato de códigos, foram inseridas na linha horizontal superior da planilha. Por exemplo, compCa (Tem Computador em Casa), gostMat (Gosta de Matemática), difMat (Tem Dificuldade com Matemática), cursoInfo (Fez Algum Curso de Informática), possSmar (Possui Smartphone), conLing (Conhece alguma Linguagem de Programação), criAPP (Já Criou Aplicativo), usoProc (Uso Processing), usoPComp (Usa o Próprio Computador). Na primeira coluna, foram inseridos os códigos representando os aprendizes, por exemplo, A1 A2, A3, A4, A5... etc. Na planilha foram lançados valores 0 ou 1, conforme a ausência ou a presença da variável nas práticas de letramentos realizadas pelos aprendizes.

A tabela de compilação de dados (abaixo) é um exemplo de planilha elaborada no Excel. Na linha horizontal superior há o grupo de variáveis compCa, gostMat, difMat, cursoInfo, possSmar, conLing, criAPP, usoProc, usoPComp. Na primeira coluna, a sequência de aprendizes de 1 a 22. O valor 0 refere à negação (não) da variável, e o valor 1 demonstra afirmação (sim).

Tabela 1 – Planilha Excel com dados a ser compilados

	compCa	gostMat	difMat	cursoInfo	possSmar	conLing	criAPP	usoProc	usoPComp
A1	1	0	0	1	1	1	1	1	1
A2	1	1	1	0	1	0	0	1	1
A3	1	0	1	0	1	0	0	1	1
A4	1	0	1	0	1	1	0	1	1
A5	1	1	0	0	1	1	0	1	1
A6	1	1	0	1	1	1	1	1	1
A7	1	0	0	0	1	1	0	1	1
A8	1	1	0	0	1	0	0	1	1
A9	1	1	0	0	1	0	0	1	0
A10	1	1	0	1	1	1	1	1	1
A11	1	0	1	0	1	0	0	1	0
A12	1	1	0	0	1	0	0	1	1
A13	1	1	0	0	1	1	1	1	0
A14	1	0	0	0	1	0	1	1	1
A15	1	0	1	0	1	1	0	1	1
A16	1	1	0	0	1	1	0	1	1
A17	1	0	1	0	1	1	1	1	1
A18	1	1	0	0	1	0	0	1	1
A19	1	1	0	1	1	1	1	1	1
A20	1	1	0	1	1	1	0	1	0
A21	1	0	1	0	1	0	0	1	0
A22	1	1	0	1	1	1	1	1	0

Com os dados digitados e a planilha criada introduzimos as informações que passaram a ser processadas no software CHIC. A partir disso, o software estatístico gera a tela da árvore de similaridades que apresenta o número de ocorrências, a média, o desvio padrão, o coeficiente de correlação, os índices de similaridades, os nós significativos e permite visualizar informações sobre o

arquivo, as frequências de pares variáveis, o coeficiente de correlação, índice e valor. A seguir apresentamos um exemplo da árvore de similaridades (Figura 1) que foi gerada a partir da tabela compilação dados.

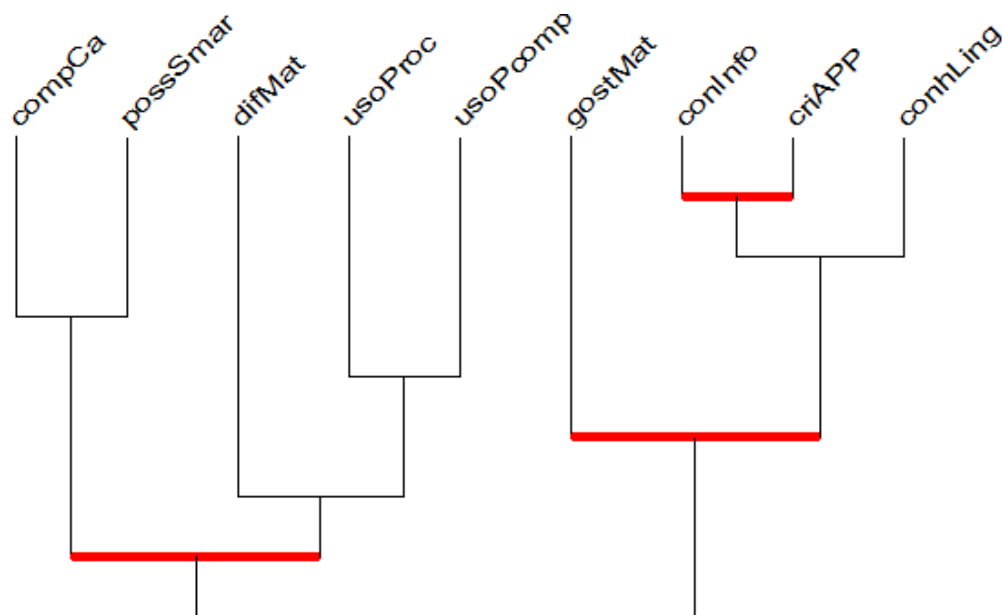


Figura 01 – Gráfico de similaridade obtido após a análise dos dados pelo software CHIC.

Para começar a compreender as correlações geradas pelo gráfico de similaridade (Figura 01) verificamos uma correlação direta (linha vermelha) entre alguns itens, como por exemplo, **conLing** (Conhece alguma Linguagem de Programação) e **criApp** (Já Criou algum Aplicativo) que, certamente estão correlacionados porque só consegue criar um aplicativo quem conhece uma linguagem de programação. Neste caso a relação é direta e, portanto, o software CHIC irá indicar o que é óbvio.

No entanto, ele também indica uma correlação em vários outros níveis diferentes entre compCa (Tem Computador em Casa), possSmar (Possui *Smartphone*), difMat (Tem Dificuldade com Matemática), usoProc (Uso Processing), usoPComp (Usa o Próprio Computador) que analisaremos neste trabalho. Por ora, estamos indicando como serão lidos e interpretados os dados gerados pelo CHIC e também constatando as relações existentes.

Estado da arte

As Tecnologias Emergentes e, principalmente, as tecnologias digitais móveis (*tablets* e *smartphones*), invadiram a nossa vida e, é óbvio, invadiram também o cotidiano escolar com uma intensidade absurda. O que nos levou a identificar práticas, projetos, produtos e soluções educacionais que nos auxiliam na preparação das novas gerações de alunos que irão “aprender a aprender”.

Deste modo, esta pesquisa tem como um de seus objetivos aprofundar as reflexões sobre as metodologias de ensino que permitem pensar sobre as atividades que envolvem alunos e professores em projetos de leitura, escrita, oralidade e, principalmente, no uso do raciocínio lógico e matemático e o que vem sendo denominado de “Pensamento Computacional”. De fato, nossa proposta é realizar uma reflexão das metodologias educacionais que estão baseadas numa perspectiva teórica-epistemológica eco-sistêmica (VALENTE; ALMEIDA, 2008, p. 16-17). Essa formulação engloba conceitos das teorias construtivista, interacionista sociocultural, afetiva e transcendental que interagem com a produção do saber a partir da vertente histórico-cultural de Lev. S. Vigotski.

Os indivíduos das gerações atuais aos quais nos referimos caracterizam-se por compreender a era digital móvel como um “*habitat natural*”. Eles consomem informações midiáticas manipulando diversos recursos e ferramentas dos dispositivos móveis, pois já nasceram imersos em um ambiente digital e estão fortemente influenciados por este meio digital e pela cibercultura (LÉVY, 1999).

Eles utilizam e aprendem em múltiplas linguagens explorando a associação de inteligências múltiplas, de forma colaborativa ou individualizada, interativa, de modo auto-didático e autônoma. Esse contexto socioeducativo é um dos reflexos das chamadas transformações da sociedade atual e da certeza que a era digital móvel está presente em nossas vidas de modo consciente e inconsciente.

Na tentativa de inserir na cultura educacional um tipo de aprendizagem que proponha atividades mais próximas da realidade do aluno, mediada por

recursos digitais, realizamos esta pesquisa que é um estudo de campo que se utiliza da linguagem *Processing* para o processo de ensino e aprendizagem.

Justificativas para a pesquisa

Esta pesquisa possui relevância educacional, pois nela, pretendemos mostrar que o uso das metodologias aplicadas ao ensino, que nasceram com as tecnologias contemporâneas da informação e comunicação, proporcionam, aos educadores e educandos, uma nova forma de ensinar e de aprender.

Essas tecnologias da informação seguida de infinitas formas de comunicação, vem provocando enorme mudanças nos modos de se fazer, de se comunicar e de se pensar dos seres humanos refletindo, substancialmente, em seus costumes e em seus hábitos diários. Segundo Pierre Lévy,

basta que alguns grupos sociais disseminem um novo dispositivo de comunicação, e todo o equilíbrio de representações e das imagens será transformado, como vimos no caso da escrita, do alfabeto da impressão, ou dos meios de comunicação e transporte modernos (1997, p. 16).

As interfaces, os dispositivos de comunicação, as diferentes formas do fazer e de agir, as linguagens e as relações com o meio ambiente e com os meios digitais são frutos do compartilhamento das pessoas com as Tecnologias Emergentes que se modificam constantemente e em cada momento cultural e histórico.

Objetivo geral

O objetivo desta pesquisa é desenvolver um estudo que permita analisar a metodologia utilizada em um contexto educacional específico com abordagem da disciplina Elementos Fundamentais da Matemática e Introdução ao Pensamento Computacional com estudantes do curso Graduação em Comunicação Social – Midialogia, da UNICAMP. Pretende-se verificar como se dá o processo de ensino e aprendizagem com base no Pensamento Computacional, suas potencialidades com o uso de linguagens de programação em especial com o uso do software *Processing* e aplicativos neste universo de

múltiplas oportunidades. Unir alunos, professores e computadores, com a finalidade de trocar experiências e informações relevantes no processo de cognição.

Objetivos específicos

1. Descrever, por meio de revisão da literatura especializada, as linhas de raciocínio que permeiam a forma tradicional de ensino e aprendizagem e o que poderia contemplar uma proposta para conduzir o Pensamento Computacional, perpassando pelos canais cognitivos, semióticos, visuais, sinestésicos e digitais.
2. Criar situações de ensino e aprendizagem, utilizando a linguagem *Processing* (linguagem de programação desenvolvida pelo *MIT – Massachusetts Institute of Technology*) que foi construída para as artes eletrônicas e comunidades de projetos visuais com o objetivo de ensinar noções básicas de programação computacional em um contexto visual e de fácil utilização.
3. A partir dos resultados obtidos através da aplicação de exercícios específicos dado para os estudantes, investigar níveis de similaridades entre as atividades realizadas juntos aos aprendizes.
4. Por fim, pretende-se analisar os dados obtidos com os exercícios e questionários aplicados a fim de avaliar se esta metodologia permite o aprendizado e se ela pode ser adotada para desenvolvimento do Pensamento Computacional.

Hipóteses

Considerando os objetivos descritos acima, esta pesquisa parte de três hipóteses fundamentais:

1. O uso de Tecnologias Emergentes no ambiente escolar muda a forma como os alunos assimilam os conteúdos, particularmente em disciplinas que tratam de conceitos mais complexos;

2. O uso das Tecnologias Emergentes contribuiu para a construção das singularidades dos aprendizes e também permite que se possam criar seus próprios projetos (aplicativos, games e sistemas computacionais).
3. O Pensamento Computacional permite utilizar estratégias de ensino e aprendizagem com as Tecnologias Emergentes. Estamos interessados na observação de atividades de aprendizagem que se baseiam em padrões estéticos gerados por estas tecnologias - em particular, a programação de meios interativos – e que apoiam o desenvolvimento do Pensamento Computacional. Nosso contexto de trabalho utiliza a linguagem de programação *Processing* que é um ambiente de programação que permite a criação de aplicações e histórias interativas, jogos e simulações e permite compartilhar estas produções no ambiente *online*.

Fundamentações teóricas

Esta pesquisa caracteriza-se por ser exploratória (SANTOS, 2004), realizando um levantamento bibliográfico e documental dos conteúdos de ensino e aprendizagem e do universo da semiótica e da complexidade. Refere-se ao uso de metodologias de análise com base na Teoria Semiótica proposta por Charles Sanders Peirce e em reflexões e problematizações com base nas teorias histórico-cultural de Lev S. Vigotski e do pensamento complexo de Edgar Morin para analisar o processo cognitivo dos alunos do Curso de Midialogia.

Também, trata-se de uma pesquisa analítica (SANTOS, 2004), pelo fato de ter como base o desenvolvimento de exercícios que foram realizados pelos alunos e permitiram refletir ao responderem questionários que foram analisados pelo software CHIC - Classificação Hierárquica, Implicativa e Coesitiva que permitiu avaliar o processo de ensino e aprendizagem nestas duas disciplinas.

O método bibliográfico (GIL, 1999) apresenta a maneira de se buscar o material informativo, selecionar e organizar previamente conhecimentos culturais e científicos que já foram elaborados por outros autores, com intuito de apontar soluções ao tema ou ao problema investigado. O método bibliográfico contribui para adquirir conhecimentos específicos na área que se pretende pesquisar,

formular e adequar citações bibliográficas para elaborar conceitos e/ou novas ideias.

O estudo da obra de Jenkins (2006) serviu para subsidiar o conceito de “Cultura da Convergência” relativos à tomada de consciência e busca pela transformação. Já em Primo (2007) buscamos problematizar sobre a concepção de interação. Em Vigotski e Peirce observamos o processo de cognição e, por fim, em Lúcia Santaella (2005) e Angel Pino (2001) refletimos como se dá o pensamento humano a partir do uso das linguagens, com base nos textos “Matrizes da Linguagem do Pensamento”, de Santaella (2005) e “As Marcas do Humano”, de Pino (2001), que foram bases teóricas e referências para a classificação dos meios de comunicação e como eles afetam o nosso processo cognitivo.

Este trabalho foi desenvolvido por meio de estudos e pesquisas, os quais foram realizados a partir de pesquisa em material bibliográfico: artigos científicos, teses, livros e periódicos, pois entendemos que, dessa forma, pudemos melhor interpretar o assunto de interesse para a desenvolvimento e transcrição do assunto, bem como a colheita de informação em aulas práticas. O cotidiano do aluno foi alvo de investigação diária. Após a colheita de informação obtida em sala de aula com ações práticas, leitura e interpretação de textos, pudemos melhor compreender as deficiências encontradas no ambiente escolar e, desta forma, contribuir para o desenvolvimento de aplicativos com a linguagem *Processing* que auxiliaram o professor e o aluno no processo ensino e de aprendizagem.

Metodologia da pesquisa

A metodologia utilizada para esta pesquisa foi o estudo de campo que foi realizado no acompanhamento das disciplinas das matérias Elementos Fundamentais de Matemática e Introdução ao Pensamento Computacional com os alunos do 1º semestre do curso de Midialogia, na Unicamp e a aplicação de questionários *online*. Estes formulários foram aplicados através da web e estiveram disponíveis na plataforma Google. Para a análise dos dados utilizamos o software CHIC - *Classification Hiérarchique Implicative et Cohésitive*, (Classificação Hierárquica, Implicativa e Coesitiva) que permitiu analisar qualitativa e quantitativa os dados levantados em aula e com os questionários.

De acordo com Morin (1987), adquirimos conhecimento inauditos sobre o mundo físico, biológico, psicológico, sociológico e, hoje, em especial com o mundo tecnológico. Na ciência há um predomínio cada vez maior dos métodos de verificação empírica e lógica. As luzes da razão parecem fazer refluir os mitos e trevas para as profundezas da mente. E, no entanto, por todo lado, erro, ignorância e cegueira progridem ao mesmo tempo que os nossos conhecimentos. Precisamos de uma tomada de consciência radical.

1. A causa profunda do erro não está no erro de (falsa percepção) ou erro lógico (incoerência), mas no modo de organização de nosso saber num sistema de ideias (teorias, ideologias);
2. Há uma nova ignorância ligada ao desenvolvimento da própria ciência;
3. Há uma nova cegueira ligada ao uso degradado da razão;
4. As ameaças mais graves em que incorre a humanidade estão ligadas ao processo cego e incontrolado do conhecimento (arma termonucleares, manipulações de todo tipo, desregramento ecológico etc.).

CAPÍTULO 1 – BASE TEÓRICA

Lev S. Vigotski (1896-1934) e Charles Sanders Peirce (1839-1914) viveram nas mesmas épocas históricas e, por isso, tiveram pensamentos que se conectam, mas não são totalmente concordantes. O primeiro dedicou-se a psicologia e a educação, já o segundo à filosofia, à lógica, à matemática e foi o criador da teoria semiótica. Eles consagraram-se como pesquisadores importantes nas suas áreas de atuação e apresentaram grandes contribuições para o entendimento do processo de cognição humana.

Aqui, optamos por relacionar diversos conceitos entre vários teóricos de linha de pesquisa diferente, reconhecendo uma tendência atual de nossas concepções que coloca a necessidade de se pensar nas questões interdisciplinares a fim de interpretar a complexidade dos modelos que abordam a cognição humana. Esta complexidade também vem sendo estudada por Edgar Morin que, assim como Peirce e Vigotski, considera as questões de interação por meio do pensamento complexo.

1.1 A cognição em Charles Sanders Peirce

A Teoria Semiótica de Charles Sanders Peirce fundamenta-se nas “três categorias do pensamento e da natureza”: primeiridade, segundidade e terceridade. Obviamente, esta categorização aplica-se ao processo de cognição humana e à transformação do conhecimento. A nossa percepção sobre as coisas do mundo tem início na observação dos fatos, que ao estimular nossa mente, coloca o “pensamento em ação”. Em busca do “repouso do pensamento” e de acordo com nossos valores, “crenças e hábitos”, produzimos novos pensamentos a partir de modelos lógicos e, assim, criamos novos fatos. O processo de cognição apresenta-se contínuo e, sucessivamente, ele é realimentado. Sem início e sem fim, a cognição e o aprendizado acontecem através do “método de investigação científica”. Para ele, o método explicita a lógica do conhecimento e da aprendizagem humana, que acontece por meio das três inferências lógicas: abdução, indução e dedução (PEIRCE, 1975, p. 85).

Entretanto, quando um fenômeno é observado surge uma dúvida que acompanha algo que é ‘novo’ para nós. Aí, várias hipóteses são consideradas a fim de se explicar este fenômeno que nos surpreendeu. Embora as hipóteses

consideradas podem estar erradas, na maioria das vezes elas são corretas e significativas. Por meio desse raciocínio acontece o “*insight*” e a criatividade que se manifesta em tudo, nas ciências, nas artes, enfim, na vida.

Para Santaella, segundo Peirce, o instinto é a capacidade humana de produzir hipóteses corretas que explicam os fenômenos que nos surpreendem. Para ele, o ser humano tem a capacidade de criar e gerar novas ideias e no processo de construção e seleção destas hipóteses está aberto à crítica e autocrítica ao avaliar estas hipóteses, de forma consciente, deliberado e controlado (*apud* SANTAELLA, 2004, p. 95-96).

Para Peirce, a verdade nunca foi absoluta e, por isso, ele sempre evitou as definições abstratas, precisas e finalizadas das ciências. Sua Teoria dos Signos está baseada no conceito de “falibilismo”⁵ que preserva a possibilidade da falha e da indeterminação, própria de todo o processo em desenvolvimento. A ciência afirma as incertezas e seu papel preponderante é formular generalizações das experiências e não simplesmente descrevê-la. “O verdadeiro homem da ciência é aquele que está disposto a abandonar todas as suas crenças no momento em que isso se prove necessário” (SANTAELLA; VIEIRA; 2008, p. 10).

Ao definir as “categorias universais do pensamento”, Peirce conclui que a primeiridade é relativa ao conceito de acaso, apresenta-se pelo que é intuitivo e mostra-se em estado puro para a consciência. Já, a secundidade, é relativa à ação e reação, se apresenta no conflito de nossa consciência que está em busca da compreensão dos fenômenos. Por fim, a terceiridade estabelece relações e, portanto, é um processo de mediação entre o signo, o objeto e seu interpretante, que possibilita as interpretações e as generalizações dos fenômenos.

A partir desse processo de semiose (ação do signo) vamos observar como se dá a transformação do conhecimento em uma mente interpretante. Nossa cognição inicia-se na percepção do fenômeno que estimula nossa mente e nos faz pensar. Como já vimos, ao buscar o “repouso do pensamento” e de acordo

⁵ Falibilismo é a doutrina filosófica segundo a qual não podemos ter certeza absoluta de qualquer forma de conhecimento. Contudo, os falibilistas não são radicais ao ponto de afirmarem que nenhuma forma de conhecimento seja válida, nem que qualquer forma de conhecimento seja válida. Para eles existem formas de conhecimento mais válidas, legítimas e frutíferas que outras, mesmo que não tenhamos certeza absoluta sobre elas.

com nossos hábitos e crenças, produzimos novos pensamentos e novos fenômenos e, desse modo, ocorre o processo de cognição e de interpretação de um signo, indefinidamente.

Santaella afirma que o Signo só funciona como tal se carregar o poder de representar, substituir o seu objeto. “O signo representa seu objeto, produz na mente desse intérprete alguma outra coisa (um signo ou quase-signo) que também está relacionada ao objeto não diretamente, mas pela mediação do signo” (SANTAELLA, 2004, p. 58). Para Peirce, nas palavras de Lucia Santaella e Jorge de Albuquerque Vieira,

o signo é qualquer coisa que é, de um lado, de tal modo determinada por um objeto e por outro lado, de tal modo determina uma ideia na mente de alguém que está última determinação, chamada de interpretante do signo, é consequentemente determinada mediatamente por aquele objeto” (CP 8.343). A definição explicita a ação do signo, a maneira como o signo age, que é chamada de semiose, a saber, sua ação é determinar uma interpretante, crescer em um outro signo (*apud* SANTAELLA; VIEIRA, 2008, p. 62).

A “dedução” é um tipo de raciocínio que está fundamentado em leis e regras pré-determinadas e que são aceitas como verdadeiras. Na “dedução” começamos por um estado hipotético das coisas que são definidas de forma abstrata pelas características que apresentam e que “é válida se e somente se existe uma relação entre o estado de coisas suposto nas premissas e o da conclusão” (PEIRCE, 1983, p. 44). A dedução é lógica e “prova que algo deve ser” (PEIRCE, 1983, p. 46). A “inferência dedutiva” tem como principal fundamento analisar os fatos pelo ponto de vista das leis; por isso, é “analítica” e, assim, é também o fim último da investigação científica que, ao conhecer a probabilidade de ocorrência de um evento, permite chegar-se a uma conclusão acertada a partir de premissas verdadeiras.

Já os raciocínios lógicos da “indução” e da “abdução” são de caráter “sintético”, porque não classificam os fenômenos de acordo com características determinadas, mas, “diante de uma sucessão de conclusões concordantes ou diante de semelhanças de fatos ou entre casos, sintetizam os dados num pensamento íntegro e único” (LAURENTIZ, 1991, p. 48).

A “inferência indutiva” observa os fenômenos e adota uma conclusão aproximada, e nos conduz à generalização dos fatos e ao estabelecimento da lei que será utilizada pela dedução. No raciocínio “indutivo”, Peirce destaca que o

“processo de investigação experimental” a que somos submetidos, diante dos fenômenos naturais e culturais, “consiste em partir de uma teoria, deduzir predições dos fenômenos e observá-los para ver o grau de concordância com a teoria”. De fato, a “indução” se apresenta com algo que é operatório (PEIRCE, 1983, p. 46).

Por fim, a “abdução” é o processo lógico que permite a criação de novas ideias. É o *“insight”* propriamente dito. A “abdução” ou “formulação de hipóteses”, diante de um conjunto de valores, apresenta as novas ideias. A “abdução” abre nossa percepção para as novas hipóteses e, assim, ela acontece quando observamos um fato curioso que é capaz de ser explicado por uma suposição; aí, adotamos esta suposição (PEIRCE, 1983, p. 150).

É através do “raciocínio abdutivo” que o homem tem novas ideias, abre novos caminhos e estabelece os novos conceitos sem compromisso algum com a validade da observação. O valor desses conceitos será estabelecido pelo processo operatório ao qual submetemos os fatos em “indução”, que finalmente serão determinados e cumpridos pela “dedução”. A “abdução” começa a partir de premissas fracas que, após passarem pelo aval experimental da “indução”, tornam-se fortes e, portanto, abalizadoras de outros pensamentos, e assim, o papel da “dedução” é analisar os fatos ou os casos a partir de regras pré-determinadas (HILDEBRAND, 1994, p. 13).

Segundo a Teoria Semiótica, a cognição tem início na percepção, que produz hipóteses em “abdução”; experimentamos a veracidade dessas hipóteses a partir da “inferência indutiva” que permite produzir conhecimento e, finalmente, define-se como um novo pensamento ou fenômeno, por meio da “inferência dedutiva”, num ciclo cognitivo interminável.

A abdução é um dos tipos de raciocínio ou argumento entre os outros dois (indução e dedução) que compõe a lógica crítica, o segundo ramo da semiótica. A abdução foi uma criação peirciana. Antes dele, só a indução e a dedução eram levadas em conta como tipos de raciocínio ou método” ... “A abdução é um quase-raciocínio, instintivo, uma adivinhação altamente falível, mas é o único tipo de operação mental responsável por todos os nossos insights e descobertas. Sem ele, o ser humano perderia a capacidade de descobrir do mesmo modo que os pássaros, sem asas, perderiam a capacidade de voar. A abdução funciona como ‘o processo para se chegar a uma hipótese explanatória’ (CP 5.171) acerca dos fatos que nos surpreendem porque não possuímos ainda uma explicação para eles (SANTAELLA; VIEIRA, 2008, p. 61).

Segundo Santaella, ao se conhecer estamos agindo sobre “um processo que transcende o indivíduo, pois o conhecimento implica compartilhamento e troca” (2008, p. 63). E assim, sempre somos obrigados pela experiência a

reconhecer que a natureza e a vida da própria espécie devem “crescer em razoabilidade e, que para isso é preciso respeitar o diferente que acaba sendo igual a nós” (*ibidem*).

1.2 O Processo de Ensino e Aprendizagem em Vigotski

As bases teóricas para as questões que envolvem o ensino e a aprendizagem e o desenvolvimento humano, além da observação dos signos, serão consideradas a partir do pensamento de Vigotski⁶. Ele afirma que a aprendizagem e o conhecimento são construídos pela mediação com o Outro, e ela acontece pela interação sociocultural. Para ele, todas as produções humanas são produções culturais e, segundo Angel Pino,

são aquelas que lhes conferem o sentido do humano, são produções culturais e se caracterizam por serem constituídas por dois componentes: um material e outro simbólico, um dado pela natureza e outro agregado pelo homem. Isso explica por que as “funções psicológicas” são funções culturais, como diz Vygotsky, ou seja, funções constituídas por dois componentes (2005, p. 91).

Também utilizaremos como fundamento para nossas reflexões o pensamento eco-sistêmico de Edgar Morin, em que a realidade não é objetiva e é constituída pela complexidade da matéria que, por sua vez, resgata “a subjetividade individual e coletiva, o caráter ativo, construtivo, afetivo e histórico dos sujeitos que aprendem, bem como a dinâmica relacional entre sujeito e o meio em que ele está inserido” (MORAES; VALENTE, 2008, p. 8).

Vigotski afirma que a cultura e a linguagem constituem os seres humanos e suas relações com o mundo de forma dialética e mediada pelo Outro. Segundo ele, o desenvolvimento do homem se dá a partir do uso da linguagem e da interação social. Para Rosana do Carmo Novaes-Pinto, Vigotski, em seu livro “Pensamento e Linguagem”, discute a inter-relação entre as “funções cognitivas superiores” e a “produção de conhecimento”. Nas palavras dela,

as funções cognitivas vão se tornando cada vez mais complexas, possibilitando aos sujeitos resolverem problemas de natureza diversas, que lhes permitam transformar os meios/bens culturais a seu favor.

⁶ **Lev Semenovich Vygotsky** (1896–1934) formou-se em Direito, mas sempre estudou literatura e psicologia do desenvolvimento. Ele desenvolveu muitas pesquisas nas áreas de educação e de psicologia e foi o primeiro pesquisador a afirmar que o desenvolvimento intelectual das crianças ocorre em função das interações sociais e condições de vida. Estudou neuropsicologia e graduou-se em Medicina.

Essa é essência de seu método dialético: a natureza exerce ações sobre o homem e influencia seu comportamento e desenvolvimento, mas o homem também é capaz de agir sobre a natureza e modificá-la (NOVAES-PINTO, 2012, p. 129).

De fato, para Vigotski, o desenvolvimento do pensamento e a “fala interior” se materializam pela linguagem, desenvolvendo-se a partir de um conjunto de alterações estruturais e funcionais que, ao serem separadas da “fala exterior”, apresentam diferenciações das funções sociais e egocêntricas desta fala. Para Moreira (1999), a partir do pensamento de Vigotski, “no desenvolvimento cultural da criança, toda função aparece duas vezes - primeiro em nível social, e depois em nível individual; primeiro, entre pessoas (interpsicológica) e, depois, se dá no interior da própria criança (intrapsicológica)” (VIGOTSKI, 1930 apud MOREIRA, 1999, p. 111). Assim, “é através da relação interpessoal concreta com outros homens que o indivíduo vai chegar a interiorizar as formas culturalmente estabelecidas de funcionamento psicológico” (OLIVEIRA, 1997, p. 38).

De fato, a interação social é fundamental para o desenvolvimento psicológico do indivíduo. Vigotski entendia que a cultura não seria algo determinado e estático, mas sim, em constante movimento de recriação e reinterpretação das informações, conceitos e significados, exercido pelos indivíduos. De acordo com Oliveira e segundo o pensamento de Vigotski, “a vida social é um processo dinâmico, onde cada sujeito é ativo e onde acontece a interação entre o mundo cultural e o mundo subjetivo de cada um” (1997, p. 38). Em sua pesquisa, baseado em sua percepção marxista, Vigotski diz que temos interações entre diferentes planos históricos: a história da espécie (filogênese), a história do grupo cultural, a história do organismo individual da espécie (ontogênese) e a sequência singular de processos e experiências vividas por cada indivíduo. Para ele, segundo Oliveira,

a maior descoberta da criança só é possível quando já se atingiu um nível relativamente elevado do desenvolvimento do pensamento e da fala. Em outras palavras, a fala não pode ser ‘descoberta’ sem o pensamento. Em resumo, devemos concluir que: 1. No seu desenvolvimento ontogenético, o pensamento e a fala têm raízes diferentes; 2. Podemos, com certeza, estabelecer, no desenvolvimento da fala da criança, um estágio pré-intelectual; e no desenvolvimento de seu pensamento, um estágio pré-linguístico; 3. A uma certa altura, essas linhas se encontram; conseqüentemente, o pensamento torna-se verbal e a fala racional (1997, p. 38).

No desenvolvimento da criança, as “funções cognitivas psicológicas” aparecem num nível social e, depois, num nível individual. E, assim, a internalização do comportamento em suas formas culturais reconstrói-se, formando a entidade psicológica do indivíduo. Resumindo estes conceitos a partir do pensamento de Vigotski,

a internalização de formas culturais de comportamento envolve a reconstrução da atividade psicológica tendo como base as operações com signos. Os processos psicológicos, tal como aparecem nos animais, realmente deixam de existir; são incorporados nesse sistema de comportamento e são culturalmente reconstituídos e desenvolvidos para formar uma nova entidade psicológica. O uso de signos externos é também reconstruído radicalmente. [...] A internalização das atividades socialmente enraizadas e historicamente desenvolvidas constitui o aspecto característico da psicologia humana; é a base do salto quantitativo da psicologia animal para a psicologia humana (1984, p. 65).

O desenvolvimento do indivíduo está fundamentado na mediação e na interação com as outras pessoas e com os elementos culturais. Nesse processo de interação, as pessoas internalizam as formas culturais de comportamento transformando, assim, sua natureza social em natureza psicológica. Desse modo, os processos psicológicos mais complexos, em que se dá a cognição, tomam forma por meio da mediação. A criança encontra soluções aos seus problemas de aprendizado de maneira independente com o “desenvolvimento real”, no entanto, num primeiro momento, a solução desses problemas acontece sob a orientação de um adulto através do “desenvolvimento potencial”. De fato, ela realiza algo quando é mediada por outra pessoa e faz isso, de maneira independente, quando adquire o conhecimento. Assim, aquilo que o aluno realiza hoje, mediado e por meio da Zona de Desenvolvimento Proximal (ZDP), ele é capaz de fazer sozinho amanhã de forma independente e, assim, é que acontece o processo de cognição para Vigotski. Voltaremos à definição de ZDP mais adiante em nosso texto.

Oliveira (1997, p. 26), a partir das reflexões de Vigotski, afirma que a “mediação, em termos genéricos, é o processo de intervenção de um elemento intermediário numa relação; a relação deixa, então, de ser direta e passa a ser mediada por esse elemento”. A relação do homem com o mundo é uma relação mediada, e é assim que as relações sociais constituem as funções psicológicas dos seres humanos que, por sua vez, são interacionistas (BOCK, 2002, p.110).

Existem elementos mediadores que criam os elos entre os organismos e o ambiente sociocultural, tornando essa mediação cada vez mais complexa.

1.3 A Mediação no Desenvolvimento Humano

Segundo Vigotski, para se entender como se dá o funcionamento psicológico do homem, é necessário compreender que tipo de mediação acontece nesse processo. De fato, a mediação, genericamente, é uma relação entre elementos intermediários que, por fazerem parte do processo de **mediação** em si, não permitem que a relação seja direta. E, assim, a relação do homem com o mundo é uma relação mediada, e o processo simples de estímulo-resposta é substituído por um processo bem mais complexo e mediado (1984, p. 40). Os elementos mediadores são de dois tipos: os instrumentos e os signos. Os instrumentos são elementos que operam entre os produtores e os objetos de seu trabalho, ampliando as possibilidades de transformação da cultura e da natureza. Assim, os objetos carregam a função para a qual foram criados e, portanto, tornam-se um “objeto social” que é mediador das relações entre os indivíduos e o mundo.

A diferença entre os instrumentos utilizados pelos homens e os utilizados pelos animais, segundo Santos, está no fato que “os animais não produzem instrumentos com objetivos específicos, não guardam os instrumentos para uso futuro, também não preservam sua função como conquista a ser transmitida a outros membros do grupo social” (2003, p. 133). Os animais modificam o ambiente para ações momentâneas, mas não conseguem desenvolver ações num processo histórico-cultural. Para eles, os instrumentos servem como ferramentas de transformação dos objetos. Já o signo, para Vigotski, é uma invenção e

o uso dos signos como meios auxiliares para solucionar um dado problema psicológico (lembrar, comparar coisas, relatar, escolher etc.) é análoga à invenção e uso dos instrumentos, só que agora no campo psicológico. O signo age como um instrumento da atividade psicológica de maneira análoga ao papel de um instrumento no trabalho (1984, p.59).

Como podemos constatar, a diferença entre instrumentos e signos está na transformação que eles processam sobre os indivíduos. Os instrumentos são ferramentas externas aos indivíduos e atuam transformando os objetos em

“objetos sociais”, já os signos atuam nos indivíduos, internamente, auxiliando os processos psicológicos e, portanto, são considerados como “instrumentos psicológicos”. Os dois atuam como mediadores da cultura e do meio social; no entanto, é preciso internalizar os signos para que aconteçam as transformações, e o processo psicológico possa agir sobre o indivíduo.

Os processos psicológicos, tal como aparecem nos animais, realmente deixam de existir; são incorporados nesse sistema de comportamento e são culturalmente reconstituídos e desenvolvidos para formar uma nova entidade psicológica (VIGOTSKI, 1984, p. 65).

No entanto, o que caracteriza o pensamento humano são os “processos mentais superiores” que são mediados por sistemas simbólicos, ou seja, ações conscientes que são controladas e determinam o pensamento abstrato, o comportamento intencional, a atenção voluntária e memorização. A linguagem é um sistema simbólico do ser humano e atua junto com o pensamento. A linguagem é a base para o relacionamento entre os seres humanos. Vigotski estabelece duas funções para a linguagem: a primeira de caráter social, e a segunda é o pensamento generalizante.

A função social da linguagem é comunicativa. A linguagem é antes de tudo um meio de comunicação social, um meio de expressão e de compreensão. Geralmente, na análise por decomposição de elementos, esta função da linguagem também se separa da intelectual e ambas eram atribuídas à linguagem, como se disséssemos, paralela e independentemente uma da outra. Sabe-se que a linguagem combina a função comunicativa com a de pensar, mas não se investigou, nem se investiga que relação existe entre ambas as funções, o que condiciona sua coincidência na linguagem, como se desenvolvem, nem como estão unidas estruturalmente entre si (VIGOTSKI, 1934 *apud* BAQUERO, 2001, p. 50).

Os homens, ao se comunicarem, criam as linguagens, as quais necessitam dos signos para se concretizarem. Os signos transmitem ideias, sentimentos, vontades, pensamentos aos outros indivíduos. Em experiências individuais, os homens, por meio da linguagem e com os signos, interagem como o Outro em seu meio ambiente social. Assim, esse processo de construção de linguagem conduz o homem ao “pensamento generalizante” que, por sua vez, torna a linguagem um instrumento para se adquirir conhecimento.

Desse modo, a relação social entre os indivíduos, ao ser mediada pela linguagem, constrói o conhecimento e o indivíduo. Referindo-se a produção de conhecimento das crianças, Vigotski diz que a linguagem tem origem, como meio

de comunicação, na criança, pela sua mediação com as pessoas que estão ao seu redor. Em seguida, ao se tornar linguagem interna, transforma-se em função mental interna, fornecendo meios para o pensamento da criança (VIGOTSKI, 1984, p.121).

O pensamento verbal e a linguagem, como sistemas de signos, surgem da necessidade de intercâmbio entre os indivíduos como uma atividade essencialmente humano que necessita de um instrumento de mediação.

a natureza do próprio desenvolvimento se transforma, do biológico para o sócio-histórico. O pensamento verbal não é uma forma de comportamento natural e inata, mas é determinado por um processo histórico-cultural e tem propriedades e leis específicas que não podem ser encontradas nas formas naturais de pensamento e fala. Uma vez admitido o caráter histórico do pensamento verbal, devemos considerá-lo sujeito a todas as premissas do materialismo histórico, que são válidas para qualquer fenômeno histórico na sociedade humana (VIGOTSKI, 1987, p. 44).

Para Vigotski, a linguagem humana desenvolve-se a partir da mediação e da interação entre os homens e entre o ambiente sócio cultural em que eles vivem e, pelo uso dos instrumentos, mudamos as ações sociais e desenvolvemos a cultura. Assim, podemos afirmar que o processo de desenvolvimento humano está intimamente relacionado às questões do ensino e da aprendizagem.⁷ Vigotski destaca que

o aprendizado é mais do que a aquisição de capacidade para pensar; é a aquisição de muitas capacidades especializadas para pensar sobre várias coisas. O aprendizado não altera nossa capacidade global de focalizar a atenção, em vez disso, desenvolve várias capacidades de focalizar a atenção sobre várias coisas (VIGOTSKI, 1984, p. 93).

Bettina S. dos Santos, a partir do pensamento de Vigotski, afirma que

é pela aprendizagem com os outros que o indivíduo constrói constantemente o conhecimento, promovendo o desenvolvimento mental, passando desse modo de um ser biológico a um ser humano. Ele ainda destaca que o desenvolvimento e a aprendizagem estão relacionados desde o nascimento da criança, sendo que a aprendizagem resulta do desenvolvimento e este não ocorre sem a aprendizagem (2003, p. 137).

⁷ O aprendizado “é o processo pelo qual o indivíduo adquire informações, habilidades, atitudes, valores, etc. a partir do seu contato com a realidade, o meio ambiente, as outras pessoas. (...) Em Vygotsky, justamente por sua ênfase nos processos sócio-históricos, a ideia de aprendizado inclui a interdependência dos envolvidos no processo” (OLIVEIRA, 1997, p. 57).

A aprendizagem é uma organização que conduz ao desenvolvimento mental, particularmente da criança e, por isso, ela desenvolve nas crianças as características humanas não naturais que são constituídas social e historicamente. Ela desperta os processos internos dos indivíduos que, ao mesmo tempo, e por meio das mediações, estão relacionados ao ambiente sociocultural desses indivíduos.

1.4 A Zona de Desenvolvimento Proximal de Vigotski

O desenvolvimento do indivíduo é um processo que acontece de fora para dentro, como já indicamos, e é determinado pela mediação com o Outro e, com isso, Vigotski sustenta-se no conceito de “Zona de Desenvolvimento Proximal (ZDP)”, que estabelece a distância entre o “desenvolvimento real”, que é determinado pela capacidade humana de resolver situações problemáticas sem ajuda dos outros, e o “desenvolvimento potencial”, que é determinado pela resolução de um problema sob a orientação de alguém. É o conhecimento de algo que temos potencialidade de aprender; no entanto, ainda não estamos completamente preparados para tal, ou seja, é algo potencialmente atingível.

O “desenvolvimento potencial” é a capacidade que o ser humano tem de realizar uma determinada atividade mediada por outro indivíduo da mesma espécie com um conhecimento superior ao seu. Nas palavras de Vigotski, o conceito de ZDP estabelece

a distância entre o nível real de desenvolvimento, determinado pela capacidade de resolver independentemente um problema, e o nível de desenvolvimento potencial, determinado através da resolução de um problema sob a orientação de um adulto ou em colaboração com outro companheiro mais capaz (VIGOTSKI, 1988 *apud* BAQUERO, 1998, p.97).

Ainda segundo ele,

A Zona de Desenvolvimento Proximal permite-nos delinear o futuro imediato da criança e seu estado dinâmico de desenvolvimento, propiciando o acesso não somente ao que já foi atingido através do desenvolvimento, como também àquilo que está em processo de maturação (VIGOTSKI, 1984, p. 97).

Com a ZDP passamos a entender como se dá o desenvolvimento interno dos indivíduos. Ao interferir nessa Zona de Desenvolvimento, o aprendizado

acontece quando a criança internaliza os conhecimentos e, aos poucos, esse conhecimento torna-se parte de suas funções psicológicas consolidadas. Hoje, a criança é capaz de fazer algo com a ajuda de alguém, amanhã ela conseguirá fazer isso sozinha. Sendo assim, a mediação, seja ela em casa, na rua e na escola, tem um papel fundamental no processo de aprendizagem.

O processo de ensino e aprendizado deve ter início a partir do nível de desenvolvimento real da criança, e o ponto de chegada deve ser o objetivo estabelecido para o aprendizado. Assim, a interferência de outros indivíduos transforma o aprendizado quando ele age diretamente na ZDP. Portanto,

a aprendizagem é um processo essencialmente social, que ocorre na interação com os adultos e os colegas. O desenvolvimento é resultado desse processo, e a escola, o lugar privilegiado para essa estimulação. A educação passa, então, a ser vista como processo social sistemático de construção da humanidade. (...) A partir destas concepções de Vygotsky, a escola torna-se um novo lugar – um espaço que deve privilegiar o contato social entre seus membros e torná-los mediadores da cultura. Alunos e professores devem ser considerados parceiros nesta tarefa social. O aluno jamais poderá ser visto como alguém que não aprende, possuidor de algo interno que lhe dificulta a aprendizagem. (...) O professor torna-se figura fundamental; o colega de classe, um parceiro importante; o planejamento das atividades torna-se tarefa essencial e a escola, o lugar de construção humana (BOCK, 2002, p. 126).

De fato, a intervenção do Outro é essencial para o desenvolvimento do indivíduo, e a interação com os colegas ou com adultos provoca intervenção no desenvolvimento de uma criança com dificuldade para deter conhecimento, mas pode adquiri-lo com essa intervenção.

1.5 O pensamento da Vigotski na Educação

Hoje, notamos que a perspectiva histórico-cultural de Vigotski (1896-1934) é uma das referências teóricas mais utilizadas na fundamentação do processo de ensino e aprendizagem e, assim, torna-se relevante para nossas reflexões. Maria Cândida Moraes e José Armando Valente, observando as perspectivas teóricas-epistemológicas, ou paradigmáticas em relação à complexidade do pensamento, constrói uma cartografia interessante. No quadro 1 verificamos como vem sendo constituído este paradigma.

Os fundamentos de nosso legado filosófico: o positivista, o interpretativo, o sócio-crítico e o que estamos chamando de paradigma emergente (MORAES; VALENTE, 2008, p.16-17).

Quadro 01- Das Perspectivas Teórico-Epistemológicas

Perspectivas Dimensões	Positivista (empírica, racionalista e clássica)	Interpretativa (construtivista, naturalista e fenomenológica)	Sociocrítica e Interacionista (construtivista e sociocrítica)	Eco-sistêmica (construtivista, interacionista, sociocultural, afetiva e transcendente)
Ontológica	- Realidade objetiva, estável, uniforme, homogênea, dada, fragmentada e previsível; - Explicação científica de natureza causal, determinista e reducionista.	- Realidade dinâmica e mutável, holística, construída, divergente, e múltipla; - Realidade inseparável do sistema de conhecimento.	- Realidade crítica e histórica, permeada de valores sociais, políticos, culturais, econômicos e éticos que se cristalizam com o tempo. - Concebe-se a realidade social sem a infraestrutura biofísica (sem vida)	- Realidade dinâmica, difusa, relacional, indeterminada e não-linear, contínua-descontínua; imprevisível. - Realidade: unidade dos contrários na qual prevalece a dinâmica do vir-a-ser sobre os modos de ser (Demo, 2005) - Realidade construída pela relação S/O. Multidimensional. - Diferentes níveis de realidade (realidade emerge como totalidade integrada). - Complexidade constitutiva da realidade, do pensamento e da ação. - Realidade: unidade global, complexa, integrada, interativa, participativa, colaborativa.
Epistemológica	- Base epistemológica: Empirismo de natureza dualista e objetiva. - Conhecimento objetivo, quantitativo, mensurável, e incontestável. - Conhecimento e descoberta. - Separação S/O. - Sujeito passivo, neutralidade do sujeito. - Processo epistemológico: regular, ordenado, progressivo, determinado pela ordem. - Conhecimento produzido e atomizado.	- Caráter fundamental: interpretativismo; - Base epistemológica: construtivismo e interacionismo simbólico. - Indivíduo como sujeito ativo na construção da realidade. - Subjetividade é fundamental. - Interatividade S/O diferente do positivismo) - Relação de interdependência S/O, marco interativo está nas relações e nos mecanismos de auto-regulação. - Conhecimento não é cópia da realidade. Resulta da ação e depende das estruturas do sujeito.	- Base epistemológica: construtivismo sociocrítico. - Produção social do saber é histórico-cultural, sendo a cultura um campo de luta para as conquistas sociais. - Conhecimento contextualizado e historicizado, sintonizado com a perspectiva de transformação social, coma construção de modelos sociais. - Condicionantes: o contexto e os interesses sociais. - Construtivismo como condição teórica, como representação conceitual concreta da natureza do conhecimento. - Conhecimento não como cópia, mas como construção do sujeito, como resultado da ação e dependente de suas estruturas. - Marco interativo: está da relação e nos mecanismos de auto-regulação.	- Base epistemológica: construtivista, interacionista fundada na intersubjetividade dialógica que gera uma base epistemológica complexa, a qual implica aceitação da natureza múltipla e diversa do sujeito e do objeto estudado, envolvendo dinâmica não-linear dialógica, interativa recursiva, e aberta. - Resgata a biopsicossócio-gênese do conhecimento humano. Conhecimento inscrito na corporeidade humana. - S/O ecologicamente indissociáveis e interdependentes. Objetividade ou a presença da intersubjetividade reveladora da impossibilidade de um conhecimento objetivo. - Somente existe objeto em relação ao sujeito o observa, que pensa, co-criação de significados. - Destacam-se os mecanismos de inter-relação, de auto-organização, de emergência, entre outros.

Metodológica	▪ Métodos quantitativos; ▪ Método único para ciências naturais e sociais: hipotético e dedutivo; ▪ Hipótese: ideia central; ▪ Dedução: operação mental; ▪ Neutralidade do sujeito; ▪ Procedimentos metodológicos básicos: controle experimental, observação sistêmica, correlação de variáveis, generalização dos resultados, construção de leis gerais a partir de premissas, ▪ Método apoiado na: causalidade linear, determinismo, relações diretas, mecânicas e lineares, asséptico com resultados confiáveis; ▪ Generalizações livres de contexto e do tempo; ▪ Confiabilidade, generalização, objetividade (critérios).	▪ Métodos qualitativos; ▪ Hermenêutica – dialética; ▪ Postula-se uma pluralidade de métodos para se interpretar a realidade (ex. estudos de casos, entrevistas, diários, etc.); ▪ Intercâmbio entre teoria e prática; ▪ Neutralidade do pesquisador; ▪ Os significados sociais somente podem ser examinados no contexto das interações individuais (diferentemente do positivismo) ▪ Critérios de rigor científico: credibilidade, confirmação e transitoriedade.	▪ Métodos qualitativos; ▪ Métodos dialéticos e participativos, resgatando as relações intersubjetivas; ▪ Busca-se a compreensão das raízes históricas da realidade; ▪ Foco na inter-relação, na descrição detalhada das situações, eventos pessoais e interações; ▪ Neutralidade científica é falácia; ▪ Dialética como relações intersubjetivas ▪ Validade consensuada.	▪ Predominância dos métodos qualitativos, mas sem negar o diálogo com os métodos quantitativos, desde que haja compatibilidade teórica e metodológica; ▪ Método como estratégia de ação aberta, adaptativa, e evolutiva do conhecimento; como caminho que se descobre ao caminhar, construído passo a passo e sujeito a bifurcações, retroações, desvios e recursões. ▪ Método apoiado na causalidade complexa; ▪ Procedimentos abertos, flexíveis, dinâmicos e revisáveis; ▪ Multimétodos, mas com prudência metodológica; ▪ Incerteza como caráter permanente na busca científica; ▪ Método transdisciplinar; ▪ Validade consensuada.
---------------------	---	---	--	--

Fonte: (MORAES; VALENTE, 2008, p. 16-17)

O Quadro das Perspectivas Teórico-Epistemológicas apresenta as dimensões ontológicas, epistemológicas e metodológicas de várias correntes de pensamento que evoluem do modelo clássico aos modelos contemporâneos e suas variações. Entre os modelos que nos interessam destacamos a Teoria Sociocrítica e Interacionista de Lev Semenovich Vigotski que tem como base a abordagem histórico-cultural, o modelo de Edgar Morin que tem uma metodologia baseada na causalidade complexa e, complementando o Quadro, acrescentamos o modelo da Teoria dos Signos de Charles Sanders Peirce que propõem uma visão do processo de conhecimento baseado em estruturas lógicas do pensamento.

É óbvio que nesta dissertação não pretendemos esgotar a discussão nem tampouco elencar todas as características existentes entre essas teorias, pois, caso tivéssemos tal intenção, com certeza não conseguiríamos realiza-la, em função da complexidade desses modelos e de suas características transdisciplinares. No entanto, indicamos a dissertação de Bruna Mazzer Nogueira (2016), “A Educação no Contexto Contemporâneo e o Pensamento Complexo” onde a pesquisadora realizou reflexões e problematizações buscando identificar as questões que envolvem a educação e o pensamento

complexo. O método adotado por ela permitiu uma análise cuidadosa e sistemática sobre a importância do caráter social no desenvolvimento humano e de suas complexidades. Com isso, espera-se deixar claro que não concordamos com a visão de que existe uma única abordagem teórica apropriada para o ensino e para a aprendizagem.

Aprender é uma atividade humana que, em razão das diferentes “teorias do conhecimento” existentes, não é entendida da mesma maneira por todos. Isso quer dizer que toda proposta de educação formal (quaisquer que sejam os recursos e ambientes utilizados) deve assumir uma certa epistemologia (PINO, 2003, p. 286).

Assim, nossa intenção é desenvolver um olhar crítico sobre as formas pelas quais a educação vem sendo realizada que nos conduzem a um aperfeiçoamento das práticas pedagógicas e da educação formal para que ela continue dando resultados positivos através dos tempos. Também nos parece coerente admitir que diferentes abordagens trazem consigo diferentes benefícios e que devem ser considerados no contexto de ensino e aprendizagem.

O objetivo principal desta pesquisa é refletir sobre uma forma e um modelo que trate do Pensamento Computacional e permita discutir alguns direcionamentos para a educação do século XXI, as urgências e as diferenças existentes entre a forma pela qual se dá o ensino e a aprendizagem atualmente.

1.6 Princípios de Morin

Segundo Silva, cada objeto criado via software cria uma nova realidade de uso das máquinas computacionais, às vezes ela se torna um equipamento de projeção de imagens, máquina de calcular ou mesmo em uma máquina de relações sociais. Em todos esses exemplos, o que produz efeito real sobre a cultura é a forma com que criamos as linguagens e as significações e a que elas estão sujeitas, ou seja, o processo comunicacional mais uma vez é estruturado entre a forma da emissão e o conteúdo a ser transmitido.

Aliada ao Pensamento Computacional as Tecnologias Emergentes tem auxiliado no processo de ensino e aprendizagem, especialmente na educação que vem se tornando uma ferramenta quase obrigatória no processo educacional. Nesta pesquisa nossa investigação tem como objetivo central

buscar compreender, através da elaboração de questionários e acompanhamento dos exercícios realizados pelos alunos, como se dá o uso das Tecnologias Emergentes e como elas podem auxiliar no processo de ensino e aprendizagem num curso de graduação em comunicação.

Em geral, estes alunos não possuem habilidades com as linguagens de programação, mas mesmo assim, com a proposta apresentada nestas duas disciplinas, eles podem executar aplicativos utilizando a programação com o *Processing*. A partir dos resultados obtidos aqui poderemos direcionar quais medidas devem ser adotadas para o ensino e aprendizagem da linguagem de programação através da matemática e lógica utilizando o pensamento computacional para estes alunos.

Os meios de comunicação e as Tecnologias Emergentes não só descentralizam as formas de transmissão e circulação do conhecimento, mas, também, projetam comportamento, estilo de vida, valores religiosos, étnicos, morais de diversas culturas. Eles são um processo global, frente ao qual não há mais a possibilidade de omissão. É preciso assumir a “Apropriação” das Tecnologias Emergentes de forma crítica, como uma dimensão estratégica da cultura.

Ao se apropriar das tecnologias atuais uma reorganização dos saberes acontece e um tipo de letramento específico vem acontecendo; é o letramento digital. Se consideramos que o uso destas tecnologias é um processo relacional onde as pessoas interagem com os objetos que, por sua vez, transformam as relações entre os usuários, suas ferramentas e seus contextos sociais.

Para desvelar o processo de “Apropriação” das Tecnologias Emergentes pelo sujeito é preciso considerar os princípios que norteiam o pensamento complexo como possíveis luzes para o conhecimento. Podem ser entendidas através dos sete princípios (MORIN, 2003, p.86):

O **princípio sistêmico ou organizacional**: cria uma ligação entre o conhecimento das partes ao conhecimento do todo, onde o indivíduo dotado de conhecimento adquirido como ser humano, apropria-se do convívio com a sociedade. Considerando a história da humanidade como uma história da fabricação, e tudo o mais, como meros comentários adicionais, torna-se possível

distinguir, grosso modo, os seguintes períodos: o das mãos, os das ferramentas, os das máquinas e dos aparelhos eletrônicos. Fabricar significa apoderar-se de algo na natureza, convertê-lo em algo manufaturado, dar uma aplicabilidade e utilizá-lo. Portanto, às fabricas são lugares onde aquilo que é dado é convertido em algo feito e com isso as informações herdadas tornam-se cada vez menos significativas, ao contrário das informações adquiridas, aprendidas, que são cada vez mais relevantes (FLUSSER, 2007, p. 36);

O **princípio holográfico**: evidencia aparentes paradoxos das organizações complexas, em que não apenas a parte está no todo, como o todo está inscrito na parte. Assim, cada célula é uma parte de um todo – o organismo global –, mas também o todo está na parte: a totalidade do patrimônio genético está presente em cada célula individual; a sociedade está presente em cada indivíduo, enquanto todo, através de sua linguagem, sua cultura, suas normas;

O **princípio do circuito retroativo**: a casualidade linear é rompida. A causa age sobre o efeito, e o efeito age sobre a causa, como na mudança de estados da matéria (do sólido ao líquido, do líquido ao gasoso e vice e versa). De modo mais complexo, “visão de mundo da ciência moderna”, a fórmula de Einstein, tudo é energia, ou seja, é a possibilidade de aglomerações da matéria. A “matéria”, nessa visão de mundo, equipara-se a ilhas temporárias de aglomerações (curvaturas) em campos energéticos de possibilidades, que se entrecruzam. Em sua forma negativa, o círculo de retroação (ou *feedback*) permite reduzir o desvio e, assim, estabilizar um sistema. Em sua forma positiva, o *feedback* é um mecanismo amplificador. Por exemplo: a violência de um protagonista provoca uma reação violenta, que, por sua vez, provoca uma reação mais violenta ainda. Inflacionárias ou estabilizadoras, são incontáveis as retroações nos fenômenos econômicos, sociais, políticos ou psicológico;

O **princípio do circuito recursivo** ultrapassa a noção de regulamentação com as de autoprodução e auto-organização. É um circuito gerador de que os produtos e os efeitos são, eles mesmos, produtores e causadores daquilo que os produz. Assim, nós, indivíduos, somos os produtos de um sistema de reprodução que vem do início dos tempos, mas esse sistema não pode se reproduzir se nós mesmos não nos tornarmos produtores com o acasalamento. Os indivíduos humanos produzem a sociedade nas interações e pelas

interações, mas a sociedade, à medida que emerge, produz a humanidade desses indivíduos, fornecendo-lhes linguagem e a cultura;

No **princípio da autonomia/dependência (auto-organização)**, os seres vivos são seres auto organizadores, que não param de se auto produzir e, por isso mesmo, despendem energia para manter sua autonomia. Como têm necessidade de retirar energia, informação e organização de seu ambiente, sua autonomia é inseparável dessa dependência; é por isso que precisam ser concebidos como seres auto ecoorganizadores. O princípio de auto ecoorganização vale especificamente, é óbvio, para os humanos – que desenvolvem sua autonomia na dependência de sua cultura – e para as sociedades – que se desenvolvem na dependência de seu meio geológico. Um aspecto chave da auto-ecoorganização viva é que ela se regenera permanentemente a partir da morte de suas células, segundo a fórmula de Heráclito, “viver de morte, morrer de vida”; e as ideias antagônicas de morte e vida são, ao mesmo tempo, complementares e antagônicas;

O **princípio do dialógico**: a dialógica permite assumir racionalmente a inseparabilidade de noções contraditórias para conceber um mesmo fenômeno complexo. Niels Bohr, por exemplo, reconheceu a necessidade de conceber partículas físicas como corpúsculos e como ondas, ao mesmo tempo. De um certo ponto de vista, os indivíduos, na medida em que desaparecem são com corpúsculos autônomos; de um outro ponto de vista – dentro das duas continuidades que são a espécie e a sociedade; e a espécie e a sociedade desaparecem quando se considera o indivíduo. O pensamento deve assumir dialogicamente os dois termos, que tendem a se excluir um ao outro;

O **princípio da reintrodução do conhecimento em todo conhecimento**: é um princípio que opera a restauração do sujeito e revela o problema cognitivo central: da percepção à teoria científica, todo conhecimento é uma reconstrução/tradução feita por uma mente/cérebro, e uma cultura e época determinadas.

Ainda segundo Morin, estes princípios indicam um modo de pensar, capaz de unir e solidificar conhecimentos separados, é capaz de se desdobrar em uma ética da união e da solidariedade entre humanos. Um pensamento capaz de não

se fechar no local e no particular, mas de conceber os conjuntos, estaria apto a favorecer o senso da responsabilidade e o da cidadania. A reforma de pensamento teria, pois, consequências existenciais, éticas e cívicas. (MORIN, 2003, p.86-96).

1.7 Pensamento Computacional

O Pensamento Computacional é um conceito em desenvolvimento que trata das questões que envolvem o processo de cognição, ensino e aprendizagem no mundo contemporâneo quando utilizamos procedimentos estéticos, tecnológicos e computacionais. Estamos interessados nas atividades de ensino e aprendizagem que se apoiam na utilização de linguagem de programação para produção de conhecimento. No caso, nosso foco principal é o aprendizado desenvolvido por meio da linguagem de programação *Processing* que é um ambiente de programação que permite a criação de narrativas interativas, jogos, simulações e produções computacionais que são desenvolvidas em linguagem de programação e que são compartilhadas numa comunidade *online*.

Nesta pesquisa pretendemos analisar as estruturas de desenvolvimento do Pensamento Computacional: conceitos, práticas e perspectivas computacionais que são elaboradas para o processo de ensino e aprendizagem. Cuny, Snyder e Wing (2010) definem o Pensamento Computacional como

os processos de pensamento envolvidos na formulação e solução de problemas que são representados de forma que possam ser efetivamente realizados por um sistema de processamento de informação (Tradução Livre).⁸

De fato, o termo “Pensamento Computacional” ou “*Computational Thinking*” foi definido pela primeira vez em um artigo por Jeannette M. Wing (2006) que afirmou que o “Pensamento Computacional” está baseado nos processos de computação que são realizados por seres humanos ou por máquinas. Ela também afirma que o pensamento computacional é uma

⁸ “the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent”

habilidade fundamental para todos, não apenas para cientistas da computação (p. 33).

A ideia que a programação de computadores ajuda no processo de aprendizagem foi identificada por Seymour Papert, em 1960 quando ele criou a linguagem de programação “Logo”. A atividade de uso da linguagem de programação tem uma importância fundamental no processo de construção do conhecimento e desenvolvimento do pensamento. Em 1971 ele argumentou que

A computação teve um profundo impacto na concretização e elucidação de conceitos em várias áreas de conhecimento: psicologia, linguística, biologia e na fundamentação em lógica e matemática. Tentarei mostrar como que essa elucidação acontece no ensino inicial destes conceitos. E assim, muito do que parece complexo para as crianças torna-se transparentemente simples; muito do que parecia ser mais abstrato e distante do mundo real transforma-se em instrumentos concretos familiarmente empregados para atingir objetivos pessoais (PAPERT, 1971, p. 3). (Tradução livre)⁹

Com o Pensamento Computacional estamos interessados nas formas como as atividades de aprendizagem, baseadas em programação por meio dos meios interativos, permitem o desenvolvimento do pensamento de jovens. Nosso contexto é o Processing onde as pessoas buscam aprender novas funções a partir do site Processing.org, uma comunidade que compartilha diversos exemplos (templates) de como os programadores podem produzir seus próprios programas.

Assim, podemos formular as seguintes perguntas: o que os alunos estão aprendendo quando utilizam as mídias interativa e a programação com o Processing? Qual é a aprendizagem que acontece quando utilizamos a programação com as mídias interativas, em vez de utilizar software de edição de vídeo ou apenas jogar um jogo? Tentando responder estas questões percebemos que o Pensamento Computacional ao ser realizado através do Processing permite um contexto e um conjunto de oportunidades que leva ao aprendizado. Há um grande interesse dos jovens no desenvolvimento de atividades de aprendizagem baseado no princípio do Pensamento

⁹ *Computation has had a profound impact by concretizing and elucidating many previously subtle in psychology, linguistic, biology, and the foundation of logic and mathematics. I shall try to show how this elucidation can be projected back the initial teaching of these concepts. By doing so much of what has been most perplexing to children is turned to transparent simplicity; much of what seemed most abstract and distant from the real world turns into concrete instruments familiarly employed to achieve personal goals.*

Computacional. Parte desse interesse é alimentado pela crescente disponibilidade de ferramentas que possibilitam que as pessoas criem seus próprios aplicativos e programas utilizando os meios computacionais. Mas, o que podemos destacar é que esse interesse está vinculado no compromisso com a aprendizagem através de atividades de programação, uma abordagem construcionista de aprendizagem que mostra a importância do envolvimento das pessoas com estas ferramentas computacionais.

Nos dois primeiros semestres de 2015 e 2016, ao estudar a atividade dos alunos do curso de Midialogia da Unicamp, desenvolvemos uma definição de Pensamento Computacional que envolve três dimensões-chave no processo de aprendizagem:

- ✓ Conceitos computacionais (os conceitos que os alunos devem possuir para terem autonomia de criarem seus próprios projetos e programas);
- ✓ Práticas computacionais (como os programadores e não programadores criam fluência no desenvolvimento de seus próprios projetos e programas),
- ✓ Perspectivas computacionais (as perspectivas de conhecimento de programação que as pessoas tem para o seu próprio desenvolvimento).

Neste estudo de campo a observação e as entrevistas com os alunos foram fundamentais para ajudar a compreender como se deu o desenvolvimento da aprendizagem. Observamos a participação nas aulas e os portfólios elaborados com os projetos desenvolvidos durante a realização da disciplina.

Conceitos sobre o Pensamento Computacional

A medida que os jovens projetam seus próprios aplicativos e realizam a programação com a linguagem Processing, verificamos que eles envolvem-se com um conjunto de conceitos computacionais (estruturas de programação com o Processing) que são comuns em muitas linguagens de programação. Identificamos sete conceitos que são muito importante para a realização de projetos que envolvem programação e que podem ser transferidos para outros contextos de programação (e não-programação). São eles: sequências, loops, paralelismo, eventos, condicionais, operadores e tratamento de dados. Para cada conceito foi pensado a elaboração de um exercício em aula.

- **Conceito de Sequências:** É um conceito-chave na programação que determinada como a atividade ou tarefa é expressa em uma série de etapas individuais ou instruções que podem ser executadas pelo computador. Como uma receita, uma sequência de instruções de programação específica ou de ação que devem ser produzidas. Por exemplo, a criação de uma linha, pode ser programada para se mover com um clique do mouse.
- **Conceito de Loop:** Os *Loops* são mecanismos para executar a mesma tarefa várias vezes. No exemplo anterior, a linha foi programada para mover-se quando o mouse estiver sendo pressionado. Assim, quando o mouse é pressionado uma ação é repetida e deste modo a linha se move cada vez que o mouse é pressionado.
- **Conceito de Evento:** Os eventos são coisas que fazem com que outra coisa aconteça. São componentes essenciais no processo de programação. Por exemplo, um botão de início de um vídeo e uma música ou de colisão de dois objetos. Isso pode fazer com que a pontuação de um jogo aumente.
- **Conceito de Paralelismo:** A maioria das linguagens modernas de programação suporta o paralelismo - sequências de instruções acontecendo ao mesmo tempo. O Processing suporta paralelismo entre objetos. Por exemplo, uma cena de festa de dança pode envolver vários personagens dançando simultaneamente, cada um com uma sequência única de instruções de dança.
- **Conceito de Condicional:** Um conceito-chave na programação é o conceito de condicional que é a capacidade de tomar decisões com base em certas condições. Isso apoia a obtenção de resultados múltiplos. O uso do comando “if” é condicional. Por exemplo, se um cubo está tocando a cor amarela, então ela deve desaparecer e reaparecer para o próximo nível do jogo; caso contrário, ele deve permanecer visível.
- **Conceito de Operadores:** Os operadores oferecem suporte para expressões matemáticas, lógicas e de sequência de caracteres, permitindo ao programador executar manipulações numéricas e de sequência de caracteres. Processing suporta uma série de operações matemáticas (incluindo adição, subtração, multiplicação, divisão, bem

como funções como seno e expoentes) e operações de sequência de caracteres (incluindo concatenação e comprimento de sequências de caracteres).

- **Conceito de Tratamento de Dados:** Os dados envolvem armazenar, recuperar e atualizar valores: variáveis (que podem manter um único número ou sequência de caracteres) e listas (que podem manter uma coleção de números ou palavras). Manter pontuação em um jogo é um elemento motivador que permite o envolvimento do jogador.

Ao projetar um aplicativo verificamos que este não é um processo direto, claro e sequencial. Ao identificar um conceito deste projeto estamos desenvolvendo um plano para realizar a solução deste projeto e assim, em seguida, devemos implementar o código para ver se resolvemos este problema. Este é um processo de adaptação em que o plano pode mudar várias vezes em resposta às aproximações da solução que queremos obter. Os programadores descrevem ciclos de criação e construção - desenvolvem um pouco, depois experimentam, e seguida desenvolvem mais um pouco e testam, assim a forma de programas acontece com base em suas experiências e novas ideias. A seguir iremos dar exemplos de ações que utilizam o software Processing e que serão explicadas no próximo capítulo com outros softwares que poderão ser utilizados para a concretização do Pensamento Computacional.

1.8 Pressupostos teóricos do processo de apropriação das Tecnologias Emergentes.

Ao observar a literatura através do entendimento do conceito de apropriação nas diversas áreas do conhecimento buscamos referências para subsidiar, iluminar, estabelecer analogias, compor um processo de análise que permitisse revelar como se dá a apropriação do conhecimento diante das Tecnologias Emergentes. Nessa perspectiva, e com base nos referenciais teóricos, foram elencados os pressupostos de uma apropriação das tecnológicas.

Segundo Borges, um acontecimento, numa perspectiva sempre crescente e contínua, num mesmo processo, se bem trabalhado pode levar a outros níveis

de apropriação mais elaborados e complexos. Para ele, a apropriação das tecnologias acontece em níveis que, apesar de suas particularidades, fazem parte de um único procedimento que deve ser estabelecido pelos princípios emocionais, técnico-operacional, de imitação, relacional, comunicacional, informacional, reflexivo e autoformacional. A seguir detalharemos cada um deles:

Emocional

Segundo Maturana (2001), o amor é a emoção que funda o social e não se esgota, ele está aí. Se não estivesse, não haveria dinâmica social e não haveria aceitação uns dos outros. A emoção do amor é constitutiva da aceitação do outro. Nós, como seres vivos sistêmicos e emocionais, a todo momento, operamos racionalmente; aprimoramos e utilizamos as mais diversas tecnologias para nossa realização pessoal e satisfação de nossos desejos. E, para satisfazê-lo, buscamos não só adquirir estes objetos que nos seduzem, atraem; potencializam os nossos sentidos de ver, ouvir, sentir e olhar; acenam com melhores possibilidades de lazer, trabalho, uso do tempo – mas também compreender com manuseá-los, utiliza-los. (BORGES, 2015, p.144).

A apropriação implica na ideia de desejo, movimento, mediação, construção, mobilização do próprio sujeito e de suas relações consigo mesmo, com o outro e com o mundo. Um processo em espiral, que tem como fio condutor o emocional, pois, o qual estando na base, é o nível que vai alimentar, permear todo o processo. No entanto, ainda é necessário que haja equipamentos tecnológicos, para que o sujeito possa dar um passo a mais no seu processo de apropriação, ou seja é preciso materializar o nível Técnico-Operacional.

Técnico – Operacional

Esse nível está diretamente relacionado à existência e disponibilidade de recursos, equipamentos tecnológicos, software e acesso à rede.

Imitação

Sem realizar uma análise profunda do termo imitação, busca-se compreender seu significado. Vigotski (1984) atribui a importância a imitação.

Segundo ele, sob a orientação de adulto, usando a imitação, as crianças são capazes de realizar muitas coisas, inclusive ações que estão muito além de suas próprias capacidades. Com a imitação, observando, compreende a ação do outro, e tenta reconstruí-la, construí-la à sua maneira, assegurado inclusive a possibilidade de superar, aprimorar, melhorar, modificar e recriar o objeto imitado. E, nesse processo, vai se apropriando dos meios que lhe permitem realizar a ação almejada. Um processo que é potencializado pela mediação do outro e pode levar ao nível mais complexo de apropriação, que é a relacional.

Em nosso estudo de campo sobre a Figura 17, podemos observar que após o professor conduzir o conteúdo abordado em forma de exercício 01, os alunos já letrados com as tecnologias puderam executar o exercício de forma a imitar o que o professor lhes mostrava, em seguida ainda em forma de imitação responderam a uma questão relacionado com que o que haviam sido alfabetizados a executar. Os indivíduos em sua totalidade responderem aos conceitos muito perto do que o professor havia explicado.

Relacional

Enquanto nossas emoções guiam momento a momento nosso agir. O relacional opera em nós instante a instante, e dão ao nosso agir seu caráter enquanto ações. É a configuração do emocionar que vivemos como Homo sapiens que especifica nossa identidade humana, não nossa conduta racional ou nosso uso de um tipo ou outro de tecnologia. O comportamento racional começa com nossa característica de viver em sociedade.

Segundo Charlot (2000), para entender um indivíduo, precisa-se entender uma atividade ou uma prática, há de se analisar o habitus, o sistema de disposições psíquicas que a baseiam. E, para conhecer o habitus, é preciso analisar as condições sociais em que ele foi construído. (p.62). Explica que o “sujeito de saber desenvolve uma atividade que lhe é própria: argumentação, verificação, experimentação, vontade de demonstrar, provar, validar” (p.61). A apropriação das tecnologias, portanto, não está nem no sujeito, nem nas próprias tecnologias, e nem no outro, mas nas relações estabelecidas entre eles.

Relação/Comunicação

A comunicação humana é um processo artificial. Baseia-se em artifícios, descobertas, ferramentas e instrumentos, a saber, em símbolos organizados em códigos. Os homens comunicam -se uns com os outros, de uma maneira caráter artificial da comunicação humana (onde o homem se comunica com outros homens por meio de artifícios) nem sempre é totalmente consciente, pois utilizamos códigos de gestos para nos comunicar (FLUSSER, 2007, p. 90). Ao comunica-se com o outro, podemos utilizar computadores/ smartphone/ rede, que seja falando, digitando, enviando e-mail, o indivíduo encontra-se em um dos níveis de apropriação.

Relação/Informação

Do ponto de vista etimológico, portanto, a manufatura corresponde ao sentido estrito do termo in+formação (literalmente, o processo de dar forma a algo). No sentido amplo, fabricar é informar. Daí deriva o sentido, menos usual, de “fabricar” como inventar ou engendrar ideias ou versões. Todo objeto manufaturado, por sua vez, tem como meta transformar as relações do usuário com seu entorno de modo a tirar proveito dele. Ao concretizar uma possibilidade de uso, o artefato se faz modelo e informação (FLUSSER, 2007, p.12).

Relação/Expressão reflexiva

Segundo (BORGES, 2015), expressão é entendida como um processo que permite ao sujeito externar, por meio das tecnologias, suas ideias, seus conhecimentos, valores, sentimentos, conjugando diversos tipos de linguagem e recursos disponibilizados na rede. É uma relação/expressão reflexiva que corresponde ao que foi denominado por Valente (2007) como letramento. A aquisição desse nível de apropriação permite que o indivíduo utilize as tecnologias digitais a favor da sua autoformação.

Autoformação

Expressa um processo em que o indivíduo encontra sua satisfação no próprio conhecimento adquirido no campo do conhecimento, da relação com o outro e da realização pessoal. Segundo Morin (2003), a educação deve contribuir para a autoformação da pessoa (ensinar a assumir a condição humana, ensinar a viver) e ensinar como se tornar cidadão. Um cidadão é definido, em uma

democracia, por sua solidariedade e responsabilidade em relação a sua pátria.

Nesse processo, o sujeito tem a consciência de que é responsável pela própria formação; de que é capaz disso e pode buscar os meios de realiza-la, permitindo-lhe o processo de transformação, que vai além do individualismo, do cognitivismo e utilitarismo do conhecimento.

Este capítulo trouxe as reflexões que levam a algumas constatações: a primeira é que as teorias de aprendizagem sempre tiveram relacionadas com a psicologia, a segunda é que os estudos relacionados à aprendizagem estão, hoje num âmbito muito mais amplo dos que as teóricas associativas. A terceira é que os teóricos como Piaget, Vygotsky, Morin e Peirce ao formular suas teóricas, não se preocupam somente com sujeito, mas passam a estudar também os meios em que vivem os signos e a união entre os três é efetivamente o que leva ao desenvolvimento cognitivo.

Movidos por estas reflexões deu-se sequência o capítulo 02 o qual serão relacionadas as questões envolvidas no aprendizado da disciplina, seguindo pelas matérias matemática e lógica que são os conteúdos iniciais para desenvolvimento de um aplicativo que em sua concepção efetiva passa pelo Pensamento Computacional.

CAPÍTULO 2 - APRENDIZADO DA DISCIPLINA

Tendo como aspecto central o Pensamento Computacional que nos leva a refletir sobre o tipo de ensino e aprendizado que estamos utilizando hoje, este estudo busca analisar uma metodologia de ensino e aprendizagem que se torna cada vez mais complexa. (Com base na teoria da complexidade de Morin e observando o trabalho de pesquisadores das áreas de Educação e Comunicação Social que utilizam o CHIC (*Classification Hiérarchique Implicative et Cohésitive* – Classificação Hierárquica, Implicativa e Coesitiva)). Pretendemos avaliar como se dá o processo de cognição de alunos que aprendem a programar em linguagem computacional a partir da lógica binária por meio das Tecnologias Emergentes.

2.1 A Linguagem Matemática

A matemática está presente em nosso dia a dia. O século XX não teria sido um dos momentos mais revolucionário da história da ciência sem os extraordinários desenvolvimentos obtidos na matemática. Os computadores não seriam o que são sem a lógica binária, a teoria dos grupos e o conceito matemática de informação. Os telefones não funcionariam se não tivesse existido o estudo estatísticos de sinais e algoritmos de digitalização e compreensão de dados. Os semáforos automáticos não seriam eficazes sem o desenvolvimento de uma área da matemática aplicada designada por investigação operacional.

No entanto, a matemática, ao mesmo tempo que se torna cada vez mais decisiva para as nossas vidas, é considerada, por muitos como uma ciência hermética e tecnicista, em que poucos conseguem se aventuram. A matemática é uma ciência fascinante, fundamental para a nossa história e onipresente no nosso dia a dia. As obras de Picasso e as transações bancárias via Internet, o número das portas das casas e o papel A4, os mapas modernos e a derrota de Hitler só foram possíveis graças a ela. As suas aplicações aparecem onde menos se espera. As histórias matemáticas são histórias de sucesso. Vejamos a seguir alguns exemplos,

2.2 Picasso, Einstein e a Quarta Dimensão

Conta-se uma história divertida sobre Picasso. Quando ele estava em sua casa aos 60 anos de idade e era um dos mais conhecidos artistas do mundo, uma senhora idosa e rica pediu-lhe um retrato. O pintor não se mostrou interessado, mas a senhora insistiu, oferecendo-se para pagar o que o artista quisesse. Picasso, enfadado, desenhou meia dúzia de riscos em um papel e entregou-os à senhora. “São cem mil francos”, disse. “Cem mil ?!”, surpreendeu-se a senhora. “Mas o senhor não demorou nem um minuto para fazer o desenho!” Com a tranquilidade dos seus 60 anos, Picasso teria replicado “Um minuto? A senhora está muito enganada. Demorei 60 anos”.

Não se sabe se a narrativa é verdadeira ou falsa, nem a idade do pintor ou o preço exato interessam para o caso. O interessante é que a história mostra que um artista aprende ao longo da vida e que é muito difícil saber de onde vêm as ideias, a técnica e o enquadramento necessário de uma determinada obra. As raízes da sua inspiração podem ser procuradas, contudo, estudando a vivência e o ambiente intelectual da sua época. Os historiadores de arte tiveram há muito a intuição de que o cubismo, um movimento artístico formalista e geometrizado, poderia ter sofrido a influência da revolução matemática e física que se operava no princípio do século XX.

O quadro *“Les Femmes d'Alger (O Versão O)”* (Figura 2) é uma das obras mais revolucionárias de Picasso. Finalizado em 1907 e medindo 2,44 por 2,34 metros a tela está exposta no Museu de Arte Moderna de Nova York, o célebre MoMA, e é uma das mais preciosas telas do acervo do museu. Ela representa um dos momentos decisivos da arte do século XX e é considerada, habitualmente, a produção que marca o nascimento do cubismo. Através dos cadernos de Picasso e de vários outros testemunhos, sabemos que o artista meditou longamente na preparação deste quadro. Nos seus esboços encontram-se inúmeros desenhos, um deles surpreende por ser a projeção de um sólido em quatro dimensões representada numa superfície plana. Em todos os esboços revela-se uma busca contínua da simplificação do corpo humano, que se aproxima de figuras geométricas simples.



Figura 02 – Les demoiselles de 1907.
Fonte: PICASSO ,1907

Finalmente, encontramos estudos de diferentes perspectivas que mostram as várias formas de se representar neste período histórico, onde a matemática fornece modelos que marcam as produções artísticas e matemáticas. As pinturas cubistas de Pablo Picasso e George Braque (1882-1963) apresentam a justaposição de diferentes perspectivas sobre a mesma tela, revelando pontos de vista distintos do mesmo objetivo. Isso foi possível de ser percebido em função da percepção matemática que existem vários modelos geométricos de representação. O cubismo explicita os vários olhares para uma cena que pode ser realizado por meio de câmeras fotografias e cinematográficas.

Outro artista que podemos destacar neste período histórico é Maurits Cornelis Escher que soube reproduzir as formas das Geometrias Não-Euclidianas. Ele era um artista holandês que viveu entre 1898 e 1972 e se tornou muito conhecido pelos trabalhos em xilogravuras e litogravuras na qual elaborou representações de figuras que são consideradas impossíveis. Ele trata de representações planimétricas que exploram metamorfoses, os modelos que propõem representar o infinito, enfim, padrões geométricos que geram representações paradoxais. As transformações geométricas que ele utiliza são: translações, rotações, reflexões e reflexões deslizantes que geram trabalhos

artísticos que são representações do espaço tridimensional em superfícies planas construindo situações impossíveis.

Para perceber como estas formulações artísticas poderiam ter origem nas preocupações científicas da época, basta reler os trabalhos do matemático francês Henri Poincaré (1854-1912), considerando um dos precursores dos modelos de representação da teoria da relatividade. Na sua *Science et Hypothèse*, uma obra de reflexão filosófica e divulgação científica publicada em 1902, explica como “se pode representar um mundo a quatro dimensões” partindo da analogia com a nossa visão, que nos “projeta na retina um quadro a duas dimensões” ” (2009, p. 126). Sabemos que os objetos têm três dimensões, explica Poincaré, porque os vemos sequencialmente em diferentes perspectivas. Assim, prossegue, “tal como se pode fazer num plano a perspectiva de uma figura a três dimensões”, pode-se “representar uma figura a quatro dimensões”. E podem-se “tomar várias perspectivas de pontos de vistas diferentes”, dando-se essa sequência de perspectivas a visão que teria um ser “que se deslocasse num espaço a quatro dimensões” (2009, p. 127).

Sabe-se que Einstein leu esta obra de Poincaré numa tradução alemã em 1904 e que essa leitura teve uma influência imensa em sua reflexão sobre o mundo físico e sobre o espaço-tempo quadridimensional. É pouco provável que Picasso tivesse lido Poincaré, mas sabe-se que um dos elementos influentes do seu grupo de amigos era Maurice Princet, um atuário com uma extensa cultura matemática. E sabe-se que esse homem, depois conhecido como “o matemático do cubismo”, lhe falava frequentemente da quarta dimensão, das geometrias não euclidianas e de outros temas científicos que fascinavam o grupo de Picasso.

O Quadro *Les demoiselles* mostra a solução de Picasso para o problema de representação quadridimensional: a pintura simultânea – e não sequencial como as representações de que falava Poincaré –, de diferentes perspectivas de um mesmo objeto. Pouco importa se o artista foi levado a criar o cubismo por inspiração direta na problemática revelada pelo matemático. A semelhança entre as preocupações geométricas de Picasso e as preocupações sobre o espaço – tempo de Poincaré e Einstein são demasiado evidentes para serem apenas coincidências. No entanto, Einstein afirmou que a visão entre o cubismo e a teoria da relatividade era diferente, a primeira representava os objetos por vários

ângulos de visão num mesmo momento, já a ideia de espaço-tempo de Einstein era de uma entidade unificada. Para ele, o espaço-tempo é composto em quatro dimensões: três espaciais e uma temporal. Assim, a teoria da relatividade unifica visões em uma representação e o cubismo apresenta várias visões num mesmo modelo de representação.

Há muito tempo que a matemática progride através de demonstrações rigorosas, com argumentos lógicos e premissas explicativas. Mas como se verifica a validade dos argumentos? Solucionar um problema matemático é como resolver um quebra-cabeça, exceto pelo fato de que nós não conhecemos com antecedência a aparência da solução final. Pode ser difícil, pode ser fácil, ou pode ser impossível de se resolver. Nós nunca saberemos até realmente conseguirmos resolvê-lo ou constatar que é impossível de se resolver.

Essa incerteza talvez seja o aspecto mais difícil de ser da Matemática. Em outras áreas do conhecimento podemos propor diversos caminhos de solução e até mudar as regras do jogo. Mesmo a própria ideia do que constitui uma solução não é claramente definida. Por exemplo, se formos incumbidos de melhorar a produtividade, de uma empresa, que métricas utilizaremos? Uma melhoria de 20% valera como solução do problema? E cerca de 10%? Em matemática e consequentemente em programação, os problemas são bem definidos e as soluções baseadas na lógica binária permite que se resolva o problema ou não.

Destacando a Lógica de Programação isso também ocorre, pois ou seu programa funciona dá uma resposta esperada ou simplesmente não funciona. Os erros podem acontecer na lógica da programação, nas escolhas das variáveis ou nas fórmulas e sintaxes erradas. Em programação, muitos são os fatores que podem nos levar a criação de um protótipo que não funcione ou que não foi criado de acordo com as expectativas esperadas.

2.3 Lógica & Matemática

A matemática e lógica se desenvolveram de forma independente, no entanto, são conhecimentos de mesma espécie. Na verdade, a partir do século XIX tornou-se impossível criar uma linha de separação entre elas. Bertrand Russell (1872-1970) defende, basicamente, que a matemática pode ser reduzida à lógica. Em outras palavras, os conceitos básicos da matemática podem ser

definidos em termos das leis fundamentais do raciocínio. Além do mais, ele argumentou que as definições matemáticas podem ser usadas em conjunção com princípios lógicos para dar origem aos teoremas matemáticos. Hoje, para o Pensamento Computacional, predominam os modelos dicotômicos e a lógica booleana que apresenta dois estados de representação que são mutuamente excludentes, as relações da tabela 2 são duais.

Tabela 2 – Sistema dicotômicos

1	0
Sim	Não
Dia	Noite
Preto	Branco
Ligado	Desligado

FONTE: DAGHLIAN ,2010).

Tradicionalmente, a lógica sempre lidou com as relações entre conceitos e proporções e com os processos pelos quais inferências válidas poderiam ser definidas como relações em formas genéricas. Como exemplo temos as seguintes formulações: “todo x é um y ; alguns z ’s são x ’s; logo alguns z ’s são y ’s” ou, como um silogismo, “todo homem é racional; Aristóteles é homem; logo Aristóteles é racional. Essas formulações são construídas para garantir automaticamente a verdade da conclusão, desde que as premissas sejam verdadeiras.

Por outro lado, alguns z ’s são x ’s não são válidas, já que podemos encontrar exemplos em que, apesar de serem verdadeiras as premissas, a conclusão é falsa. Por exemplo: “todo homem é um mamífero; alguns animais com chifre são mamíferos; logo, alguns animais com chifre são homens”. Desde que algumas regras sejam seguidas, a validade de um argumento não dependerá das sentenças. Por exemplo: o mordomo matou o milionário ou sua filha o matou; sua filha não o matou; logo, o mordomo o matou. Com estas premissas produz-se uma dedução válida.

A solidez do argumento não depende de nossa opinião sobre o mordomo nem sobre a relação entre o milionário e sua filha. A validade aqui é garantida pelo fato de proposições de formas gerais do tipo “se p ou q e não q , então p ” produz uma verdade lógica.

Podemos perceber que, nos dois primeiros exemplos, as variáveis x , y e z desempenham papéis bem semelhantes aqueles das variáveis em equações matemáticas (marcam o lugar onde expressões podem ser inseridas, da mesma maneira que valores numéricos são inseridos para variáveis em álgebra). Assim, a verdade na inferência “se p ou q e não q , então p ” pode lembrar os axiomas da Geometria de Euclides. Mesmo assim, foi necessário passar quase dois milênios de contemplação da lógica para que os matemáticos levassem esta ciência a sério.

No dia a dia empregamos todo o tipo de argumento com os mais variados conteúdos (políticos, religiosos, morais etc). Aqui, nosso objetivo é convencer o interlocutor que estamos certos em nossos argumentos. Podemos pensar em lógica como o estudo da validade dos argumentos, focalizando a atenção não no conteúdo, mas sim na sua forma ou na sua estrutura. A lógica, também chamada de formal, simbólica ou ainda matemática, pode ser tratada por três concepções:

- lógica com um sistema de regras;
- lógica como um conjunto de leis;
- lógica como estrutura linguística.

Podemos associar a cada concepção um determinado período histórico da evolução desta ciência. A grosso modo, estes três períodos correspondem, respectivamente, às três concepções de lógica: 1º) período grego (século IV a.C. até o início do século XIX); 2º) período booleano (século XIX e primeira década do século XX) e o 3º) período contemporâneo.

A busca de regras que assumem a validade de um argumento dominou o primeiro período, que se inicia por volta de IV a.C. com a primeira sistematização conhecida como lógica, uma coleção de tratados denominada *Organon*. Leibniz apresentava seu sistema em uma determinada perspectiva e a partir de uma ideologia universal, próxima da concepção contemporânea e, com um modo de se operar os símbolos muito próximo das concepções do cálculo algébrico.

No segundo período, a partir do século XIX, a lógica começou a evoluir num sentido mais matemático ou, mais precisamente mais algébrico. A evolução nesse sentido representa uma mudança nas concepções sobre a lógica que passou então a buscar as suas leis como um resultado do paralelo de fórmulas

algébricas com o cálculo das classes. Essa aproximação da lógica com a álgebra deu-se sob a influência de George Boole (1815-1864), Augustus de Morgan (1806-1871) e outros. Neste momento histórico surgem grandes reflexões sobre os princípios lógicos e uma concepção referente à natureza da matemática. G. Frege (1848-1925) e C. S. Peirce (1839-1914) introduziram novas formulações e reflexões sobre a lógica e matemática. Frege desenvolveu um sistema lógico por um método linguístico (cálculo proposicional). Ele acreditava que poderíamos deduzir a matemática da lógica e, assim, buscou mostrar que a matemática poderia ser definida em termos lógicos. Já para o filósofo e lógico Peirce,

a verdadeira lógica está baseada numa espécie de observação do mesmo tipo daquela sobre a qual se baseia a matemática, e essa é quase a única, ou senão a única ciência que não necessita de auxílio algum de uma ciência da lógica (PEIRCE 1975, p. 21).

Assim, a lógica passa a ocupar seu espaço enquanto ciência do raciocínio ao lado da matemática. A partir das reflexões de A. Tarski (1901-1983), A. Turing (1912-1954), A. Church (1903-1995), E. Zermelo (1871-1953), e muitos outros, as discussões sobre os objetos lógicos e matemáticos começam a ser formuladas. Podemos observar que a Lógica e os modelos abstratos passaram a ser estudados a partir de A. L. Cauchy (1789-1857), N. H. Abel (1802-1829) e K. Weierstrass (1815-1897), que discutem os fundamentos destas ciências, buscando encontrar apoios sólidos para a aritmética, a álgebra, o cálculo diferencial, o cálculo integral, enfim, toda a análise matemática. Para Boyer (1974, p. 94),

o método axiomático é o caminho lógico para a aritmetização da análise, onde, a noção de espaço vetorial transforma nosso modo de perceber, operar e pensar sobre as geometrias. A 'dissociação entre objetos e operadores' passa a ser o principal aspecto 'para a constituição de uma estrutura vetorial' (HILDEBRAND, 2001, p. 49).

As reflexões sobre a lógica e a matemática intensificam com B. Riemann (1826-1866) que ao observar “à curvatura dos espaços riemannianos” afirmou que a geometria não deve ser pensada por pontos e isso nos conduziu aos princípios da teoria da relatividade de Einstein. Por outro lado, o conceito de “cortes de Dedekind” estabelecem a separação entre a análise matemática e a geometria e estas teorias passam a ser pensadas de forma abstrata e lógica.

Por fim, o terceiro momento, o período contemporâneo começa a partir do início do século XX – mais precisamente a partir da publicação em três volumes

dos *Principia Mathematica*, de A. N. Whitehead (1861-1947) e B. Russell (1872-1970) entre outros. É nesse período que o enfoque linguístico-formal se impõe, ou seja, a lógica passa a ser vista como uma linguagem, ou como um sistema de signos, pressupõe uma sintaxe (regras ou leis de combinação dos signos) e uma semântica (interpretação e significado dos signos). Peirce afirma que a Lógica é igual a Semiótica e cria toda a sua Teoria dos Signos com um modelo de pensamento lógico. Assim, hoje, existem vários modelos lógicos, cada um associado a uma determinada estrutura linguística. A partir de 1930, até nossos dias, a evolução da lógica caminha em uma direção de maior integração à matemática atingindo uma complexidade técnica elevada e ampliando consideravelmente o seu domínio de conhecimento e suas aplicações. Finalizando a evolução das ciências lógicas, constatamos que existem vários modelos de se pensar a lógica: lógica clássica, lógica difusa, lógica para completa, lógica para consistente desenvolvida pelo brasileiro Newton Costa. Assim, “encontramos inúmeros modelos lógicos que nos permitem mostrar a infinidade de interpretações possíveis que estão diante de nós” (HILDEBRAND, 2001, p. 50).

Tabela 3 – Sinopses das várias lógicas

Sinopse dos vários Modelos Lógicos	
Clássica	Cálculo de predicados de primeira ordem a) Teoria dos conjuntos. b) Teoria dos tipos. c) Teoria de categorias como fundamento da matemática.
Não clássica	Complementares da clássica
	a) Epistêmica clássica. • Lógica de crença. • Lógica do conhecimento. b) Modal clássica. c) Clássica de ação etc. d) Intencionais clássica etc.
	Heterodoxas
	a) Para completas. b) Para consistentes. c) Não aléticas. d) Quânticas. e) Relevantes. f) Modais para consistentes. g) Epistêmicas para completas. h) Indutivas para consistentes etc.

Fonte: (BISPO, 2014.)

Em programação a lógica e matemática se entrelaçam em conceitos e formas para que instruções em um computador escritas em uma linguagem de programação possam ser executadas. Independente da linguagem em que um programador escreve um programa ou aplicativo, todas têm regras que regulamentada o uso de palavras e da pontuação. Essas regras são chamadas de **sintaxe** da linguagem.

Um programa sem erros de sintaxe pode ser executado em um computador, mas ele pode não produzir os resultados corretos. Para que um programa funcione como queremos, o computador precisa receber instruções em uma sequência específica, não se deve deixar instruções de fora e não se deve inserir instruções irrelevantes. Fazendo assim, quando da execução do programa em uma lógica correta o programa do computador terá, a partir das instruções recebidas, condição de executar o programa corretamente.

O computador se tornou uma ferramenta fundamental para o desenvolvimento de toda e qualquer atividade humana. Pode parecer exagero tal assertiva, mas imaginem se uma explosão eletrônica destruísse todos os computadores eletrônicos baseados em silício no mundo. Teríamos um caos no sistema financeiro, transporte, comunicação, produtivos, administrativos, defesa etc. Resumindo, regrediríamos alguns séculos e, certamente, um novo sistema computacional iria surgir.

2.4 Desenvolvimento de um aplicativo

Segundo Pino (2003), um programa de computador nada mais é que um algoritmo escrito em uma linguagem específica. Todo o nosso trabalho estará voltado para a ideia de construção de algoritmos que serão traduzidos em programas de computador. Examinaremos vários algoritmos para resolver diferentes tipos de problemas. Vamos começar o trabalho analisando os passos para o desenvolvimento de um aplicativo, a partir da leitura da proposta de um problema. Essa fase é dividida em etapas conforme segue:

2.4.1 Etapa 01 – Estratégia

Antes da preocupação em resolver um problema, devemos ter certeza da compreensão exata e completa sobre a proposta do problema. O trabalho inicial

consiste em leitura cuidadosa e análise minuciosa do enunciado, estabelecendo o conjunto de informações correspondente à entrada de dados (informações disponíveis) e qual será o resultado desejado (informações que se pretende obter), ou seja, a saída de informações. Estas especificações podem não ser suficiente para o entendimento do problema, mas são absolutamente necessárias.

2.4.2 Etapa 02 – Algoritmo

A partir do processo que deve ter sido exaustivamente analisado e modelado na etapa anterior, descrevemos o algoritmo, utilizando o recurso de diagrama de blocos ou pseudo-linguagem. Nesta etapa devemos fazer uma simulação do algoritmo para percebermos e corrigirmos eventuais erros de lógica ou erros de cálculo.

2.4.3 Etapa 03 – Implementação

A preparação do programa é iniciada com o esboço de tela que será a interface entre o usuário e o aplicativo. Depois realizamos a escrita dos códigos a partir do algoritmo obtido na etapa anterior. Em seguida, testamos o programa corrigido eventuais erros não detectados nas etapas anteriores.

2.5 Paradigmas de programação

Uma linguagem de programação é determinada pelo grau de liberdade que o usuário da linguagem tem em relação as particularidades dos computadores e até que ponto lhe é permitido pensar em termos associados ao problema que está sendo trabalhado. Na realidade, o desenvolvimento de linguagem de programação não progrediu dessa maneira, mas através de diferentes caminhos, à medida que abordagens alternativas ao processo de programação (chamados de paradigmas de programação) emergiram e foram buscadas. Consequentemente, o desenvolvimento histórico das linguagens de programação é mais bem retratada por um diagrama de múltiplas linhas.

Programação não é apenas para engenheiros ou para programadores. Muitas pessoas pensam que a programação é apenas para pessoas que são bons em matemática e outras disciplinas técnicas. É possível criar diferentes

tipos de linguagens de programação e ambientes que envolvem pessoas com mentes visuais e espaciais.

Em 1960 o Logo era uma linguagem alternativa inicial, projetada no final dos anos 1960 por Seymour Papert como um conceito de linguagem para as crianças. O Logo tornou possível que crianças programassem muitos meios diferentes, incluindo uma tartaruga robótica e imagens gráficas na tela. Outro exemplo contemporâneo é o ambiente de programação Max desenvolvido por Miller Puckette na década de 1980. Max é diferente de línguas típicas; seus programas são criados por ligação de caixas que representam o código de programa, em vez de linhas de texto. Tem entusiasmo gerado a partir de milhares de músicos e artistas visuais que utilizam como base para a criação de áudio e software visual.

Da mesma forma interfaces gráficas de usuário abriu o caminho da computação para milhões de pessoas, ambientes de programação alternativas vai continuar a nos permitir que as novas gerações de artistas e designers possam trabalhar diretamente com programas. Espera-se que o *Processing* possa incentivar muitos artistas e designers a utilizar o software e que irá estimular o interesse em outros ambientes de programação construídas para as artes.

2.5.1 Logo

Desenvolvida em 1967, por Seymour Papert, educador matemático no MIT e adaptada para o português em 1982, na Unicamp, pelo Núcleo de Informática Aplicada à Educação, a linguagem logo fornece importante ferramenta de aprendizagem ao ensino de lógica as crianças. A principal diferença entre Logo e as demais linguagem de programação é que foi desenvolvida para ser usada por crianças e para que as crianças possam com ela aprender coisas. Nela está embutida a filosofia da educação não diretiva, de inspiração piagetiana, em que a criança aprende explorando seu ambiente.

Logo é uma linguagem simples e poderosa. Sua simplicidade está no fato, de ser fácil de aprender: pessoas alfabetizadas, de qualquer idade podem programar em seu primeiro contato com ela. A proposta da linguagem Logo era colocar a criança para comandar um robô ou uma representação de robô na tela

do computador. Um dos primeiros robôs controlado pela Linguagem Logo lembrava a forma de uma tartaruga, a partir daí a tartaruga passou a ser o símbolo dessa linguagem. A partir de comandos como "para frente 100" (pontos), e "para direita 45" (graus), movimenta-se o robô pelo espaço. Quando o robô está representado na tela do computador, a tartaruga deixa um rastro por onde passa, criando assim um desenho. O termo Logo é derivado do grego "*logos*" que significa "palavra", uma referência aos comandos da linguagem. A figura 3 demonstra o ambiente de programação mais conhecido do Logo é o da tartaruga, que era inicialmente um pequeno robô controlado por comandos de computador.



Figura 3 – O robô “tartaruga” – Linguagem *Logo*.
Fonte: (PAPERT,1980).

2.5.2 Matlab

O MATLAB (do inglês *Matrix Laboratory*) criado em 1970 por Cleve Moler, então presidente do departamento de ciência da computação da Universidade do Novo México. Ele logo se espalhou para outras universidades e encontrou um forte uso no âmbito da comunidade matemática aplicada. Jack Little, um engenheiro, conheceu a linguagem MATLAB, durante uma visita feita por Moler a Universidade de Stanford em 1983. Reconhecendo o seu potencial comercial, ele juntou-se a Moler e Steve Bangert. Eles reescreveram MATLAB em C¹⁰, em 1984 fundaram a MathWorks e prosseguiram no seu desenvolvimento. As

¹⁰ C. Linguagem de programação da década de 1970.

bibliotecas reescritas ficaram conhecidas como LAPACK¹¹. É um software de computação numérica de análise e visualização de dados. Embora seu nome signifique Laboratório de Matrizes, seus propósitos atualmente são bem mais amplos. Ele nasceu como um programa para operações matemáticas sobre matrizes, mas ao longo dos anos transformou-se em um sistema computacional bastante útil e flexível.

Seu ambiente de trabalho é fácil de ser utilizado, pois os problemas e soluções são escritos em linguagem matemática e não na linguagem de programação tradicional, como muitos outros softwares utilizam. Assim o MATLAB é uma ferramenta e uma linguagem de programação de alto nível, e tem como principais funções: construção de gráficos e compilação de funções, manipulação de funções específicas de cálculo e variáveis simbólicas.

Além disso, o MATLAB possui uma grande quantidade de bibliotecas auxiliares (“Toolboxes”) que otimizam o tempo gasto para realizar tarefas, uma vez que, o usuário poderá utilizar muitas funções já definidas, poupando o tempo de criá-las. Por outro lado, infelizmente, os programas feitos são difíceis de serem executados num ambiente fora do MATLAB. Figura 4, mostra como criar um gráfico de contorno de superfície utilizando o software MATLAB

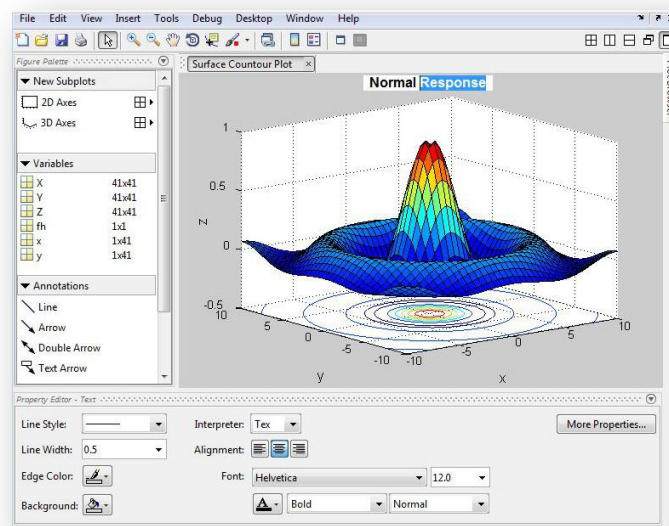


Figura 4 – Como editar e codificar mapas sem codificação.
Fonte:(MATHWORKS, 2014)

¹¹ LAPACK. Biblioteca auxiliar desenvolvida para trabalhar em conjunto com o software Matlab.

2.5.3 Python

Em 1991, o matemático Guido Van Rossu criou a linguagem de programação Python, escreveu na documentação oficial da linguagem um breve histórico do início do seu desenvolvimento. Conta que no final de 1990 nos laboratórios do CWI, foi criado o Python como uma linguagem de scripts para o Sistema Operacional Distribuído Amoeba. Evolucionada da linguagem ABC, que tinha um foco para usuários como físicos, engenheiros e linguistas, ela nasce para ser uma linguagem para os leigos em computação também a utilizarem. Em 1995, continuando o trabalho na CNRI nos Estados Unidos e depois em 2000 na BeOpen.com, formando a BeOpen PythonLabs (WIKIPÉDIA, 2015).

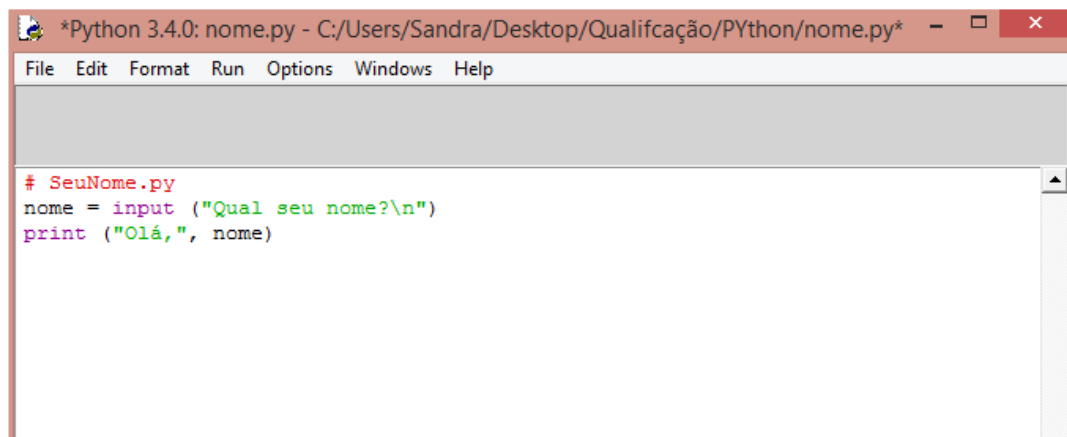
Python é uma linguagem multiplataforma: em geral, o Python pode ser executado no Windows e em sistemas derivados do Unix, como Linux, BSD e o de MAC OS X, por meio da cópia de arquivo, ou arquivos, que compõem o programa na máquina-alvo, sem nenhuma “construção” ou compilação necessária. É possível criar programas Python que usam funcionalidades de uma plataforma específica, mas é raro que isso seja realmente necessário, já que quase todas as bibliotecas padrão do Python e as de terceiros são completa e transparentemente multiplataforma.

Python é, provavelmente, a linguagem de programação popular mais fácil e agradável de lidar. O código de Python é simples para ler e escrever, e consegue ser conciso sem ser algo enigmático. Python é uma linguagem muito expressiva, o que significa que podemos normalmente escrever em muito menos linhas de código em comparação ao que seria necessário escrever em uma aplicação equivalente, digamos, em C++¹² ou Java¹³. Figura 5 mostra um exemplo de um programa em Python que pede o usuário digite seu nome e após os dados digitados será impresso na tela do programa a frase “Olá, e o nome”

¹² C++, Linguagem de programação criada na década de 1970.

¹³ Java Linguagem de programação criada na década de 1970

Figura 5 – Ambiente de desenvolvimento Python: Fonte IDLE 3.4.0



Geralmente, você pode criar programas que tenham mais de uma só linha, de modo que o Python vem com um editor para escrever programas. Figura 6 mostra um exemplo de programa que manipula a classe ColorSpiral.py, que usa variáveis para alterar a cor , o tamanho e o angulo para girar as imagens espirais.

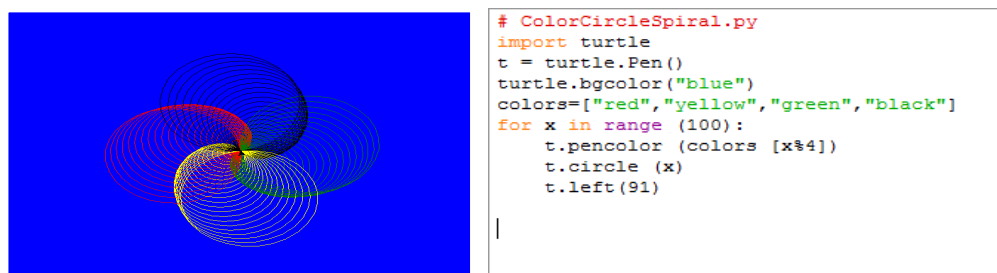


Figura 6 – Aplicativo com figura Geométrica:

Fonte: (Software IDLE 3.4.0, 2016)

2.5.4 Scratch

Scratch é uma linguagem de programação desenvolvida por Lifelong Kindergarten Group no Media Lab, MIT (com financiamento da National Science Foundation, Intel Foundation, Nokia e do consórcio de pesquisa do MIT Media Lab). Este aplicativo possibilita a criação de histórias interativas, jogos e animações bem como o compartilhamento das criações na Web. Tudo pode ser feito a partir de comandos que devem ser agrupados de modo lógico ().

A tela inicial do Scratch é composta por (A) uma área que apresenta e possibilita a escolha dos grupos de comandos desta linguagem de programação, (B) uma área edição que possibilita a criação do projeto, ou a programação de

Assim a criação de um projeto no Scratch requer a escolha de comandos da linguagem de programação, a edição de um projeto que envolve a programação utilizando elementos gráficos para compor o “palco” da estória, a definição de scripts ou “rotinas de ações” a partir do uso de comandos, especificação de parâmetros, sprites (objetos), trajes e sons. Figura 7 demonstra o ambiente de desenvolvimento e como as variáveis são usadas na animação do jogo.



2.5.5 App Inventor

68

Os blocos da ferramenta "MIT App Inventor", são parecidos com um puzzle, encaixando uns nos outros criando um conjunto de regras e operações que são desenvolvidas ao longo da utilização da aplicação criada. Para que as regras e operações sejam realizadas é necessário que o utilizador da aplicação interaja com esta através da interface gráfica. Aí são colocados os elementos de entrada e saída de informações. Figura 8 demonstra a criação de um projeto usando o cenário do jogo Minecraft.

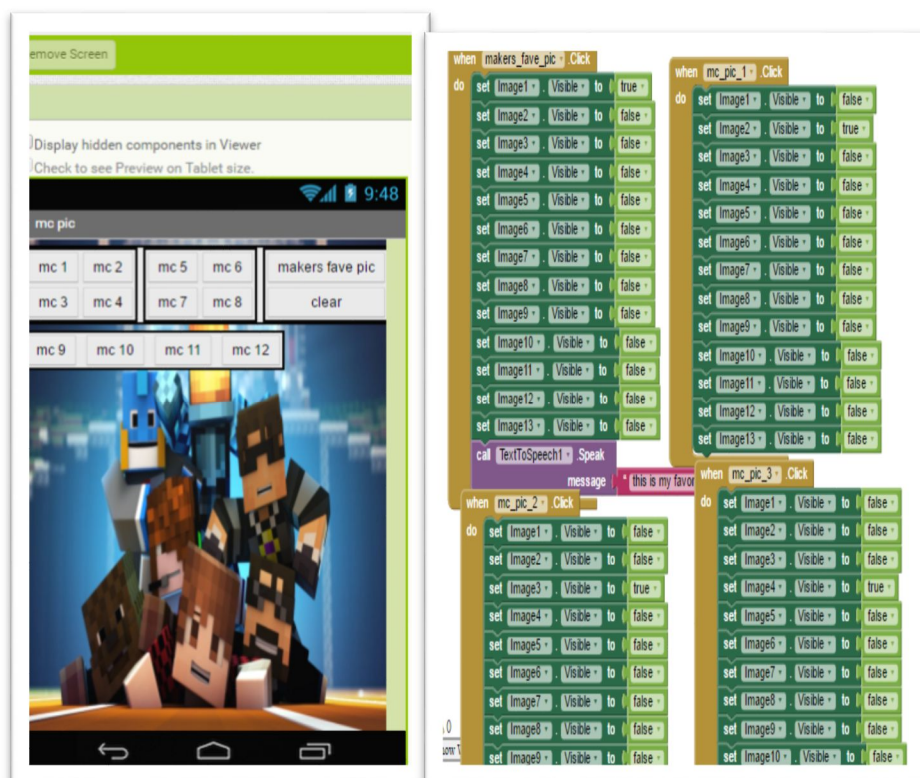


Figura 8 – Ambiente de desenvolvimento do App Inventor2
Fonte: (AI2.APPINVENTOR.MIT.EDU, 2015)

2.5.6 Alice

Alice é um ambiente de programação em 3D inovador que torna mais fácil para criar uma animação para contar uma história, jogar um jogo interativo, ou um vídeo para compartilhar na web. Alice é uma ferramenta de ensino disponíveis gratuitamente projetado para ser a primeira exposição de um aluno para orientada a objetos de programação. Ele permite que os alunos a aprender conceitos fundamentais de programação no contexto da criação de filmes animados e jogos de vídeo simples. Em Alice, objetos 3-D (por exemplo,

peçoas, animais e veículos) preencher um mundo virtual e os alunos a criar um programa para animar os objetos.

Na interface interativa de Alice, o aluno arrasta e larga telhas gráficas (cenários disponibilizados no software) para criar um programa, onde as instruções correspondem a declarações padrão em uma linguagem de programação orientada a produção, tais como Java, C++ e C#. O software Alice permite ver imediatamente como seus programas de animação podem correr, permitindo compreender facilmente a relação entre as instruções de programação e o comportamento dos objetos em sua animação. Ao manipular os objetos em seu mundo virtual, os estudantes ganham experiência com todas as construções de programação normalmente ensinadas em um curso de programação introdutório. Figura 09 demonstra o ambiente de desenvolvimento do software Alice

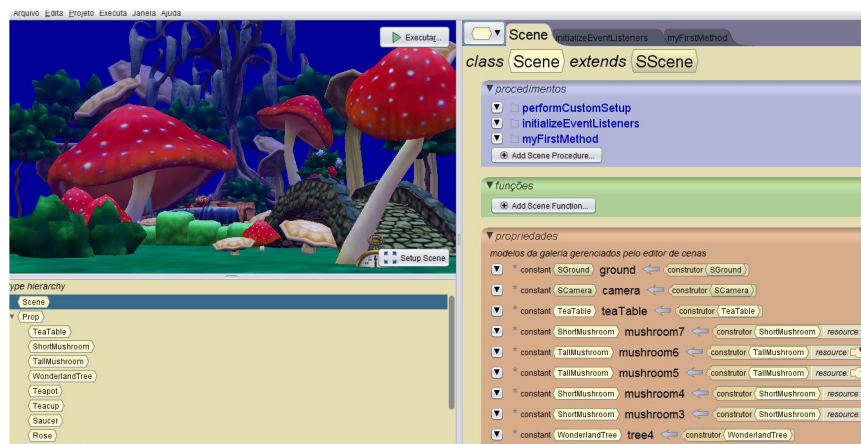


Figura 9 – Ambiente de desenvolvimento Alice
Fonte: (SOFTWARE ALICE 3.2, 2016.)

2.5.7 Arduino

Em termos prático, um Arduino é um pequeno computador que pode ser programado para processar entradas e saídas de um dispositivo e os componentes externos conectado a ele (Figura 10). O Arduino é o que chamamos de plataforma de computador física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de hardware e software.

O Arduino pode ser utilizado para desenvolver objetos interativos independentes, ou pode ser conectado a um computador, a uma rede, ou até

mesmo à Internet para recuperar e enviar dados do Arduino e atuar sobre eles. Em outras palavras, ele pode enviar um conjunto de dados recebidos de alguns sensores para um site, dados estes que poderão, assim, ser exibidos na forma de um gráfico (McROBERTS, 2011).

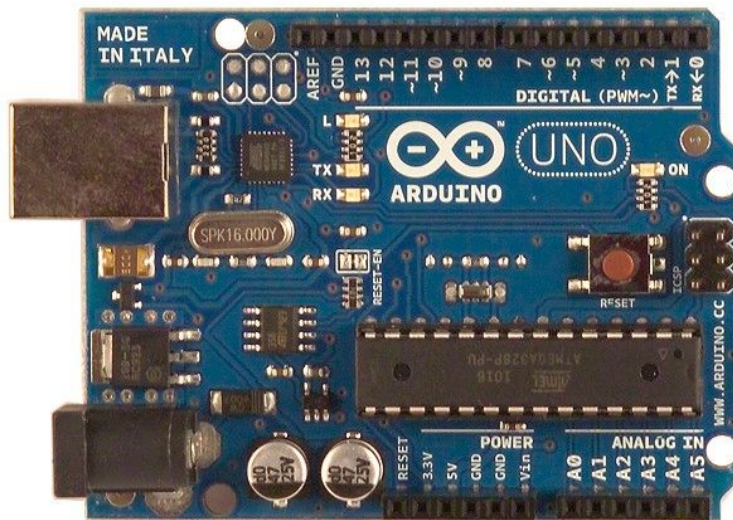


Figura 10 – Placa Arduino Uno.
Fonte:(McROBERTS, 2011.)

2.5.8. Processing

O *Processing* surgiu em 2001 e foi criado por Ben Fry e Casey Reas, na época eles eram alunos de doutorado do *MIT Media Lab*, sob a orientação do Professor John Maeda e membros do grupo de pesquisa *Aesthetics and Computation*. O objetivo de Fry e Reas era elaborar uma linguagem de programação simples, que inspirasse o processo criativo nas artes visuais e pudesse ser utilizada por artistas e designers. (PROCESSING.ORG, 2015)

O *Processing* teve como inspiração direta um outro projeto que foi liderado por Maeda e lançado em 1999: a linguagem *Design By Number* – DBN (Figura 11). Durante o doutorado Fry e Reas trabalharam na manutenção e no desenvolvimento do DBN e essa experiência serviu como base para o projeto do *Processing*. Enquanto trabalhava no DBN, Fry criou vários experimentos em outras linguagens de programação (*Phyton* e *Scheme*) e algumas funcionalidades visuais, que, no entanto, não cabiam nele. Interessados em combinar as capacidades mais complexas realizadas por Fry nos seus

experimentos com o aspecto educacional do DBN, Fry e Reas criaram o *Processing*.

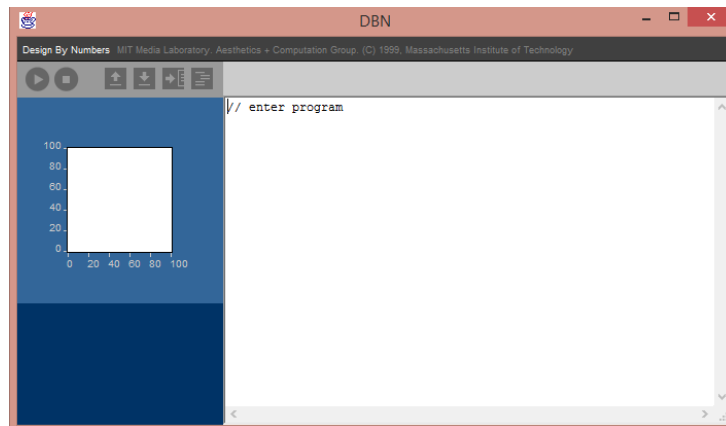


Figura 11 – Designer By Number.
Fonte: (Elaboração própria - SOFTWARE DBN, 2016.)

O *Processing* é uma linguagem de alto nível, orientada a objetos, composta de dois aspectos fundamentais: um ambiente de desenvolvimento (IDE) e a linguagem de programação em si, foi criado utilizando como base o Java e, por isso, tem muito potencial, havendo muito pouco que não seja possível fazer com ela. Permite a escrita de programas que gráficos (sketches) sem a complexidade e a estrutura rígida normalmente presente na linguagem Java. Figura 12 a seguir mostra exemplo de código em *Processing* onde as imagens geradas na tela são criadas com a movimentação do mouse.

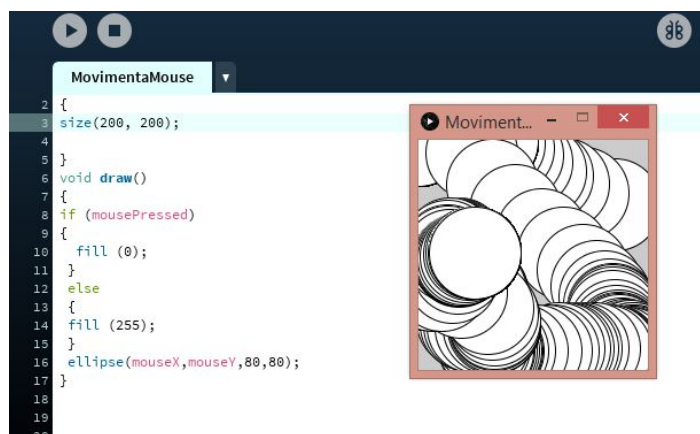


Figura 12 – Programa em *Processing* que captura o movimento do mouse na tela.
Fonte: (Elaboração própria – SOFTWARE PROCESSING, 2016.)

Como o *Processing* é voltado para as artes visuais, no output do programa é primordial um desenho ou representação visual – diferentes de outras linguagens de programação que privilegiam o output em texto. No universo da

programação existe uma tradição: o primeiro programa escrito em uma nova linguagem deve produzir o texto “*Hello, World!*” (Olá Mundo!). No Processing, todavia, um programa *Hello, World!* é, normalmente, uma imagem, como a linha mostrada na Figura 13.

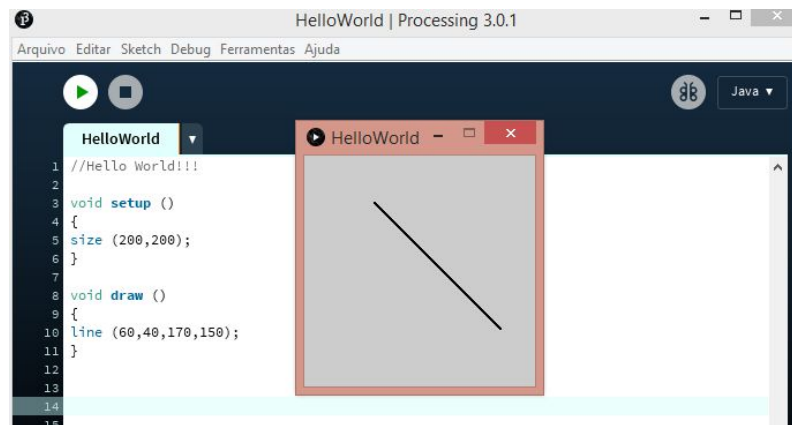


Figura 13 – Programa em Processing denominado *Hello, World!*.
Fonte: (Elaboração própria – SOFTWARE PROCESSING, 2016.)

A linguagem *Processing* não representa uma ruptura radical em relação às linguagens de programação clássicas. Diferentemente das linguagens de programação analisadas neste capítulo, o *Processing* é uma linguagem textual. Isso pode representar para algumas pessoas um nível maior de dificuldade. No entanto, o Processing reposiciona a programação textual e linear de uma maneira que a torna alcançável para pessoas que se interessam por ela, mas que se sentem intimidadas ou desinteressadas pelo tipo de programação que acontece nos departamentos de ciência da computação (REAS; FRY, 2006). Com o Processing é possível produzir imagens com comando mais simples, enquanto que no Java, um programa para realizar a mesma tarefa requer mais linhas de código, como pode ser observado abaixo:

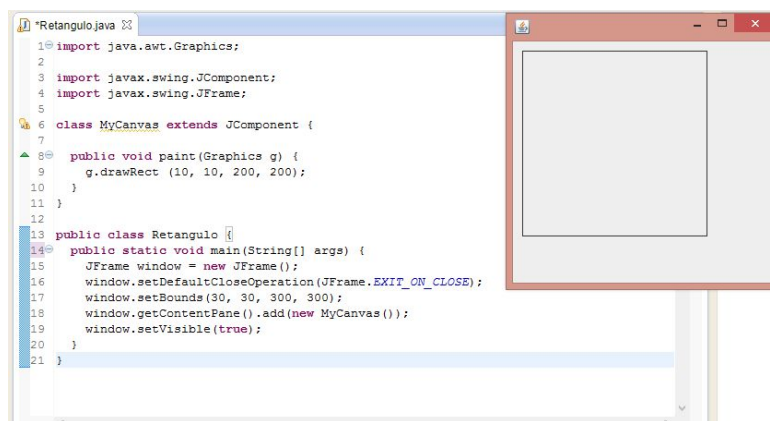


Figura 14 – Código em Java para desenhar um retângulo e imagem gerada pelo código
(Elaboração própria – SOFTWARE ECLIPSE, 2016.)



Figura 15 – Código em *Processing* para desenhar um retângulo e imagem gerada pelo código.
(Elaboração própria – SOFTWARE PROCESSING, 2016.)

Como é possível observar a figura 14 traz um programa escrito na linguagem Java para gerar um retângulo, já a figura 15 mostra a codificação para reproduzir o mesmo retângulo usando o software Processing, o *Processing* simplifica a escrita do código-fonte para a produção de imagens. A programação possui simplicidade para os programadores que desejam criar efeitos visuais a partir do código ou que ainda estão aprendendo a programar.

Segundo Maeda (2006), a simplicidade é uma qualidade que não apenas desperta a fidelidade apaixonada pelo design de um produto, mas também se tornou uma ferramenta estratégica-chave para empresas confrontarem suas próprias complexidades intrínsecas.

A manipulação (uso de computador) e a interação (com outros indivíduos) seriam as propulsoras dessa construção de significados. No contato com o

computador, ao inserir o software, o aprendiz “precisa contar com o fato de que esse trabalho conjunto com o aparelho em si carrega muito mais coisas que se manifestam com o acaso” (BYSTRINA, 1995, p. 24). Quais seriam essas “coisas”? Esse processo envolve informações que configuram, para Bystrina (1995), a base do processo da comunicação, sendo armazenadas interna (na mente humana, por exemplo) ou externamente (em um CD). Sobre essas informações, constroem-se textos (linguais e hiperlinguais).

O aluno sempre poderá aprender algo, sempre buscará novas informações e sempre encerrará a atividade com um outro conhecimento que não o inicial conhecimento real. Um conhecimento potencial completo, total, por sua vez, pode indicar que o indivíduo parou de adquirir informações, refletir sobre elas e construir seu conhecimento, em outras palavras, uma espiral interrompida. Isso é possível enquanto houver vida e capacidade de aprender?

A partir dessa relação entre as Tecnologias Emergentes e aprendizagem, abordo, no Capítulo 4, Desenvolvimento da pesquisa, e a ementa da matéria Elementos fundamentais da matemática e Introdução ao Pensamento computacional no Curso de Midialogia da Universidade de Campinas.

3. DESENVOLVIMENTO DA PESQUISA

Neste capítulo apresentaremos os objetivos e ementas das disciplinas Elementos Fundamentais da Matemática e Introdução ao Pensamento Computacional, conteúdo programático, metodologia, análise dos dados, o estudo da disciplina, o tempo, local e sujeitos de investigação e coleta de dados. Em 2015 no início do 1º semestre foi aplicado aos alunos dois questionários elaborados na plataforma do Google através do formato de formulário online, utilizando a ferramenta *Google Forms* (GOOGLE, 2015), se mostrou um recurso bastante pertinente para coletar dados a distância e de forma precisa e rápida, a fim de traçar um diagnóstico da turma, aplicamos aos 28 alunos matriculados no curso. E no 1º semestre de 2016, a fim de traçar um novo diagnóstico da turma, aplicamos aos 31 alunos matriculados no curso uma nova pesquisa para melhorar a análise sobre o processo de ensino e aprendizagem.

3.1 Objetivos e Ementa da Disciplina

Neste capítulo serão abordados o conteúdo programático abordados no curso de Midialogia nas matérias Elementos Fundamentais da Matemática e Introdução ao Pensamento Computacional.

3.1.1 Objetivos Gerais da Disciplina:

A matemática é a ciência da observação dos padrões da natureza e da cultura. Sua evolução acontece associada às formas e aos meios de comunicação e, conseqüentemente, ao desenvolvimento das linguagens estabelecidas por estes meios. O objetivo desta disciplina é observar, compreender e analisar os modelos e padrões de representação dos espaços geométricos e topológicos matemáticos nos vários momentos históricos de nossa cultura. De fato, pretende-se estudar os eixos de similaridades entre as representações matemáticas e as imagens geradas pelas tecnologias emergentes por meio do pensamento computacional e através de uma linguagem de programação. Na disciplina utilizou-se a linguagem de

programação gráfica de código aberto: o software *Processing* com o objetivo de implantarmos estratégias que levassem os alunos a desenvolverem o Pensamento Computacional.

3.1.2 Objetivos Específicos das Disciplinas

- Conhecer os conceitos e fundamentos matemáticos e os padrões de representação da natureza e da cultura;
- Conhecer os conceitos fundamentais dos algoritmos e das linguagens de programação;
- Adquirir capacidade de documentação e descrição de um programa de computador;
- Introduzir o *Processing* como ferramenta geral de manipulação de sistemas computacionais;
- Realizar produções computacionais utilizando o software de código aberto: *Processing*.

3.1.3 Conteúdo Programático

Temática 01: a primeira temática a ser tratada nas disciplinas foram os conceitos que envolvem as artes, matemática e as suas representações e modelos. A disciplina pretende apresentar as ciências, particularmente a matemática e as artes como forma de conhecimento humano que são pensadores por meio de modelos e imagens.

Temática 02: a segunda temática são as linguagens matemática e as artes. A disciplina pretende mostrar as relações entre as representações matemáticas e as artes por meio das similaridades entre estas duas linguagens que geram conhecimento. Ao ver a matemática através das imagens geradas pelos podemos verificar suas relações com as produções artísticas de cada momento histórico: período pré-industrial, industrial mecânico e industrial eletrônico e digital.

Terceira 03: programação e processamento de imagens. Nesta temática será desenvolvido os conceitos básicos, implementação de algoritmos e aplicação de métodos e modelos lógicos em sistemas computacionais para processamento de imagens e desenvolvimento do Pensamento Computacional.

Será utilizado o software *Processing* (Open Source) para desenvolvimento de produções geradas por linguagem de programação. Os alunos aprenderam técnicas de elaboração de desenho estáticos, generativo, animação e processamento de imagens com vídeo por meio do software *Processing*.

3.1.4 Metodologia

A disciplina foi ministrada em dois momentos: o primeiro é teórico e a partir de modelos conceituais matemáticos e suas imagens, apresentamos a matemática com uma linguagem de produção de conhecimento. Já no segundo instante, apresentamos uma linguagem de programação de código aberto: *Processing* para desenvolvimento de produtos computacional e, de forma prática, realizar produtos computacionais utilizando as Tecnologias Emergentes.

3.2 Análise dos dados no software CHIC

Os dados resultantes da análise-interpretativa foram associados às categorias, compilados em uma planilha do Excel e inseridos no software CHIC para gerar gráficos de similaridade. Atualmente, o software CHIC permite realizar um método estatístico multidimensional utilizado em estudos qualitativos de regras de associação – atender tipos distintos de variáveis, quantificar a significação dos valores qualitativos, constância da regra associada, de ordenações de classes de regra, a tipicidade e contribuição de sujeitos ou categorias de sujeitos à constituição de certas regras (VALENTE, 2015, p. 55).

O software CHIC representa, por meio de gráficos, caminho de regras e uma hierarquia de regras sobre regras, a partir de intervalos determinados. Assim, mediante essa ferramenta de estatística multidimensional, os resultados foram sintetizados e estruturados com o intuito de contribuir para a análise do processo de construção de letramentos e dos níveis de letramentos. As ocorrências e não ocorrências de determinadas categorias foram fixadas em variáveis e tratadas pelo software para gerar regras de associação, que podem ser visualizadas em gráficos denominados como árvore de similaridades.

Para gerar as árvores de similaridades foi necessário criar planilhas no Excel. As variáveis, em formato de códigos, foram inseridas na linha horizontal superior da planilha. Por exemplo, CP, CT, CC, CT, SemCo, Co, Desco, Reco, como vimos anteriormente. Na primeira coluna, foram inseridos os códigos representando os aprendizes, por exemplo, A1, A2, A3, Na planilha foram lançados valores 0 ou 1, conforme a ausência ou a presença da variável dos atores em relação a resposta sobre as questões abordadas.

A Tabela 04 é um exemplo de planilha elaborada no Excel. Na linha horizontal superior há um grupo de variáveis CompCa (Computador em casa), gostMat (Gosta de matemática), difMat (Dificuldade matemática), cursoInfo(Curso informatica), possSmar (possui Smarthphone), ConLing (Conhecimento em Linguagem de programação), CriAPP (criação aplicativo), usoProc (uso Processing), usoPComp (uso próprio computador). Na primeira coluna, a sequência de aprendizes de 1 a 22. O valor 0 é referente à negação (não) da variável, e o valor 1 demonstra afirmação (sim) da variável.

TABELA 04 – Compilação de dados. Exemplo de uma planilha

	compCa	gostMat	difMat	cursoInfo	possSmar	conhLing	criAPP	usoProc	usoPcomp
A1	1	0	0	1	1	1	1	1	1
A2	1	1	1	0	1	0	0	1	1
A3	1	0	1	0	1	0	0	1	1
A4	1	0	1	0	1	1	0	1	1
A5	1	1	0	0	1	1	0	1	1
A6	1	1	0	1	1	1	1	1	1
A7	1	0	0	0	1	1	0	1	1
A8	1	1	0	0	1	0	0	1	1
A9	1	1	0	0	1	0	0	1	0
A10	1	1	0	1	1	1	1	1	1
A11	1	0	1	0	1	0	0	1	0
A12	1	1	0	0	1	0	0	1	1

A13	1	1	0	0	1	1	1	1	0
A14	1	0	0	0	1	0	1	1	1
A15	1	0	1	0	1	1	0	1	1
A16	1	1	0	0	1	1	0	1	1
A17	1	0	1	0	1	1	1	1	1
A18	1	1	0	0	1	0	0	1	1
A19	1	1	0	1	1	1	1	1	1
A20	1	1	0	1	1	1	0	1	0
A21	1	0	1	0	1	0	0	1	0
A22	1	1	0	1	1	1	1	1	0

As planilhas foram salvas em arquivos “CSV (separado por vírgulas) ”.

Após isso, elas foram abertas e processadas pelo software CHIC e, assim, geramos as árvores de similaridades. A tela que apresenta o número de ocorrências, a média, o desvio padrão, o coeficiente de correlação, os índices de similaridades, os nós significativos e as telas para visualização das informações sobre o arquivo: frequência de pares variáveis, o coeficiente de correlação, índice e valor. O exemplo de árvore de similaridades, figura 13, foi gerado a partir da tabela 03.

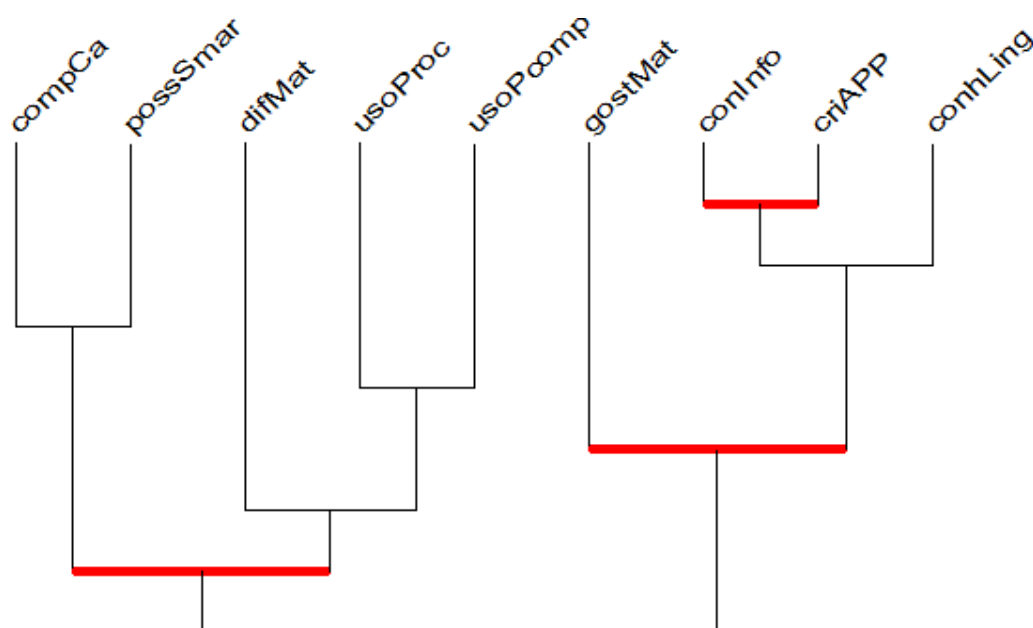


Figura 16 – Exemplo de árvore de similaridades do software CHIC.

Para a análise de similaridades foram identificadas as intersecções que correspondem às convergências entre as variáveis. Quando o gráfico é apresentado de forma horizontal, as similaridades mais fortes são representadas por linhas horizontais próximas às variáveis, conforme a convergência entre compCa, possSmar, difMat seguida conlInfo, criaApp que cria uma nova relação entre gostMat, conlInfo, criaApp e sem une a conhLing. Quando aparece a linha vermelha, o software CHIC classifica como um nó significativo entre ocorrências. No caso, há uma forte relação entre as variáveis compCa, possSmar e difMat. Em seguida, observamos que as variáveis conlInfo, criaApp, estão relação de forma mais forte, interligando com a gostMat, conlInfo, criaApp, conhLing. As três variáveis compCa, possSmar e difMat estão conectadas por um nó significativo com as variáveis conlInfo, criaApp, seguidas pelas gostMat, conlInfo, criaApp que possuem outro nó significativo.

A utilização do CHIC favoreceu a visualização de variáveis com ocorrências similares. A partir da identificação de similaridades, foi possível estabelecer categorias, ou seja, agrupar unidades de variáveis de acordo com o processo de construção do processo de ensino e aprendizagem.

3.3 O estudo da Disciplina Elementos Fundamentais de Matemática e Introdução ao Pensamento Computacional

Esta subsecção apresenta o início de intermináveis capítulos que poderão ser escritos a respeito da formação proposta, apoiada sobre três pontos, quais sejam: 1) observação espontânea de material didático visando à sensibilização dos alunos quanto à produção do projeto final, com análise dos relatórios e dos trabalhos finais; 2) Elaboração de questionários avaliativos do conteúdo apresentado aos alunos; e 3) Análise dos projetos e das respostas enviadas aos alunos (questionários elaborados em formato google docs e relatórios finais), materiais e projetos elaborados pelos estudantes envolvidos na pesquisa.

Apresento aqui o processo e os resultados dessa formação, iniciando pelo material didático eleito para servir de base à aprendizagem dos alunos do componente curricular Introdução ao Pensamento computacional no curso de Midialogia, habilitação em Multimeios, da Universidade de Campinas (UNICAMP), no primeiro semestre de 2016.

3.3.1 O tempo da investigação

O estudo de campo, ocorreu no 1º Semestre de 2015 e 1º Semestre de 2016 com encontros semanais, as segundas-feiras na aula de Elementos Fundamentais de Matemática, ministrada pelos professores Dr. Hermes Renato Hildebrand e Dr. José Armando Valente com duração de quatro horas. No 1º semestre de 2016 a disciplina passou a se chamar “Introdução ao Pensamento Computacional”, no entanto, tratou, basicamente do mesmo conteúdo programático com pequenas alterações apenas. As observações de campo ocorriam nas segundas-feiras sempre no período das 14h00 às 18h00 na UNICAMP. Eram observados os exercícios que eram organizados pelo professor na abordagem de assuntos referente à matéria.

No decorrer do semestre foram elaborados oito exercícios propostos aos alunos e, no final do semestre, os alunos confeccionaram projetos finais utilizando todo o conhecimento adquirido durante o semestre. Durante este intervalo de tempo foram realizados questionários sobre o conteúdo e sobre as principais questões observadas durante o semestre. Estes questionários foram

respondidos pelos alunos e no final do semestre os dados foram tabulados e analisados.

3.3.2 O Local

As atividades ocorreram nos laboratórios de informática da Universidade Estadual de Campinas, na cidade de Campinas - SP. O laboratório de informática tem acesso à internet, onde os alunos faziam uso dos computadores do laboratório e outros faziam o uso de seu próprio notebook durante as aulas.

3.3.3 Sujeitos da Pesquisa

Da primeira etapa da pesquisa participaram 22 sujeitos adultos. Eram 14 mulheres e 6 homens. Na segunda etapa participaram 30 sujeitos adultos: 20 mulheres e 10 homens. Todos eram alunos ingressantes no 1º ano do Curso de Midialogia.

3.3.4 Coleta de dados

Dois questionários foram elaborados na plataforma do Google através do formato de formulário online, utilizando a ferramenta *Google Forms* (GOOGLE, 2015), que se mostrou um recurso bastante pertinente para coletar dados a distância e de forma precisa e rápida.

Questionário 1

Inicialmente foi criado o Questionário 1 (Apêndice 1) com perguntas elaboradas no sentido de identificar os atores e seu perfil de dificuldade com relação a matéria matemática e seu perfil tecnológico, conforme mostra a figura a seguir. Para Gil (1991), a elaboração de questionários, sequer algumas regras a serem seguidas. O ponto inicial é que as questões devem ser as mais claras possíveis e de fácil interpretação, devem permitir uma interpretação única, e devem tratar de uma ideia por vez. Esses foram os princípios adotados para a elaboração e reelaboração das perguntas. Também devemos levar em consideração o nível de informação do usuário.

Questionário 2

Num segundo momento foi criado o Questionário 2 (Apêndice 2) com perguntas elaboradas no sentido de identificar o conhecimento adquirido pelos atores com a resolução dos exercícios propostos em sala de aula.

Questionário 2 – Exercício 01

No primeiro exercício do Questionário 2 utilizamos o exemplo da Figura 01 (Figura 17 a seguir) que tratou de conceitos básicos. Algo fundamental para qualquer indivíduo que manipula uma linguagem de programação e deve estar apto a utilizar os procedimentos para programar: deve conhecer a **sintaxe básica, definição de uma variável e criar um método para resolver os exercícios.**

As sintaxes utilizadas neste primeiro procedimento forma: criação de comentários (insere `//` - barras invertida na horizontal) ou `/* comentário */` (entre barras invertida na horizontal com dois asteriscos) quando se quer criar um comentário em bloco; devemos declarar as variáveis a serem utilizadas na programação; algumas funções básicas que são utilizadas como `size()`, `point()`, `noStroke()` e as funções de movimento das figuras como `translate()`, `rotate()`, `pushMatrix()` e `popMatrix`.

Foi disponibilizado o código original do exercício 01 aos alunos e solicitado que alterassem algumas informações disponíveis no código original, e que as alterações fossem visualizadas no resultado. Todas estas alterações foram efetuadas pelos atores durante a aula do professor, posteriormente foi disponibilizado através do Google Forms uma pergunta direcionada as principais funções utilizadas na construção do aplicativo que usa conceitos de transparência e movimentação de uma figura.

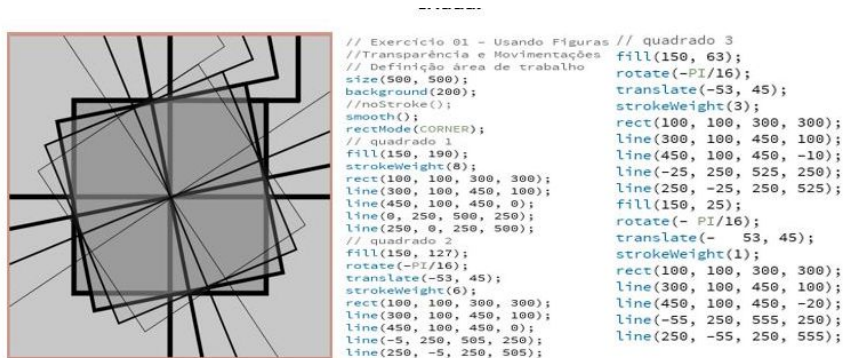


Figura 17 - figura com transparência e movimento com codificação;

Após a realização da codificação no exercício 01 perguntou-se: Explique o que você conseguiu entender do bloco a seguir:

```
size(500,500);
background(200);
smooth();
rectMode(CORNER);
fill(150,190);
strokeWeight(8);
rect(100,100,300,300);
line(300,100,450,100);
line(450,100,450,0);
line(0, 250,500, 250);
line (250,0,250,500);
```

Segundo Morin (2000), ao se buscar construir referências ao contexto global e complexo, devemos identificar o que conhecedor sabe do mundo. Como dizia François Recanati, “a compreensão dos enunciados, longe de se reduzir a uma mera decodificação, é um processo não modular de interpretação que mobiliza a inteligência geral e faz amplo apelo ao conhecimento do mundo” (MORIN, 2000, p. 38). Dessa maneira, há uma correlação entre a mobilização dos conhecimentos em conjunto e uma ativação da inteligência geral.

CAPÍTULO 4 - Análise dos Dados

Neste capítulo apresentaremos a análises dos dados extraídos a partir dos questionários respondidos e documentos preenchidos pelos alunos que participaram desta pesquisa, bem como o levantamento de dados do estudo de campo. No início do 1º semestre de 2015, a fim de traçar um diagnóstico da turma, aplicamos aos 28 alunos matriculados no curso dois questionários (Apêndices A e B). Buscamos verificar preferências, habilidades e conhecimentos que os alunos tinham com relação aos itens pesquisados e qual era o contato deles com as áreas do conhecimento da disciplina: matemática, lógica e linguagem de programação. Destaco que nem todos os alunos responderam o questionário (24 alunos).

4.1 Análise do Público Alvo

No início do 1º semestre de 2015, a fim de traçar um diagnóstico da turma, aplicamos aos 28 alunos matriculados no curso dois questionários (Apêndices A e B). Conforme segue:

Questionados sobre se gosta de matemática, 12 (50%) responderam que gostam de matemática. Dos 12 restantes (50%), responderam que não gostam de matemática.



Figura 18. Sobre o gostar da matéria matemática.

Questionados sobre se possui dificuldade com a matemática, 9 (37,5%) responderam que possuem dificuldade com a matemática. Dos 15 restantes (62,5%), responderam que não possuem dificuldade com a matemática.



Figura 19. Sobre dificuldade com matemática

Questionados sobre se já fizeram algum curso de informática, 5 (21%) responderam que já fizeram curso de informática. Dos 19 restantes (79%), responderam que não fizeram curso de informática.

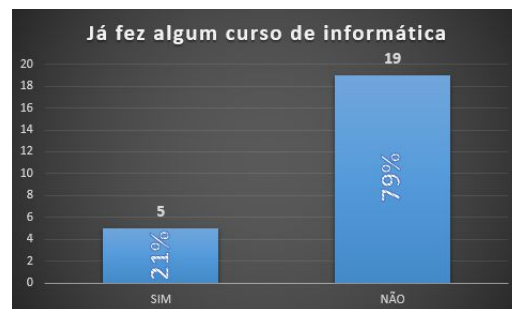


Figura 20. Sobre já fez algum curso de informática

Questionados sobre conhece alguma linguagem de programação, 13 (54%) responderam que conhece alguma linguagem de programação. Dos 11 restantes (46%), responderam que não conhece alguma linguagem de programação.



Figura 21. Sobre conhece alguma linguagem de programação.

Questionados sobre já criou algum aplicativo, 7 (29%) responderam que já criaram algum aplicativo. Dos 17 restantes (71%), responderam que não criaram nenhum aplicativo.



Figura 22. Já criou algum aplicativo

Questionados sobre se usaram o próprio computador para elaborar os exercícios, 17 (71%) responderam que usaram o próprio computador para elaborar os exercícios. Dos 7 restantes (29%), responderam que não usaram o próprio computador para elaborar os exercícios.

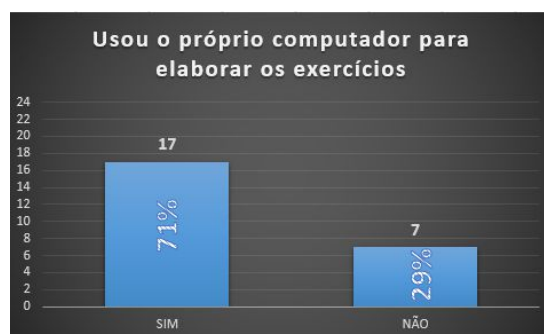


Figura 23. Sobre usou o próprio computador para elaborar os exercícios

Como as questões abordadas no 1º semestre de 2015 faziam parte de um levantamento piloto e dado continuidade ao levantamento de dados sobre como o uso das TIC (Tecnologia da Informação e Comunicação) podem se fundir no processo ensino e de aprendizagem no 1º semestre de 2016. Elaborei novas perguntas, a fim de traçar um diagnóstico mais detalhado da nova da turma, apliquei aos 31 alunos matriculados no componente curricular dois questionários constantes no Apêndice 2. Busquei averiguar perfil social, preferências, habilidades e conhecimentos que os alunos tinham com relação aos itens questionados, e qual seu contato com as áreas do conhecimento sobre as matérias matemática, lógica e linguagem de programação.

No início do 1º semestre de 2016, a fim de traçar um diagnóstico da turma, aplicamos aos 31 alunos matriculados no curso dois questionários (Apêndice B). Conforme segue:

Questionados sobre que tipo de instituição concluiu o ensino médio, 15 (48%) responderam que concluíram o ensino médio na rede particular. Dos 16 restantes (52%), responderam que concluíram o ensino médio na rede pública.

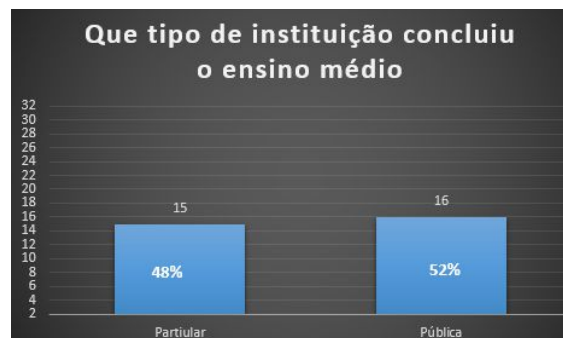


Figura 24. Que tipo de instituição concluiu o ensino médio

Questionados sobre se possui computador em casa, 29 (94%) responderam que possuem computador em casa. Dos 02 restantes (6%), responderam que não possuem computador em casa.



Figura 25. Possui computador em casa

Questionados sobre se Gosta de matemática, 21 (68%) responderam que Gostam de matemática. Dos 10 restantes (32%), responderam que não Gostam de matemática.



Figura 26. Gostam de matemática

Questionados sobre se apresentam dificuldade com a matéria matemática, 21 (68%) responderam que Apresentam dificuldade com a matéria matemática. Dos 10 restantes (32%), responderam que não apresentam dificuldade com a matéria matemática.

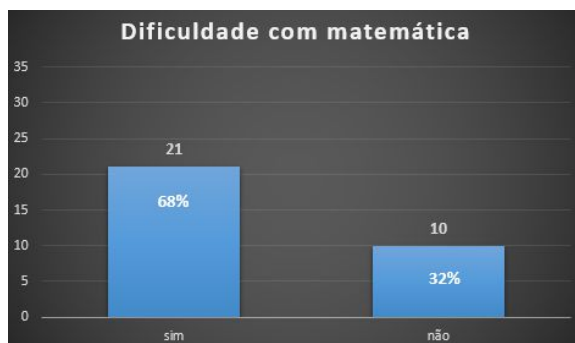


Figura 27. Dificuldade com a matéria matemática

Questionados sobre se já fez algum curso de informática, 11 (35%) responderam que sim, já fizeram algum curso de informática. Dos 20 restantes (65%), responderam que não fizeram nenhum curso de informática.



Figura 28. Fizeram algum curso de informática

Questionados sobre se já teve a matéria lógica, 16 (52%) responderam que sim, já tiveram a matéria lógica. Dos 15 restantes (48%), responderam que não já tiveram a matéria lógica.



Figura 29. Já teve a matéria lógica

Questionados sobre se conhece alguma linguagem de programação, 7 (23%) responderam que sim, conhecem alguma linguagem de programação. Dos 24 restantes (77%), responderam que não conhecem nenhuma linguagem de programação.

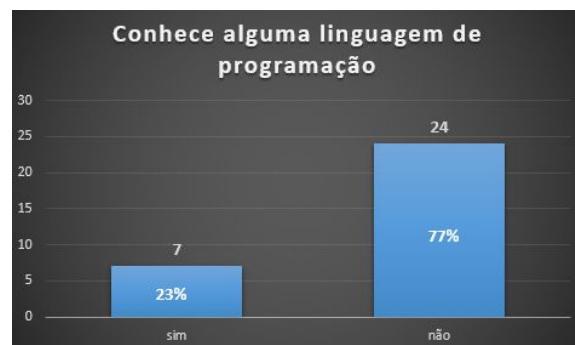


Figura 30. Conhece alguma linguagem de programação

Questionados sobre se sabe a diferença entre as duas formas de produção para Web, 2 (6%) responderam que sim, sabem qual a diferença entre as duas formas de produção para Web. Dos 29 restantes (94%), responderam que não sabem a diferença entre as duas formas de produção para Web.

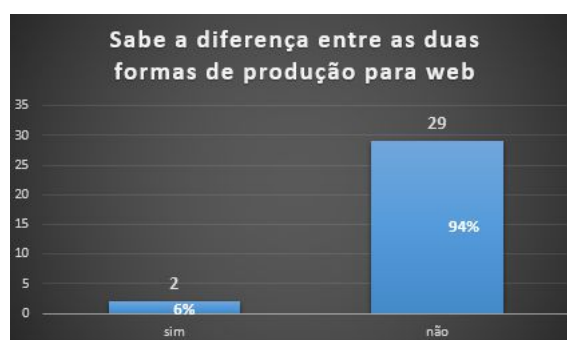


Figura 31. Conhece alguma linguagem de programação

Questionados sobre se já criou algum programa de computador, 4 (13%) responderam que sim, já criaram algum programa de computador. Dos 27 restantes (87%), responderam que não criaram nenhum programa de computador.

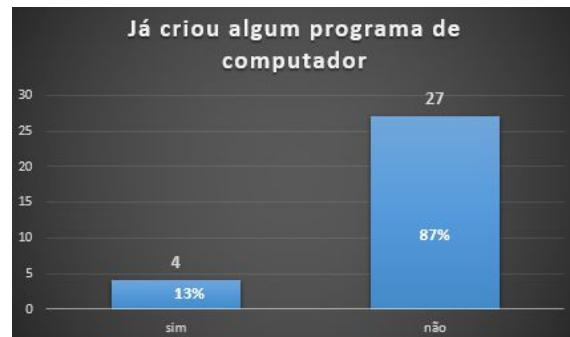


Figura 32. Conhece alguma linguagem de programação

Questionados sobre se já criou algum aplicativo, 2 (6%) responderam que sim, já criaram algum aplicativo. Dos 29 restantes (94%), responderam que não criaram nenhum aplicativo.



Figura 33. Já criou algum aplicativo

Questionados sobre se conhece o *Processing*, 7 (23%) responderam que sim, conhecem o *Processing*. Dos 24 restantes (77%), responderam que não conhecem o *Processing*.

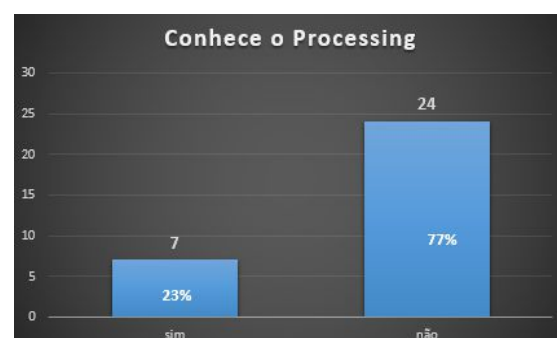


Figura 34. Conhece o *Processing*

Questionados sobre se conhece alguma linguagem de código aberto, 6 (19%) responderam que sim, conhecem alguma linguagem de código aberto. Dos 25 restantes (81%), responderam que não conhecem nenhuma linguagem de código aberto.

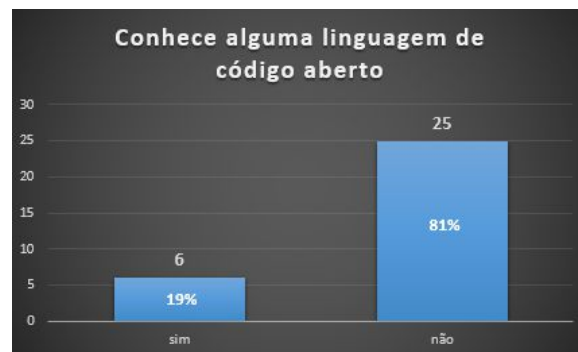


Figura 35. Conhecem alguma linguagem de código aberto

Questionados sobre se quando encontra alguma dificuldade (em relação a matéria matemática) o que faz, 1 (3%) respondeu que desiste do problema, 21 (68%) responderam que tentam resolver o problema, 7 (23%) responderam que pedem ajuda aos amigos. Dos 2 restantes (6%), responderam que procuram ajuda nas redes sociais.

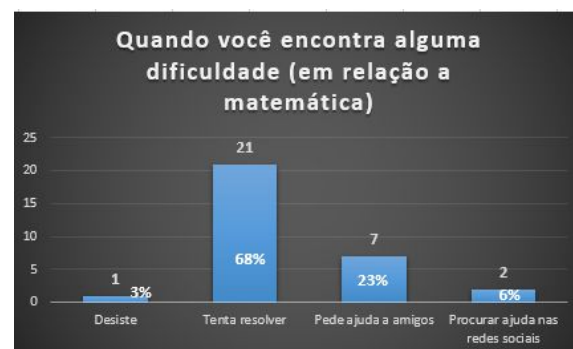


Figura 36. Quando você encontra alguma dificuldade (em relação a matemática), o que faz?

Questionados sobre se quando tenta resolver um problema e procura ajuda nas redes sociais, quais as redes sociais usa. 14 (45%) responderam que usam as redes sociais: Facebook, Youtube, Whatsapp, Instagram e Twitter, 12 (39%) responderam que usam as redes sociais: Youtube, Twitter, Whatsap e outras, 4 (13%) responderam que usam outras redes sociais. E 1 (3%) restante, respondeu que procura as redes sociais Google+ e outras.



Figura 37. Redes sociais mais usadas

O questionário utilizado para diagnosticar o conhecimento dos alunos quanto às Tecnologias Emergentes em especial quanto a utilização do software *Processing*, respondido no primeiro levantamento realizado por 24 alunos, permitiu verificar, quantitativamente, a percepção quanto à aprendizagem dos alunos que não possuíam habilidade em programação, em específico para a criação de um cenário ao utilizar o software *Processing*. O diagnóstico mostrou que, por meio das Tecnologias Emergente, é possível criar atividades que mesmo para aqueles que não possuem habilidade em programação, há possibilidade de letramento quanto ao uso do software.

Para Valente (1998), a atividade de programar é subjacente às ferramentas propostas e seu uso pelos *não programadores*, é um processo cíclico. Meio pelo qual o indivíduo constrói sua base de dados (programa), a partir de hipóteses prévias (sobre o domínio do problema, sobre o ambiente, sobre a linguagem) e a questiona (interroga). A partir da execução da base de dados pelo computador, há um feedback que leva os *não programadores* a uma reflexão e reformulação de seu conjunto de hipóteses, continuando o ciclo de programação.

A crença de que o uso do software *Processing* como Tecnologia Emergente auxiliou a aprendizagem quando da criação dos exercícios propostos, foi pontada por 24 alunos (100%) como a ferramenta que os auxiliou na resolução dos exercícios propostos. O fato dos alunos assistirem as aulas no laboratório onde dispunham de computadores para a criação das atividades e em alguns casos os alunos faziam uso do seu laptop durante as aulas ajudou no letramento com o software, pois houve a facilitação do entendimento, houve contato entre as pessoas, grande quantidade de informações, praticidade,

ruptura da barreira espacial, correlação da quantidade de recursos e aprendizagem, atendimento a variedade de interesses e necessidades individuais, envolvimento de outros sentidos na apreensão de informações, autonomia e criação de didática própria, qualidade, estímulo, linguagem de fácil entendimento, otimização do processo educativo, contato com o tema da aula na prática, melhora do processo de aprendizagem, economia de tempo, conhecimento mais acessível, interação e criatividade.

4.2 Análises de similaridade e implicativa

Como exposto nos Capítulos 1 e 3, partimos para a análise dos resultados obtidos com aplicação do software Chic e as categorias descritas na subseção a seguir. Optei em analisar as repostas abertas feita aos alunos da turma do 1º semestre de 2016 do Curso de Midialogia, visando detectar o conhecimento do aluno com relação a utilização do software Processing, as principais dificuldades encontradas para a criação dos exercícios: 1) cenário, 2) Mandala, 3) Criação do projeto Ping Pong, e 4) Projeto final. A diferenças entre as similaridades e as implicações entre as categorias selecionadas, buscando uma avaliação mais completa da aprendizagem dos alunos envolvidos na pesquisa.

4.2.1 Grupos e categorias

A divisão dos grupos de categorias, refere-se às questões respondidas pelos alunos e a quantidade se justifica pela diversidade das respostas. As categorias descritas a seguir foram definidas a partir do olhar teórico-prático, apresentado nesta dissertação, sobre as respostas dos alunos aos questionários aplicados constantes da seção Apêndices.

4.2.1.1 Grupo conhecimento em linguagem de programação (CLP)

Este grupo foi estabelecido a partir da manifestação dos alunos quanto a seu conhecimento prévio sobre linguagem de programação. Para tanto, defini as duas categorias a seguir.

CLP1 – Ignora (o aluno não conhece nenhuma linguagem de programação);

CLP2 – Conhece (o aluno teve contato ou manipulou alguma linguagem de programação).

4.2.1.2 Alfabetização em linguagem de programação

Este grupo baseia-se nos aspectos que os alunos estão alfabetizados quanto a utilização das linguagens de programação C++, Java, VB, HTML e outras. Os alunos mostraram possuir alfabetização em algumas tecnologias antes de iniciar o trabalho com o software *Processing*, indicando uma opinião preestabelecida, tanto dos alunos que possuíam algum conhecimento prévio, quanto dos que ignoravam o software *Processing*. O grupo abrange as categorias abaixo.

ALPC – C++ (o aluno teve contato ou manipulou a ling. de programação C++);

ALPJ – JAVA (o aluno teve contato ou manipulou a ling. de programação JAVA);

ALPVB – VB (o aluno teve contato ou manipulou a ling. de programação VB);

ALHTML – HTML¹⁴ (o aluno teve contato ou manipulou a linguagem HTML);

ALPO – OUTRAS (o aluno teve contato ou manipulou outras ling. de programação).

4.2.1.3. Alfabetização em HTML (AHTM)

Este grupo baseia-se nos aspectos que os alunos estão alfabetizados quanto a linguagem de marcação de texto HTML. Perguntamos aos alunos sobre o conhecimento em relação a HTML, onde os alunos responderam sim e não. O grupo abrange as categorias abaixo.

AHTML1 – Ignora (o aluno não conhece a linguagem HTML);

AHTML2 – Conhece (o aluno teve contato ou manipulou a linguagem HTML).

4.2.1.4 Já criou programa para computador (JCPC)

Este grupo baseia-se nos aspectos que o aluno já criou algum programa para computador e está alfabetizado quanto a linguagem de alguma linguagem

¹⁴ HTML: *HyperText Markup Language*, que significa *Linguagem de Marcação de Hipertexto*) é uma linguagem de marcação utilizada na construção de páginas na Web. Documentos HTML podem ser interpretados por navegadores. Apesar do HTML não ser uma linguagem de programação este foi mencionado pelos alunos.

de programação. Foi perguntado aos alunos sobre se já havia criado algum programa para computador, onde os alunos responderam sim e não. O grupo abrange as categorias abaixo.

JCPC1 – Ignora (o aluno não criou nenhum programa de computador);

JCPC2– Conhece (o aluno já criou algum programa de computador).

4.2.1.5 Já criou algum aplicativo (JCAPP)

Este grupo baseia-se nos aspectos que o aluno já criou algum aplicativo e está alfabetizado quanto a linguagem de alguma linguagem de programação. Foi perguntado aos alunos sobre se já havia criado algum app, no qual os alunos responderam sim e não. O grupo abrange as categorias abaixo.

JCAPP1 – Ignora (o aluno não criou nenhum APP de computador);

JCAPP2– Conhece (o aluno já criou algum APP).

4.2.1.6 Conhece o software *Processing* (CSP)

Este grupo baseia-se nos aspectos que o aluno conhece o software *Processing*. Foi perguntado aos alunos sobre se já conhecia o software *Processing*, onde os alunos responderam sim e não. O grupo abrange as categorias abaixo.

CSP1 – Ignora (o aluno não conhece o software *Processing*);

CSP2 – Conhece (o aluno conhece o software *Processing*).

4.2.2 Percepção sobre as matérias matemática e lógica

Este grupo baseia-se nos aspectos que os alunos atribuíram seu conhecimento sobre a diferença entre as matérias matemática e lógica antes de iniciar o primeiro contato com a matéria introdução ao pensamento computacional, indicando uma opinião preestabelecida, tanto dos alunos que possuíam algum conhecimento prévio, quanto dos que ignoravam a diferença entre as matérias. O grupo abrange as categorias abaixo.

MML1 - Matemática e lógica (o aluno considera que as matérias de matemática e lógica dependem uma da outra).

MML2 - Matemática e lógica como elementos diferentes (o aluno considera que as matérias de matemática e lógica possuem funções distintas).

MML3 - Não sabe informar (o aluno não conseguiu expressar sua opinião sobre as diferenças entre as matérias matemática e lógica).

4.2.3 Percepção sobre o pensamento computacional

Este grupo baseia-se nos aspectos que os alunos atribuíram sua opinião sobre o que você entende por pensamento computacional antes de iniciar o primeiro contato com a matéria introdução ao pensamento computacional, indicando uma opinião preestabelecida, tanto dos alunos que possuíam algum conhecimento prévio, quanto dos que ignoravam o assunto pensamento computacional. O grupo abrange as categorias abaixo.

PC1 - O que é o pensamento computacional (o aluno considera que o pensamento computacional está fortemente ligado ao uso de um computador ou aparelho eletrônico).

PC2 - Pensamento voltado a elementos de programação (o aluno considera que o pensamento computacional está presente na linguagem de programação e elementos que fazem pensar em lógica).

4.2.4 Percepção sobre: O que faz quando encontra dificuldade com a matéria matemática (DMM)

Este grupo baseia-se nos aspectos que os alunos atribuíram sua resposta sobre o que fazem quando encontram dificuldade em relação a matéria matemática, indicando uma opinião preestabelecida, tanto dos alunos que possuíam algum conhecimento prévio, quanto dos que ignoravam o assunto matemática. O grupo abrange as categorias abaixo.

DMM1 - Desiste (o aluno desiste de resolver os problemas relacionados com a matéria matemática).

DMM2 - Procura a ajuda de amigos (o aluno procura a ajuda de amigos para resolver os problemas relacionados com a matéria matemática).

DMM3 - Procura a ajuda de em redes sociais (o aluno procura a ajuda em redes sociais para resolver os problemas relacionados com a matéria matemática).

DMM4 - Tenta resolver o problema (o aluno tenta resolver os problemas relacionados com a matéria matemática).

4.2.5 Percepção durante o processo de aprendizagem durante a criação do projeto final

Após terem sido “apresentados” ao *Processing* e orientados sobre os comandos básicos do software, bem como terem dado início à manipulação dos exemplos disponíveis no site do Processing.org e após o termino da criação do projeto final os alunos tiveram que elaborar um relatório onde tiveram que refletir sobre sua aprendizagem, informando ainda quais os foram pontos positivos e o negativo encontrados durante a elaboração do projeto. Conforme relatório constante do Apêndice C. O relatório foi enviado por e-mail aos 31 alunos matriculados no curso, onde 27 alunos responderam. De qualquer maneira, para uma análise qualitativa e comparativa do processo de aprendizagem, entre o início do curso e o período em que foi analisado o relatório gerou as categorias abaixo:

4.2.5.1 Percepção sobre o uso do *Processing* (UP)

Este grupo baseia-se nos aspectos que os alunos atribuíram às tecnologias enates de iniciar o trabalho com a criação de um mini aplicação utilizando o software *Processing*, indicando a opinião preestabelecida, tanto dos alunos que possuíam algum conhecimento prévio em programação, quanto dos que ignoravam programação. O grupo abrange as categorias abaixo.

UP1 - Conseguiu aprender sozinho (o aluno considera que conseguiu aprender sozinho durante criação do projeto final).

UP2 - Aperfeiçoou seu conhecimento (o aluno considera que conseguiu aperfeiçoar seu durante criação do projeto final).

UP3 - Encontrou diversão durante a aprendizagem (o aluno considera que encontrou diversão durante criação do projeto final).

UP4 - Conseguiu atingir os objetivos (o aluno considera que conseguiu atender os objetivos durante criação do projeto final).

UP5 - Utilizou projetos existentes para criar o seu projeto (o aluno considera que utilizou a codificação de projetos existentes no site do Processing.org para a criação do projeto final).

UP6 - Ficou satisfeito com o resultado obtido (o aluno considera que conseguiu alcançar os objetivos e está satisfeito com a criação do projeto final).

UP7 - Voltou a programar (o aluno considera que conseguiu voltar a programar com a criação do projeto final).

UP8 - Aprendeu a utilizar o *Processing* (o aluno considera que aprendeu a utilizar o *Processing* com a criação do projeto final).

UP9 - Se familiarizou com os códigos do *Processing* (o aluno considera que se familiarizou com o *Processing* para a criação do projeto final).

UP10 - Aprimorou seus conhecimentos (o aluno considera que aprimorou seus conhecimentos com a criação do projeto final).

UP11 - Modificação (o aluno considera que as modificações feitas nos projetos existentes no site do Processing.org foram muito importante para a criação do projeto final).

UP12 - Edição de fotografia (o aluno considera que conseguiu aprender e realizar inteiramente por conta própria a edição da fotografia, o que lhe permitiu conquistar uma autonomia com o uso do *Processing* com a criação do projeto final).

UP13 - Nova função (o aluno considera que adquiriu uma nova função ao programar com o *Processing* para a criação do projeto final).

UP14 - Pouco conhecimento (o aluno considera que apesar do pouco conhecimento em programação cumpriu a proposta do trabalho a medida em que desenvolvia o projeto final).

UP15 - Muito interessante (o aluno considera que foi muito interessante estudar o processo de formação de um jogo e que ampliou seus conhecimentos durante a elaboração do projeto final).

UP16 - Grande aprendizado (o aluno considera que houve um grande aprendizado em feitos que antes eram realizados sem pensamento lógico e matemático que foi trabalhado durante a elaboração do projeto final).

UP17 - Compreendeu elementos matemáticos (o aluno considera que compreendeu alguns elementos matemáticos mais profundo com a alteração do código base durante a elaboração do projeto final).

UP18 - Adquiriu conhecimento em programação (o aluno considera que adquiriu conhecimento em programação durante a elaboração do projeto final).

UP19 - Aperfeiçoamento pessoal (o aluno considera que houve aperfeiçoamento pessoal durante a elaboração do projeto final).

4.5.3 Análise de similaridade

A partir da categorização, detalhada na subseção anterior, elaborei uma tabela em Excel, a partir das respostas obtidas dos alunos. Em seguida, carreguei o arquivo no Chic. O emprego do Chic gerou uma árvore de similaridade nitidamente dividida em 14 grupos envolvendo 43 categorias, indicados pelas chaves em vermelho na Figura 38.

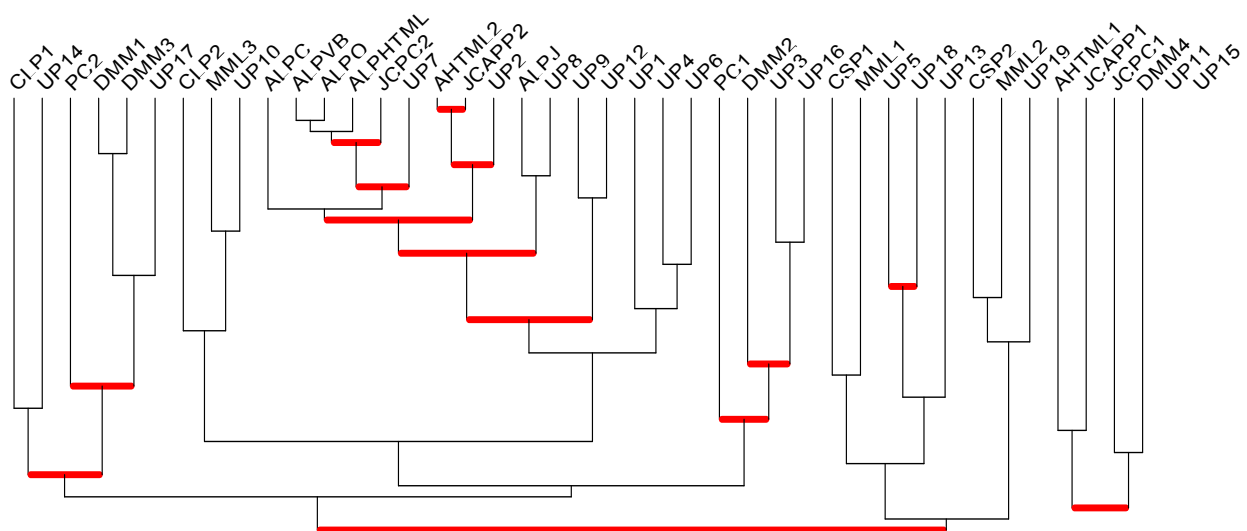


Figura 38 – Árvore de similaridade gerada pelo CHIC

A similaridade obtida a partir das respostas foi igual a 1, ou seja, total, em diversas relações entre as categorias, em 2 níveis, como indicado no Quadro 1. Outras relações apresentaram similaridade correspondente ao intervalo $< 1 \text{ e } \geq 0.00129985$, aproximadamente, como explicitado a seguir, que também se mostram significativas para esta análise (Quadro 3).

Quadro 2 – Classificação dos níveis com similaridade 1

Classificação ao nível: 1:	(AHTML2 JCAPP2) similaridade: 1
Classificação ao nível: 2:	(ALPVB ALPO) similaridade: 1

Quadro 3 – Classificação dos níveis com similaridade $< 1 \text{ e } \geq 0.00129985$

Classificação ao nível 3: ((ALPVB ALPO) ALPHTML) similaridade: 0.999919
Classificação ao nível 4: (((ALPVB ALPO) ALPHTML) JCPC2) similaridade: 0.999878
Classificação ao nível 5: (DMM1 DMM3) similaridade: 0.999833
Classificação ao nível 6: ((AHTML2 JCAPP2) UP2) similaridade: 0.999666
Classificação ao nível 7: (((ALPVB ALPO) ALPHTML) JCPC2) UP7) similaridade: 0.999331
Classificação ao nível 8: (UP9 UP12) similaridade: 0.998933
Classificação ao nível 9: (ALPC (((ALPVB ALPO) ALPHTML) JCPC2) UP7)) similaridade: 0.998701
Classificação ao nível 10: ((ALPC (((ALPVB ALPO) ALPHTML) JCPC2) UP7)) ALPJ) similaridade: 0.997379
Classificação ao nível 11: (((ALPC (((ALPVB ALPO) ALPHTML) JCPC2) UP7)) ALPJ) UP8) similaridade: 0.996943
Classificação ao nível 12: (((ALPC (((ALPVB ALPO) ALPHTML) JCPC2) UP7)) ALPJ) UP8) ((AHTML2 JCAPP2) UP2)) similaridade: 0.995994
Classificação ao nível 13: (MML3 UP10) similaridade: 0.993256
Classificação ao nível 14: (UP3 UP16) similaridade: 0.993256
Classificação ao nível 15: (UP4 UP6) similaridade: 0.98856
Classificação ao nível 16: (UP5 UP18) similaridade: 0.985318
Classificação ao nível 17: (CSP2 UP19) similaridade: 0.974946
Classificação ao nível 18: (((ALPC (((ALPVB ALPO) ALPHTML) JCPC2) UP7)) ALPJ) UP8) ((AHTML2 JCAPP2) UP2)) (UP4 UP6)) similaridade: 0.958663
Classificação ao nível 19: (PC2 UP17) similaridade: 0.951527
Classificação ao nível 20: (((ALPC (((ALPVB ALPO) ALPHTML) JCPC2) UP7)) ALPJ) UP8) ((AHTML2 JCAPP2) UP2)) (UP4 UP6)) (UP9 UP12)) similaridade: 0.951333
Classificação ao nível 21: ((DMM1 DMM3) DMM2) similaridade: 0.927136
Classificação ao nível 22: (CLP2 (MML3 UP10)) similaridade: 0.901957
Classificação ao nível 23: (MML2 (UP3 UP16)) similaridade: 0.901957
Classificação ao nível 24: (CLP1 UP1) similaridade: 0.886102
Classificação ao nível 25: ((PC2 UP17) UP14) similaridade: 0.880704
Classificação ao nível 26: (CSP1 MML1) similaridade: 0.857863
Classificação ao nível 27: (DMM4 UP13) similaridade: 0.815444
Classificação ao nível 28: ((MML2 (UP3 UP16)) PC1) similaridade: 0.790698
Classificação ao nível 29: (AHTML1 JCAPP1) similaridade: 0.758183
Classificação ao nível 30: ((CLP2 (MML3 UP10)) (((ALPC (((ALPVB ALPO) ALPHTML) JCPC2) UP7)) ALPJ) UP8) ((AHTML2 JCAPP2) UP2)) (UP4 UP6)) (UP9 UP12)) similaridade: 0.737483
Classificação ao nível 31: ((CSP1 MML1) (UP5 UP18)) similaridade: 0.725375
Classificação ao nível 32: ((PC2 UP17) UP14) ((DMM1 DMM3) DMM2)) similaridade: 0.639428
Classificação ao nível 33: ((CLP2 (MML3 UP10)) (((ALPC (((ALPVB ALPO) ALPHTML) JCPC2) UP7)) ALPJ) UP8) ((AHTML2 JCAPP2) UP2)) (UP4 UP6)) (UP9 UP12)) ((MML2 (UP3 UP16)) PC1)) similaridade: 0.614332
Classificação ao nível 34: (JCPC1 (CSP2 UP19)) similaridade: 0.613585

Classificação ao nível 35: (((CSP1 MML1) (UP5 UP18)) (DMM4 UP13)) similaridade: 0.315916
Classificação ao nível 36: ((CLP1 UP1) (AHTML1 JCAPP1)) similaridade: 0.28476
Classificação ao nível 37: (((((CLP2 (MML3 UP10)) (((((ALPC (((ALPVB ALPO) ALPHTML) JCPC2) UP7)) ALPJ) UP8) ((AHTML2 JCAPP2) UP2)) (UP4 UP6)) (UP9 UP12))) ((MML2 (UP3 UP16)) PC1)) ((CSP1 MML1) (UP5 UP18)) (DMM4 UP13))) similaridade: 0.141935
Classificação ao nível 38: ((JCPC1 (CSP2 UP19)) (((PC2 UP17) UP14) ((DMM1 DMM3) DMM2))) similaridade: 0.130743
Classificação ao nível 39: (((CLP1 UP1) (AHTML1 JCAPP1)) ((JCPC1 (CSP2 UP19)) (((PC2 UP17) UP14) ((DMM1 DMM3) DMM2)))) similaridade: 0.00129985

A partir da classificação dos 39 níveis de similaridade, inicia-se a interpretação dos principais grupos gerados pela (Figura 38). No grupo de similaridades selecionado, no nível 1, a relação se estabelece entre as categorias o aluno já manipulou a linguagem HTML (AHTML2) e o aluno já criou um aplicativo (JCAPP2) – (Figura 39). Um nó de similaridade foi gerado para a partir das categorias.

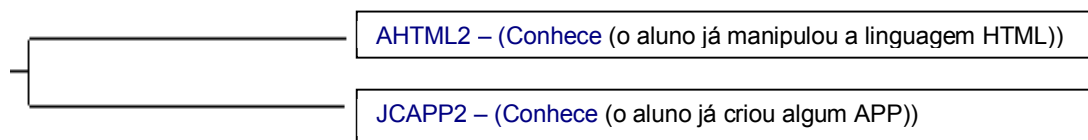


Figura 39 – Nível 1 de similaridade
Fonte: Elaboração própria

No grupo de similaridades selecionado, no nível 1, a relação se estabelece entre as categorias o aluno já manipulou a linguagem de programação Visual Basic (ALPVB) e o aluno já manipulou outra linguagem de programação (ALPO) (Figura 40). Um nó de similaridade foi gerado para a partir das categorias.

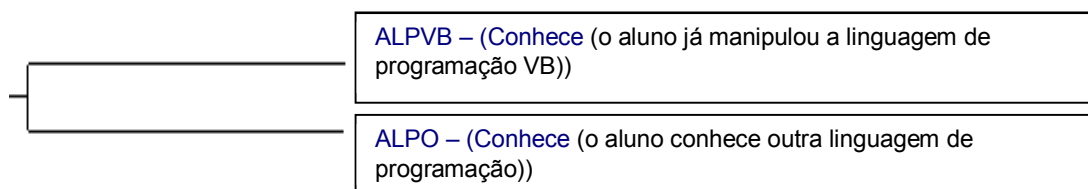


Figura 40 – Nível 2 de similaridade
Fonte: Elaboração própria

No grupo de similaridades selecionado, no nível 3, a relação se estabelece entre as categorias: o aluno já manipulou a linguagem de programação Visual Basic (ALPVB) - o aluno já manipulou outra linguagem de programação (ALPO)

– e o aluno já manipulou a linguagem de programação HTML (ALPHTML) (Figura 41). Dois nós de similaridades foram gerados para a partir das categorias.

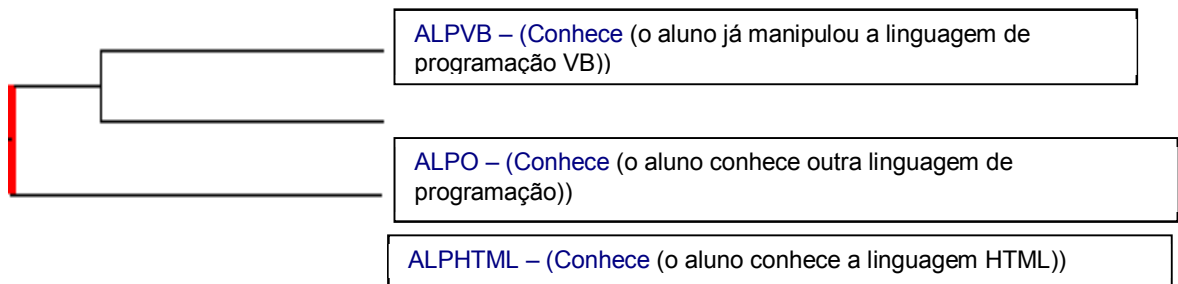


Figura 41 – Nível 3 de similaridade
Fonte: Elaboração própria

No grupo de similaridades selecionado, no nível 4, a relação se estabelece entre as categorias: o aluno já manipulou a linguagem de programação Visual Basic (ALPVB) - o aluno já manipulou outra linguagem de programação (ALPO) – e o aluno já manipulou a linguagem de programação HTML (ALPHTML) – e o aluno já criou um programa de computador (JCPC2) (Figura 42). Três nós de similaridades foram gerados para a partir das categorias.

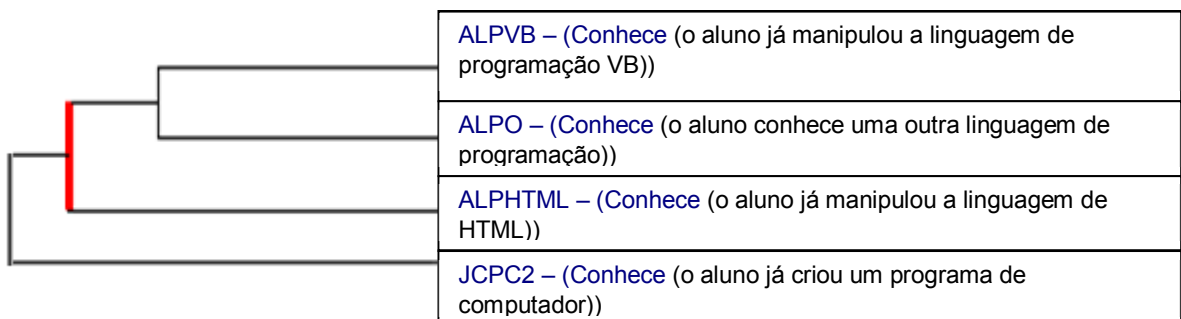
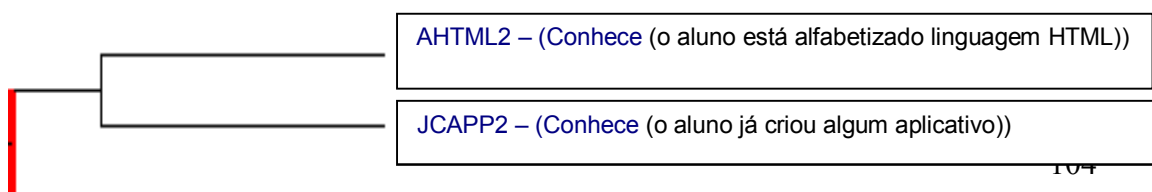


Figura 42 – Nível 4 de similaridade
Fonte: Elaboração própria

No grupo de similaridades selecionado, no nível 6, a relação se estabelece entre as categorias: o aluno está alfabetizado com a linguagem (AHTML2) - o aluno já criou um aplicativo (JCAPP2) – o aluno considera que conseguiu aperfeiçoar seu conhecimento durante a criação do projeto final (UP2) (Figura 43). Dois nós de similaridades foram gerados para a partir das categorias.



UP2 – (Aperfeiçoou seu conhecimento (o aluno considera que conseguiu aperfeiçoar seu conhecimento durante a criação do projeto final))

Figura 43 – Nível 6 de similaridade
Fonte: Elaboração própria

No grupo de similaridades selecionado, no nível 8, a relação se estabelece entre as categorias o aluno se familiarizou com os códigos do *Processing* (UP9) e o aluno trabalhou com edição de imagens (UP12) – (Figura 44). Um nó de similaridade foi gerado para a partir das categorias.

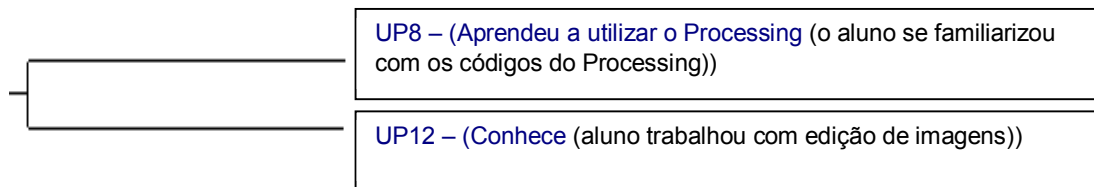


Figura 44 – Nível 8 de similaridade
Fonte: Elaboração própria

No grupo de similaridades selecionado, no nível 11, a relação se estabelece entre as categorias: o aluno já teve contato com a linguagem de programação C++ (ALPC) - o aluno já manipulou a linguagem de programação VB (ALPVB) – o aluno conhece uma outra linguagem de programação (ALPO) - o aluno já manipulou a linguagem de HTML (ALPHTML) - o aluno já criou um programa de computador (JCPC2) - o considera que conseguiu voltar a programar com a criação do projeto final (UP7) - o aluno já manipulou a linguagem de programação Java (ALPJ) - o aluno se familiarizou com os códigos do Processing (UP8) - (Figura 45). Sete nós de similaridades foram gerados para a partir das categorias.

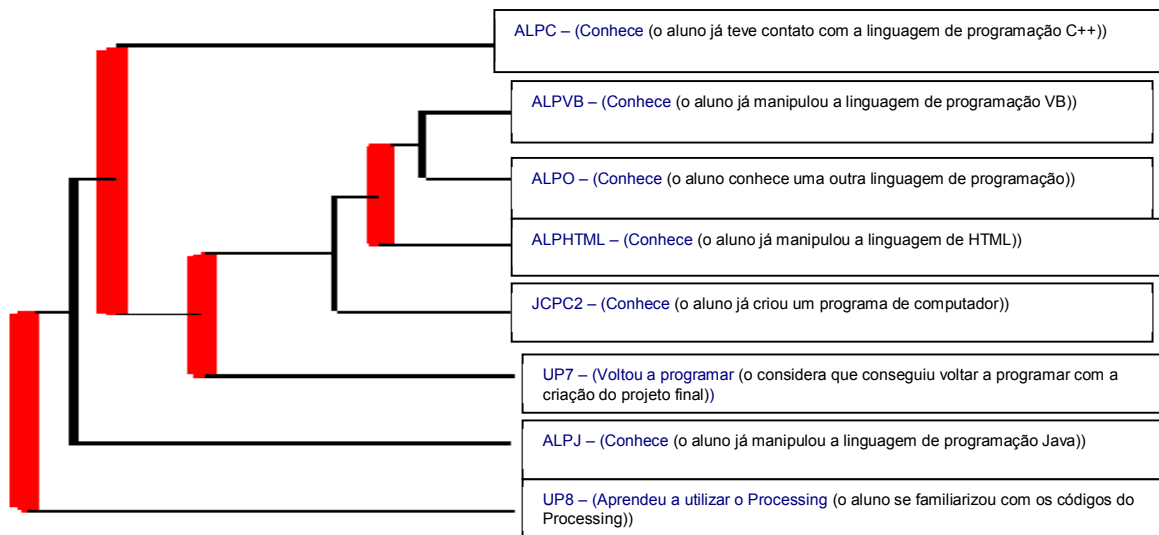


Figura 45 – Nível 1 de similaridade

No grupo de similaridades selecionado, no nível 14, o aluno considera que encontrou diversão durante a criação do projeto (UP3) e o aluno considera que houve um grande aprendizado em feitos que antes eram realizados sem pensamento lógico e matemático que foi trabalhado durante a criação do projeto final (UP16) – (Figura 46). Um nó de similaridade foi gerado para a partir das categorias.

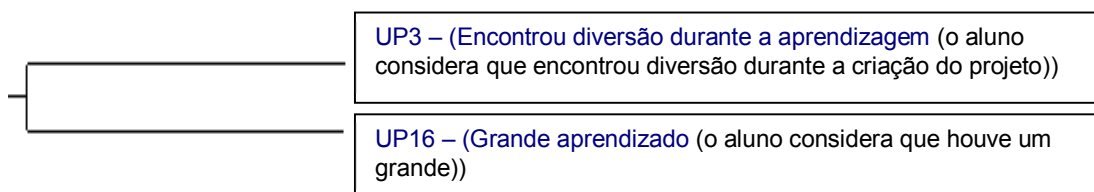


Figura 46 – Nível 14 de similaridade

No grupo de similaridades selecionado, no nível 17, o aluno conhece o software Processing (CSP2) e o aluno considera que houve aperfeiçoamento pessoal durante a elaboração do projeto final (UP19) – (Figura 47). Um nó de similaridade foi gerado para a partir das categorias.

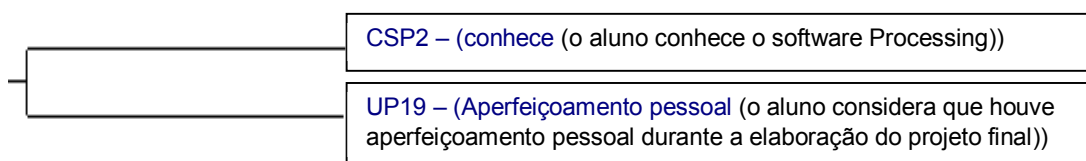


Figura 47 – Nível 17 de similaridade

No grupo de similaridades selecionado, no nível 20, a relação se estabelece entre as categorias: o aluno já teve contato com a linguagem de programação C++ (ALPC) - o aluno já manipulou a linguagem de programação VB (ALPVB) – o aluno conhece uma outra linguagem de programação (ALPO) - o aluno já manipulou a linguagem de HTML (ALPHTML) - o aluno já criou um programa de computador (JCPC2) - o considera que conseguiu voltar a programar com a criação do projeto final (UP7) - o aluno já manipulou a linguagem de programação Java (ALPJ) - o aluno se familiarizou com os códigos do *Processing* (UP8) - o aluno está alfabetizado com a linguagem de HTML (AHTML2) – o aluno já criou um aplicativo (JCAPP2) – O aluno considera que conseguiu aperfeiçoar seu conhecimento durante a criação do projeto final (UP2) – O aluno considera que conseguiu os objetivos durante a criação do projeto final (UP4) – O aluno considera que conseguiu os objetivos e está satisfeito com a criação do projeto final (UP6) – O aluno considera que se familiarizou com o *Processing* para a criação do projeto final (UP9) – O aluno trabalhou com edição de imagens (UP12) - (Figura 24). Quatorze nós de similaridades foram gerados para a partir das categorias.

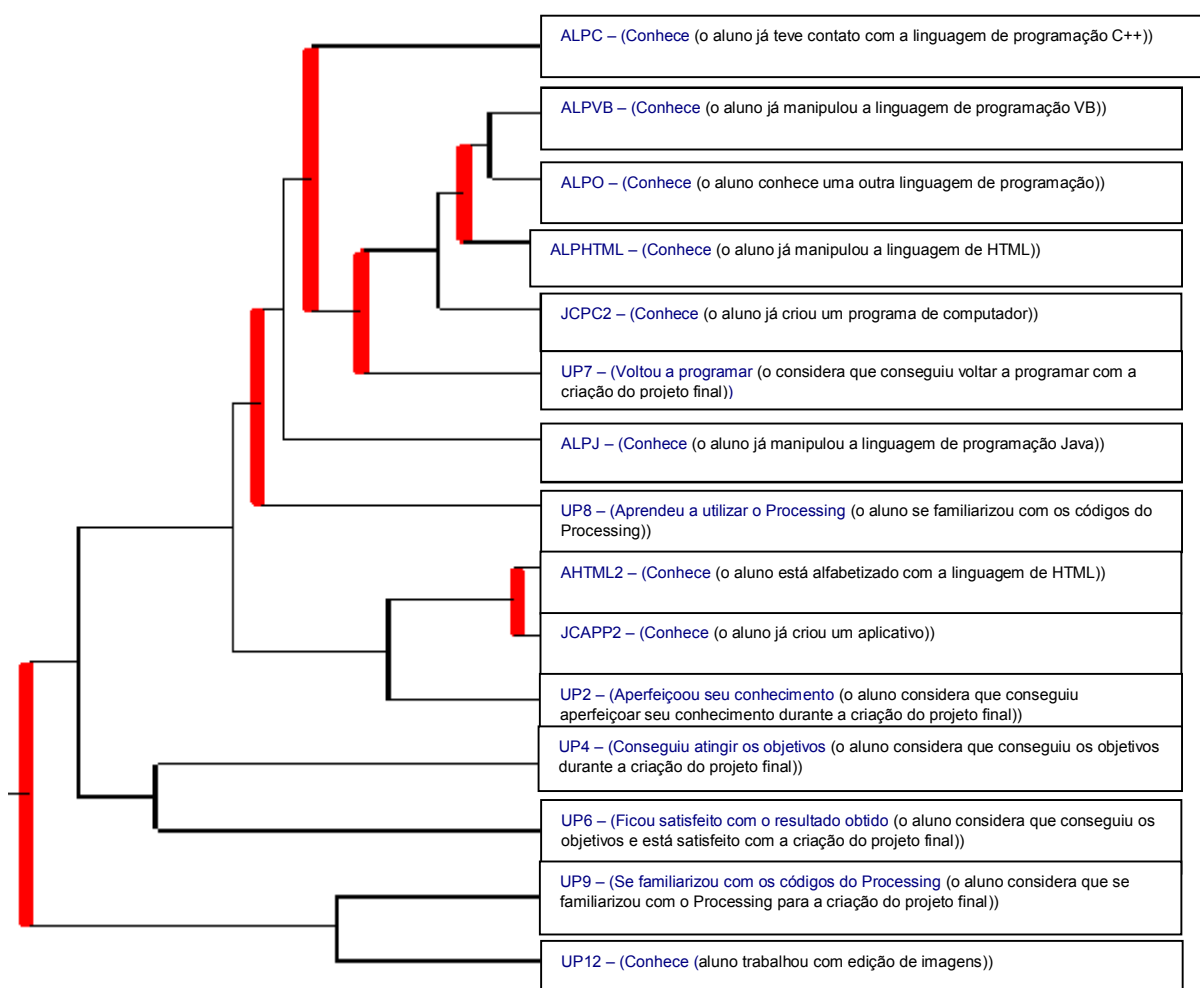


Figura 48 – Nível 20 de similaridade

A partir da análise dos nós mais significativos gerados da análise árvore de similaridade, realizada a partir das árvores geradas pelo Chic, reforçam algumas das hipóteses levantadas durante o estudo de campo sobre o uso das Tecnologias Emergentes no ambiente escolar e apresentada nesta dissertação, indicada na seção 2.3 e detalhadas nas subseções do Capítulo 5, na tentativa de responder de maneira positiva algumas questões que causam muito interesse, referentes especificamente de como as Tecnologias Emergente podem auxiliar docentes e discentes no ambiente escolar.

Uma das questões leva a pensar que a criação de um projeto digital que em seu desenvolvimento (produção) utilize de uma linguagem de programação e por meio de software apropriado (Processing) e que esses alunos envolvidos no projeto devem possuir habilidades em programação, habilidade que é cada vez mais utilizada nas diversas áreas envolvendo Tecnologia.

As análises das respostas dos aprendizes evidenciam a necessidade de repensar que uma adequação deve ser ajustada quando a metodologia utilizada durante o processo de ensino e de aprendizagem, diante do atual cenário, que exige dos profissionais os conhecimentos diversos sobre mídias digitais para o desempenho de suas atividades profissionais. Uma formação capaz de auxiliar a aprendizagem tem base no emprego das Tecnologias Emergentes no processo educacional, tanto como fim, correspondendo à exigência do mercado de trabalho, quanto como meio, promovendo um aprender fazendo, com base no que diz Piaget (1977), o que leva a pensar na formação de um profissional apto a produzir material os diversas de habilidades.

Os alunos, que se mostraram abertos à proposta, ao responderem ao primeiro questionário, expressando suas considerações a respeito do uso do software *Processing* e sua utilização na criação do projeto final proposto pelo professor, também carregavam certos conceitos, com base em sua formação e vivência. Algumas das relações pressupostas se confirmaram ao longo da interpretação das análises, como aquelas estabelecidas, por exemplo, entre as categorias o aluno conhece o software *Processing* e o aluno considera que houve aperfeiçoamento pessoal durante a elaboração do projeto final (Figura 49, p. 116). E outras me surpreenderam que geradas através da árvore de similaridade.

CONSIDERAÇÕES FINAIS

O uso da computação na criação e exibição de conteúdos diversos não é nenhuma novidade, atualmente grande parte dos objetos e produtos de nosso cotidiano também surgem a partir dos mesmos princípios: códigos matemáticos programáveis e executáveis. São informações armazenadas nos meios magnéticos e linguagens computacionais que, a partir de uma determinada lógica, ao ser interpretada e executada, gera produções e resultados.

As tecnologias sempre serviram como meio, método e inspiração para as produções humanas e, por isso, realizo esta pesquisa a fim de detectar como se dá hoje, o uso das Tecnologias Emergentes que ao se aliarem as metodologias de ensino e aprendizagem, por meio de linguagem de programação, especificamente o *Processing* pode ser uma forte ferramenta para o ensino e aprendizagem.

Procurou-se elucidar nesta pesquisa quais aspectos contribuíram com o processo de apropriação e de letramento digital ao se usar as Tecnologias Emergentes. A identificação, a compreensão e o entendimento desses aspectos, deverão contribuir para reflexões metodologias sobre o ensino, que tem nas tecnologias digitais um suporte para sua realização.

Identificou-se alguns padrões que são responsáveis por estes processos de ensino e aprendizagem. Eles podem ser divididos em sete fases: emocional, técnico-operacional, imitação, relação-comunicação, relação-informação, relação-expressão reflexiva e autoformação.

Esta pesquisa evidenciou que entre os indivíduos estudados havia uma informação com relação ao conhecimento com a matéria matemática. Para o primeiro estudo de campo 50% dos entrevistados não gostavam de matemática e 37,5% tinham dificuldade com matemática. Já para o segundo estudo de campo 32% dos entrevistados não gostavam de matemática e tinham a mesma porcentagem de dificuldade com a matemática.

Outra questão evidenciada entre os indivíduos estudados foi que sobre a realização de curso de informática e o conhecimento de alguma linguagem de programação. Para o primeiro estudo de campo 21% dos entrevistados não haviam feito nenhum curso de informática e 46% não conhecem nenhuma linguagem de programação. Já para o segundo estudo de campo 65% dos

entrevistados não haviam feito nenhum curso de informática e 77% não conheciam nenhuma linguagem de programação. Partindo deste cenário imaginava-se que esses indivíduos não conseguiriam realizar os exercícios propostos pelo professor.

Os resultados obtidos nesta pesquisa, leva a afirmar que mesmo diante destas e de outras deficiências e defasagem de matérias, os alunos tiveram condições de realizar as atividades propostas pelo professor, mesmo que com auxílio do professor ou de colegas.

Como explicitado no Capítulo 3, esta pesquisa mostra que a interação mediada por meio da apropriação das Tecnologias Emergentes, em específico o uso do *Processing*, como ferramenta de aprendizagem, produziu um resultado louvável no cenário da aprendizagem, pois como mostra as dezenove categorias encontradas nos depoimento dos alunos em relação ao uso do software, estas justificam que mesmo onde os alunos encontravam dificuldades com linguagem de programação, estes conseguiram produzir o projeto final proposto pelo professor.

A partir da árvore constante da Figura 38 (p. 101), que abrange um número maior de categorias na análise de similaridade do Chic, é possível delinear outras considerações, algumas relacionadas à interação, destaque da seção 4.2, mas em grande parte focadas nas Tecnologias emergentes e na relação dos aprendizes – e da aprendizagem – com elas, especialmente o uso do software Processing

O fato de os alunos perceberem na produção do projeto final a eficiência e a possibilidade de aperfeiçoamento pessoal e se sentirem felizes com os resultados obtidos indica que as Tecnologias emergentes podem contribuir para a formação, levando a refletir sobre sua relevância na educação, mas tendo como evidência a capacidade de pensar, imaginar e arriscar-se na atividade criadora e não o simples cumprimento de tarefas.

Muitos estudantes ainda possuem certa desconfiança com relação às Tecnologias, pois muitas vezes os aprendizes enxergam que em alguns cenários são somente executores de tarefas ou copista de códigos já prontos. É preciso estar atento à demanda daqueles que as veem como auxiliares no processo de construção de conhecimento sobre a orientação docente. Isto é evidenciado nas respostas dos aprendizes, que mesmo não tendo habilidade

com programação e em alguns casos encontrando dificuldades para programar estes dizem “**Aperfeiçoei meus conhecimentos**” e “**fiquei satisfeito**”, com a construção do projeto final, como indicado na análise exposta na Figura 48 (p. 107), que demonstra que as ferramentas digitais oferecem vantagens significativas no processo de aprendizagem mediante acompanhamento docente.

Quanto aos objetivos específicos, a pesquisa resultante nesta dissertação possibilitou atingi-los à medida que permitiu:

- **Criar situações de ensino e aprendizagem**, utilizando a linguagem *Processing* (linguagem de programação desenvolvida pelo *MIT – Massachusetts Institute of Technology*) que foi construída para as artes eletrônicas e comunidades de projetos visuais com o objetivo de ensinar noções básicas de programação computacional em um contexto visual e de fácil utilização. Para tanto, a aplicação das Tecnologias Emergentes deve ser planejada e a metodologia aplicada deve ser revista para que haja o resultado esperado, o envolvimento de todos – docente e aprendizes é vital – para que se efetive enquanto dispositivo de interação e troca de informação entre ambos, caso contrário, configura apenas uma alternativa de recurso pedagógico, que pode ou não ser empregada. Conforme apresentada em detalhes ao longo do Capítulo 5, o emprego das Tecnologias Emergentes foi condição para o desenvolvimento do processo de aprendizagem na referida matéria. Saliento que a metodologia estudada no desenrolar desta pesquisa é somente uma das maneiras de empregar as Tecnologias Emergentes no referido curso, cabendo a cada docente estabelecer os dispositivos e softwares a serem empregados e como adotá-los, de acordo, inclusive, com a infraestrutura disponível, com seu conhecimento sobre uma ou outra tecnologia e com o envolvimento dos alunos;

- **explicitar como as Tecnologias Emergentes podem auxiliar no processo de ensino e de aprendizagem**. A avaliação da aprendizagem realizada pelos alunos permitiu evidenciar que o uso do software *Processing* os auxiliaram no tocante a produção do projeto final, especificando em um arquivo digital salvo ora em arquivo com extensão .txt ou em arquivo com extensão .pde e enviado ao ambiente digital **Ensino aberto** – portfólio individual do aluno;

Para que os objetivos fossem atingidos, no entanto, foi necessário muito mais que uma análise das respostas dos alunos e acampamento dos trabalhos criados em ambiente digital a meus questionamentos.

Diante de tais considerações, esta dissertação justifica-se à medida que contempla: a) uma reflexão sobre a importância da interação mediada pelas Tecnologias Emergentes na formação profissional do curso de Midialogia que, no mercado de trabalho, que se depara com novas demandas tecnológicas; b) os futuros profissionais devem ser autodidata, pois esta habilidade deve estar presentes nos estudantes em cursos universitários; e c) uma abordagem transdisciplinar, envolvendo comunicação, educação e tecnologias, áreas indissociáveis em uma pesquisa como a apresentada nesta pesquisa.

A formação dos conceitos no indivíduo é resultado de uma atividade complexa, na qual todas as funções intelectuais fundamentais participam. No entanto, este processo não pode ser reduzido à associação, à tendência, à imagética, à interferência ou às tendências determinantes, todas são indispensáveis, mas não são suficientes se não se empregar signos ou a palavra como meios pelos quais dirigimos as nossas operações mentais, controlamos os seus cursos e o canalizamos para a solução do problema com que nos defrontamos (VIGOSTKI, 2001, p. 61).

Porém, a tarefa, por si só, não explica o mecanismo de desenvolvimento que tem por resultado a formação do conceito. O que prova que esta investigação é a de intentar a compreensão das relações intrínsecas entre as tarefas externas e a dinâmica do desenvolvimento e considerar o início dos conceitos como função do crescimento cultura e social global dos indivíduos, que não afeta apenas o conteúdo, mas também o seu modo de pensar (VIGOSTKI, 2001, p. 61).

Apoiado no Pensamento Complexo, devemos considera a aleatoriedade e a desordem presentes no universo, na rede e, conseqüentemente, na vida do homem contemporâneo. O pensamento linear pressupõe isolamento parte/todo e uma organização sequencial de dados, acontecimentos, fenômenos. Por muitas décadas essa estratégia foi considerada apropriada para que o ensino e a aprendizagem pudessem ser realizados de maneira bem-sucedida, mas a emergência da rede trouxe para sociedade atual a consolidação de novas formas de interação entre pessoas e entre pessoas e conteúdo.

É inegável que a presença das Tecnologias Emergentes tem provocado transformações importantes na organização econômica, social e cultural. Isso pode ser observado desde a maneira como interagimos socialmente, como acessamos a informação, como procedemos com as transações bancárias etc. No entanto, como afirma Buckingham (2007), a maior parte do tempo estamos utilizando um potencial muito limitado dessas tecnologias, uma vez que nos restringimos ao uso de “softwares de escritório” como os processadores de texto, planilhas etc. Ainda estamos na fase de entender e explorar essas tecnologias como se fossem sofisticadas máquinas de escrever, de acessar a informação e de se comunicar (VALENTE, 2015).

Referências

- AMADO, P. **Introdução à programação gráfica**. Manual de apoio às sessões de formação em Introdução à programação Gráfica usando Processing; Universidade do Porto, 2006.
- ARNHEIM, R. **Arte e percepção visual**. São Paulo: Pioneira Thomson, 1998.
- BISPO, C. A. F. CASTANHEIRA L.B.; FILHO O.M.S. **Introdução à lógica matemática**; Editora Cenage Learning, 1ª edição, São Paulo, 2014;
- BOYER, C. B. **História da matemática**. Traduzido por Elza F. Gomide. São Paulo: Edgard Blucher, 1974.
- BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. **AERA 2012**, Vancouver, Canada, 2012.
- BYSTRINA, I. **Tópicos de Semiótica da Cultura**. Tradução: Norval Baitello Junior e Sônia B. Castino; notas e transcrições: Solange Silva; edição: Luiz Lasbeck. São Paulo: CISC/PUC-SP. Maio de 1995. Aulas de Ivan Bystrina. Préprint.
- CUNY, J.; SNYDER, L. e WING, J. M. **Demystifying computational thinking for noncomputer scientists**, 2010. Disponível em <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>. Acesso em 10/11/2016.
- CHARLOT, B. **Da relação com o saber às práticas educativas**; Cortez Editora; 1ª edição, São Paulo, 2013;
- CRATO, N. **A matemática das coisas**; Livraria da Física; 1ª edição, São Paulo, 2009;
- DAGHLIAN, J. **Lógica e Álgebra de Boole**; Editora Atlas; 4ª edição, São Paulo, 2010;
- DELORS, J. **Educação: um tesouro a descobrir**. São Paulo, Cortez, 1998.
- DERTOUZOS, M. L. **O que será: como o novo mundo da informação transformará nossas vidas**. São Paulo: Companhia das Letras, 1997.
- GADOTTI, M. **Perspectivas atuais da educação**. Porto Alegre: Artes Médicas, 2000.
- GIL, A. C. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 1991.
- GRAS, R.; AG ALMOULOU, S. A implicação estatística usada como ferramenta em um exemplo de análise de dados multidimensionais. In: **Revista do**

Programa de Estudos Pós-graduação em Educação Matemática – PUC-SP, vol. 4, n. 2. p. 75-88. São Paulo: Educ, 2002.

HILDEBRAND, H. R. **As imagens matemáticas**. Texto inédito. São Paulo: Tese de Doutorado apresentada no Departamento de Comunicação e Semiótica da Pontifícia Universidade Católica de São Paulo – PUCSP, 2001.

HUIZINGA, J. **Homo Ludens**. O Jogo como elemento da cultura. São Paulo: Perspectiva, 2001.

FLUSSER, V. **O mundo codificado**. São Paulo: Editora Cosacnaify, 2015.

FRENKEL, E. **Amor e matemática**. Rio de Janeiro: Editora Casa da Palavra, 2014.

LIVIO, M. **Deus é matemático?** Editora Record; 3ª edição, Rio de Janeiro, 2012.

LEMOV, D. **Aula nota 10**: 49 técnicas para ser um professor campeão de Audiência. São Paulo: Editora Da Boa Prova, 2011.

LÉVY, P e MASSAD, A. A Caminho da Inteligência Coletiva. In: **Revista Fórum**, São Paulo, set. 2002. Disponível em:

<http://www.revistaforum.com.br/revista/7/inteligenciacoletiva.htm>. Acesso em 10 set 2004.

LÉVY, P. A Inteligência Coletiva. In: **Webinsider**. 09 set. 2002. Disponível em:

<http://webinsider.uol.com.br/vernoticia.php/id/1420>. Acesso em: 20 nov. 2004.

_____. **As tecnologias da inteligência**. Rio de Janeiro: 34, 1993.

_____. **Cibercultura**. São Paulo. São Paulo: Editora 34, 1999.

LOPES, J. J. **Ensinar em ambientes virtuais I**: Quarta Parada - Hipertextos e Mapas.

MAEDA, J. **As Leis da simplicidade** São Paulo: Editora Novo Conceito, 2007.

MAIA, F.I. **No jardim dos letramentos**: tomadas de consciência e poéticas em rede na cultura da convergência. Tese de (doutorado)26 desenvolvida na Unicamp – Programa de Pós-graduação em Artes Visuais. CAMPINAS/SP: UNICAMP, 2011.

MARCONI M. A.; LAKATOS, E. M. **Fundamentos de metodologia científica**. 5ª Edição. São Paulo: Atlas, 2003.

MATURANA, H.; VARELA F. **A árvore do conhecimento**. São Paulo. Editorial PSY, 1995.

_____. **Cognição, ciência e vida cotidiana**. Belo Horizonte. Editoria UFMG, 2001.

_____. **Emoções e linguagem na educação e na política.** 3ª reimpressão, Belo Horizonte. Editoria UFMG, 2002.

McROBERTS, M. **Arduino básico.** São Paulo: Novatec, 2011.

MORAES, M. C.; VALENTE, J. A. **Como pesquisar em educação a partir da complexidade e da transdisciplinaridade?** São Paulo: Paulus, 2008.

MORIN, E. **Introdução ao pensamento complexo.** Porto Alegre: Sulina, 2015.

_____. **Os sete saberes necessários à educação do futuro.** 2ª edição. São Paulo: Editora Cortez, 2000.

_____. **A cabeça bem-feita.** 8ª edição. Rio de Janeiro: Bertrand, 2003.

_____. **Ciência com consciência.** 8ª edição. Rio de Janeiro: Bertrand, 2005.

OAKIM, P. **Arte e programação na linguagem *Processing*.** Dissertação (Mestrado desenvolvido no TIDD) Programa de Pós-graduação de Tecnologia da Inteligência e Desing Digital da PUC/SP. São Paulo: PUCS/SP, 2015.

PAPERT, S. *Teaching Children Thinking*, In: **Logo Memo nº 2**, 1971. Disponível em: https://archive.org/stream/bitsavers_mitaiaimAI_471587/AIM-247_djvu.txt. Acessado em 28 de jul. 2016.

PEIRCE, C. S. **Semiótica e filosofia:** com tornar clara nossas ideias. São Paulo: Editora Abril, 1975.

_____. **Coleção os pensadores:** Vida e obra de Peirce. São Paulo: Editora Abril Cultural, 1983.

PINKER, S. **Como a mente funciona.** 3ª edição. São Paulo: Companhia da Letras, 2015.

PINO, A. **As marcas do humano:** às origens da constituição cultural da criança na perspectiva de Lev S. Vigotski. São Paulo: Cortez Editora, 2005.

_____. **Técnica e semiótica na era da informática.** Contrapontos, v. 3, n. 2, p. 283-296, mai/ago 2003.

PRENSKY, M. **Aprendizagem baseada em games digitais.** São Paulo: Senac/SP, 2012.

SANTAELLA, L. **O que é semiótica.** São Paulo: Brasiliense, 1985.

_____. **A assinatura das coisas:** Peirce e a literatura. Rio de Janeiro: Imago, 1992.

_____. **Matrizes da linguagem e do pensamento:** sonora, visual e verbal. São Paulo: Iluminuras/Fapesp, 2001.

_____. **Semiótica aplicada.** São Paulo: Thomson, 2002.

_____. **Linguagens líquidas na era da mobilidade**. 2ª. ed. São Paulo: Paulus, 2011.

_____. **Comunicação e semiótica**. São Paulo: HACKER EDITORES, 2004.

_____. **Navegar no ciberespaço: o perfil cognitivo do leitor imersivo**. 4 ed. São Paulo: Paulus, 2011.

SEVERINO, A. J. **Educação, sujeito e história**. São Paulo: Olho d'Água, 2001.

TONUS, M. **Interação no processo de aprendizagem em comunicação social**. In: PRIMO, Alex et al (Orgs.). *Comunicação e interações*. Porto Alegre: Sulina, 2008.

TURKLE, S. **A vida no ecrã: a identidade na era da internet**. Tradução de Paulo Faria. Ed. Relógio d'água, 1997.

VALENTE, J. A. **Conhecimento repensando a educação**. 2ª edição. São Paulo. Unicamp/NIED, 1998.

_____. **Uso do Chic na formação de educadores**. Rio de Janeiro. Letra capital, 2015.

VYGOTSKY, L. S. **Pensamento e linguagem**. São Paulo: Martins Fontes, 1993.

_____. **A formação social da mente: o desenvolvimento dos processos psicológicos superiores**. São Paulo: Martins Fontes, 2007.

WING, J.M. **Computational thinking**. *Communications of the ACM*, 49(3), p.33-35, 2006.

APENDICE A

Questionário 01, elaborado no Google Forms, através do link

(<https://docs.google.com/forms/d/1jPdWOqTSe65KE8JQyVBBL97K8KWBvUY-Ztf8afzmeYE/edit#>)



Case - Unicamp

Pesquisa a alunos Unicamp - Matéria Fundamentos da matemática - Uso Processing

***Obrigatório**

1-) Você possui computador em casa? *

- ☒ Sim
☐ Não

2-) Você gosta de matemática? *

- ☒ Sim
☐ Não

3-) No ensino Fundamental/Médio, você possuía dificuldade com a matemática? *

- ☒ Sim
☐ Não

4-) Você já fez algum curso de informática? *

- ☒ Sim
☐ Não

5-) Você possui um Smartphone (telefone inteligente)? *

- ☒ Sim
☐ Não

6-) Você conhece alguma linguagem de programação? *

- ☒ Sim
☐ Não

7-) Já criou algum aplicativo fazendo uso de linguagem de programação? *

- ☒ Sim
☐ Não

8-) Se você respondeu sim para a pergunta anterior. Qual linguagem de programação você conhece? *

- ☒ Java
☐ VB
☐ C#
☐ PHP
☐ C
☐ C++
☐ Python
☐ Outras

9-) O uso da ferramenta Processing durante as aulas práticas te ajudou na resolução dos exercícios propostos?? *

- ☐ Sim
☐ Não

10-) Durante as aulas você utilizou seu próprio computador para elaborar os exercícios? *

- ☐ Sim
☐ Não

APENDICE B

Questionário 02, elaborado no Google Forms, através do link

(<https://docs.google.com/forms/d/1jPdWOqTSe65KE8JQyVBBL97K8KWBvUY-Ztf8afzmeYE/edit#>)



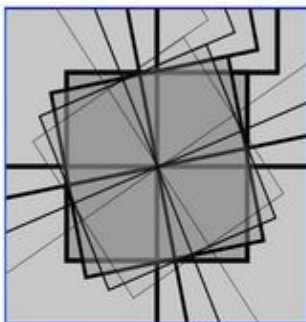
Case - Unicamp - parte 02

Levantamento de dados uso Processing

Identifique -se. +

No exercício abaixo, explique o que você conseguiu entender do bloco do código a seguir: `size(500,500); background(200); smooth(); rectMode(CORNER); fill(50,10); strokeWeight(3); rect(100,100,300,300); line(300,100,450,100); line(450,100,450,0); line(0, 250,500, 250); line(250,0,250,500);` +

Exemplo - exercício 01 - Usando Figuras, Transparência e Movimentações

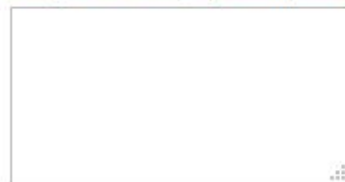


No aula sobre criação de cenário. Você encontrou dificuldade para criar o cenário sugerido pelo professor? Cite quais foram as principais dificuldades encontradas. +

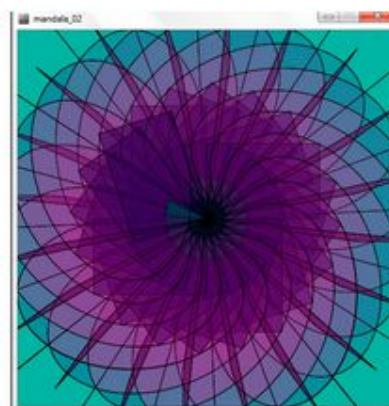
Exemplo - exercício 02 - Criação cenário



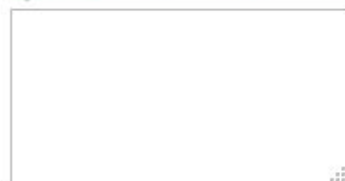
No aula sobre criação da Mandala. Use das funções `if`, `else` e `for`. No bloco de código do código a seguir: `background(#FFFFFF); size (700, 700); translate(-----); noFill (); strokeWeight(5); stroke (#FC0000 ); for(int i = 0; i<6; i++) { ellipse(0, 0, 350, 100); rotate (PI/6); } Qual/quais propriedade(s) inserido(s) na linha translate para que a mandala pudesse ser criada? +`



Exemplo exercício - criação mandala (comando `if/else/for`)



No aula criando animação, o bolinha rebatendo nas 4 bordas com roquete. Qual a função das linhas: `void draw ();` `void setup ();` `void mousePressed ();` respectivamente. +



Exemplo exercício - bolinha rebatendo nas 4 bordas com roquete



APENDICE C

Resposta alunos ao questionário 02, exercício 01. (1º semestre 2015).

D1	A tela aberta terá dimensões de 500x500 pixels, com um plano de fundo em escala de cinza. O rectMode estabelece o canto esquerdo como referência para desenhar o quadrilátero, que terá preenchimento também cinza e com alguma transparência. O quadrilátero tem centro 250, 250 e laterais de 300 px. As linhas terão espessura 8 (determinada por strokeWeight), e serão desenhadas quatro linhas, segundo as coordenadas determinadas.
D2	size(500,500) = tamanho da tela a ser aberta pelo processing; background(200) = determinação da coloração de fundo; smooth() = desenha formas sem contorno; rectMode(CORNER) = desenha quadrado; fill(150,150) = preenche a forma com determinada cor; strokeWeight(8) = determina a grossura da linha; rect(100,100,300,300) = determina posição da forma na tela e dimensões de altura e largura; line(300,100,450,100) = determina posição da linha na tela pelas coordenadas do ponto inicial e final; line(450,100,450,0) = linha que inicia em determinado ponto e termina coincidindo com o lado esquerdo da tela; line(0,250,500,250) = linha com ponto inicial do lado esquerdo;
D3	"Size" define o tamanho do quadro do sketch a ser criado, "background" determina a cor do fundo, "smooth" serve para atenuar as linhas desenhadas, "fill" preenche determinada figura com a cor inserida entre parênteses, "strokeWeight" define a grossura da linha a ser desenhada, "rect" desenha retângulos com as coordenadas e dimensões especificadas entre parênteses e line desenha linhas.
D4	size (500,500) significa o tamanho da área de trabalho, em outras palavras, o tamanho de tela que, no caso, significa 500x500 pixels. background(200) é a cor de fundo da tela. smooth(): transparência rectMode(CORNER): o desenho formado ficará no centro da tela fill(150,190) são as cores de preenchimento do desenho strokeWeight(8) é o contorno do desenho rect(100,100,300,300): são as coordenadas de um quadrado line(300,100,450,100): uma linha será formada cujas coordenadas serão: x1=300, y1=100, x2=450 e y2=100 line(450,100,450,0): uma linha será formada cujas coordenadas serão: x1=450, y1=100, x2=450 e y2=0 line(0, 250,500, 250): uma linha será formada cujas coordenadas serão: x1=0, y1=250, x2=500 e y2=250 line(250,0,250,500): uma linha será formada cujas coordenadas serão: x1=250, y1=0, x2=250 e y2=500
D5	É criada uma imagem de tamanho 500x500 pixels a qual possui um fundo com a cor cinza (correspondente no sistema rgb), com linhas "esfumadas" e um retângulo no centro. é determinado uma cor para preenchimento desse retângulo e a grossura das linhas que formam a figura. Por fim, são feitas linhas que cruzam essa figura nas coordenadas determinadas
D6	Tela com tamanho de 500x500; fundo na cor cinza;desenhos de retângulos e linhas que se intercalam, cuja espessura das linhas/sua cor é modificada pelo comando Stroke.
D7	O código faz um desenho com tamanho de tela de 500 com o fundo de cor cinza, com um retângulo e quatro linhas, preenchidos e com a espessura da linha de 8

D8	size (500,500) - tamanho do quadro(janela) é de 500 x 500 pixels background (200) - plano de fundo terá a cor 200 smooth () - função que deixa as linhas mais suaves rectMode (CORNER) - função que deixa as bordas das linhas arredondadas fill (150,190) - preenche alguma forma criada com a cor 150, com a transparência 190 strokeWeight (8) - espessura da linha rect (100,100,300,300) - retângulo que tem como ponto de origem o ponto (100,100) e possui 300 pixels de largura e 300 pixels de comprimento line (300,100,450,100) - linha que se inicia no ponto (300,100) e vai até o ponto (450,100) line (450,100,450,0) - linha que se inicia no ponto (450,100) e vai até o ponto (450,0) line(0, 250,500, 250) - linha que se inicia no ponto (0,250) e vai até o ponto (500,250) line(250,0,250,500) - linha que se inicia no ponto (250,0) e vai até o ponto (250,500)
D9	Primeiramente, determina-se o tamanho da imagem e a cor do fundo, nesse caso, um cinza 200. Em seguida, altera-se a cor do interior de um quadrado, que será criado com o comando line, sendo o primeiro número o ponto esquerdo superior, o segundo, o direito superior, o terceiro, o direito inferior e o quarto, o esquerdo inferior. Posteriormente, criam-se as linhas simples.
D10	O plano de fundo é um quadrado de tamanho 500x500, de cor 200. Há um retângulo colocado no modo CORNER, na posição 100,100, de tamanho 300x300, com contorno de tamanho 8, com preenchimento de cor e modo smooth "ativado". A primeira função de linha do código é uma reta de 180° e a segunda função de linha é uma reta de 90°, perpendicular à primeira. As terceira e quarta linhas são perpendiculares uma a outra também.
D11	O tamanho da tela do programa é 500 por 500 pixels, o fundo será da cor (200,0,0), a função smooth() faz com que as bordas das formas geométricas desenhadas sejam 'suaves', o rectMode(Corner) faz com que o programa interprete os dois primeiros parâmetros de rect() como as coordenadas da aresta superior esquerda; o fill preenche a forma geométrica com a cor desejada; o rect desenha um retângulo com de acordo com os parâmetros descritos; as funções line desenharam linhas de acordo com os parâmetros descritos.
D12	"size(500, 500)" é a sequência que aumenta o tamanho da figura. "background(200)" muda a cor do fundo para o cinza. "smooth()" atenua as linhas. "rectMode(CORNER)" muda o ponto de referência do quadrado para o canto. "fill(150,190)" pinta dentro do quadrado. "strokeWeight(8)" faz as linhas de contorno terem grossura 8. "rect(100,100,300,300)" são as coordenadas do quadrado desenhado no plano cartesiano. "line(x0,y0,x1,y1)" são as coordenadas das linhas que saem do quadrado no plano cartesiano.
D13	Cria-se um arquivo de 500x500, e um desenho constituído de linhas e um quadrado de contorno preto. Em seguida, a cada rotate gerado, copiando o conjunto de linhas e quadrados, a grossura das linhas diminui.
D14	Determina o tamanho da tela para 500 pixels de altura e largura. Determina a cor do fundo 200 em RGB Suaviza as bordas Preenche o que vier abaixo na cor 150,190 em RGB Determina a largura da linha para 8 Desenha um retângulo na posição 100 dos eixos X e Y, de com 300 de altura e largura. Desenha 3 linhas, em diferentes posições
D15	size(500,500); proporções 500 x 500 background(200); define a cor do fundo smooth(); retira a borda rectMode(CORNER); monta um quadrado fill(150,190); provoca o efeito do "esfuminho" strokeWeight(8); grossura da linha rect(100,100,300,300); monta um quadrado line(300,100,450,100); define uma linha line(450,100,450,0); define uma linha

	<code>line(0, 250,500, 250);</code> define uma linha <code>line(250,0,250,500);</code> define uma linha
D16	Com esse bloco de código será criado uma tela de tamanho 500x500 pixels (<code>size(500,500);</code>), com cor de fundo cinza (<code>background(200);</code>). Dentro dessa tela deve haver um quadrado cinza com transparência e as linhas que o delimitam serão "esfumaçadas" com a espessura de 8 pixels (<code>smooth(); rectMode(CORNER); fill(150,190); strokeWeight(8); rect(100,100,300,300);</code>), junto com este quadrado estarão quatro linhas "esfumaçadas" e com espessura de 8 pixels também (<code>line(300,100,450,100); line(450,100,450,0); line(0, 250,500, 250); line(250,0,250,500);</code>).
D17	o tamanho da janela que abriga o programa é de 500 por 500. A cor do plano de fundo é a que corresponde ao número 200, um cinza. O <code>smooth</code> garante a definição das bordas da figura. Ativar o <code>rectMode (CORNER)</code>, significa que quando a função para desenhar o retângulo (<code>rect</code>) foi declarada, os dois primeiros números, no caso 100 e 100, são as coordenadas x e y para o canto superior esquerdo do quadrado. Os próximos dois números, 300 e 300, são respectivamente a largura e altura do retângulo. As quatro funções <code>line</code> desenharam quatro linhas em posições diferentes, os primeiros dois números do parenteses marcam o x e y, respectivamente, do início da linha, e os dois últimos números marcam o x e y do fim da linha.

APENDICE D

Resposta alunos ao questionário 02, exercício 02. (1º semestre 2015)

D1	Demorei um pouco para entender como funcionavam as operações de rotate e translate, e que elas agiam acumuladamente umas sobre as outras. Mas isso foi inicialmente, antes de assimilar a lógica de programação.
D2	Não encontrei dificuldade na criação da forma inicial mas sim na rotação simétrica e periódica da forma.
D3	A maior dificuldade encontrada foi desenvolver um desenho harmônico a partir de coordenadas, pois muitas vezes a figura desenhada não aparecia no local onde eu esperava. Quanto aos comandos não houve dificuldades muito significativas, pois desenvolvi um cenário a partir de códigos simples
D4	Sim, encontrei algumas dificuldades. As maiores foram: - Delimitar as condições de quantas vezes e como as formas geométricas apareceriam no desenho
D5	Sim. A principal dificuldade encontrada foi posicionar as figuras desejadas na imagem (problemas com as coordenadas). Além disso, por não termos conhecimento prévio sobre a função de repetição de imagens, a construção do cenário foi muito mais lenta visto que foi montado figura por figura.
D6	Como foi o primeiro exercício e eu não tinha familiaridade com programação, eu senti dificuldades, principalmente porque eu estava usando o comando Translate, que mudava todas as coordenadas da tela. Quando eu descobri esse erro, o exercício fluiu melhor.
D7	A principal dificuldade ocorreu devido a ser a primeira vez que utilizava o código do processing, desconhecendo como utiliza-lo. Assim, a cada objeto que pretendia colocar era necessário pesquisar como era o código e entende-lo.
D8	Não, apenas foi trabalhoso pois exigiu o "cálculo" dos pontos adequados para a criação do cenário.
D9	As maiores dificuldades foram na criação de triângulos, já que a distância para determinação dos pontos ainda não havia sido entendida.
D10	No exercício do cenário não encontrei grandes dificuldades, mas também não foi muito fácil porque era a primeira vez que eu estava lidando com alguns tipos de formas geométricas no processing, como triângulo e elipse.
D11	Um pouco. Como era uma das minhas primeiras experiências com Processing, usei o método da tentativa e erro para alcançar o objetivo, o que levou algum tempo. Muitas vezes me confundia com as coordenadas e comandos.
D12	Sim. Não conseguia virar os quadrados,
D13	Encontrei apenas algumas dificuldades para colocar elementos nos lugares desejados, uma vez que é difícil acertar a proporção entre os elementos e o cenário em geral.
D14	A maior dificuldade foi em compreender a coloração dos objetos
D15	Na rotação e translação das linhas em volta do quadrado.
D16	Não foram encontradas grandes dificuldades. O maior problema foi ter que testar as coordenadas para que as partes da figura se encaixassem.
D17	Não.

APENDICE E

Resposta alunos ao questionário 02, exercício 03. (1º semestre 2015)

D1	(height/2, width/2) ou (350, 350)
D2	Não absorvi.
D3	Em "translate" são inseridas as coordenadas do centro da figura para que a origem do plano cartesiano seja movida para lá, a fim de facilitar a criação do desenho e possibilitar a rotação do mesmo em torno de seu próprio eixo.
D4	Não sei responder
D5	as propriedades inseridas foram translate (width/2,height/2) para que a mandala fosse iniciada no centro da imagem e não no canto superior esquerdo, que é a posição inicial a origem (0, 0)
D6	Translate {height/2;width/2}.
D7	width/2, height/2
D8	translate (350,350)
D9	Na linha translate foram colocadas as medidas dos quadrados e elipses, sendo esse comando responsável pela alteração da posição inicial da forma.
D10	Não lembro desse exercício.
D11	For.
D12	(height/2; width/2)
D13	Height/2,Width/2
D14	translate(height/2, widht/2)
D15	Não lembro.
D16	translate (width/2, height/2);
D17	(width/2, height/2)

APENDICE F

Resposta alunos ao questionário 02, exercício 04. (1º semestre 2015)

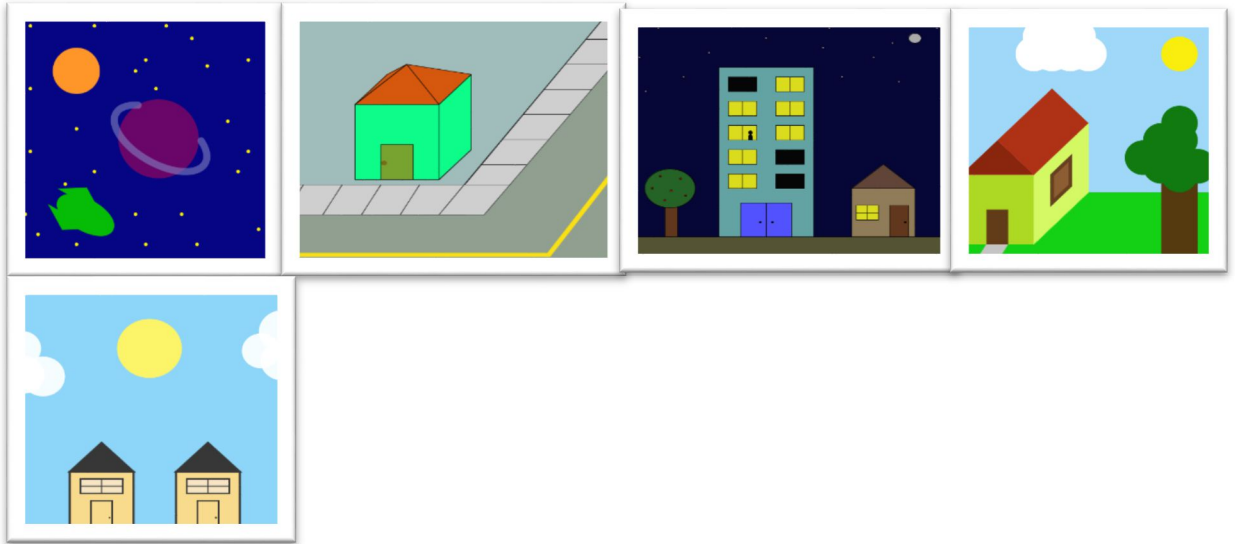
D1	Não realizei esse exercício.
D2	Criar comandos que se repetirão/permanecerão ativos até a finalização do programa. Void Draw é responsável por desenhar a bolinha. void mousePressed define a resposta do programa quando o botão direito do mouse é clicado.
D3	void draw: cor do fundo, tamanho da elipse e as condições para a bolinha bater nos quatro cantos void setup: tamanho da imagem e cor do fundo void mousepressed: analisa a posição cartesiana do mouse quando clicado
D4	void setup () estabelece as configurações permanentes (tamanho da tela, tamanho da linha, cores, etc), necessárias para a execução do void draw (). Este, por sua vez, estabelece o que realmente será desenhado/elaborado. Void mousePressed () determina o que acontece com o código caso o mouse seja pressionado.
D5	Void setup dita as configurações que serão necessárias no void draw, este que dita as configurações dos desenhos realizados na tela. O void mousePressed dita o que ocorre quando o mouse clica em determinado local da tela.
D6	void draw (): comando para desenhar a bolinha void setup (): comando para que o movimento da bolinha seja executado void mousePressed (): a bolinha apenas se movimenta se uma tecla do mouse for pressionada
D7	void draw(); - toma as próximas linhas como base para a criação de um desenho void setup() - toma as próximas linhas como base para definir as propriedades iniciais (tal como size ou background) void mousePressed(); - essa função começa a rodar assim que o mouse é pressionado
D8	void draw (); realiza a ação de desenhar void setup (); realiza ações de movimento void mousePressed () executa a ação quando o mouse é pressionado.
D9	Void draw: inicia as configurações da animação, refere-se aos desenhos, as formas, as cores (ao estilo). Void setup: define os padrões da animação, como ela vai funcionar, seus comandos. Void MousePressed: define o que acontecerá quando o usuário clicar com o mouse na tela.
D10	void draw() = repete o desenho enquanto o programa é rodado void setup() = aplica as definições iniciais do programa void mousePressed() = comando para a utilização do mouse pressionado para realizar comando
D11	Void draw () é a parte do programa que se definem as características fixas do programa, já o void setup () é a parte da animação propriamente dita. Void mousePressed () serve para definir o que acontece de animação no programa quando o mouse é pressionado.
D12	void draw: determinação da cor do fundo, do tamanho da elipse, grossura da linha da "raquete", e criação das variáveis e condições relativas ao movimento da bolinha e da raquete void setup: determinação do tamanho da imagem e da cor do fundo void mousePressed: inicia o jogo quando o mouse é clicado
D13	void draw () "desenha" o movimento imaginário da bolinha, movimentando-a em linha reta. void setup () permite os outros comandos void. void mousePressed () indica um comando para quando o mouse for clicado.

D14	void setup(); é utilizado para as configurações que não precisam se repetir (ex: tamanho da tela (size(x,y);). void draw(); é utilizado para que as configurações da bolinha continue se repetindo para que ela se mova. void mousePressed(); é utilizado para que se possa configurar alguma função para quando o mouse fosse pressionado.
D15	Não sei.
D16	void setup: Determina configurações como cor, tamanho, etc void draw: Repete o código dentro dele enquanto o programa estiver rodando void moudePressed: Executa os códigos dentro dele enquanto o mouse estiver pressionado
D17	"Void" serve para configurar códigos com movimento, mas não me recordo da função de cada um especificamente.

APENDICE G

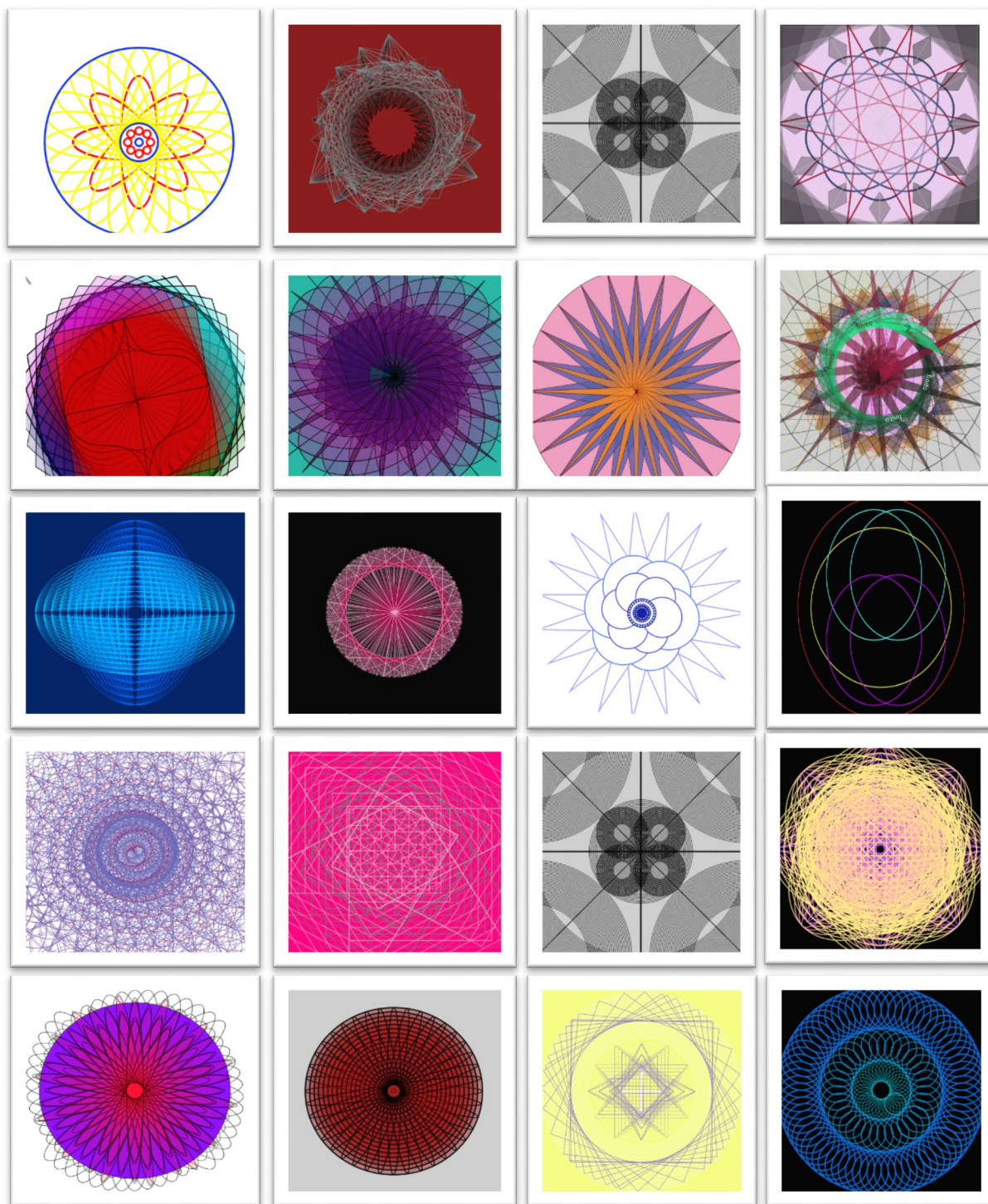
Cenários produzidos pelos alunos – 1º semestre 2015

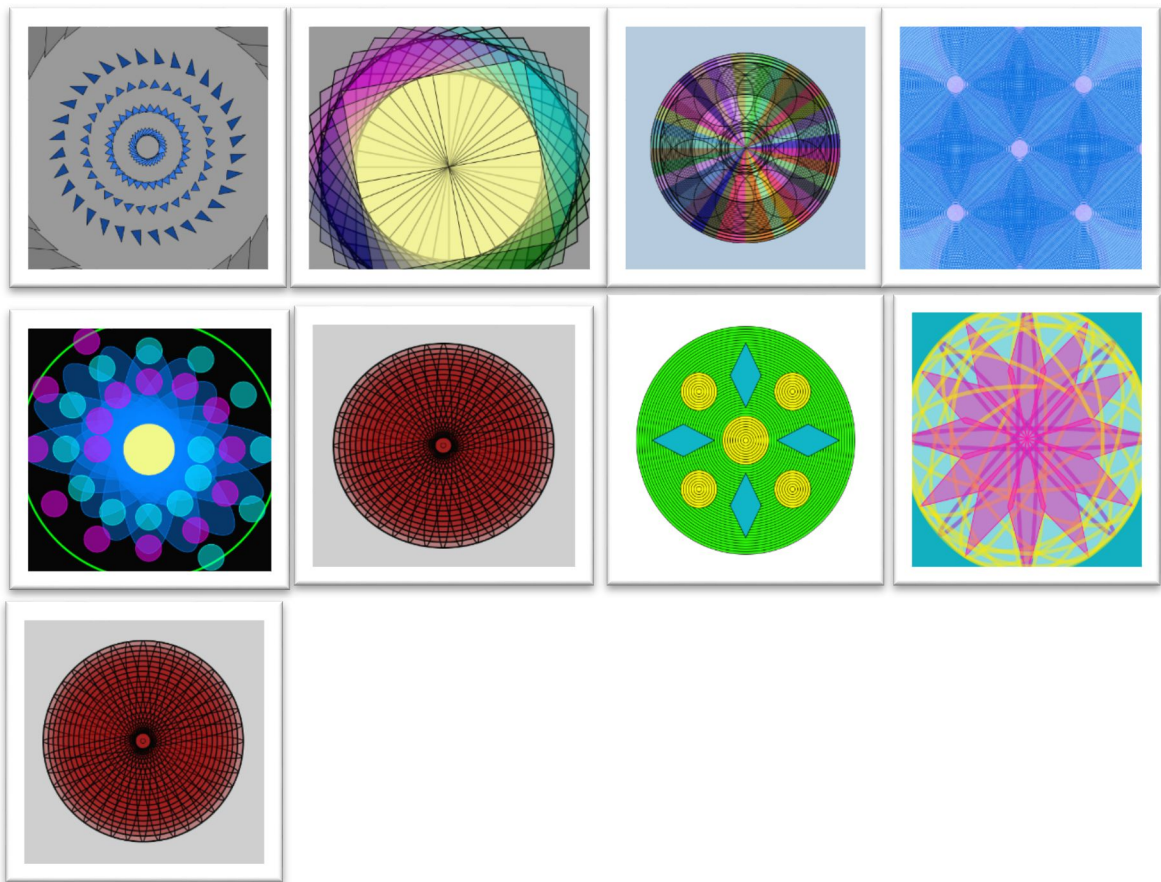




APENDICE H

Mandalas produzidas pelos alunos – 1º semestre 2015





APENDICE I

Questionário 02, elaborado no google Forms, através do link

(https://docs.google.com/forms/d/17eR76HIYG6xr3_m2Njo_lahfSdMX7As5qGsd922pj0w/edit)



*Obrigatório

1. a. Por favor, identifique-se: *

Escolher ▼

1.b. Caso Você não encontre sua identificação, digite abaixo seu nome completo.

Sua resposta

1.c. Informe seu email. *

Sua resposta

1.d. Informe seu telefone.

Sua resposta

2. Qual a origem do seu ensino médio? *

- ☐ Escola pública
- ☐ Escola particular

3. Você possui computador em casa? *

- ☐ Sim
- ☐ Não

4. Você gosta de matemática? *

- ☐ Sim
- ☐ Não

5. Você possui alguma dificuldade com a matemática? *

- ☐ Sim
- ☐ Não

6. Você já fez algum curso de informática? Qual? *

Sua resposta

7. Você possui smartphone? *

☐ Sim

☐ Não

8. Já teve a matéria lógica em alguma fase do seu período escolar? *

☐ Sim

☐ Não

9. Você conhece alguma das linguagens de programação abaixo? Qual? *

☐ Java

☐ Visual Basic

☐ C

☐ C++

☐ PHP

☐ Pure Data

☒ Nenhuma

10. Caso você conheça alguma linguagem de programação diferente das anteriores? Informe qual?

Sua resposta

11. Você sabe fazer página de internet em html. *

☐ Sim

☒ Não

12. Você sabe a diferença entre as duas formas de produção para web? *

☐ Sim

☐ Não

13. Você já criou algum programa para computador? *

☐ Sim

☐ Não

14. Você já criou algum aplicativo? *

☐ Sim

☐ Não

15. Você conhece o Processing? *

☐ Sim

☐ Não

16. Você conhece alguma linguagem de código aberto? *

☐ Sim

☐ Não

17. Qual a diferença entre matemática e lógica? *

Sua resposta

18. O que você entende por pensamento computacional? *

Sua resposta

19. Dê um exemplo de situação concreta em que você coloca em ação o pensamento computacional? *

Sua resposta

20. Qual a distinção entre pensamento computacional e raciocínio lógico? *

Sua resposta

21. Quando você encontra alguma dificuldade (em relação a matemática). O que acontece? *

☐ Desiste

☐ Tenta resolver o problema

☐ procura a ajuda de amigos

☐ procura a ajuda em redes sociais

22. Quando você tenta resolver um problema e procura ajuda nas redes sociais, quais das redes abaixo você usa com mais frequência. *

☐ Facebook

☐ Youtube

☐ Whatsap

☐ Instagram

☐ Twitter

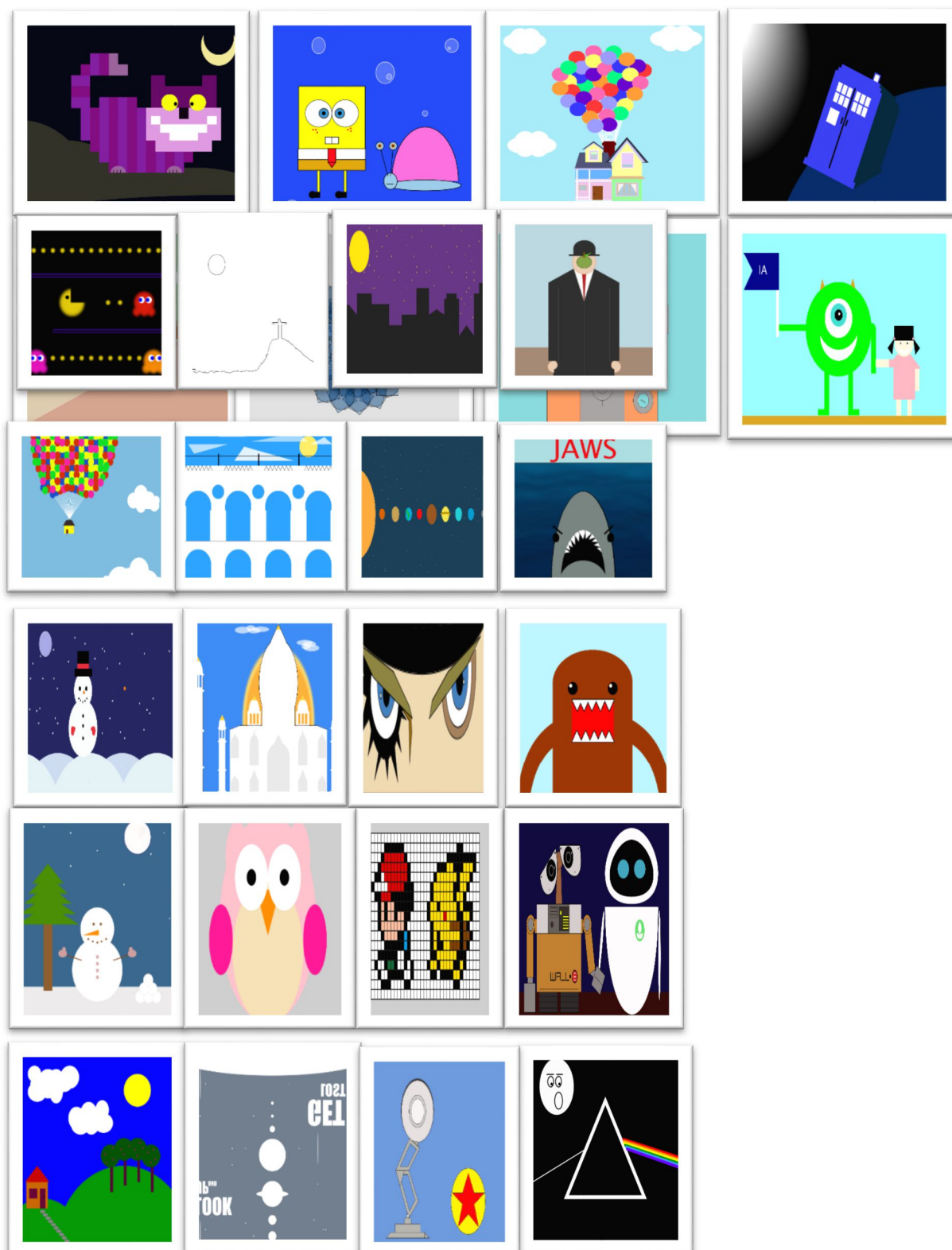
☐ Google+

☐ Outras

23. Depois de analisar o conteúdo programático da disciplina "Introdução ao Pensamento Computacional" o que você pode sugerir para acrescentar na disciplina? *

APENDICE J

Cenários produzidos pelos alunos – 1º semestre 2016



APENDICE L

Mandalas produzidas pelos alunos – 1º semestre 2016





APENDICE M

ATOR		PROJETOS	PONTOS POSITIVOS E NEGATIVOS
D1	CAS - CONSEGUIU APRENDER SOZINHO Encontrou diversão durante a aprendizagem PUP3 - CONSEGUI ALCANÇAR O OBJETIVO Pouco conhecimento em programação Utilizou os projetos existente como base para criar o seu FCP - FALTOU CONHECIMENTO EM PROGRAMAÇÃO	Jogo Rainbow Snake	PONTOS POSITIVOS Como ponto positivo considero o fato de ter aprendido nesse processo algumas funções e detalhes da linguagem matemática utilizada em programação. Esses detalhes são essenciais para qualquer programação meramente elaborada, portanto esse aprendizado vai me permitir criar novos projetos no <i>Processing</i> futuramente com menor dificuldade. Além disso, consegui sanar minhas dúvidas acerca da interação humano computador, mesmo que de forma simples, programei um jogo que me agrada e alcancei meu objetivo. Assim, posso dizer que, mesmo com todos os problemas já citados, o processo de criação foi bem divertido e cheio de aprendizados. Por fim, destaco a satisfação que senti ao ver algo que imaginei funcionando, o que considero o principal motivo para considerar o resultado final satisfatório. PONTOS NEGATIVOS Acredito que o principal ponto negativo nessa produção foi o fato de que meu conhecimento de programação é insuficiente e defasado; Isso me impossibilitou de fazer um jogo completamente do zero, como gostaria, me fez recorrer a códigos já prontos na tentativa de aprender algo a mais e restringiu a minha criatividade pois só pude fazer o pouco que era possível com as funções e códigos que compreendi. O tempo também foi um elemento primordial para as dificuldades encontradas. Devido à greve não tivemos as aulas necessárias para o desenvolvimento desse trabalho nem o devido tempo para desenvolvê-lo de forma autodidata com perfeição. Por isso, alguns detalhes que considero como erros no meu jogo (como a falta de um menu ou o fato da <i>Snake</i> poder passar sobre o título) não puderam ser consertados por falta de conhecimento da minha parte nessa linguagem.
D2	AP - APRIMOROU CONHECIMENTO	Manipulação de cores em mandala	PONTOS POSITIVOS Apesar do não desenvolvimento do jogo, como era querido inicialmente, na tentativa de cria-lo pude entender melhor a função <i>random</i>, o que provocou o surgimento da ideia para esse projeto. Além disso, aprimorei meus conhecimentos sobre o sistema RGB e sobre a função <i>for</i>. E apesar de não ter atingido minha ideia inicial (jogo), obtive sucesso no desenvolvimento desse projeto e o concluí de maneira satisfatória. PONTOS NEGATIVOS A pouca quantidade de aulas da disciplina CS107 comprometeu o resultado final, uma vez que antes de realizar esse projeto tentei desenvolver um jogo mas não obtive sucesso pela complexidade dos recursos que

			deveria utilizar. Se não fossem os imprevistos, a maior quantidade de aulas e, consequentemente, de matéria possibilitaria o desenvolvimento de um jogo bem elaborado e garantia uma maior satisfação pessoal com o projeto.
D3	CAS - CONSEGUIU APRENDER SOZINHO Encontrou diversão durante a aprendizagem PUP3 - CONSEGUI ALCANÇAR O OBJETIVO E se sente satisfeito com o resultado. Trabalhou com imagens FCP (faltou conhecimento em programação)	Unicórnio Fofo	PONTOS POSITIVOS O principal ponto positivo considero ter sido o fato de ter adquirido, durante esse processo, aprendizados sobre algumas funções e detalhes da linguagem matemática utilizada na programação. As ferramentas utilizadas no <i>Processing</i> são essenciais para qualquer tipo de programação meramente elaborada. Destaco também a satisfação que senti ao ver algo que imaginei e desenvolvi funcionando, o que considero o principal motivo para da minha satisfação final com o produto. Posso dizer que consegui ultrapassar as barreiras de minhas dificuldades já citadas anteriormente e fazer com que o processo de criação fosse bem divertido e inesperado. PONTOS NEGATIVOS O principal ponto negativo nessa produção foi o fato de meu conhecimento em programação ser insuficiente e defasado, o que me impossibilitou de fazer um jogo complexo e com mais funções, como gostaria. Isso restringiu a minha criatividade, pois só pude fazer o pouco que era possível com as funções e códigos que compreendi. O tempo pode ser dito também como uma grande barreira encontrada. Devido à greve pela qual passou a Unicamp no meu 1º semestre, não tivemos o número de aulas esperado, e por isso acabamos por não obter um amplo e desenvolvido conhecimento sobre o <i>Processing</i>. Além disso, a greve também dificultou a comunicação com os professores, tornando nossa experiência quanto a criação de um jogo um tanto autodidata. O intervalo entre a proposta do trabalho e a entrega foi muito curto, fazendo com que ambos, relatório e animação, fossem feitos o mais rápido possível e não apresentassem resultados tão bons quanto poderiam ter.
D4	Voltou a programar. Se divertiu fazendo o jogo. Aprendeu a utilizar o <i>Processing</i>. Trabalhou com imagens. Limitou -se a usar as funções disponíveis no software.	Clique se for capaz	PONTOS POSITIVOS Pude voltar a programar e me divertir fazendo um jogo, e vou me divertir ainda mais vendo meus amigos desavisados tentando vencer o jogo. Aprendi como utilizar o <i>Processing</i> para a criação de jogos. PONTOS NEGATIVOS O <i>Processing</i> tem algumas limitações e essas limitações me impediram ou me atrapalharam de fazer algumas coisas, como por exemplo, tive que modificar o formato de todas as imagens para .JPG, pois o <i>Processing</i> não rodava com imagens .PNG.

D5	Sentiu dificuldade para conseguir programar as funções. Trabalhou com vídeo	Ping Pong Faustão	PONTOS POSITIVOS Com esse desafio de elaborar um jogo com base em alguma programação já feita, aprendi a entender o código alheio e ver nele diversas possibilidades de aprimoramento. Também aprendi melhor para que serve as funções <i>if</i> e <i>else</i> e quando devo usa-las. PONTOS NEGATIVOS Apesar de termos tido algumas aulas de <i>Processing</i>, elas não foram suficientes para que eu conseguisse programar as funções que eu queria, então tive que recorrer diversas vezes ao <i>site</i> e pedir ajuda ao meu namorado para efetuar o comando devidamente. Além disso, não consegui fazer com que o jogo funcione sem fazer alguns comandos no <i>Processing</i> antes de executá-lo. Para que o jogo execute de maneira correta, é sempre necessário <i>linkar</i> o vídeo no fichário, pois caso não feito isso, não aparece no final se o jogador perder.
D6	Trabalhou com imagens Classe Triangulator.pde Possui conhecimento em programação e não teve dificuldade para criar o projeto.	Delaunay Filtro	Não descreveu pontos positivos e negativos. Fui motivada pelo meu interesse em matemática e computação, decidi fazer um filtro de Delaunay para transformar fotografias em imagens trianguladas para o meu trabalho final da disciplina CS107. Para isso, escolhi no <i>OpenProcessing</i> códigos e algoritmos que me ajudariam nessa tarefa. Como já possuía conhecimentos em programação, não encontrei muitas dificuldades na execução desse trabalho. Também não foi difícil encontrar os códigos que me auxiliaram no desenvolvimento no site <i>OpenProcessing</i>. Apesar de já saber programar e ter programado diversos tipos de programa em diversas linguagens, nunca tinha programado nada que envolvesse o processamento de imagens. Acabei concluindo que o <i>Processing</i> é uma ótima opção para projetos visuais utilizando imagens e fiquei satisfeita com o resultado do trabalho. Acredito que os conhecimentos adquiridos nesse projeto serão de grande utilidade ao longo do curso de Midialogia.
D7	Trabalhou com imagens Se familiarizou com os comandos do <i>Processing</i>.	Homem Vitruviano, de Leonardo da Vinci	Durante a realização deste trabalho, pude me familiarizar ainda mais com os comandos do <i>Processing</i>, além de ter aprendido que é possível, sim, realizar uma manipulação de fotografias através de seus comandos. Ao misturar elementos artísticos com matemáticos, através da fotografia e da programação, vivenciei, de maneira prática, mais um exemplo de convergência entre esses dois meios, que é algo que foi deveras frisado pelos professores em suas aulas durante o semestre, através de pesquisas, apresentações e o próprio uso que foi ensinado do <i>Processing</i>.
D8	Conclui seu projeto.	Jogo da cobra” (snake game)	Por fim, concluo que minha interpretação do código foi bem sucedida e isso possivelmente me ajudará a compreender próximas programações feitas pelo <i>OpenProcessing</i>. Algumas passagens eu não compreendi e procurei no site do programa os significados de algumas funções, mas no total foi o que eu esperava.

D9	Aprimorou habilidades.	Micro-animações inspiradas em estéticas artísticas: Pontilhismo, Impressionismo e Futurismo	PONTOS POSITIVOS Como pontos positivos, acredito que aprimorei minhas habilidades de programação e de interpretação de códigos em geral. Além disso, estabeleci uma relação interessante entre tecnologia e arte, reinterpretando características de pinturas através de um <i>software</i> moderno e que pode ser interpretado e visto por todos aqueles que tiverem interesse. PONTOS NEGATIVOS Como pontos negativos posso citar a falta de organização e antecipação para realizar o trabalho, o que causou atrasos e também fez com que eu não pudesse explorar ou tentar realizar atividades mais complexas que a que eu realizei. Ademais, também penso que as aulas teóricas não me forneceram uma base suficiente para realizar tal projeto. Alguns códigos eu ainda não havia estudado e tive que interpretá-los sozinho, o que acusou mais atraso no meu trabalho.
D10	A modificação deste código foi muito proveitosa para meu aprendizado.	Snake De Stijl	PONTOS NEGATIVOS A produção desse jogo teve seus empecilhos. Devido as conjunturas da universidade, a falta de aulas prejudicou bastante o projeto, tornando a sua realização do zero algo impossível. Fora isto, a falta de tempo devido ao acúmulo de diversos trabalhos finais de semestre em muito contribuiu para a simplicidade deste projeto. PONTOS POSITIVOS A modificação deste código foi muito proveitosa para meu aprendizado, principalmente em relação as funções void em todas suas categorias (<i>setup</i>, <i>draw</i>, <i>reset</i>). Me fez rever também diversos conceitos básicos, aprendidos no início das aulas que haviam sido esquecidos por mim. Além do mais, os conhecimentos do movimento estético De Stijl também foram de grande proveito meu, tenho certeza que as regras de proporção e cor poderão ser aplicadas em diversos produtos durante minha graduação.
D11	Projeto serviu como um bom aprendizado a respeito de como funciona o <i>Processing</i> .	Ping Pong	DISCUSSÃO O jogo apresenta bom funcionamento, como uma mecânica bem aplicada e a presença de um placar. O que poderia vir a ser visto como um ponto negativo, que é a simplicidade do jogo, era um dos objetivos a ser atingido, uma vez que as funções utilizadas foram trabalhadas em sala de aula. CONSIDERAÇÕES FINAIS Este projeto serviu como um bom aprendizado a respeito de como funciona o <i>Processing</i>. Apesar de ainda estar em um nível bem básico, considero que essa iniciação em programação será de grande valia ao longo de minha vida acadêmica. Fundamental agradecer a ajuda da aluna Paola Champi, e de seu namorado, que me instruíram sobre etapas em que estava tendo dificuldades, como inserir o "Game Over".

D12	Conseguir aprender e realizar inteiramente por conta própria a edição da fotografia, me permitiu conquistar uma autonomia com o uso do <i>Processing</i>. SDCP - SENTIU DIFICULDADE COM PROGRAMAÇÃO.	Posição de imagens	PONTOS POSITIVOS Conseguir aprender e realizar inteiramente por conta própria a edição da fotografia, me permitiu conquistar uma autonomia com o uso do <i>Processing</i>, e foi uma vitória que me deixou animada com a universidade, mesmo que esteja vivendo em um momento trágico da vida. Mesmo com a imensa defasagem que sinto ao trabalhar com a ferramenta, acredito que consegui obter resultados satisfatórios no meu trabalho final. PONTOS NEGATIVOS Senti uma enorme dificuldade no decorrer da realização do trabalho e da interpretação dos códigos, devido a uma enorme defasagem que talvez pudesse ter sido amenizada com mais aulas, e com a orientação dos professores da matéria, que acabei não procurando por motivos pessoais.
D13	Ponto positivo foi ter aprendido uma nova função ao programar com o <i>Processing</i>. Código muito complexos.	Keep Calm and Tickle the Doge” para “Keep Calm and Tickle Agostinho Carrara	PONTOS NEGATIVOS A minha surpresa por ter se deparado com um código muito complexo foi decisivamente um dos maiores pontos negativos na hora de realizar o jogo, pois não pude modificá-lo do jeito que eu gostaria. PONTOS POSITIVOS Apesar de ser contrastante, um dos pontos positivos foi a facilidade com a qual eu pude modificar o que eu tinha escolhido modificar, porque não me trouxe futuros problemas e também outro ponto positivo foi ter aprendido uma nova função ao programar com o <i>Processing</i>: a inserção de arquivos de imagens no código, que é bem útil em qualquer tipo de projeto a ser realizado no <i>Processing</i> e também foi um ponto positivo ter conhecido outras novas funções, por mais que eu não tenha conseguido aprender a usá-las ainda.
D14	Pude aprender novas funções no <i>Processing</i>, aperfeiçoei e revisei funções que aprendi em sala como o <i>void</i>, <i>void draw</i> e <i>void setup</i>, com isso sinto que minha relação com o programa está caminhando para um possível domínio das técnicas Falta de conhecimento em linguagem de programação	Império Contra Ataca	PONTOS POSITIVOS Com o desenvolvimento desse trabalho eu pude aprender novas funções no <i>Processing</i>, aperfeiçoei e revisei funções que aprendi em sala como o <i>void</i>, <i>void draw</i> e <i>void setup</i>, com isso sinto que minha relação com o programa está caminhando para um possível domínio das técnicas. PONTOS NEGATIVOS Ainda não domino a linguagem, estou melhor claro, mas com a conclusão do semestre imaginei que não teria tantos problemas em programar, esse trabalho me mostrou que ainda tinha e tenho várias dificuldades, muitas foram solucionadas ao longo do trabalho mas ainda existe vários outros.
D15	Apesar de meu pouco conhecimento prévio sobre programação, consegui cumprir a proposta desse trabalho devidamente, na medida em que, a partir do desenvolvimento desse jogo,	Ping Pong	Acredito que, apesar de meu pouco conhecimento prévio sobre programação, consegui cumprir a proposta desse trabalho devidamente, na medida em que, a partir do desenvolvimento desse jogo, aprendi ao menos o básico sobre programação e pude compreender um pouco mais sobre sua importância em nosso cotidiano. Aprendi a utilizar comandos mais complexos no <i>Processing</i>, tais

	aprendi ao menos o básico sobre programação e pude compreender um pouco mais sobre sua importância em nosso cotidiano Falta de conhecimento em linguagem de programação		como <i>boolean</i>, <i>float</i>, <i>void display</i>, <i>void move</i>, <i>void setLocation</i>, entre outros. Para tanto, creio que o maior ponto negativo ao fazer esse trabalho foi justamente minha inexperiência com programação; no entanto, esse também é um dos grandes pontos positivos, pois pude entender como funciona, basicamente, uma linguagem de programação. Porém, sem dúvidas, se eu tivesse buscado entender mais sobre a linguagem Java ao longo do curso e se houvesse mais tempo para realização desse trabalho, o resultado final seria mais satisfatório e elaborado, visto que optei por desenvolver um jogo simples e tradicional justamente porque o nível de dificuldade seria menor e estaria ao meu alcance. Por fim, realizar esse trabalho foi extremamente satisfatório, apesar das dificuldades, e me despertou interesse em entender melhor sobre linguagens de programação
D16	Foi muito interessante estudar o processo de formação de um jogo simples e ampliar os conhecimentos sobre o assunto. Também foi proveitoso testá-lo e melhorar todos os aspectos que julguei necessários.	Clique Contador	PONTOS POSITIVOS: Foi muito interessante estudar o processo de formação de um jogo simples e ampliar os conhecimentos sobre o assunto. Também foi proveitoso testá-lo e melhorar todos os aspectos que julguei necessários. PONTOS NEGATIVOS: Apesar do resultado, foi um desafio entender a linguagem utilizada para compor o jogo e desenvolver os códigos utilizados. O resultado foi bastante simples, talvez mais do que poderia ter sido.
D17	Se tornou de grande uso e envolveu um grande aprendizado em feitos que antes eram realizados por mim sem pensamento lógico e matemático.	Explore o Noya	Deste modo, a experiência com o <i>Processing</i>, mesmo que breve e dificultada por diversos fatores, se tornou de grande uso e envolveu um grande aprendizado em feitos que antes eram realizados por mim sem pensamento lógico e matemático.
D18	Conseguiu alcançar os objetivos foi relativamente fácil. Dúvidas em encontrar um exemplo para trabalhar.	Jogo da Velha	PONTOS POSITIVOS: Alcançar os meus objetivos foi relativamente fácil, pois escolhi partir de um jogo já existente e decidi realizar ações condizentes com as habilidades que possuo. Não precisei de muito tempo para ler e entender o código, traduzir as observações da autora e as mensagens no jogo e fazer alterações estéticas, o que é algo muito positivo. PONTOS NEGATIVOS: No site <i>OpenProcessing</i> encontrei muitas opções de jogos da velha e, por não ter gostado de nenhum logo de início e ter ficado em dúvida sobre qual escolher, gastei um pouco mais de tempo do que eu gostaria.
D19		Hold the Door	Não fez relatório
D20	Como ponto positivo eu considero o que eu aprendi durante a feitura do jogo, pois apesar das dificuldades com programação, eu pude expandir meu conhecimento no assunto falta de conhecimento em linguagem de programação.	Uma Homenagem a Star Wars	PONTOS NEGATIVOS: O ponto negativo do projeto foi a dificuldade que eu, particularmente, tenho com programação e entender essa linguagem computacional, apesar de estar feliz com meu projeto, gostaria de ter feito algo maior e talvez mais complexo. Mas ainda pretendo me reunir com o professor Hermes Renato Hildebrand e pedir ajuda em futuros

			projetos por gostaria muito de ampliar meu conhecimento nessa área. PONTOS POSITIVOS: Como ponto positivo eu considero o que eu aprendi durante a feitura do jogo, pois apesar das dificuldades com programação, eu pude expandir meu conhecimento no assunto. Outro ponto foi poder trabalhar com duas coisas que gosto muito, que são jogos e a série Star Wars.
D21	Através dos dias pude aprender inúmeras ações, são estas: produção de um produto, disciplina administrativa dos horários estipulados, design para os jogos e programação do <i>Processing</i>. Já possuía habilidade em programação. Falta de conhecimento em linguagem de programação.	Jogos Retrô: “Au Times”	PONTOS POSITIVOS: Através dos dias pude aprender inúmeras ações, são estas: produção de um produto, disciplina administrativa dos horários estipulados, design para os jogos e programação do <i>Processing</i>. Na produção de um produto, tive que me preocupar com o jogador. Para isso precisei pensar como um jogador (“Jogabilidade”, eficiência e simplicidade foram os meus alvos principais). Estipular cargas horárias e prazos desafiaram a disciplina e a responsabilidade para cumprir os horários e os prazos. A cobrança de mim mesma foi necessária. PONTOS NEGATIVOS: Existiram aspectos negativos durante a produção do jogo que são importantes de serem mencionados, pois estas experiências negativas poderão servir para futuros produtos. Não foi possível desenvolver todos os jogos desejados, dos quatro jogos apenas três tornaram-se reais. Durante o desenvolvimento, a junção dos jogos em um só programa mereceu muito cuidado e tempo para não comprometer nem os jogos e nem o programa que os unia, foi um procedimento muito estressante e dificultoso. A minha deficiência na linguagem em JAVA e no <i>Processing</i> serviu de limitador. Em muitos momentos tive empecilhos e bloqueios para complementar o programa.
D22	O aprendizado adquirido acerca das entranhas do funcionamento do computador. Ou seja, a execução desse trabalho fez com que percebesse que o conhecimento obtido fez, realmente, jus ao nome da matéria: Introdução ao Pensamento Computacional.	Recolha as Frutas Baseado no exemplo gotas de chuva	PONTOS POSITIVOS: O principal ponto positivo na realização desse trabalho foi o aprendizado adquirido acerca das entranhas do funcionamento do computador. Ou seja, a execução desse trabalho fez com que percebesse que o conhecimento obtido fez, realmente, jus ao nome da matéria: Introdução ao Pensamento Computacional. Desse modo, pude compreender como subsiste o funcionamento do computador, o fato de ser uma potente calculadora e ter conhecimento das funções mais simples de programação. Destarte, considero bastante válida a experiência vivenciada não só ao longo de toda a execução da matéria CS107, mas também na execução de um produto mais elaborado e por conta própria.
D23	Pude compreender alguns elementos matemáticos mais profundos com a alteração do código base	Jogo: Flappy Bird	PONTOS POSITIVOS: A criação do jogo, me ajudou muito quanto a rever conceitos iniciais do curso, assim como aprofundar em algumas seções. Especialmente na criação de <i>sprites</i> para o modo alternativo e na parte sonorização, pude

	Falta de conhecimento em linguagem de programação		obter uma maior profundidade do que aquela vista na sala de aula. Além disso, pude compreender alguns elementos matemáticos mais profundos com a alteração do código base. PONTOS NEGATIVOS: No geral, senti muita dificuldade ao redigir o projeto e fazer as respectivas alterações nele. Sinto que me faltou uma base consideravelmente enorme, e que não teria conseguido obter um produto final caso tivesse que iniciar o desenvolvimento a partir do que sabia. Nisto, gostaria de ter estudado mais sobre a parte programacional e ter tido uma base lógica maior, mas, reconheço que parte desta dificuldade veio em função das faltas de aulas e de uma orientação mais compreensível e direcionada.
D24	Não conseguiu terminar o projeto. Me considero satisfeito com o resultado final, pois consegui fazer com que o programa compilasse, apesar de todos os erros que ele apresentava inicialmente quando eu o escolhi para alterar e para apresenta-lo como trabalho final.	Quiz	Apesar de todas as modificações feitas, não consegui fazer que após o fim das 10 perguntas surgisse a tela final com o placar do jogador e com o resultado se o mesmo ganhou ou perdeu. Após a última pergunta o programa se fecha imediatamente. No entanto, tentando modificar para consertar o problema, o código deixava de compilar, então decidi deixa-lo assim e explicar no relatório o erro do código. Apesar deste erro, me considero satisfeito com o resultado final, pois consegui fazer com que o programa compilasse, apesar de todos os erros que ele apresentava inicialmente quando eu o escolhi para alterar e para apresenta-lo como trabalho final.
D25	Ponto positivo é a aquisição de conhecimento de programação. Falta de conhecimento em linguagem de programação.	Ping Pong	PONTOS POSITIVOS O primordial ponto positivo é a aquisição de conhecimento de programação. Praticamente desconhecia programação e admito que temi ao adentrar no projeto — felizmente, programei um jogo funcional de <i>Pong</i>. Houve, também, um exercício de lógica presente durante o ato de programar — vide o tópico “<i>Conteúdo matemático</i>” —, benéfico à saúde mental e o prazer do ato de criar. PONTOS NEGATIVOS Os aspectos negativos são de cunho estético que minha habilidade como programadora não conseguiu vencer. Desejei criar um menu, mas nem sabia por onde começar; os dizeres de vitória, encaminhados pelo código de Singhal (2014), no entanto, poderiam ser substituídos por imagens. Infelizmente, a inserção dessas imagens de finalização retardava incrivelmente a bola após a pontuação do quarto ponto. Além disso, elas impediam o término do jogo, pois o placar computou valores maiores que cinco.
D26	Pontos positivos posso citar um aproveitamento dessa matéria para meu aperfeiçoamento pessoal nessa plataforma e no desenvolvimento de jogos,	O Jogo mais Difícil Do Mundo	PONTOS POSITIVOS Como pontos positivos posso citar um aproveitamento dessa matéria pra meu aperfeiçoamento pessoal nessa plataforma e no desenvolvimento de jogos, algo que nunca tinha realizado em meios eletrônicos. Apesar de

	algo que nunca tinha realizado em meios eletrônicos. Apesar de apenas conhecer o básico da plataforma <i>Processing</i>, consegui criar um jogo com o que aprendi em sala de aula Falta de conhecimento em linguagem de programação		apenas conhecer o básico da plataforma <i>Processing</i>, consegui criar um jogo com o que aprendi em sala de aula. Além disso, esse projeto demonstrou que é possível reviver relações antigas em plataformas novas e que continuam em desenvolvimento. PONTOS NEGATIVOS Como pontos negativos posso citar uma demora na escolha do projeto a ser desenvolvido e, uma vez escolhido, como esse seria executado. Por não ter um domínio muito grande sobre a plataforma <i>Processing</i>, demorei a escolher o que seria executado porque queria algo relativamente fácil de ser desenvolvido, mas ao mesmo tempo legal e que poderia ser utilizado por um alto número de indivíduos. Fora isso, tive dificuldade em encontrar e entender alguns códigos necessários para esse projeto, problemas esses que foram solucionados com uma grande quantidade de tempo extra, o que atrasou a entrega do trabalho.
D27	O desenvolvimento do projeto foi divertido. Houve um grande aprendizado em todo o processo. Já possuía habilidade em programação. Falta de conhecimento em linguagem de programação.	Attack of the balloons	PONTOS NEGATIVOS Durante todo o processo de produção foram encontrados diversos obstáculos. Primeiramente não consegui codificar a minha ideia original, que seria fazer o jogador estourar os balões por meio de disparos. Passei muito tempo tentando aprender um método de fazer isso dar certo, mas não entendia muitos dos códigos que encontrava, por tanto, tive que adaptar um pouco o funcionamento do <i>game</i>. Por mais que já tivesse um contato prévio com programação, ainda senti dificuldade em codificar minhas ideias utilizando o <i>Processing</i>. PONTOS POSITIVOS Apesar do pouco tempo disponível, e demais empecilhos, houve também pontos positivos na elaboração desse <i>game</i>. Como tenho grande fascínio desde a infância por esse universo, é sempre um prazer realizar qualquer atividade que tenha relação com isso. Assim, posso dizer que foi um projeto bem divertido de ser elaborado, tanto na parte de criação, quanto ver ela pronta, e funcionando. É sempre muito satisfatório ver uma criação sair do plano das ideias e ser posta em prática, tendo um resultado positivo no final. Como sempre, também houve um grande aprendizado em todo o processo. Desde a conclusão do meu TCC durante o ensino técnico, não tive mais um contato com projetos voltados para programação, e o desenvolvimento desse <i>game</i> fez com que eu relembrasse conceitos já esquecidos, e aprendesse alguns novos, além de criar um maior aprendizado sobre o <i>Processing</i>, que pode vir a ser útil para novos projetos seja durante a graduação, ou depois da mesma.