



UNIVERSIDADE DE BRASÍLIA  
INSTITUTO DE ARTES  
DEPARTAMENTO DE ARTES VISUAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM ARTES VISUAIS

TEÓFILO AUGUSTO DA SILVA

# Arte Computacional: O que reter, como reter? O que guardar, como guardar?

Brasília  
2022

TEÓFILO AUGUSTO DA SILVA

# Arte Computacional: O que reter, como reter? O que guardar, como guardar?

Tese apresentada ao Programa de Pós-Graduação em Artes Visuais do Instituto de Artes da Universidade de Brasília, para obtenção do título de Doutor em Artes Visuais.

**Área de Concentração:** Métodos, Processos e Linguagens

**Linha de Pesquisa:** Arte e Tecnologia (AT)

**Orientadora:** Profa. Dra. Suzete Venturelli

Brasília  
2022

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) e da Universidade Federal do Sul e Sudeste do Pará (Unifesspa) por meio do Programa PRODOUTORAL (Edital 04/2019 – PROPIT Unifesspa).

O autor também foi aprovado na Bolsa CAPES/DAAD para doutorado sanduíche na *Universität der Künste – Berlin (UdK)* sob orientação do Prof. Dr. Alberto de Campo.

Ficha catalográfica elaborada automaticamente,  
com os dados fornecidos pelo(a) autor(a)

da Silva, Teófilo Augusto  
dT314a      Arte Computacional: O que reter, como reter? O que  
guardar, como guardar? / Teófilo Augusto da Silva;  
orientador Suzete Venturelli. -- Brasília, 2022.  
195 p.

Tese (Doutorado - Doutorado em Arte) -- Universidade de  
Brasília, 2022.

1. Arte Computacional. 2. Protocolo. 3. Preservação. 4.  
Conservação. 5. Mídias Instáveis. I. Venturelli, Suzete,  
orient. II. Título.

Dedico este trabalho à minha família que sempre me surpreende e aquece meu coração, me permitindo errar e crescer, contudo mais importante, me faz querer ser uma pessoa melhor. Em especial, quero dedicar este trabalho à minha esposa, Denise Rodrigues Lima, que pode ser quase uma coautora desta pesquisa, pelo apoio na leitura do texto, no suporte emocional e no companheirismo do dia a dia. Te amo!

No processo de escrita deste texto, nasceu em 30 de janeiro de 2021, o meu filho Apollo Messner da Silva e Lima. Dedico este trabalho, também, para mostrar a ele que mesmo em pequenas quantidades, todo ser humano, tem a obrigação de contribuir para o futuro de todos. Te amo meu filho.

Ao retornar para o Pará, tive uma perda muito grande com o falecimento de uma de minhas filhas de patas. Pretinha você sempre estará em meu coração.

## **Agradecimentos**

Este espaço é muito importante e eu tenho tantas pessoas a agradecer que quase poderia escrever uma nova tese. Esse percurso começa com o apoio inegável da Profa. Suzete Venturelli, que é uma mentora, uma amiga colossal e uma mãe de coração. Estendo este carinho também ao Prof. Dr. Alberto de Campo, que me orientou na pesquisa durante o Doutorado Sanduíche, em Berlim (estendo também os agradecimentos a adorável esposa Dra Marianne, que além de uma pessoa muito culta é uma cozinheira de mão-cheia). Ao Prof. Cleomar Rocha, por ter acreditado no meu entusiasmo de recém-empossado como professor, proporcionando voltar para Marabá com um projeto de vida no estudo da arte e tecnologia, um logo e um título para o laboratório e me premiou ainda, com uma amizade natural. À Profa. Dra. Anita Jori, que me recebeu no Flusser Archiv da UdK. À conservadora do ZKM, Morgane Stricot, que foi muito receptiva e uma excelente anfitriã. À conservadora do Museu Reina Sofia de Madrid, Dra. Arianne Vanrell Vellosillo, por uma tarde maravilhosa conversando sobre sua tese que desde o início esteve em minhas referências. Ao Prof. Dr. Lino García Morales, por sempre responder rapidamente às minhas dúvidas e colaborar demais com a base teórica desta tese. Ao Prof. Alexandre “Alix” Silva dos Santos Filho, grande incentivador da fundação do Media Lab/Unifesspa. Para a Profa. Tânia Moreira, por ter sido minha primeira vice-coordenadora do laboratório enfrentando os primeiros obstáculos. Um grande agradecimento ao meu amigo, companheiro de pesquisa e co-coordenador do Media Lab/Unifesspa Prof. Claudio Maciel Coutinho que sempre me presenteia com conselhos maravilhosos trazidos num português impecável e com um humor temperado de ironia e otimismo. Aos colegas do Media Lab/UnB por terem sido tão pacientes e parceiros em várias obras: Artur Cabral (em especial por sua contribuição com o texto), Phralade Hargreaves e Lorena Ferreira. Ao amigo Alexandre Rangel, que contribuiu para que eu me compreendesse como artista. Aos amigos Prof. Dirlenvalder Loyolla e Prof. Dennis Sávio por me ajudarem lendo e opinando no texto. Diante de tudo que passamos eu e minha família quando no período fora do Brasil, uma pessoa se destacou demais, Bruno Monteiro de Melo, meu querido amigo estava presente para nós nos momentos mais angustiantes que um estrangeiro pode vivenciar. Obrigado demais! Quero agradecer, também, aos professores do Programa de Pós-Graduação em Artes Visuais da Universidade de Brasília, aos meus colegas professores da Faculdade de Artes Visuais e os Técnicos da Universidade Federal do Sul e Sudeste do Pará que estão lá sustentando e trabalhando pelo curso enquanto seus colegas saem para a capacitação.



## **Kunst Muss Nicht. (RW.12)**

ZIELINSKI, Siegfried. (2012)

52.49097999437469, 13.358781588738387 (Geolocalização do Medienhaus).

# Resumo

Esta tese visa contribuir para a conservação da intenção estética das obras computacionais. As obras artísticas computacionais representam um limiar entre fazeres artísticos e científicos que representam uma forma diferente de refletir sobre o mundo e sobre os produtos culturais denominados tecnologias. Para a sociedade atual em que os processos computacionais estão cada vez mais hibridizados, com interfaces cada vez mais “naturais”, a obra de arte computacional, fruto da tradução de algoritmos em uma linguagem de máquina é ponto de reflexão sobre como nós seres humanos nos relacionamos com estes códigos e traduções entre nosso querer (intenção) e os resultados sensoriais (obras de arte computacionais). Os artistas que se utilizam das Linguagens de Programação para a produção de sua obra devem se atentar para a natureza da tecnologia que sempre caminha para a sua obsolescência com objetivo de evoluir para uma nova versão de si, isso nos direciona a uma problemática na preservação histórica da obra de arte computacional que até então tem tido como resultado a perda de muitas destas obras que não mais podem ser executadas em computadores atuais e cujo hardwares e softwares originais não mais são encontrados. Ao artista cabe preocupar-se não com a manutenção tecnológica, mas com a preservação da instrução poética, ou seja, devemos encontrar uma forma de deixar claro as intenções estéticas dos artistas com suas criações. Esta tese apresenta uma sugestão de um protocolo de conservação estética de obras computacionais a partir de reflexões históricas e ontológicas que demonstram a relação entre as tecnologias e a mensagem que cada obra intenciona proporcionar. Para tanto ela está dividida em duas partes: A primeira denominada “O que reter? O que guardar?” apresenta os três primeiros capítulos (“Uma história da Arte Computacional”, “A Arte e as Tecnologias da Informação e da Computação” e “Uma ontologia da Arte Computacional”) e tem por objetivo refletir sobre os processos formadores da Arte Computacional e a essência da mesma para entendermos o que deve ser preservado. Já a segunda parte “Como reter? Como guardar?” com o quarto capítulo (“Experiências em Conservação de Obras em Mídias Instáveis”) e quinto capítulo (“Protocolo de Conservação de Obra de Arte Computacional”), discorrem sobre as experiências práticas de conservação, as teorias específicas para esta problemática e a apresentação do modelo C7Art, que recebe este nome por ser derivado do Modelo C4 mas utilizar um mínimo de sete níveis e ter seu uso destinado aos objetos artísticos. No quinto capítulo ainda são apresentados os testes realizados com o protocolo nas obras EVO\_CIRCUITO (Media Lab/UnB) e FUTURE VOICES (SN4TP/UdK).

**Palavras-chave:** Arte Computacional, Protocolo, Preservação, Conservação, Mídias Instáveis.

# Abstract

*This thesis aims to contribute for the conservation of the aesthetic intention of computational artworks. Computational artworks represent a threshold between artistic and scientific activities and that represents a different way of reflecting on the world and on cultural products called technologies. For today's society in which computational processes are increasingly hybridized, with increasingly "natural" interfaces, the computational artworks are the result of the translation of algorithms into a machine language and this is a point of reflection on how we human beings we relate to these codes and translations between our will (intention) and sensory results (computational works of art). Artists who use Programming Languages in their artworks must pay attention to the nature of technology that always walks towards its obsolescence in order to evolve into a new version of itself, this leads us to a problem in historical preservation of the computational work of art that until then has resulted in the loss of many of these works that can no longer be executed on current computers and whose original hardware and software are no longer found. It is up to the artist to be concerned not with technological maintenance, but with the preservation of poetic instruction, that is, we must find a way to make clear the aesthetic intentions of artists with their creations. This thesis presents a suggestion of a protocol for the aesthetic conservation of computational works from historical and ontological reflections that demonstrate the relationship between technologies and the message that each work intends to provide. Therefore, it is divided into two parts: The first one called "What to retain? What to keep?" presents the first three chapters ("A history of Computational Art", "The Art and Technologies of Information and Computing" and "An ontology of Computational Art") and aims to reflect on the formative processes of Computational Art and the essence of it, to understand what should be preserved. The second part "How to retain? How to save?" with the fourth chapter ("Experiences in Conservation of Works in Unstable Media") and fifth chapter ("Protocol for the Conservation of Computer Art Work"), they discuss practical conservation experiences, specific theories for this problem and the presentation of the C7Art model, which receives this name because it is derived from Model C4 but uses a minimum of seven levels and is intended for use in artistic objects. The fifth chapter also presents the tests carried out with the protocol in the EVO\_CIRCUITO (Media Lab/UnB) and FUTURE VOICES (SN4TP/UdK) works.*

**Keywords:** *Computational Art, Protocol, Preservation, Conservation, Unstable Media.*

# Sumário

RESUMO .....	8
ABSTRACT .....	9
SUMÁRIO .....	10
LISTA DE IMAGENS .....	12
LISTA DE TABELAS .....	16
INTRODUÇÃO .....	17
PARTE 1: O QUE RETER? O QUE GUARDAR? .....	26
<b>1. PRIMEIRO CAPÍTULO: UMA HISTÓRIA DA ARTE COMPUTACIONAL .....</b>	<b>27</b>
<b>1.1 A REVOLUÇÃO COMPUTACIONAL SOCIOCULTURAL .....</b>	<b>27</b>
1.1.1. <i>Computadores e Máquinas</i> .....	29
1.1.2. <i>O Computador Moderno</i> .....	32
1.1.3. <i>De transistores para chips</i> .....	34
<b>1.2. O ALGORITMO E AS LINGUAGENS DE PROGRAMAÇÃO .....</b>	<b>35</b>
1.2.1. <i>Primórdios da Linguagem de Programação</i> .....	38
1.2.2. <i>Programas Armazenados e como resolver problemas no código</i> .....	40
1.2.3. <i>Programação como ciência, computação como hobby</i> .....	42
1.2.4. <i>Conexão, Internet e Multimídia: o computador no centro de tudo</i> .....	45
<b>1.3. OS LAÇOS DA HUMANIDADE .....</b>	<b>47</b>
1.3.1. <i>Internet e a onipresença digital</i> .....	48
<b>2. SEGUNDO CAPÍTULO: A ARTE E AS TECNOLOGIAS DA INFORMAÇÃO E DA</b>	
<b>COMPUTAÇÃO .....</b>	<b>52</b>
2.1. <i>CYBERNETIC SERENDIPITY: O EQUILÍBRIO DA RIGIDEZ DAS CIÊNCIAS COM A FLUIDEZ DAS ARTES</i> .....	55
2.2. <i>WALDEMAR CORDEIRO E GIORGIO MOSCATI: AVANT-GARDE DA ARTE-CIÊNCIA NO BRASIL</i> .....	59
2.3. <i>DESDOBRAMENTOS</i> .....	63
2.4. <i>SEGUNDA GERAÇÃO FINAL DE 1970 ATÉ O SÉCULO XXI: FORMAÇÃO ACADÊMICA, LABORATÓRIOS E INSTITUTOS DE PESQUISA E CONSERVAÇÃO</i> .....	66
2.5. <i>A PRÓXIMA GERAÇÃO</i> .....	72
<b>3. TERCEIRO CAPÍTULO: UMA ONTOLOGIA DA ARTE COMPUTACIONAL .....</b>	<b>82</b>
<b>3.1. PRINCÍPIOS DE ONTOLOGIA DA ARTE COMPUTACIONAL .....</b>	<b>82</b>
3.1.1. <i>O Computador como objeto estético</i> .....	93
3.1.2. <i>Sobre o código como objeto estético</i> .....	96
PARTE 2: COMO RETER? COMO GUARDAR? .....	103
<b>4. QUARTO CAPÍTULO: EXPERIÊNCIAS EM CONSERVAÇÃO DE OBRAS EM MÍDIAS</b>	
<b>INSTÁVEIS .....</b>	<b>104</b>
4.1. <i>TEORIA DA EVOLUÇÃO CONSERVATIVA</i> .....	104
4.2. <i>ESTRATÉGIAS UTILIZADAS: ANÁLISE DE CASOS</i> .....	106
4.2.1. <i>Substituição</i> .....	108
4.2.2. <i>Migração</i> .....	109
4.2.3. <i>Emulação, Virtualização, Simulação e Containerização</i> .....	110
4.2.4. <i>Reinterpretação</i> .....	111
4.2.5. <i>(Re)Criação</i> .....	113

4.3.	<b>BOLSA DOUTORADO-SANDUÍCHE NA UDK – BERLIN</b>	113
4.3.1.	<i>Relato de Visita Técnica à ZKM</i>	115
4.3.2.	<i>Relato de entrevista com pesquisadores</i>	121
4.3.3.	<i>Participação na disciplina de Arte Generativa</i>	123
5.	<b>QUINTO CAPÍTULO: PROTOCOLO DE CONSERVAÇÃO DE OBRA DE ARTE COMPUTACIONAL</b>	124
5.1.	<b>MODELO C7ART</b>	124
5.1.1.	<i>Princípios direcionadores</i>	124
5.1.2.	<i>Modelo C4</i>	127
5.1.3.	<i>Adaptando os modelos</i>	128
5.1.4.	<i>Diagrama de 7 níveis</i>	129
5.1.5.	<i>Documentação</i>	130
5.1.6.	<i>Etapas do protocolo para conservação estética</i>	133
5.2.	<b>ESTUDO DE CASO EM CRIAÇÃO ESTÉTICA E DOCUMENTAÇÃO E APLICAÇÃO DO PROTOCOLO</b>	137
5.2.1.	<i>EVO_CIRCUITO (Maio de 2018)</i>	137
5.2.2.	<i>FUTURE VOICES</i>	165
	<b>CONSIDERAÇÕES FINAIS</b>	181
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	183
	<b>REFERÊNCIAS MULTIMÍDIA</b>	187
	<b>APÊNDICES</b>	189
	<b>APÊNDICE I. VERSONIFICAÇÃO DE OBRA DE ARTE COMPUTACIONAL</b>	190
	<b>APÊNDICE II. LEGENDA DE FLUXOGRAMA DE PROCESSOS</b>	194
	<b>APÊNDICE III. EXEMPLOS DE DOCUMENTOS PARA AS RESPOSTAS I E II DO QUESTIONÁRIO DA FASE I DO PROTOCOLO</b>	195

## Lista de Imagens

<b>Figura 1:</b> Protótipo da Máquina Diferencial. Fonte: London Science Museum.....	30
<b>Figura 2:</b> Tear de Jacquard exposto no Museu Alemão de Tecnologia. Fonte: Acervo próprio. Foto apreendida no dia 16 de janeiro de 2022. ....	32
<b>Figura 3:</b> Gabarito de fluxograma para programação. Fonte: Acervo próprio. ....	41
<b>Figura 4:</b> Imagem em ASCII gerada a partir do site <a href="http://www.text-image.com">www.text-image.com</a> .....	45
<b>Figura 5:</b> Ampliação do M da imagem anterior.....	45
<b>Figura 6:</b> Diagrama da disposição dos nós da ARPANET em 1975. Fonte: CAMPBELL-KELLY et al., 2018, p. 201. ....	50
<b>Figura 7:</b> Imagem colocada junto ao sumário do Catálogo da Exposição. Fonte: MASON, 2018.....	55
<b>Figura 8:</b> CORDEIRO, Waldemar; MOSCATI, Giorgio, BEABÁ, 1968. Software, cartões perfurados. (Kassab, 2009).....	61
<b>Figura 9:</b> Waldemar Cordeiro e Giorgio Moscati. Da esquerda para a direita, 'Transformação em grau zero', 'Transformação em grau 1' e 'Transformação em grau 2', 1969   Foto: Reprodução (Giorgio Moscati e Waldemar Cordeiro: quando o computador encontrou a Arte, 2011.....	62
<b>Figura 10:</b> Laboratórios do Media Lab - MIT. Fonte: Media Lab MIT (n.d.).....	74
<b>Figura 11:</b> Notas no caderno de Reas. Imagem disponibilizada no artigo "A Moderne Prometheus". (2018).....	74
<b>Figura 12:</b> "Hello World" codificado em JAVA. Fonte: REAS; FRY, 2018.....	75
<b>Figura 13:</b> "Hello World" codificado em Processing Fonte: REAS; FRY, 2018. ....	76
<b>Figura 14:</b> Retângulo cinza sobre fundo branco. Fonte: Acervo próprio (2020).....	76
<b>Figura 15:</b> Captura de tela do framework Processing com o código da figura anterior e seu resultado. ....	77
<b>Figura 16:</b> Aplicação em WPF do Visual Studio em C# e XAML. Fonte: Acervo próprio. (2020). ....	78
<b>Figura 17:</b> <i>Código do XAML descrevendo o retângulo e suas características.</i> Fonte: Acervo próprio (2020). ....	78
<b>Figura 18:</b> Exemplo de um programa feito no MAX. Fonte: Cycling74, n.d.....	79
<b>Figura 19:</b> Diagrama das instruções do computador. Acervo do autor. ....	84
<b>Figura 20:</b> VILAR, Pilar Montero; GARCÍA, Lino. THE CHALLENGES OF DIGITAL ART PRESERVATION. E-Conservation, n. May, 2018. p. 2. ....	88

<b>Figura 21:</b> Um still de Ent- (2021) HEANEY, Libby. Fonte: DAMN magazine. Disponível em: <a href="https://www.damnmagazine.net/calendar/libby-heaneys-ent/">https://www.damnmagazine.net/calendar/libby-heaneys-ent/</a> . Acesso em: 09/05/2022.....	89
<b>Figura 22:</b> MORALES, Lino García. Teoría de La Conservación Evolutiva: conservación y restauración del arte de los nuevos medios. 1. ed. Online: BoD - Books on Demand, 2019. Disponível em: <a href="http://www.bod.com.es">www.bod.com.es</a> p. 28. ....	90
<b>Figura 23:</b> Still da saída visual do Hydra após a inserção de apenas uma única linha de código. Fonte: Site do Hydra. Código utilizado como exemplo na documentação..	98
<b>Figura 24:</b> Still da saída visual do Hydra após a modificação dos parâmetros da linha de código. Fonte: Site do Hydra. Código utilizado como exemplo na documentação..	99
<b>Figura 25:</b> Still da obra Flocking (lado esquerdo) no site <a href="http://code-poetry.com">code-poetry.com</a> . HOLDEN & KERR (s.d.). ....	100
<b>Figura 26:</b> Still da obra Flocking (lado direito) no site <a href="http://code-poetry.com">code-poetry.com</a> . HOLDEN & KERR (s.d.). ....	100
<b>Figura 27:</b> Diagrama das Morfologias Comparativas entre obras. Fonte: Morales, Lino. (2019), p. 56 .....	105
<b>Figura 28:</b> Um dos diversos espaços de armazenagem para materiais de reposição para as obras do acervo permanente da ZKM. Fonte: Acervo próprio, 31 janeiro 2022. ....	109
<b>Figura 29:</b> Still do videoguia em libras, disponibilizado pelo Itaú Cultural em seu canal do YouTube, sobre a obra BEABÁ. Fonte: Canal Itaú Cultural YouTube. Disponível em: << <a href="https://www.youtube.com/watch?v=pQ90gVX7cqY&amp;t=107s">https://www.youtube.com/watch?v=pQ90gVX7cqY&amp;t=107s</a> >>. Acesso em: 21 maio 2022.....	110
<b>Figura 30:</b> Ambiente no terceiro andar do prédio com espaços destinados à manutenção das obras que ocorriam no andar térreo. <b>Fonte:</b> Acervo próprio (2022). ....	117
<b>Figura 31:</b> Imagem da capa do manual da obra Billow V de Daniel Canogar. Fonte: Fornecido o arquivo digital pelo sr. Diego Mellado.....	118
<b>Figura 32:</b> Still do vídeo da 23ª Jornada de Conservação de Arte Contemporânea do Museu Reina Sofia demonstrando o movimento que a pessoa que estiver instalando a obra deve fazer com a mão para conectar a fiação. <b>Fonte:</b> YouTube. Disponível em: << <a href="https://www.youtube.com/watch?v=XvpM1OQrA6k">https://www.youtube.com/watch?v=XvpM1OQrA6k</a> >>. Acesso em: 25 maio 2022. Local do still no vídeo: 2h46'45".....	119
<b>Figura 33:</b> Os sete níveis básicos do protocolo C7Art. Fonte: Acervo próprio.....	129
<b>Figura 34:</b> Um dos Diagramas de Fluxo originais de Gordon Pask. Fonte: (PASK, 1971). ....	131

<b>Figura 35:</b> Instruções poéticas a serem observadas ao início da diagramação. Fonte: Acervo próprio, 2022.....	134
<b>Figura 36:</b> <i>Sugestão de iconografias a serem utilizadas nos diagramas.</i> Fonte: Acervo próprio. ....	136
<b>Figura 37:</b> EVO_CIRCUITO montado em Buenos Aires durante o Simpósio Internacional de Mídias Digitais Interativas. Fonte: Acervo próprio (2018). ....	138
<b>Figura 38:</b> Parte interna do corpo do EVO_CIRCUITO, parte superior. Podemos visualizar as trilhas compostas de papel alumínio. Fonte: Acervo próprio.....	138
<b>Figura 39:</b> Parte interna do corpo do EVO_CIRCUITO, parte inferior. Esta parte do corpo fica exposta verticalmente em uma superfície lisa. O motor teve que compensar o peso extra da peça pela gravidade. Fonte: Acervo próprio. ....	139
<b>Figura 40:</b> <i>Prahlada Hargreaves testando as reações das tintas condutivas em tela de pintura.</i> Fonte: Acervo próprio. (2018) .....	139
<b>Figura 41:</b> Diagrama do inicial do corpo do EVO_CIRCUITO (HARGERAVES, 2018). Fonte: Acervo próprio. ....	140
<b>Figura 42:</b> Os quatro PAD's utilizados no EVO_CIRCUITO. Fonte: Acervo próprio. ....	142
<b>Figura 43:</b> Diagrama EVO_CIRCUITO nível L1. Fonte: Acervo próprio. (2022) .....	146
<b>Figura 44:</b> Diagrama EVO_CIRCUITO nível L2. Fonte: Acervo próprio (2022) .....	148
<b>Figura 45:</b> Diagrama EVO_CIRCUITO nível L4. Fonte: Acervo próprio (2022) .....	149
<b>Figura 46:</b> Diagrama EVO_CIRCUITO nível L5. Fonte: Acervo próprio (2022) .....	151
<b>Figura 47:</b> Diagrama EVO_CIRCUITO nível L6. Fonte: Acervo próprio (2022) .....	152
<b>Figura 48:</b> Diagrama EVO_CIRCUITO nível L7a. Fonte: Acervo próprio (2022) .....	154
<b>Figura 49:</b> Diagrama EVO_CIRCUITO nível L7a1. Fonte: Acervo próprio (2022) ....	155
<b>Figura 50:</b> Diagrama EVO_CIRCUITO nível L7a2. Fonte: Acervo próprio (2022) ....	157
<b>Figura 51:</b> Diagrama EVO_CIRCUITO nível L7a3. Fonte: Acervo próprio (2022). ...	158
<b>Figura 52:</b> Diagrama EVO_CIRCUITO nível L7b. Fonte: Acervo próprio (2022) .....	160
<b>Figura 53:</b> Diagrama EVO_CIRCUITO nível L7b1. Fonte: Acervo próprio (2022) ....	161
<b>Figura 54:</b> Diagrama EVO_CIRCUITO nível L7b2. Fonte: Acervo próprio (2022) ....	162
<b>Figura 55:</b> Diagrama EVO_CIRCUITO nível L7b3. Fonte: Acervo próprio (2022) ....	163
<b>Figura 56:</b> Diagrama EVO_CIRCUITO nível L7b4. Fonte: Acervo próprio (2022) ....	164
<b>Figura 57:</b> Diagrama Future Voices nível L1. Fonte: Acervo próprio. (2022). ....	168
<b>Figura 58:</b> Diagrama Future Voices nível L2. Fonte: Acervo próprio (2022) .....	170
<b>Figura 59:</b> Diagrama Future Voices nível L4. Fonte: Acervo próprio (2022) .....	171
<b>Figura 60:</b> Diagrama Future Voices nível L6. Fonte: Acervo próprio (2022). ....	173
<b>Figura 61:</b> Diagrama Future Voices nível L7a - relativo ao código _00loadMe_FV.scd. Fonte: Acervo próprio (2022). ....	174

<b>Figura 62:</b> Diagrama Future Voices nível L7b - relativo ao código director_FV.scd. Fonte: Acervo próprio (2022). .....	176
<b>Figura 63:</b> Diagrama Future Voices nível L7b1 - relativo ao código director_FV.scd - Global Function \dateSeed. Fonte: Acervo próprio (2022). .....	177
<b>Figura 64:</b> Diagrama Future Voices nível L7b2 - relativo ao código director_FV.scd - Global Function \adjustTime. Fonte: Acervo próprio (2022). .....	178
<b>Figura 65:</b> Diagrama Future Voices nível L7b3 - relativo ao código director_FV.scd - Global Function \hourClock. Fonte: Acervo próprio (2022).....	179

## Lista de Tabelas

**Tabela 1:** Gerações da Arte Computacional. Fonte: Acervo da Pesquisa. .... 54

**Tabela 2:** Etapas de implantação do protocolo. Fonte: Arquivo próprio. .... 145

**Tabela 3:** Etapas de implantação do protocolo. Fonte: Arquivo próprio..... 167

## Introdução

Nos últimos setenta anos a criação artística e os espaços de fruição estabeleceram uma ligação íntima com todas as tecnologias da computação e da comunicação disponíveis. Experiências artísticas foram produzidas junto aos primeiros computadores e estas levaram a criação de novas tecnologias que resultaram em novas experiências num ciclo infundável.

Os teóricos, os curadores, os conservadores, os museus e galerias, enfim todo o campo das artes, aos poucos tiveram que se adaptar a esta nova expressão que se transformava não apenas pelo movimento natural das experiências estéticas, mas também pelo processo de obsolescência dos aparelhos, peças e componentes que constituíam aquilo que se entendia como a Obra de Arte.

À semelhança do modelo crítico flusseriano com as categorias pré-industriais (FLUSSER, 2011, p. 40), as categorizações das artes visuais tradicionais já não acomodavam estas novas expressões, visto que muitas delas além de transitarem junto a Ciências da Comunicação e da Computação e as Engenharias Elétrica, Mecânica, Mecatrônica, Eletrônica; ainda, por volta dos anos 1990, passaram a interagir com a Biologia, a Química, a Física Quântica entre outras. Toda essa inter e transdisciplinaridade retirou a possibilidade de classificar as “artes da nova mídia” (*New Media Art*) dentro da classificação adotada até então.

Outra questão se deu quando os museus e galerias passaram a querer adquirir estas obras, ou seja, quando esta passou a se inserir no fluxo econômico da arte. Além da importância histórico-cultural que estas obras representam, passaram também a serem ativos econômicos em muitas coleções e o discurso da efemeridade guiada pela obsolescência passou a representar um risco para qualquer um que quisesse adquirir estas obras.

Ampliou-se então o debate sobre a forma de conservação destas expressões e é neste ponto que se insere esta tese. O processo de conservação destas obras passou por diversas fases e algumas daquelas que conseguiram ser conservadas, foram feitas após o falecimento dos artistas responsáveis pela criação, levando ao questionamento de se a intenção estética original do(a)s artista(s) foi respeitada.

As três grandes estratégias para se lidar com uma peça artística (Conservação, Restauração e Preservação) deveriam sempre partir da vontade de seus criadores. Entretanto, a produção de uma obra cujo suporte está em constante modificação é diferente da pintura ou escultura que tem suportes tradicionais. Estas últimas sofrem de deterioração química ou física em um processo natural, e nisso o setor de conservação dos museus e galerias já atuam para evitar seu desgaste, porém para as “novas mídias” não são apenas estes dois processos que levam ao desuso de um suporte, há também este fenômeno da obsolescência que é incontável por fazer parte da própria

essência das tecnologias da comunicação e da computação desenvolverem novas funcionalidades ao eliminarem características de suas versões anteriores.

Ainda com relação as três grandes estratégias, a Preservação é a atitude voltada para não se utilizar o instrumento, equipamento ou aparelho de forma que ele se mantenha em seu estado original. Estratégia útil para elementos cujo movimento de equilíbrio e entropia significa excluir o ser humano da equação, mas inútil quando se trata de equipamento eletro-eletrônico cujos componentes metálicos enferrujam e as peças eletrônicas oxidam quando não há atividade. A Conservação é a ação de manter a obra funcional com o mínimo de esforço ou impacto no objeto artístico. Para as obras das “novas mídias” este processo tem sido utilizado com êxito enquanto existirem peças de substituição dentre aquelas cuja tecnologia já se encontra defasada. Contudo, dentro da Restauração está o maior desafio à manutenção da intenção poética do(a)s artista(s) visto durante a pesquisa que deu origem a esta tese.

Segundo Cesare Brandi (2019) a restauração é qualquer intervenção voltada a retornar um produto da atividade humana à sua eficiência original. No caso da obra de arte, essa eficiência está ligada à sua capacidade de transmitir a intenção original do(a) artista, ou, no interior da teoria da Estética Informacional (MOLES, 1990), sua capacidade informativa.

Brandi ainda acrescenta que as restaurações seguirão dois caminhos, aquele relativo à restauração dos manufatos industriais e aquele relativo às obras de arte. O desafio demonstrado ao longo desta tese, está no fato de que a obra de arte tecnológica é talvez a mais dependente das mudanças tecnológicas de toda a História da Arte. Essa afirmação considera que na história da computação e das mídias comparada com a própria história da arte tecnológica demonstra que os limites estéticos foram se expandindo conforme uma nova descoberta ou invenção tecnológica era divulgada.

Assim, partimos da teoria de Brandi, e afirmamos que o restauro da funcionalidade do suporte da obra de arte tecnológica seria um “lado secundário ou concomitante, e jamais primário e fundamental que se refere à obra de arte como obra de arte” (BRANDI, 2019, p. 26). Isso nos leva ao objetivo da tese que é o de investigar as formas de documentação do processo criativo de artistas computacionais na busca pela preservação da intenção estética.

Dois exemplos nos inspiraram a realizar esta investigação. A primeira foi a comunicação “Acertos e desacertos na preservação de acervo em Arte Computacional Interativa” apresentada no Encontro Internacional de Arte e Tecnologia de 2016, a professora e artista computacional Tania Fraga narra sua relação com o processo de documentação de suas obras<sup>1</sup>.

---

<sup>1</sup> **Nota do autor:** Seu esforço, apresentado também no artigo de mesmo título e divulgado nos anais do evento (FRAGA, 2016), vem em um crescente histórico desde o ano de 1970 quando ainda era aluna de arquitetura na Universidade Federal de Minas Gerais (UFMG) em que “antevi[u] as possibilidades para o desenho e o projeto ao entender os fundamentos computacionais que possibilitam sua realização (UFMG: programação em linguagem FORTRAN)” (FRAGA, 2016, p. 261), projeto que apesar de lhe render prêmio, não conseguiu realizar a conservação dos seus documentos de trabalho, restando apenas o certificado da premiação.

A artista menciona desde 1980 ter conservado fotos e slides de projetos iniciais de arte computacional quando ingressou no corpo docente da UnB, bem como *scripts* produzidos na época que foram armazenados digitalmente em um algoritmo de compactação, que criados com tecnologia obsoleta, não podem ser abertos por softwares contemporâneos, apenas um computador com o sistema operacional DOS 3.0 (que lançado em 1984 foi descontinuado em 1988) poderia realizar este processo (FRAGA, 2016, p. 262).

Os desafios enfrentados por Fraga para conservar suas obras aumentaram de dificuldade junto com a crescente disponibilidade de tecnologias de software e de hardware (tanto na sua evolução de versão como na criação de novos equipamentos e programas): “A partir deste período guardei as máquinas (exceto a da *Silicon Graphics*, guardei os óculos de cristal líquido, mas eles não funcionam mais).” (Fraga, 2016, p. 264). Para tentar fazer frente a todas as possibilidades, imagináveis ou não, do presente ou do futuro, ela passou a guardar os sistemas operacionais em “Máquinas Virtuais”<sup>2</sup>.

A partir de 2005, paulatinamente, todas as versões dos programas criados, da máquina virtual da linguagem Java e da API Java 3D utilizados passaram a ser arquivados como backups em vários discos rígidos, com cópias em CDs, com todos os arquivos necessários para rodá-los. A partir de 2014 tem sido guardada, também, as versões das máquinas virtuais para 64 e 32 BITS, para Windows, Unix e Macintosh. (FRAGA, 2016, p. 265).

O segundo exemplo parte do artista, pesquisador e professor que lidou com processos de documentação e conservação por ocasião de uma rerepresentação de uma de suas obras, Gilberto Prado. Em 2014, sua obra “Desertesejo” foi escolhida pela curadoria da exposição Singularidades/Anotações para exibição (PRADO; CUZZIOL, 2019, p. 245). Todavia, a obra criada entre os anos de 1999 e 2000 não podia mais ser executada com uma tecnologia quatorze anos mais recente. O *plugin* de VRML (*Virtual Reality Modeling Language*) utilizado pela obra tornou-se obsoleto nos navegadores de Internet mais recentes. A obra no ano em que foi exibida (2000) foi considerada inovadora (CUZZIOL; PRADO, 2019, p. 117), não apenas na performance da computação gráfica, mas também na capacidade de interação humano-máquina. Desde sua criação até a sua seleção para o evento de 2014, não apenas a capacidade gráfica dos computadores aumentou, mas também os dispositivos de *output* visual – óculos de realidade virtual, projetores de vídeo ou mesmos os monitores – melhoraram sua performance em relação à qualidade da imagem tais como: representação de cores, contrastes e resolução.

Neste sentido, para se ater a mesma experiência estética o artista decidiu não se utilizar das tecnologias recentes, que poderiam provocar uma outra experiência imersiva diferente da inicial.

---

<sup>2</sup> **Nota do autor:** Máquina Virtual é a simulação de um Sistema Operacional com todos os softwares e drives escolhidos pelo usuário instalados. Seu uso permite que haja múltiplos sistemas operacionais ou várias configurações do mesmo com objetivo de testar ou caso determinado software somente rode em um sistema específico ou configuração específica.

Visto que uma simples apresentação de uma documentação em vídeo estava igualmente descartada pela falta de interatividade, o artista e a curadoria decidiram pelo caminho da restauração do *Desertesejo* (PRADO; CUZZIOL, 2019, p. 246).

O trabalho de restauro foi intenso, como todos os ambientes do trabalho tinham que ser remodelados em 3D, texturas, sons e iluminação recriados, avatares reconstruídos etc. Consequentemente, mesmo com a criação dos novos ambientes desenvolvidos por programas diferentes, tanto a aparência quanto a experiência do trabalho anterior foram mantidos e apresentados ao público na exposição em 2014 (PRADO; CUZZIOL, 2019, p. 246).<sup>3</sup>

O trabalho de restauro apresentou duas grandes vantagens quando se analisado o esforço para que a experiência estética permanecesse inalterada: (1) o grande volume de documentação da obra, como: vídeos, fotos, anotações do artista e dezenas de artigos científicos; que em conjunto constituem uma descrição da obra e da experiência de quem a vivenciou; e (2) a participação ativa do artista no processo do restauro, coordenando a reconstrução dos elementos que compõem o mundo imersivo do *Desertesejo*.

Os dois casos descritos acima são indícios de um problema que se apresenta para artistas computacionais: a preservação da intenção poética em suas obras em oposição à obsolescência em relação à tecnologia de computadores, periféricos e *softwares*.

Ou seja, os artistas ao pretenderem preservar suas intenções e investigações estéticas, encontram dificuldades em expor trabalhos antigos. Estes obstáculos vão desde a perda de anotações e documentos, até a total incapacidade de acessar determinados arquivos armazenados em *softwares* obsoletos ou em dispositivos de armazenamento inutilizados pelo tempo.

A pesquisa então, neste sentido, mostra que há maneiras diferentes de resolução deste problema tanto na teoria quanto na prática. O objeto de arte computacional pode ser fruto de discussão para as instituições de guarda e exposição (como galerias e museus), para o mercado de arte em geral, além dos artistas. Esta tese é elaborada a partir do ponto de vista do artista, suas intenções e poéticas. Neste sentido, destaca que para a preservação das suas obras estes elementos devem ser considerados e respeitados no momento da restauração e na própria conservação da obra.

Em 2013, Bernhard Serexhe coordenou o projeto “Digital Art Preservation” realizado na ZKM | Karlsruhe, Alemanha em que cita:

O problema geral abordado aqui, ou seja, a preservação e transmissão da cultura digital genuína de nosso tempo para as gerações futuras, está destinado a se tornar uma questão decisiva para todas as instituições culturais no século XXI, independentemente de apoiarem os padrões de valores tradicionais, estruturas operacionais e objetivos, ou eles próprios procuram fazer parte da vanguarda do progresso tecnológico.

<sup>3</sup> **Nota do autor:** Tradução nossa do original em inglês.

O projeto digital art conservation foi iniciado na ZKM | Karlsruhe em resposta a urgência desta questão, tão fundamental para a arte digital [...]. (SEREXHE, 2013, p. 13)

O problema apresentado por Serexhe traz como responsáveis principais “as instituições culturais”, porém isso não exclui a visão do artista. O livro “Digital Art Preservation” é dividido em duas partes iniciais: a primeira apresenta artigos concentrados nas teorias gerais da conservação de obras baseadas em computador e artigos cujo ponto de vista era a das instituições com abordagens tecnicistas e metodológicas. A segunda parte procura uma aproximação prática da conservação ouvindo artistas, programadores e teóricos. O livro finaliza com uma série de estudos de caso e um capítulo dedicado ao ensino do modo de pensar conservacionista nas escolas de arte.

No ano seguinte, em 2014, o simpósio “Futuros Possíveis” realizado na FAU-USP com parceria da Ars Electronica (Linz, Áustria) também iniciou o debate sobre

estratégias e metodologias para armazenamento e preservação de arte digital e processos de digitalização de acervos, incluindo também estudos sobre novas formas de organização e disponibilização das informações em sistemas de visualização de dados” (BEIGUELMAN; MAGALHÃES, 2014)

Na ocasião deste simpósio, Arianne Vanrell Vellosillo, restauradora no Museu Nacional Centro de Arte Reina Sofia em Madri, apresentou o trabalho intitulado “Estratégias de conservação e humanidades digitais”, derivado de sua tese de doutorado “Novas estratégias para a conservação das coleções de arte com elementos tecnológicos: propostas metodológicas de humanidades digitais” defendida em 2015. No artigo referente à comunicação ela cita:

A iminência e a rapidez dos processos de deterioração e obsolescência dos elementos de fabricação, das linguagens informáticas e dos suportes de conservação, armazenagem, visualização e reprodução das obras baseadas na tecnologia são objetos de numerosas discussões e propostas por parte de grupos de pesquisa de todo o mundo. (VELLOSILLO, 2014, p. 138)

No mesmo evento, Prado participa com palestra/artigo “Arte digital, diálogos e processos”, em que faz uma retrospectiva dos seus próprios trabalhos artísticos, individuais ou coletivos com o grupo “Poéticas Digitais”, e diante das transições de tecnologia que vivenciou faz duas perguntas: “O que reter, como reter? O que guardar, como fazê-lo?” (PRADO, 2014, p. 122), demonstrando clara preocupação quanto ao que o artista deve fazer caso deseje preservar suas obras, além do recorrente questionamento de se deve preservar.

Outra inspiração para a pesquisa foi o livro “Teoria da Conservação Evolutiva” (MORALES, 2019) de Lino García Morales da Universidad Politécnica de Madri em que o professor e pesquisador apresenta cartografias das novas mídias<sup>4</sup>, um trabalho catalográfico e taxionômico das

---

<sup>4</sup> **Nota do autor:** Deste ponto em diante o termo “novas mídias” será substituído pelo termo “mídias instáveis” independentemente do uso por teóricos citados, por considerarmos este termo mais adequado à esta expressão artística. Quando o termo “novas mídias” for utilizado por qualquer teórico de maneira a ser descrita ou conceituada ou analisada, então faremos uma observação.

obras de arte e tecnologia acrescido de reflexões sobre como se dá o processo de restauro e conservação das mesmas.

Em exemplos brasileiros é possível citar o trabalho do professor Pablo Gobira em que na coordenação de seu grupo de pesquisa (LabFront) na Universidade Estadual de Minas Gerais (UEMG) cuja produção bibliográfica é relevante para o tema, além de organizar um evento anual de Arte Tecnológica.

A professora Giselle Beiguelman da Universidade de São Paulo (USP) - também organizadora do evento Futuros Possíveis, já citado acima - constantemente é chamada a palestrar sobre sua pesquisa de conservação da Net Art e os desafios envolvidos na espécie de “meta-preservação” em que a própria Internet deve ser a principal responsável pelo armazenamento e acessibilidade de materiais nela mesma produzidos e veiculados.

O site da Fundação Itaú Cultural contém uma enciclopédia sobre arte em geral e que compreende também a parte de arte e tecnologia, na qual cita alguns artistas (brasileiros e estrangeiros), suas obras e currículo. A Fundação se preocupa em como abordar a restauração das obras computacionais em seu acervo. Fato que ocorre principalmente desde o restauro das obras “Beabá” de Waldermar Cordeiro e Giorgio Moscati e “Desertesejo” de Gilberto Prado.

Outra referência a ser destacada é o livro “Digital art through the looking glass, new strategies for archiving, collecting and preserving in digital humanities (GRAU; HOTH; WANDLVOGT, 2019). Ele apresenta uma coletânea de artigos de pesquisadores do tema e dentre eles dois artigos dos autores Diego Mellado Martínez e Patrícia Falcão se despontam, pois apresentam estudos de caso em metodologias de conservação de obras computacionais.

O primeiro com o título “A System Engineer’s Perspective for the RE-creation of Media Art in n-Cha(n)-t by David Rokeby”, defende a metodologia da re-criação da obra de arte como parte do esforço de conservá-la para o futuro, inspirado na “Teoria da Restauração” de Cesare Brandi. Para tanto apresenta a integração das ferramentas de engenharia de sistemas com a prática de conservação das obras de arte digital e tecnológica, passando a respeitar a experiência estética proposta pelo artista, mas sem negar que não há como impedir o movimento de obsolescência dos equipamentos e *softwares*.

O segundo intitulado “*Preservation of Software-based Art at Tate*” discute os limites das intervenções que aparelhos culturais de exposição artística podem ter ao tentar conservar as obras que adquiriram sem invalidar a vontade do artista. No processo, ela descreve uma sequência de procedimentos e ações que devem ser seguidas para assegurar uma maior “longevidade” das obras de arte computacionais, dando o exemplo da Emulação como possibilidade.

A existência de um interesse crescente em projetos e pesquisas que se propõem a investigar a questão da conservação das obras de arte computacionais representa um indicativo da relevância

do tema. Contudo, enfatizamos aqui que há uma enorme diversidade de obras de arte tecnológicas, uma vez que dentro deste campo entende-se as obras que se utilizam de processos digitais e analógicos que tenham uma ligação com as inovações tecnológicas. Assim, podem se enquadrar igualmente a biocibernética, a *web-art*, a *software art*, a *hardware art*, a *game art*, entre outros, citando inclusive as artes mais tradicionais que se aproveitam dos instrumentos tecnológicos como a pintura e a escultura digital, a fotografia digital, a performance mediada por computador, o cinema interativo, entre outros.

A categorização das expressões artísticas, muitas vezes nomeadas pelos seus suportes, pode ocasionar em equívocos ou imprecisões na conservação de experiências estéticas, trocando, por exemplo, a conservação do objeto (*hardware*) em detrimento da experiência interativa do usuário com a obra, limitando assim a experiência a um “disseram ser assim” ao invés de um “eu senti isso”. Exemplificando com o termo “Arte Computacional” cujo entendimento mais comum tem sido de uma arte com ou mediante um aparelho computador, quando na realidade: “Arte computacional nem sempre recorre ao computador ou a operações lógico-matemáticas; às vezes está fundamentada na intuição somente, na qual a arte é uma matemática sem lógica e verdade” (VENTURELLI, 2017, p. 8).

Quanto a questão da nomenclatura entendemos ser necessário apontar que nosso entendimento é de que o termo “arte de novas mídias” é errôneo a começar pelo fato de ser datado (continuaremos a considerar novas até quando; e quando novas surgirem o que se dá com as antigas?). Em nossa pesquisa fomos apresentados a dois termos que entendemos serem mais condizentes: Mídias Variáveis apresentado pelo Museu Guggenheim (DEPOCAS *et al.*, 2003) e Mídias Instáveis apresentado pelo laboratório V2 Lab da Holanda com o “Manifesto das Mídias Instáveis” (1987). O primeiro demonstra a variedade de suportes disponíveis para estas expressões artísticas, o segundo aponta para a natureza destes suportes e as discussões em que orbitam. Por questões poéticas, utilizaremos o termo “arte das mídias instáveis” ao longo desta tese.

Retornando ao trabalho da professora Suzete Venturelli sobre arte computacional, ela afirma que: “o objeto do artista computacional é o software” (VENTURELLI, 2017, p. 64), o que nos remete a uma investigação sobre a gênese da obra de arte computacional. Este desafio mais próprio do caminho da ontologia, se desenvolve como ponto inicial na definição do campo epistemológico que iremos abordar assim como as metodologias de investigação aplicadas. Convém salientar que apesar de não se encontrar em nosso rol de habilidades o domínio filosófico ou histórico, consideramos que para um rigor na análise do objeto investigado haja necessidade que se sedimente estas bases. O desafio aqui está na própria natureza da composição da área de investigação artística – Arte/Ciência/Tecnologia – que encontra aspectos de um racionalismo e do

estruturalismo no campo da ciência e da tecnologia, equilibrado com a relação empírica e fluídica do artista com sua obra e os conceitos ali trabalhados.

Assim, é relevante responder a uma ligeira modificação da pergunta do título da tese, “O que reter, o que guardar?”. Essa pergunta direciona para reflexões sobre a essência da obra de arte computacional, pois se a essência for o código e a forma como ele está escrito, a obra poderia perdê-la quando aquela tecnologia, no caso a linguagem de programação em que o software foi escrito, ficasse obsoleta e descontinuada. Isso poria a arte sob influência direta do mercado econômico, limitando a experiência estética sobre determinada obra a um período apenas da existência humana, negando às gerações futuras a memória daquelas experiências de fruição estética.

Diante da experiência adquirida ao longo dos quatro anos desta pesquisa que se iniciou em 2018, as entrevistas feitas, os documentos catalogados, o período que empreendemos a pesquisa em outras instituições fora do Brasil, elaboramos um protocolo para conservação estética da arte computacional, que leva em consideração as idiosincrasias da computação e das artes, e reflete um esforço para contribuir com ações de institutos como o ZKM, assim como tentar trazer mais um instrumento para o(a)s artista(s).

Para chegarmos à apresentação deste protocolo e do modelo de diagramação descritiva para as obras de arte computacional que elaboramos, esta tese foi dividida em duas partes, desmembrando a pergunta título.

A primeira parte intitulada “O que reter? O que guardar?”, traz os três primeiros capítulos e tem a preocupação de entender o que é aquilo que faz a Arte Computacional a Arte Computacional:

O Primeiro Capítulo, “Uma história da arte computacional”, realiza um apanhado histórico com recorte na velocidade cada vez crescente com que as tecnologias da comunicação e da computação eram substituídas por versões mais novas, inclusive quando isso é demandado pelo mercado para movimentar o consumo de bens.

No Segundo Capítulo, “A Arte e as Tecnologias da Informação e da Computação”, procuramos descrever como houve a aproximação dos artistas com aquelas que foram chamadas de “novas mídias” por mais de setenta anos por conta do seu apelo para o futuro. Descreve as três gerações de artistas computacionais e os principais centros de pesquisa e eventos em que se reúnem.

No Terceiro Capítulo, “Uma ontologia da Arte Computacional”, executamos um trabalho de investigação ontológica sobre a Arte Computacional para entendê-la como um objeto de pesquisa. A premissa aqui é de que para poder conservar ou restaurar algo, é necessário saber quais os elementos que compõe a coisa alvo da conservação ou da restauração.

A segunda parte, “Como reter? Como guardar?” apresenta em dois capítulos as estratégias adotadas para a conservação e como elas refletem na preservação da intenção poética do(a)s artista(s), além de apresentar o protocolo e o modelo C7Art de diagramação em fases da descrição da poética da obra de arte computacional.

No Quarto Capítulo: “Experiências em Conservação de Obras em Mídias Instáveis”, resgatamos as experiências adquiridas ao longo da trajetória do doutoramento, em especial com as ações e contatos estabelecidos durante o período de Doutorado-Sanduíche realizado de Abril de 2021 a Abril de 2022, primeiro de forma virtual (devido as restrições da pandemia) após presencialmente em Berlim, Alemanha, na Universidade de Artes de Berlim – UdK, de onde pudemos realizar visitas técnicas no Centro de Arte e Mídia Karlsruhe (ZKM) em Karlsruhe, Alemanha; no Museu Reina Sofia e na Universidade Politécnica de Madri, em Madri, Espanha. Apresentando assim as experiências dos pesquisadores destas instituições frente a conservação de obras computacionais e a preservação das poéticas.

O Quinto e último Capítulo, “Protocolo de Conservação de Obra de Arte Computacional, apresenta o resultado prático das observações das formas de documentação de artistas entrevistados em forma de um protocolo, representando sugestões sobre o que deve ser feito como documentação caso seja intenção do(a)s artista(s) realizar a conservação de sua obra. O modelo de diagramas C7Art, inspirado no modelo C4 de arquitetura de software, foi criado para permitir uma troca de informações mais clara entre o(a)s criador(a)(es) da obra e terceiros que possam se interessar pelo funcionamento da mesma por alguma razão específica.

Entendemos que essa tese não representa o fechamento total da pesquisa, já que inúmeros desdobramentos podem ser feitos. Assim, esta tese representa o estado da arte deste momento específico, em que se concluí a pesquisa de doutorado apresentando aquilo que nos propusemos a realizar desde o momento da apresentação do anteprojeto, um protocolo para conservação de arte computacional.

# Parte 1: O que reter? O que guardar?

A Parte 1 aborda os capítulos: “Uma história da arte computacional”, “A Arte e as Tecnologias da Informação e da Computação” e “Uma ontologia da arte computacional” que têm como proposta resgatar a história da Arte Computacional a partir da intersecção entre as histórias do computador e da Arte, citando os desdobramentos tecnológicos que repercutiram na forma como as linguagens de programação foram desenvolvidas e o estabelecimento da Ontologia da Arte Computacional.

Procura-se estabelecer uma ligação entre os eventos precursores do surgimento da Arte Computacional, em relação aos institutos de pesquisa, aos cientistas e aos artistas tanto no Brasil quanto no exterior. Esta etapa da pesquisa contribuiu significativamente para a reflexão sobre a necessidade de conservação das obras de Arte Computacional, diante das perdas históricas ocorridas tanto pela obsolescência dos equipamentos e sistemas, quanto pela falta de organização e documentação por parte dos seus criadores artistas e/ou cientistas.

# 1. Primeiro Capítulo: Uma história da arte computacional

## 1.1 A revolução computacional sociocultural

No decorrer da pesquisa, diversas vezes e sob diversas égides, verificou-se que as introduções de diversos artigos e livros, possuíam um denominador comum, que lidavam com a relação do computador com algum aspecto da cultura e destacavam a menção de um “impacto” ou “revolução” que o computador representou em diferentes setores da sociedade humana, como pode-se observar na citação de Manuel Castells <sup>5</sup>(2006):

Ao redor deste núcleo de tecnologias da informação, definido em um sentido mais amplo, houve uma constelação de grandes avanços tecnológicos, nas duas últimas décadas do século XX, no que se refere a materiais avançados, fontes de energia, aplicações na medicina, técnicas de produção (já existentes ou potenciais, tais como a nanotecnologia) e tecnologias de transportes, entre outros. Além disso, o processo atual de transformação tecnológica expande-se exponencialmente em razão de sua capacidade de criar uma interface entre campos tecnológicos mediante uma linguagem digital comum na qual a informação é gerada, armazenada, recuperada, processada e transmitida. Vivemos em um mundo que, segundo Nicolas Negroponte, se tornou digital.(CASTELLS, 2006, p. 67–68).

Nesse mesmo viés, para Edmond Couchot:

Ao longo da última década, as tecnologias do cálculo automático – conjunto de aplicações extremamente diversificadas da informática, agrupadas sob o termo *numérico* – aumentaram ainda mais seu controle sobre o mundo das imagens. (COUCHOT, 2003, p. 11)

Warren Sack adota uma postura mais dramática, quando diz que “a Revolução computacional pode ser encarada como uma reescrita do mundo” (SACK, 2019, p. 1). O impacto mencionado se refere à uma profunda e intensa mudança sociocultural, econômica e política provocada pelo que Castells denomina de “penetrabilidade, ou seja, por sua penetração em todos os domínios da atividade humana, não como fonte exógena de impacto, mas como o tecido em que essa atividade é exercida” (CASTELLS, 2006, p. 68).

Para o Castells, os computadores e todas as tecnologias da informação e da computação, que surgem em paralelo ou concomitantemente, representaram para a cultura o mesmo que as novas fontes de energia tiveram para a primeira e segunda Revolução Industrial. Mas, Castells considera que, ao contrário das outras Revoluções, a dependência dos novos conhecimentos e

---

<sup>5</sup> **Nota do Autor:** Inclusive, a posição do sociólogo (CASTELLS, 2006, p. 67) sobre os domínios científicos e do pensamento humano que deveriam ser incluídos no rol das tecnologias da informação, que adiciona a engenharia genética junto com a computação (software e hardware) e a microeletrônica, pode ser um importante marco para a caracterização da “Arte e Tecnologia” como categoria estética.

informações não fica centralizado em um grupo pequeno por muito tempo, pois na Revolução Informacional os novos conhecimentos e informações são “conhecimentos e dispositivos de processamento/comunicação da informação, em um ciclo de realimentação cumulativo entre a inovação e o uso” (CASTELLS, 2006, p. 69). Além disso, ao contrário das tecnologias anteriores, essas se difundiram de maneira uniforme (em questão de adoção não em questão de proporção populacional, já que isso depende de fatores econômicos) entre todos os povos e culturas.

Por isso, a história da criação e das transformações da tecnologia computacional, contada aqui a partir, principalmente, das obras de Martin Campbell-Kelly (CAMPBELL-KELLY *et al.*, 2018), de Raul Wazlawick (WAZLAWICK, 2016) e de Manuel Castells (CASTELLS, 2006), procura apresentar não apenas uma descrição óbvia das tecnologias inventadas, mas da motivação por trás destas invenções, e das consequências diretas das mesmas na cultura.

O impacto provocou mudanças na metodologia de pesquisas tecnológicas realizadas, que surgem tanto nas universidades ou em centros de pesquisa em empresas, e se tornaram protagonistas no processo de invenção e descoberta. Por exemplo, a invenção da Internet surgiu de um projeto de comunicação militar descentralizado para se tornar um sistema multimidiático global que facilitou a ligação e a cooperação em pesquisa de diversos centros de pesquisa e universidades (CASTELLS, 2006, p. 82–89).

Entre outros espaços da cultura, o computador também penetrou nas Artes. Para Grau, Hoth, Wandl-Vogt, os artistas digitais utilizam as mesmas tecnologias que influenciam nosso dia a dia, a saber “telecomunicações, mídias sociais e aplicativos mobiles” (GRAU; HOTH; WANDL-VOGT, 2019, p. 10). Assim estes artistas digitais investigam nossa sociedade tentando contornar o efeito “caixa-preta”<sup>6</sup> destas tecnologias, que pré-determina um *modus operandi*.

Por isso, entende-se que no contexto cultural e para as Artes, o computador ampliou o universo de possibilidades estéticas, quando utilizado como parte do processo criativo da mídia interativa, ou ainda como uma ferramenta que auxilia a produção da arte tradicional como a pintura, a escultura, a fotografia etc.

Por outro lado, para os autores, os computadores são um fenômeno sociocultural no qual o papel do artista digital se destaca, no desenvolvimento da tecnologia:

Através de métodos transdisciplinares na interseção da arte, ciência e tecnologia, a arte digital combina criação artística com pesquisa inovadora e desenvolvimento tecnológico e, portanto, conecta a história da arte com métodos digitais e fenômenos socioculturais contemporâneos. (...) Os artistas digitais têm contribuído para o desenvolvimento de análises computacionais por meio da estética e de disposições

---

<sup>6</sup> **Nota do autor:** o conceito de “caixa-preta” vai ser bem desenvolvido por Vilém Flusser em “Filosofia da Caixa Preta”, mesmo não sendo ele o primeiro a mencionar o conceito, e tem por significado qualquer sistema fechado, aquele que as funções ocorrem independentemente de interferência externa, cuja única função daquele que se relaciona com a “caixa-preta” é a de fornecer *inputs* e obter os *outputs*.

experimentais em arte-ciência-tecnologia em sua forma artística. (*Op Cit, IDEM, IBIDEM*).

Outro ponto relevante que se destaca no contexto sociocultural é a forte ligação do processo de desenvolvimento do computador, da linguagem de programação e da Internet com as pesquisas acadêmicas, que provocou o interesse dos artistas na qualificação através de programas de pós-graduação ou em cursos de graduação, no Brasil, que já possuem arte computacional em sua grade.

Entretanto, nos últimos anos, constata-se no contexto sociocultural que artistas digitais autodidatas surgiram em função da educação informal, facilitada pela diminuição do custo frente à capacidade dos computadores e o aparecimento de *softwares* mais específicos para a produção artística, assim como o desenvolvimento de linguagens de programação de alto nível, que são mais próximas da linguagem humana. Nesse sentido, Suzete Venturelli afirma que buscou evidenciar que o artista computacional é programador e, portanto, se diferencia de outros artistas na atualidade, pois ele é ao mesmo tempo artista e programador, mesmo não existindo ainda nos currículos atuais de arte a formação adequada, que passa pelo ensino superior. Para a autora os artistas computacionais se formam nas universidades, na sua maioria são doutores ou estão estudando para isso, e trabalham em centros de pesquisa relacionados a questões da arte, ciência e tecnologia computacional. Para a autora “É importante frisar que no cerne de tudo isso está o computador, peça fundamental de aproximação dessas áreas de conhecimento.” (VENTURELLI, 2017, p. 161).

### **1.1.1. Computadores e Máquinas**

O computador, seja ele denominado como ferramenta, instrumento, equipamento, aparelho ou dispositivo, impactou os séculos XX e XXI ao provocar transformações profundas nos seus aspectos culturais, simbólicos, informacionais, comunicacionais, entre outros. As mudanças ocorridas, por exemplo, na transformação de um equipamento totalmente mecânico para os equipamentos microeletrônicos, fazem parte de uma trajetória histórica do desenvolvimento da ciência e da tecnologia.

O surgimento dos computadores está relacionado às necessidades de contabilizar estoques e vendas, desde o período das Grandes Navegações, onde o fluxo comercial aumentou em uma curva íngreme (cf. CAMPBELL-KELLY *et al.*, 2018; WAZLAWICK, 2016). Além disso, neste período pré-industrial, as cidades começaram a crescer e a necessidade de contabilizar essa massa urbana e suas conseqüentes implicações para as políticas públicas ficou evidente para os políticos (cf. CAMPBELL-KELLY *et al.*, 2018; WAZLAWICK, 2016).

Assim, as primeiras experiências para construção de um aparato computacional ocorreram ainda durante o período da primeira revolução industrial e sua principal função era executar cálculos assim como os computadores humanos realizavam. Hoje, a função computacional

é executada em outra camada de interface do aparelho, o que permite ao mesmo mediar outros processos de linguagem, desta maneira podemos avaliar o computador atual como um equipamento muito mais complexo do que uma máquina, pois apresenta a capacidade de lidar com “símbolos”, tornando-o uma “Máquina Semiótica” (NÖTH, 2008).

Se estabelece então uma importante crítica conceitual já que o conceito de máquina, aplicado ao computador, remete aos primeiros instrumentos movidos à vapor e depois à eletricidade, que substituíram trabalhadores manuais em suas atividades artesanais. Feitos de materiais diversos, os instrumentos, tinham nas engrenagens seu denominador comum, peças que dificilmente encontrar-se-á num aparelho computacional contemporâneo.

Como mencionado acima, a força motriz para a pesquisa de equipamentos computacionais era a automatização de cálculos, Cambell-Kelly (2018), menciona que desde o final do século XVIII o governo britânico investia na produção de “Tabelas Náuticas”, que eram feitas com periodicidade anual e publicadas com o nome de “*Nautical Almanac*”, aumentando com isso a precisão navegacional consideravelmente. A produção de tabelas náuticas era uma preocupação tão importante para o governo britânico que passou a investir na pesquisa do cientista Charles Babbage, que não conseguiu concluir a pesquisa, mas apresentou um protótipo denominado de Máquina Diferencial (**Figura 1**), que está exposto no Museu de Ciências de Londres.



**Figura 1:** Protótipo da Máquina Diferencial. Fonte: London Science Museum.

Apesar de não ter conseguido apresentar mais que um protótipo da Máquina Diferencial, Babbage passa a desenvolver o projeto da Máquina Analítica sem apoio estatal. Esta, ao contrário da sua irmã mais velha, não estaria condicionada apenas ao cálculo de tabelas marítimas, nela todos os cálculos seriam possíveis de serem executados. Ambas as máquinas exigiam uma confecção metalúrgica e mecânica muito precisa e delicada, que ainda não estava disponível na Inglaterra do século XIX. Como resultado, a Máquina Analítica desenhada por Babbage só viria a ser construída no século XX, em comemoração ao centenário de Babbage e se encontra em exposição no Museu de Ciências de Londres.

Para qualquer estudioso do tema lidar com uma história do computador é um trabalho muito extenso, já que não se trata de uma única invenção. O computador como conhecemos hoje é um equipamento muito longe de qualquer das primeiras experiências, como as máquinas de Babbage e se apresenta como uma amálgama de várias tecnologias computacionais e midiáticas.

As máquinas de Babbage têm como referência o Tear de Jacquard (CAMPBELL-KELLY *et al.*, 2018), que foi uma importante invenção para a indústria da tecelagem, porém sua forma única de inserção das instruções de tecelagem foram clara inspiração para a programação de cartões perfurados que seria utilizado na teórica Máquina Analítica e na Indústria Computacional primitiva. Tratava-se, portanto, de um Tear que podia ser programado por uma sequência de cartões perfurados, produzindo uma infinidade de padrões nas tapeçarias de maneira automática

A Máquina Analítica herdava então características que veremos apenas no Computador moderno: (1) a capacidade de ser programada para diversas tarefas com cartões perfurados; (2) uma separação entre a computação aritmética (*Mill* = Processador) e a guarda dos números (*Store* = Memória); e (3) um *output* impresso do resultado. E, para completar a metáfora utilizada por Babbage entre sua máquina e a indústria têxtil, Lovelace escreve em "*Rascunho da Máquina Analítica*": "Nós Podemos dizer mais apropriadamente que a Máquina Analítica tece padrões assim como o tear de Jacquard tece flores e folhas"(MENABREA *et al.*, 1843).



**Figura 2:** Tear de Jacquard exposto no Museu Alemão de Tecnologia. Fonte: Acervo próprio. Foto apreendida no dia 16 de janeiro de 2022.

### 1.1.2. O Computador Moderno

Apenas em 1937, no artigo “*On computable numbers, with an application to the Entscheidungsproblem*” escrito por Alan Turing, a tecnologia de uma máquina capaz de ser programada e reprogramada para realizar diversos tipos de atividade retornou a discussão.

Neste artigo, Turing descreve uma máquina, que mais tarde seria conhecida como “Máquina de Turing”, capaz de ler e escrever símbolos em uma fita de comprimento infinito sendo controlada por uma tabela de instruções. Esta máquina, com sua capacidade de ser programada, seria uma máquina universal capaz de computar qualquer cálculo, e isso se dava pelo fato de que a máquina era capaz de reconhecer que determinado símbolo se referia a uma parte específica de um cálculo, e uma vez transformando conceitos em símbolos e permitindo-os serem calculados como números, a Máquina de Turing poderia realizar virtualmente qualquer tipo de tarefa do pensamento humano.

Em um segundo artigo, “*Computing machinery and intelligence*” (TURING, 1950), este escrito já após a Segunda Guerra Mundial (SGM), Turing aborda a questão da “capacidade de pensar” dos computadores, o que antecipa novamente reflexões sobre os limites do computador que iremos explorar mais à frente no texto.

Turing fundou um grupo de estudos de computadores no *National Physical Laboratory* após a SGM. Os anos da SGM foram importantes para o desenvolvimento de diversas áreas da ciência,

entre eles os computadores. A IBM já possuía computadores mecânicos baseados em cartão perfurado, e outros computadores analógicos eram utilizados em conjunto com calculadoras de mesa (CAMPBELL-KELLY *et al.*, 2018, p. 67). O primeiro computador digital eletrônico também surgiu como um projeto de pesquisa durante a guerra nos EUA, apesar de ter sido concluído oficialmente somente seis semanas após o término da mesma em 1945. O ENIAC (*Electronic Numerical Integrator and Computer* ou Computador eletrônico numérico integral) podia realizar 5000 cálculos por segundo. Todavia, o ENIAC possuía um gigantesco problema de design para um período de guerra em que a velocidade é um fator importante para determinar o rumo das batalhas: gastava-se muito tempo para reprogramar o computador ao final de um cálculo. (CAMPBELL-KELLY *et al.*, 2018, p. 67).

Em 1944, John Von Neumann, cientista matemático tomou conhecimento do projeto do ENIAC e ficou interessado nas possibilidades de cálculo, mas verificou igualmente os problemas que o computador possuía: muito grande, calculava números com poucos dígitos e tinha uma reprogramação muito demorada. Os engenheiros então retornaram à pesquisa, e elaboraram o EDVAC (*Electronic Discrete Variable Automatic Computer*) que utilizava um *delay* de mercúrio para armazenar pulsos elétricos que eram analisados como números, e utilizava o conceito de programa armazenado, em que o computador tinha que armazenar tanto o programa base quanto os números que deveria analisar<sup>7</sup>.

Neste momento também é “plantada a semente da linguagem de programação e a inteligência artificial” (Campbell-Kelly *et al.*, 2018, p. 76), já que o fato do computador tratar suas instruções como dado permitia que você utilizasse programas para criar outros programas. Outra inovação trazida pelo EDVAC, foi a utilização de metáforas biológicas para as partes do computador, Von Neumann ao escrever sobre o EDVAC descreveu suas cinco partes como os órgãos do computador (Controle Central, controle aritmético central, memória, input e output) e assim acidentalmente se cria o termo “memória de computador” que veio a substituir o termo “armazém” que ainda era utilizado desde Babbage (CAMPBELL-KELLY *et al.*, 2018, p. 76). Ainda, passou-se a utilizar o código binário que aumentava a capacidade de armazenamento numérico em comparação com o ENIAC que utilizava código decimal.

Após o fim da SGM o mundo não teve nenhum conflito de escala planetária que envolvesse diversas nações, mas a disputa de hegemonia política entre o Ocidente Capitalista e o Oriente Comunista trouxe investimentos para a tecnologia e a ciência. A corrida armamentista da chamada Guerra Fria capitaneada pelos EUA e pela União Soviética, fez com que o desenvolvimento de

---

<sup>7</sup> **Nota do autor:** Este conceito é encontrado ainda hoje nos softwares contemporâneos em que algumas linhas de código inicial são feitas para a execução do mesmo, sendo, inclusive, um dos principais obstáculos encontrados na restauração de obras computacionais, uma vez que é necessário um conhecimento mais aprofundado do próprio sistema operacional para saber os requisitos utilizados por determinado software e como levar esses requisitos para outro sistema.

computadores e outros equipamentos eletroeletrônicos, bem como sistemas de comunicação entrasse em uma curva nunca vista de desenvolvimento. Da Revolução Agrícola para Revolução Industrial passaram-se 8.000 anos, mas da invenção da lâmpada elétrica (1847) para a invenção do ENIAC e do EDVAC nem 100 anos se passaram, o que seria um prenúncio da velocidade de obsolescência envolvida nesta tecnologia.

A partir dos anos 1950, as diversas empresas que fizeram a história da computação contemporânea surgiram e desenvolveram os computadores para serem máquinas de processamento de dados e não simples máquinas de calcular<sup>8</sup>.

### 1.1.3. De transistores para chips

Outras inovações na computação foram celebradas ao longo do tempo, como o desenvolvimento de múltiplos pontos de entrada em um único computador (*inputs*), as estações de trabalho, com a inserção de um teclado para entrada dos dados diretamente no equipamento que excluía a necessidade de cartões perfurados, para aproveitar a capacidade de cálculo dos computadores em um sistema multitarefas. Os transistores foram substituídos por circuitos integrados em 1965, e estes por sua vez sofreram um processo de miniaturização que persiste ainda hoje, dando início à microeletrônica. Por isso, em 1970 era possível comprar um microcomputador pessoal do tamanho de um eletrodoméstico por um valor dez vezes menor de um computador de igual processamento de 1965, que ocupava uma sala.

Em 1969, a constante demanda por cientistas e engenheiros para desenvolver computadores menores, que eles pudessem usar em suas casas para criar seus próprios softwares e testar suas próprias invenções, atraiu a atenção e pequenas empresas surgiram somente para atender a essa demanda do mercado. As grandes empresas, por sua vez, abriram divisões de Pesquisa e Desenvolvimento (P&D) apenas com esse fim. Neste mesmo período, surge a indústria de vídeo games (a Atari, a primeira destas, é fundada em 1971), que será responsável por exigir cada vez maior miniaturização das placas, circuitos, capacitores, transistores e outros

---

<sup>8</sup> **Nota do Autor:** Eckert e Mauchly, mesmos engenheiros que criaram o ENIAC e o EDVAC, fecharam um acordo com o Censo americano para desenvolver um computador semelhante ao ENIAC, esse computador foi chamado de UNIVAC (*Universal Automatic Computer*). Nos dois anos após o fim da SGM as tecnologias paralelas utilizadas para o melhoramento de computadores não tinham se estabilizado, as linhas de *delay* de mercúrio não eram confiáveis o bastante para uso comercial nos computadores. Além disso, o problema da entrada dos dados nos computadores tinha de ser resolvido e para isso os engenheiros desenvolveram uma fita magnética, mas para implementar essa tecnologia era necessário desenvolver uma fita especial e servomotores que agentassem uma rápida aceleração e uma freada brusca. No começo de 1951 o UNIVAC estava funcionando produzindo um calor de 120 kilowatts ocupando o segundo andar do prédio da empresa. Outros computadores foram sendo desenvolvidos, como o Model 702 da IBM, empresa que apesar de chegar atrasada ao mercado de computadores, conseguiu se manter enquanto outras empresas faliram. No final dos anos 1950, a IBM já dominava quase hegemonicamente o mercado, principalmente com o IBM 1401, que já usava transistores, ao invés de tubos, e uma memória interna, ao invés de fitas magnéticas. Também era acompanhado de uma unidade de impressora (IBM 1403), sendo também muito menor do que os computadores anteriores. Ao final dos anos 1960, um computador IBM podia ser encontrado em todas as grandes firmas e indústrias dos EUA e em muitas outras ao redor do mundo, o que impulsionou o mercado de programação, uma vez que programando para um IBM você com certeza alcançaria uma grande parte do mercado.

componentes. Este processo de miniaturização foi um dos maiores facilitadores para a invenção do computador pessoal (Campbell-Kelly *et al.*, 2018, p. 224).

Desde Babbage as inovações no setor de computação passaram por intervalos menores. Mas, até a invenção do Computador Pessoal (um microcomputador que passava a ser utilizado por usuários comuns) os intervalos não foram tão curtos e com saltos tão gigantescos. Há pouco mais de uma década era comum o uso de CD-ROMs como mídias de armazenamento de dados e hoje dificilmente um computador (agora portátil) possui um leitor de DVD ou Blu-Ray (os substitutos derradeiros do CD-ROM). Somente nos últimos quarenta anos vimos: o desenvolvimento de um sistema operacional base para o computador e a conseqüente proliferação de soluções diferentes para cada equipamento ou tipo de usuário, uma interface visual de interação usuário – máquina, a guerra de desenvolvimento de software, e conseqüentemente de hardware, entre a Apple e a Microsoft / Intel, a abertura da internet para o mundo, a corrida por novos jogos e suas capacidades de realismo, a tecnologia de Realidade Virtual, a miniaturização das memórias RAM e ROM, a reposição dos HDs por SSDs, a invenção do computador quântico entre outros. Esse processo de inovação, e conseqüente obsolescência das tecnologias anteriores, pode ser exemplificada ao se perceber que a mesma capacidade de um computador da década de 1960 pode ser encontrada em um *Raspberry Pi* (um Single-Board Computer ou Computador de placa única) ou outro tipo de microcontrolador, que possuem o tamanho de um cartão de crédito e custa entre 5 a 10% do valor de um notebook de uso doméstico.

## 1.2. O Algoritmo e as Linguagens de Programação

Neste ponto do texto é importante definir o conceito de “algoritmo”, “programa” e “linguagem de programação”, uma vez que tem se tornado corriqueiro o uso indiscriminado dos mesmos como sinônimos. Nesta tese tem-se o “algoritmo” como qualquer série de instruções que se utilizam de dados de entrada e produzem (calculam?) um resultado possível de ser analisado. Um “programa” ou “código” possui um ou vários algoritmos, também chamados de rotinas (algoritmos principais) e sub-rotinas (algoritmos derivados dos primeiros). Uma linguagem de programação é o conjunto de expressões que obedece a regras semânticas e de sintaxe para construir um programa que irá fornecer ao computador instruções para manipulação dos dados.

O desenvolvimento técnico do hardware computacional se deu, em grande parte, por seus usuários terem enxergado o potencial de uso geral para o aparelho, levando-o de uma máquina de calcular automática, para um aparelho manipulador de códigos, uma máquina informacional (cf. CAMPBELL-KELLY *et al.*, 2018; MOLES, 1990). Ontologicamente esta se insere como uma nova

categoria de máquina, pois o significado mais difundido sobre a máquina é oriundo ainda da Revolução Industrial e se trata, segundo Flusser (2011), do instrumento que passa a ser “técnico”:

Instrumentos são prolongações de órgãos do corpo: dentes, dedos, braços, mãos prolongados. Por serem prolongações, alcançam mais longe e fundo a natureza, são mais poderosos e eficientes. Os instrumentos simulam o órgão que prolongam: a enxada, o dente; a flecha, o dedo; o martelo, o punho. São “empíricos”. Graças a revolução industrial, passam a recorrer a teorias científicas no curso de sua simulação de órgãos. Passam a ser “técnicos”. Tornam-se, destarte, ainda mais poderosos, mas também maiores e mais caros, produzindo obras mais baratas e mais numerosas. Passam a chamar-se “máquinas” (FLUSSER, 2011, p. 39).

As potencialidades de funcionamento do computador, catalisadas pelo algoritmo e as linguagens de programação, permite-nos perceber que eles não são máquinas no sentido do termo industrial. Para Flusser

[...] os aparelhos, embora produtos industriais, já apontam para além do industrial: são objetos pós-industriais. Daí perguntas industriais (por exemplo, as marxistas) não mais serem competentes para aparelhos. A nossa dificuldade em defini-los se explica: aparelhos são objetos do mundo pós-industrial, para o qual ainda não dispomos de categorias adequadas (FLUSSER, 2011, p. 40).

Desta maneira, historicamente, o desenvolvimento de novas potencialidades de hardware levou a novas possibilidades de instrução a serem direcionados a seu funcionamento, criando assim funcionalidades e conseqüentemente a necessidade de linguagens de programação mais complexas. Convém salientar que essa complexidade se dá pelas possibilidades de interação humano-computador, porque inclusive a prática da codificação é facilitada por aplicativos como: os de autocompletar as linhas de código, as orientações por cores, designs de organização de rotinas e sub-rotinas.<sup>9</sup>

Podemos então afirmar que a complexidade dos *outputs* não necessariamente se comparada a uma codificação extremamente complexa. Como indício ver-se-á ao final do século XX as linguagens programadas visualmente por meio de caixas e ligações entre elas (denominada de *Low Code* ou *No Code*), estas linguagens contam com uma interface cada vez mais amigável sem necessariamente ser digitada qualquer linha de código.

O algoritmo, antes algo concentrado dentro de um aparelho frio em uma sala separada em uma grande empresa, universidade ou centro de pesquisa, passa a ser onipresente na vida de cada indivíduo do planeta: em aparelhos computadorizados cada vez mais miniaturizados (*tablets*, *smartphones*, *smartwatches*) e passa inclusive a comandar filtros empíricos do usuário para o Mundo, determinados, no início, pelos nossos hábitos (aprendizado de máquina) e depois os

---

<sup>9</sup> **Nota do autor:** Recentemente foi lançada um Add-On chamado “GitHub CoPilot” que utiliza as informações de um repositório (GitHub) analisados por uma Inteligência Artificial (IA) que compara com o código que está sendo escrito pelo desenvolvedor e dá sugestões para completar com sub-rotinas já completamente desenhadas. Cf.: <https://copilot.github.com/>.

reforçando-os (inteligência artificial), toma conta das Redes Sociais digitais, dos anúncios nos sites e até mesmo em e-mails que recebemos (cf. COUCHOT, 2003, p. 12).

Essa onipresença algorítmica também colabora para a necessidade de investigarmos mais à fundo esse fenômeno e a necessidade é tão palpável que vários países europeus já cogitam implantar ou já implantaram em seu currículo de Ensino Básico a Introdução à Linguagem da Programação, ou seja, estamos já nos preparando para uma virada no conhecimento humano em que aparelhos e seres humanos se aproximarão cada vez mais na sua comunicação.

O computador se torna um coautor de feitos do mundo, e assim conhecer e entender as linguagens de programação nos permitem lidar com os limites impostos pelos algoritmos, escapar das amarras da linguagem de programação e, como artistas, tentar subverter esses limites forçando o aparelho a funcionalidades não previstas ou demonstrando potencialidades criativas de seus desenvolvimentos.

Como veremos adiante, programar os primeiros computadores era um trabalho extremamente técnico. Fosse um equipamento mecânico ou elétrico sua programação muitas vezes estava concentrada em manipular partes do corpo central do computador (mudanças de entradas de fios, modificação de válvulas, botões ou dimmers (botões que aumentam a intensidade de algum sinal rotacionando-o em sentido horário e diminuindo no sentido contrário) e sendo tão delicado e de conhecimento tão específico, o acesso ao equipamento e ao conhecimento eram limitados a poucas pessoas.

O desenvolvimento das tecnologias de processamento de dados tornou cada vez mais dispendioso as potencialidades de cálculo que eram desperdiçadas com este tipo de entrada única de dados<sup>10</sup>, disso surge a necessidade de ampliar o uso da máquina e de torná-la mais facilmente reprogramável.

A modificação da interface de entrada e saída de dados foi uma das etapas, uma segunda etapa estava em tornar a linguagem de programação dos aparelhos mais próxima da comunicação dos seres humanos. Os termos linguagem de baixo nível é usado para as linguagens mais próximas do entendimento do funcionamento estrutural do aparelho e linguagens de alto nível é usado para as linguagens escritas semelhante as linguagens verbais.

Para Warren Sack (2019), professor da Universidade da Califórnia e autor do livro “*The Software Arts*”, a linguagem de programação trata-se de uma tradução entre a linguagem humana e a linguagem do computador (SACK, 2019, p. 5–30).

Historicamente, é possível dizer que esta posição foi primeiramente rascunhada no século dezessete com o propósito de desenvolver uma linguagem artificial e filosófica

---

<sup>10</sup> **Nota do autor:** um exemplo seria um aparelho com um potencial de cálculo  $n$  que em sua única entrada de cálculo é inserido um cálculo mais simples, mas trabalhoso, que demanda pouca capacidade de processamento, em nosso exemplo imaginário um esforço  $n/100$ , mas que ocupa o tempo de trabalho do aparelho, leia-se enquanto ele não terminar este cálculo, nenhum outro pode ser inserido.

que seria usada para unir as artes liberais (por exemplo, lógica, gramática e retórica, as artes liberais da linguagem) e as artes mecânicas (por exemplo aquelas praticadas por artesãos nas oficinas produzindo alfinetes, meias, fechaduras, armas e jóias). Resumindo, essa linguagem artificial se tornou o que nós hoje conhecemos como linguagens de programação para computador (op cit., idem, p.2)

Um ponto a se considerar para reflexão está na tradução da palavra “Translate” utilizada por Sack: que pode ser traduzida em português como “Tradução” ou como “Translação” dando um sentido de movimento, que também pode ser uma boa metáfora para a relação Humano x Computador, pois como poderemos ver mais à frente, não se trata de uma simples decodificação sígnica, mas de todo um entendimento de como o computador “vê” aquele dado que entra no sistema.

Devemos então retomar a história de como o algoritmo e as linguagens de programação evoluíram junto com a própria linha de desenvolvimento do hardware e como isso modificou o status do computador de máquina para coautor de trabalhos criativos.

### **1.2.1. Primórdios da Linguagem de Programação**

Como apresentado anteriormente, os equipamentos de computação mecânicos como a Máquina Diferencial, eram capazes de apenas um tipo de cálculo, assim como computadores analógicos eram utilizados para apenas uma função. Seu programa, tomado como no conceito flusseriano (FLUSSER, 2011), já continha com sua própria programação e qualquer mudança em suas funções algorítmicas, envolveria criar uma nova máquina. Já a Máquina Diferencial atuaria realizando seus cálculos utilizando cilindros marcados com numerais como *input* e *output*, ou seja, os números bases eram inseridos acima e ao lado podia-se ver o resultado.

A necessidade de se criar uma máquina universal, continha a necessidade que a mesma pudesse aceitar novas instruções para lidar com os dados inseridos de maneira a atender a todo e qualquer cálculo, fossem tabelas de marés, de estrelas, ou do censo.

Inspirado, então, no Tear de Jacquard, Charles Babbage projetou a Máquina Analítica, como citado, para que a mesma recebesse suas instruções de funcionamento por meio de cartões perfurados. Ada Lovelace então descreveu um conjunto de códigos de cartões perfurados para a Máquina Analítica no único documento escrito sobre a mesma “Rascunho do Engenho Analítico”<sup>11</sup>.

Sack (2019, p. 57) apresenta a ideia de que o trabalho de Babbage e Lovelace foi a tradução de uma “linguagem de trabalho de operações” para uma “linguagem de máquina”, o que Lovelace deu o nome de “ciência de operações”. Sack parte do princípio que ao escreverem a “*Encyclopédie*”, Diderot, D’Alambert e os outros que trabalharam com eles, precisaram criar uma forma de tornar escrito o que os artistas falavam de seus trabalhos e máquinas.

---

<sup>11</sup> Traduzido pelo autor do original “*Sketch of the Analytical Engine*”.

Meu argumento é que Diderot, d’Alambert, e os enciclopedistas são com certeza os parteiros da linguagem para descrever o trabalho de artesãos e suas máquinas e isso, com o correr dos séculos, se tornou a raiz do que hoje conhecemos como linguagens de programação. Ao imprimir como texto e diagramas o que os artesãos apenas falaram na oficina, os enciclopedistas parearam as artes mecânicas e as liberais. (SACK, 2019, p. 60)

Os termos “linguagem de trabalho” e “linguagem de máquina” são denominações usadas por Sack para determinar duas estratégias utilizadas pelos enciclopedistas franceses para inserir no compêndio os trabalhos de artistas e artesãos (arte para Sack é utilizada como na Grécia de Sócrates). O primeiro termo ele define como “a linguagem – o texto e a fala – utilizado para descrever os processos e produtos do trabalho” (SACK, 2019, p. 57) e o segundo termo é “a ‘linguagem de trabalho’ aplicada no design e análises de máquinas” (*op cit., idem, ibidem*). E ainda alerta que “[q]uando uma máquina é designada para substituir um humano no processo de trabalho, as ações performadas pelo humano devem ser traduzidas em linguagem de máquina” (*op cit., idem, ibidem*).

É importante citar, neste momento, que a visão de Ada Lovelace e Babbage sobre suas máquinas de computar eram de máquinas capazes apenas de fazer o que lhe era ordenado fazer (GIANNETTI, 2006, p. 29). “O Engenho Analítico não tem a menor pretensão de originar nada. As coisas que sabemos como lhe ordenar fazer, essas sim, pode fazê-las todas” (Lovelace apud Giannetti, 2006, p. 29). O problema colocado para ser resolvido até então pelo Computador era o cálculo<sup>12</sup>.

A máquina de Turing apesar de ser chamada de universal também apenas realizava cálculos baseados em sua estrutura eletromecânica. Apenas em seu artigo de 1950, “*Computing Machinery and Intelligence*” que Turing iria mencionar a possibilidade de o computador aprender (o que seria o computador tomar decisões a partir dos resultados que ele encontra), todavia, apenas 50 anos depois foi criado um computador que pudesse armazenar dados o suficiente para passar pela Prova de Turing e entrar no Jogo da Imitação. O algoritmo computacional só encontrou uma evolução quando houve uma mudança na percepção do que o computador podia fazer, de uma máquina de calcular para uma máquina de linguagem. O ENIAC, o EDVAC, o UNIVAC, todos foram computadores cujo único objetivo era realizar cálculos matemáticos, com operações cada vez mais complexas tornando-as naquele momento calculadoras de toneladas de peso.

---

<sup>12</sup> Mas, as decepções da vida de Babbage ao não ter construído efetivamente a Máquina Analítica e por todos os exemplares de similares à Máquina Diferencial não terem conseguido suplantar os requisitos de sua demanda de mercado, fez com que afinal os cientistas e engenheiros do século XIX e início dos XX afastassem-se do caminho digital e começassem a construir modelos e máquinas de comportamento análogos para estudar determinados fenômenos, o que hoje conhecemos como computação analógica (Campbell-Kelly *et al.*, 2018, p. 46).

É importante destacar um fato sobre estes computadores iniciais de que eles não possuíam nenhum sistema de instrução de funcionamento internalizado, o que conhecemos hoje por Sistema Operacional (SO). As instruções de funcionamento e os números a serem calculados eram inseridos todas as vezes, mesmo que o mesmo programa fosse rodado com números diferentes, a fita deveria repetir todo o programa.

### 1.2.2. Programas Armazenados e como resolver problemas no código

O primeiro computador a possuir seu conjunto de instrução de funcionamento previamente armazenado foi o EDSAC da Universidade de Cambridge, mas desenvolver os programas para que o computador era um trabalho extenuante (CAMPBELL-KELLY *et al.*, 2018, p. 167): após escrever os programas o programador deveria testar e verificar, ao encontrar erros (fato corriqueiro) o indivíduo deveria retornar ao equipamento de perfuração para localizar os erros (o tamanho das máquinas às vezes as colocava em lados diferentes de um prédio). O fato de o computador ter um programa de instruções previamente guardado auxiliava na localização destes erros, mas exigia este trabalho de teste e reteste. O erro passou a se chamar *Bug* (existe a história de um inseto, em inglês “bug”, que impediu o funcionamento de uma máquina ao vir a morrer dentro do mecanismo interno) e o processo para corrigir esses erros foi então chamado de Debug.

A criação de softwares e a resolução dos *bugs* que surgiam se tornava cada vez mais complexo conforme a tecnologia mudava. Tornou-se comum a todos os computadores seguir um padrão binário para suas instruções internas; o 0 significa o “Não” e o 1 o “Sim”, isso em termos físicos corresponde a sinais elétricos, o 1 indica a presença do sinal, o 0 a ausência. Multiplique isso por todas as possibilidades de entrada de dados de um computador e agora minimize isso a ponto de caber em um processador como os processadores atuais que podem ter 1 cm<sup>2</sup> de tamanho ou menos<sup>13</sup>.

Em Campbell-Kelly *et al* está um exemplo sobre como funciona essa correlação entre o que o programador deseja e como o computador deve “enxergar”: “Por exemplo no EDSAC, a instrução simples ‘adicione o número curto na memória de localização 25’ era armazenado como 11100000000110010.” (CAMPBELL-KELLY *et al.*, 2018, p. 168). Obviamente cálculos mais complicados e longos ficariam inviáveis para serem escritos com instruções assim, então os programadores passaram a utilizar notações para referenciar a instruções. No exemplo acima, os programadores do EDSAC utilizariam a seguinte notação: **A 25 S**, “A” para adição, 25 a localização da memória, e “S” para indicar ser um número curto.

Maurice Wilkes e David Wheeler foram os responsáveis pela escrita e adoção do “*Initial Orders*” (IO), Instruções Iniciais que eram inseridas na memória do computador (op. cit., p. 168). O

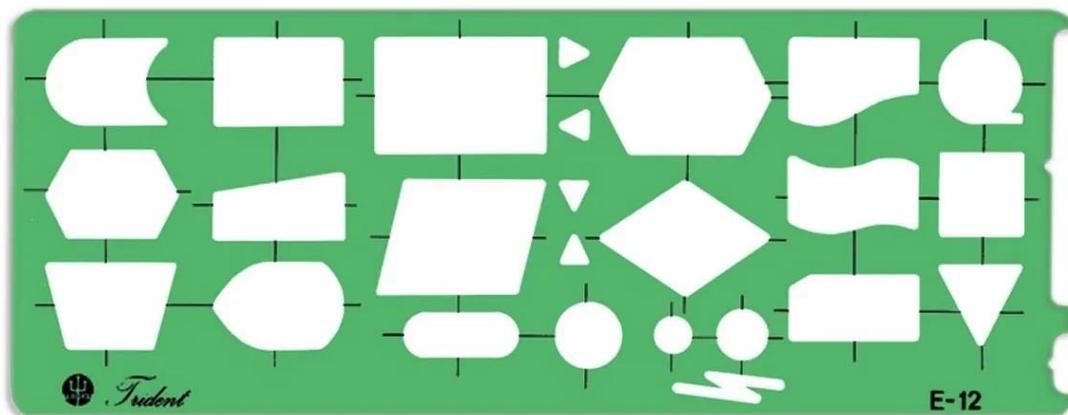
---

<sup>13</sup> A título de curiosidade, um processador Intel i7 de 11ª Geração possui nada menos que 3 bilhões de transistores.

IO seria carregado antes de qualquer outro programa na memória do computador e teria a responsabilidade de transformar as notações em código binário. Mesmo com esse procedimento inicial automatizado persistia a possibilidade do erro. Em termos atuais os erros em um código podem representar um pouco de atraso em uma aplicação, mas na época em que um único computador existia para toda uma empresa, universidade ou até mesmo país e havia uma única entrada de dados, o “tempo de máquina” era muito caro, e todos os novatos da programação iriam passar pelo mesmo ritual de aprendizado de programação que era cometer erros banais que iriam criar um processo lento e custoso.

O Grupo de Cambridge então chegou ao conceito de “biblioteca de sub-rotina”, já que perceberam que em todos os programas várias instruções eram comuns e rotineiras; desta maneira, o grupo percebeu que se estas operações fossem convertidas em pequenos programas que pudessem ser armazenados e quando necessário copiado pelos programadores para seus próprios programas.

Mesmo com a notação e com as bibliotecas de sub-rotina, ainda seria muito complicado escrever um programa sem se perder diante de números e letras. Para facilitar foi utilizada uma técnica de fluxogramas que visava traduzir os problemas a serem resolvidos pelo computador em passos a serem cumpridos pelo mesmo (algoritmo). A primeira vez que a técnica foi descrita foi em “*Process Charts*”(GILBRETH; GILBRETH, 1921) do casal Frank e Lilian Gilbreth e serviria para estudar todos os processos e maximizar o tempo de produção. Foi von Neumann quem começou a aplicar a técnica aos programas de computador em 1950 (Campbell-Kelly *et al.*, 2018, p. 170).



**Figura 3:** Gabarito de fluxograma para programação. Fonte: Acervo próprio.

Assim, uma série de formas geométricas, linhas e setas que as ligam podiam criar uma representação gráfica dos passos dos processos de um algoritmo e seu processamento por um computador (cf. **Apêndice II**). Pontos de Decisão são espaços de bifurcação do código (*If Then*) e representados por losangos, Entradas (*Inputs*) e Saídas (*Outputs*) são simbolizados por

paralelogramos, ações como transporte (setas grossas), armazenamento em banco de dados ou memória (triângulo pequeno para baixo), extrair esses dados do banco de dados ou da memória (triângulo pequeno para cima), todas encontram algum tipo de representação gráfica que ajudava na organização das notações dentro do programa.

Conforme o potencial de processamento e a memória dos computadores se desenvolviam, o IO de Wilkes e Wheeler já não dava conta de todas as variáveis, assim os vários laboratórios de pesquisa e indústrias de computadores começaram a pensar em sistemas de programação automática que permitiria os programadores a escreverem um programa como se estivesse escrevendo em um tipo de linguagem parecida com o inglês e o computador mesmo se responsabilizasse de transformar esses códigos em código binário para a sua instrução básica. Neste caso, este sistema tinha como objetivo facilitar a transferência da ideia do programa do fluxograma para um programa escrito.

Em 1953 – 1954, um pesquisador da IBM (John Backus) produziu um sistema chamado “*Formula Translator*”, que ficou conhecido como FORTRAN a primeira “linguagem de programação” propriamente dita, tinha como importante característica a de que podia ser utilizada em qualquer computador que possuísse um tradutor, independente da marca do equipamento. FORTRAN se tornou então a linguagem de programação científica mais importante da história da computação até hoje, devido a forma como se tornou padrão na indústria.

Até então não havia espaço no mercado para programadores individuais em suas casas, já que toda empresa ao comprar um computador também contratava seus próprios programadores. Com o tempo isso foi ficando muito caro, o número de programadores era escasso e o seu salário subia junto com a demanda, então muitas empresas começaram a comprar programas já prontos ao invés de escrever seus próprios. A criação destes mercados e o desenvolvimento de mais linguagens de programação ajudaram nos anos 1960 – 70 na criação de uma cultura de programação fora de grandes empresas, que junto a uma cultura “hacker” que se formou, levou ao desenvolvimento de computadores domésticos.

### **1.2.3. Programação como ciência, computação como hobby.**

No final de 1950 e início de 1960, a indústria de computadores ficou “refém” dos programadores, uma vez que eles eram poucos no mercado e a experiência necessária para desenvolver um programa de computador que não fosse apenas um desperdício de tempo de processamento (e com isso desperdício de salários, energia, equipamentos) era muito alta. O estereótipo do programador “nerd” de sandálias, barba desleixada e roupas não formais demonstrava como estes se sentiam endeusados e sabiam que as empresas não podiam preterir deles, sendo comum que histórias de rompantes e histeria destes mesmos indivíduos fossem

contadas com desfecho semelhante em que as empresas acabavam sendo extremamente tolerantes.

Nesta mesma década, para tentar resolver essa falta de oferta da mão de obra, iniciaram-se os cursos de Ciência da Computação nas universidades norte-americanas, mas ainda a experiência era um ponto de gargalo. De maneira que a indústria e a pesquisa acadêmica tornaram como ponto de atenção principal a diminuição desta experiência necessária para criar programas complexos.

O FORTRAN, o COBOL, e outras linguagens demonstraram que o próprio computador deveria se tornar um aliado na criação de programas e sub-rotinas para diversas atividades humanas, mas ainda se criavam programas em caráter de experiências empíricas. Assim, começou a se difundir a ideia de aplicar conhecimentos e técnicas da engenharia na programação, surgindo o conceito da “Engenharia de Software” e da “Metodologia de Design Estruturado”.

Os defensores da engenharia de software enfatizam a necessidade de impor disciplina industrial à prática informal e idiossincrática dos programadores. Eles rejeitaram a noção que grandes projetos de softwares eram inerentemente ingeríveis e recomendavam, todavia, que desenvolvedores de software adotem métodos e técnicas emprestadas da indústria tradicional. O objetivo definitivo seria um tipo de “fábrica de software” completo com partes intercambiáveis (ou “componentes de software”), produção mecanizada, e uma grande força de trabalho amplamente desqualificada e rotinizada. As ferramentas usadas para alcançar esse objetivo incluíam design estruturado, métodos formais, e modelos de desenvolvimento.

(CAMPBELL-KELLY *et al.*, 2018, p. 185).

Outro ponto importante para esta tese foi a mudança de paradigma sobre a escrita de um programa: inicialmente via-se um software como um projeto de único uso, ele continuava sendo executado até não ser mais útil, depois que o problema investigado se tornava mais complexo um novo programa deveria ser escrito do zero. Assim o conceito de desenvolvimento e verificação foi criado e a programação tornou-se mais orgânica, se desenvolvendo conforme o programador recebia o *feedback* de seu uso.

Neste mesmo período iniciou-se a “Revolução da Miniaturização” e pouco a pouco os gigantescos computadores passaram a serem substituídos por versões menores e mais potentes e as estações de trabalho ligadas à um gigantesco núcleo davam lugar aos microcomputadores pessoais que agora começavam a entrar nas residências e a serem disponibilizados nas universidades e depois nas escolas, devido as possibilidades exploradas de redes. A linguagem BASIC também se espalhou e novas experiências de uso do computador começaram a ser testadas.

O primeiro computador para “*hobbie*”, o Altair 8800 chegou ao mercado em 1975. Era vendido pelo correio e vinha desmontado para que o entusiasta pudesse comprar, mas não havia nenhum mecanismo de *input* amigável, a programação dele era feita em código binário diretamente

em uma série de alavancas que ficavam enfileiradas na parte frontal do aparelho, e o único *output* do aparelho eram luzes que mudavam o padrão quando o programa rodava corretamente.

Como equipamento não teve uma representatividade expressiva, mas iniciou uma cultura de modificação (*Maker e Hacker*) já que grupos de entusiastas passavam a desenvolver periféricos e “add-ons” para inserir novas funcionalidades ao equipamento básico, como: um display para visualizar os comandos de entrada e dados de saída; melhoria na capacidade de memória; um acessório para gravar os dados em fita magnética, entre outros.

Dentro desta cultura Hacker que em 1975, Steve Wozniak e Steve Jobs, montaram o primeiro computador Apple baseado em um chip processador da MOS Technology. Este era tão básico quanto o Altair 8800, não dispo de sequer de uma fonte elétrica própria, mas em 1976, já haviam vendido 200 equipamentos para a comunidade entusiasta, e com isso Jobs percebeu a existência de um mercado específico para esses microcomputadores que não apenas o de curiosidade para entusiastas, mas um equipamento para ser utilizado em casa para entretenimento e auxílio nas tarefas domésticas.

Para tanto, ele deveria parecer com um eletrodoméstico: teriam de ser uma unidade envolta em um case plástico, dispo de um teclado para *input*, ter a possibilidade de no mínimo ser ligado a uma tela (uma televisão, por exemplo) para visualizar o resultado da computação e ter algum tipo de memória de longa duração para armazenamento de dados; mas o aspecto mais importante neste ponto da discussão era que devia ter um software que pudesse ser apelativo ao público não-entusiasta.

Ainda não se tinha muita noção das potencialidades do computador, mesmo porque ainda não existiam muitos softwares de uso específico, sendo a maioria criada por seus próprios usuários para resolver problemas pessoais. Os primeiros programas prontos a serem comercializados de maneira mais ostensiva foram os *games* (jogos) puxados pela comunidade de videogames como o Atari. A possibilidade de tornar o computador um equipamento multiuso e o uso do mesmo para jogos permitiu que as pessoas passassem a se acostumar com o ambiente computacional e sua interface.

Aos poucos softwares para diversos usos foram se firmando no mercado retirando a utilidade de várias máquinas de uso singular e agregando mais funções ao computador: um software de tabelas financeiras que substituía as calculadoras; e um processador de texto com tecnologia “WYSIWYG” (*What You See Is What You Get*, ou o que você vê é o que você tem) em que na tela aparecia o texto exatamente como ficaria no papel impresso.

Alguns anos mais tarde, o microcomputador deixou de ser um equipamento apenas para entusiastas, caiu nas graças do público comum que começou a perceber todas as potencialidades de uso para o computador. Neste momento, grandes empresas começaram a produzir os PC’s

(*Personal Computers*): IBM, Dell, Compaq, Olivetti, Hitachi, Toshiba e Gateway; além da Apple (não tão expressiva quanto os gigantes formados no início da história do computador comercial) que produzia seus computadores próprios.

### 1.2.4. Conexão, Internet e Multimídia: o computador no centro de tudo

Até então, o output visual de um computador era “*All-Type*” ou seja, a saída de todos os resultados era representada nas telas em formato de caracteres e mesmo imagens eram criadas pela ilusão visual da disposição de caracteres do código ASCII, hoje chamada de Imagem ASCII com uma linha artística destinada a explorar esse segmento: a ASCII Art (**Figura 6 e Figura 7**).

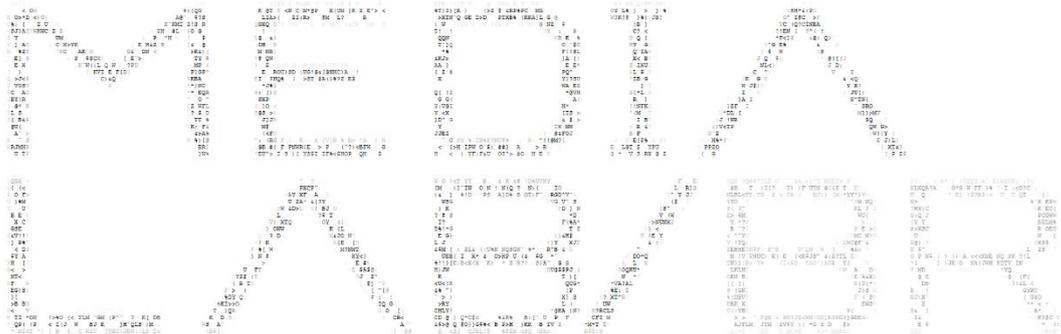


Figura 4: Imagem em ASCII gerada a partir do site [www.text-image.com](http://www.text-image.com).



Figura 5: Ampliação do M da imagem anterior.

Neste período o usuário se comunicava com o computador por meio do que é chamado de *Disk Operating System* (DOS), sendo o mais notório dentre eles o MS-DOS da Microsoft. Mas em 1980, foi desenvolvida uma implementação aos sistemas operacionais para torná-los mais amigáveis ao público comum: a GUI, *Graphical User Interface*, Interface Gráfica de Usuário. Os usuários de Apple Macintosh já vinham usufruindo deste ambiente, mas os usuários de PC's só passaram a utilizar essa modalidade com a criação do Windows.

O uso do DOS era feito por meio de linhas de comando, que exigia um bom tempo de aprendizado, um certo tempo para digitar todas as linhas necessárias às atividades específicas e esteticamente era limitado. Enquanto no sistema gráfico, a visualização de todo o conteúdo de um diretório pelo usuário dava-se na inicialização do software gestor de arquivos por meio de um duplo clique do mouse no ícone do mesmo (geralmente representado por uma pasta). Para termos de comparação, de forma a obter o mesmo resultado, o usuário do DOS precisava digitar um comando semelhante a **DIR C:\NOME DA PASTA\NOME DA SUBPASTA**. Qualquer letra ou símbolo errado ou fora de lugar o computador respondia com uma mensagem de erro e bastava ao usuário ter de repetir todo o procedimento.

O GUI era a estratégia para criar uma experiência mais natural do usuário no uso do computador. Apesar de haver um período de adaptação do usuário com a relação dos movimentos do mouse e do ponteiro (movimentos horizontais num plano horizontal que se transformam em movimentos em um plano vertical) e com a entrada de dados pelo teclado (cursos de digitação eram comuns no início da inserção dos computadores domésticos), bem como os atalhos do mesmo, ainda assim era mais rápido e intuitivo do que digitar uma série de linhas de código e se perder diante de informações dispostas de maneira monocromática (CAMPBELL-KELLY *et al.*, 2018, p. 258).

O GUI era as vezes conhecido como interface WIMP, que representava seus componentes críticos: Windows [Janelas], Icons [Ícones], Mouse e Pull-down menus [Menus Suspensos]. A ideia principal na interface era a de usar a “metáfora de desktop” para qual as pessoas comuns poderiam reagir, e para qual estava distante de uma computação técnica. A tela mostrava um desktop idealizado no qual pastas e documentos e instrumentos de escritório, como cadernos de notas e calculadoras. Todo os objetos no desktop eram representados por “ícones” - por exemplo, um documento era representado por uma pequena imagem de uma página de digitação, uma imagem de uma pasta representava uma série de documentos e por assim vai. Para examinar um documento, o indivíduo simplesmente usaria o mouse para mover o ponteiro na tela para selecionar o ícone apropriado; clicando no ícone iria fazer com que uma “janela” se abrisse na tela para que o documento pudesse ser visto. Quanto mais documentos eram selecionados e abertos, as janelas se amontoavam, assim como documentos em uma mesa real iriam acumular um sobre o outro. (Campbell-Kelly *et al.*, 2018, p. 258).

Por volta de 1989, os desenvolvimentos tecnológicos da indústria de software iriam se descolar da indústria de hardware em termos de lançamento – até então qualquer versão nova de

um software esperava o lançamento de um novo hardware – com isso, o ciclo de obsolescência se tornou mais rápido:

Muito do crescimento acelerado da indústria de software no final dos anos 1980, incluindo o da Microsoft, era devido a sempre crescente sofisticação de seus produtos. Pacotes de aplicações eram rotineiramente atualizados mais ou menos a cada dezoito meses, gerando receita adicional enquanto os usuários trocavam por pacotes mais sofisticados, fazendo melhor uso do hardware rapidamente melhorado. Windows não era uma exceção. (Campbell-Kelly *et al.*, 2018, p. 266)

As consequências deste ciclo de obsolescência para a Arte Computacional serão descritas mais tarde. Por hora, cabe caracterizar o ciclo: Uma melhoria na capacidade de um hardware cria um vácuo de processamento/memória, ou anula qualquer tecnologia de software existente, fazendo com que a indústria de software crie novos produtos, começando por atualizações de sistemas operacionais, depois mudanças em sistemas operacionais exigem mudanças nos softwares específicos e mudanças nos softwares específicos pode ocasionar com que arquivos iniciados em versões anteriores possam não ser mais abertos, como na situação da professora Tânia Fraga descrita na introdução deste trabalho.

Como percebemos, as mudanças obtidas no *hardware* e no *software* transformaram o computador de uma máquina gigantesca feita apenas para grandes empresas em um equipamento multiuso que passava a se ocupar da gestão de informações, do entretenimento, além de cálculos. Mas um fato tornou a convergência entre as linhas de desenvolvimento da computação e das mídias mais evidente: o surgimento da Internet nos anos 1990.

### 1.3. Os laços da humanidade

Antes do computador se tornar algo que possamos carregar no bolso, que controla os sinais de tráfego, câmeras de vídeo pela cidade, inserido nos automóveis para ajudar a gerenciar a energia elétrica e o combustível; o computador era um objeto único e quase que totalmente *offline*. Mas o potencial de reserva de dados, compartilhamento de pesquisas, compartilhamento de capacidade de processamento de informações, representado por diversos microcomputadores espalhados pelo mundo, quando considerado como um único fenômeno, era muito promissor na sua exploração.

O desenvolvimento de uma forma de conectar os computadores espalhados não mais em um prédio, mas em todo um país ou entre nações seria um dos pontos-chaves do que se chamou de Novas Tecnologias da Informação e da Comunicação (NTIC's).

Desde a I Guerra Mundial que o desenvolvimento das tecnologias da comunicação tomou dianteira na importância de pesquisas científicas e tecnológicas. A passagem dos cabos de telégrafo entre os continentes diminuiu o tempo de resposta entre governos e empresas em meses (uma carta, se chegasse, levava meses para sair do interior dos EUA até a Europa). A rede

telefônica também se estendeu muito depois das Guerras Mundiais, aproveitando inclusive os cabos telegráficos, mas os maiores desenvolvimentos pré-internet foram do Rádio e da Televisão. O rádio foi uma das primeiras tecnologias de comunicação de massa, tendo tido grande importância na I Guerra Mundial e se popularizando após isso, fazendo com que em questão de décadas todos os lares possuíssem pelo menos um aparelho radiofônico.

Em seguida tivemos a inserção da televisão que trouxe a voz do rádio junto com a imagem do cinema e se aproveitando do apelo visual a televisão e o cinema se tornaram importantes veiculadores da cultura ocidental para o mundo todo. A distribuição das redes de televisão e a influência militar trouxeram recursos para criar uma rede de satélites artificiais em torno do planeta, aos poucos os países de cada lado dos blocos econômicos da Guerra Fria, utilizavam a corrida espacial para enviar satélites que ampliavam o alcance das ondas de televisão e depois das redes de telefonia. E foi exatamente por conta da Guerra Fria que surgiu a Internet ou a Grande Rede Mundial de Computadores.

### **1.3.1. Internet e a onipresença digital**

Desde antes da *Encyclopédie* de Diderot, D'Alambert e outros, que o mundo sonhava com um repositório de todo o conhecimento humano. A Internet representou a formação desta “Biblioteca de Alexandria Mundial”, uma expansão da “galáxia de Gutemberg” citada por Marshall McLuhan ao destacar a representatividade da difusão do texto impresso para a cultura humana. Para descrever a importância que a Internet representou para a ampliação de toda a comunicação humana, Manuel Castells retornou à invenção do alfabeto (CASTELLS, 2006, p. 413) e a comparou com o a comunicação mediada por computador (CMC) possibilitada pela Internet.

A “formação de um hipertexto e uma metalinguagem que, pela primeira vez na história, integra no mesmo sistema as modalidades escrita, oral e audiovisual da comunicação humana” (CASTELLS, 2006, p. 414) apresenta em termos de experiência estética não apenas uma abertura de exposição de trabalhos artísticos em proporções planetárias, mas também de todo um repertório sógnico específico da vida no mundo digital, assim como acesso instantâneo aos signos das múltiplas culturas e identidades espalhadas pelo Globo e que encontram um representante na rede. A Internet e a CMC se popularizaram pela ação da comunicação entre universidades.

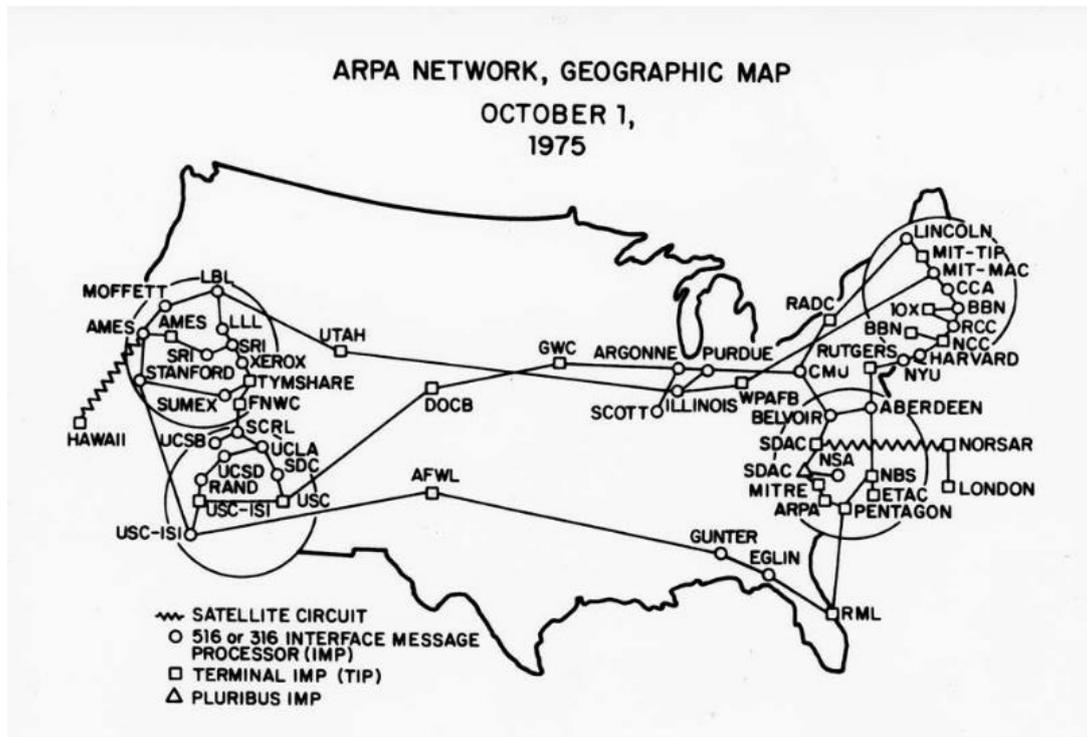
Em meio à Guerra Fria, os EUA temiam que um ataque da União Soviética pudesse desestabilizar a comunicação no país e, no caso de os ataques serem mais abrangentes, informações importantes, não apenas militares, mas também políticas e científicas, pudessem se perder com a destruição dos espaços em que estivessem armazenadas (CASTELLS, 2006, p. 428). Com isso incentivou pesquisas que fizessem com que as informações pudessem ser dispensadas

em vários lugares ao mesmo tempo e que se uma linha fosse interrompida outras linhas poderiam fazer a ponte, mantendo a conexão sempre ativa e as informações sempre preservadas.

Voltando um pouco no tempo, parte da preocupação dos construtores de computadores era poder aproveitar a capacidade de processamento de um computador permitindo que vários terminais pudessem dar entrada em dados e receber suas informações processadas também de maneira separada. Esta ideia evoluiu para um pensamento de terminais ligados à distância (não apenas no mesmo recinto ou prédio do computador) mas em outros prédios, cidades ou mesmo do outro lado do país. Foi assim que por volta de 1966 a tecnologia de redes de computadores estava pronta para ser explorada (CAMPBELL-KELLY *et al.*, 2018, p. 280).

Para evitar uma consequência tão destrutiva quanto a perda do conhecimento, não apenas os órgãos militares estavam inseridos, mas várias universidades norte-americanas, e o projeto Arpanet tinha como objetivo principal a comunicação entre quatro grandes centros de computadores: o da Universidade da Califórnia de Los Angeles e de Santa Barbara, a Universidade de Utah e o Instituto de Pesquisa de Stanford. O projeto Arpanet era um grande projeto do Governo norte-americano para melhorar a capacidade de processamento de informação e cálculos para fins diversos, mas inicialmente motivado por motivos militares tendo sido implementado ainda ao final da SGM. O caso, é que ao envolver as universidades no projeto para a junção destes centros de computação, diversos alunos de graduação e pós-graduação desenvolviam seus estudos dentro deste projeto, o que para Campbell-Kelly *et al* acabou por criar uma cultura ímpar e de certa forma anárquica que foi reforçada pela cultura hacker dos anos 1970 que já vimos anteriormente e que contribuiu para as características iniciais da Internet.

A rede foi pensada então como as redes telegráficas, contendo “nós” que convergiam as informações para então distribuí-las e as universidades passaram a agir como estes nós. Em 1970, já existiam vinte e três nós de hospedagem; em 1994 com a invenção de World Wide Web (WWW baseado em protocolos de localização física de computadores na rede por meio de um endereço virtual) existiam centenas de servidores públicos para acesso à internet só nos EUA e em 1999 a Internet conectava 63 milhões de computadores-servidores (CASTELLS, 2006, p. 431).



**Figura 6:** Diagrama da disposição dos nós da ARPANET em 1975. Fonte: CAMPBELL-KELLY *et al.*, 2018, p. 201.

E assim, podemos concluir com a citação abaixo:

A INTERNET ESTAVA para se tornar indiscutivelmente o mais importante desenvolvimento em computação de meados de 1990 em diante. Mas isso foi possível apenas porque na década anterior o computador pessoal havia se transformado de uma tecnologia não amigável para algo que se aproximava de um eletrodoméstico de informação. Companhias de software providenciaram aplicativos que transformavam o computador pessoal de propósito geral em uma máquina de propósito especial que ia ao encontro das necessidades de usuários específicos – quer se fossem analistas de negócios utilizando as tabelas, indivíduos usando processadores de texto, ou jogadores. Enquanto muitas destas aplicações simplificaram usar um computador – por exemplo, fazendo com que um programa processador de texto se parecesse com uma máquina de escrever – foi a interface gráfica de usuário que abriu o computador para a sociedade em geral. Ao invés de teclar comandos complicados, os usuários poderiam agora simplesmente apontar e clicar por seu caminho por entre o mais complexo software, enquanto desconhece essencialmente tudo sobre o funcionamento interno de um computador. (CAMPBELL-KELLY *et al.*, 2018, p. 273)

O primeiro capítulo representa um esforço de caracterizar o caminho histórico do computador e das linguagens de programação, ressaltando a velocidade cada vez mais célere com que cada uma das tecnologias era substituída por outra no processo de obsolescência. Alguns artistas perceberam as alterações que este aparelho iria produzir e iniciaram suas experiências no campo do que ficou conhecido durante muito tempo como Arte das Novas Mídias (*New Media Art*). No próximo capítulo

iremos explorar, ainda de maneira histórica, a forma como a Arte acompanhou os desdobramentos da tecnologia, e como fez para suplantar parte da obsolescência.

## 2. Segundo Capítulo: A Arte e as Tecnologias da Informação e da Computação

A introdução sobre História do Computador e História da Programação no primeiro capítulo visa contribuir para que tenhamos um exercício descritivo prévio na construção da ontologia da obra de arte computacional. Diante do fato inegável que esta tese é fruto de uma pesquisa em Artes Visuais, também somos confrontados com o fato de que todo este espaço de pesquisa em que estamos envolvidos e que acostumamos a nos referenciar a ele no Brasil como “Arte e Tecnologia”, tem suas raízes em outras linhas históricas, tais como: a da Computação, a do Algoritmo, a das Mídias e da Arte.

Por exemplo, foi no apelo da construção de aparatos mecânicos e depois elétricos e depois eletroeletrônicos que podemos escalonar o desenvolvimento da Câmera Fotográfica, sendo que sua patente foi depositada em 1839, mesmo período em que a Revolução Industrial se fazia vigente (1760 a 1840 mais ou menos). Historicamente, as experiências artísticas acompanharam os desenvolvimentos tecnológicos da sociedade humana e com o desenvolvimento do computador não foi diferente neste sentido.

Para este estudo, o importante retomar o conceito flusseriano de Programa (FLUSSER, 2011) que ele tem uma gênese cultural, não é fruto de uma linguagem hermética baseada na matemática, mas sim uma série de instruções que são passadas culturalmente, ou seja, na criação do indivíduo e na sua própria vivência em sociedade. Convém ressaltar que para o mesmo, a palavra aparelho que hoje comumente ainda vemos associada a um objeto, é na realidade algo mais abstrato, mas podemos conceituar o aparelho como um espaço (físico, virtual ou metafórico) reservado para a atuação dessas instruções.

Um dos importantes marcos para a área “Arte e Tecnologia” foi a invenção de uma câmera fotográfica, para Flusser, apesar de ser mencionada apenas no terceiro capítulo do livro “Filosofia da Caixa Preta” (Flusser, 2011, p. 37) também a câmera aparece como objeto de estudo para um tratado sobre liberdade e sobre o funcionamento da sociedade pós-industrial.

Ainda, é no capítulo 3 que ele realiza uma separação da ciência em duas categorias que não se anulam ou se rivalizam, como hoje temos com as ciências duras e as ciências leves,

Grosso modo, há dois tipos de objetos culturais: os que são bons para serem consumidos (bens de consumo) e os que são bons para produzirem bens de consumo (instrumentos). Todos os objetos culturais são bons, isto é: são como devem ser, contêm valores. Obedecem a determinadas intenções humanas. Esta, a diferença entre as ciências da natureza e as da cultura: as ciências culturais procuram pela intenção que se esconde nos fenômenos, por exemplo,

no aparelho fotográfico, portanto, segundo tal critério, o aparelho fotográfico parece ser instrumento. (Flusser, 2011, p. 38)

Ele continua,

Instrumentos têm a intenção de arrancar objetos da natureza para aproximá-los do homem. Ao fazê-lo, modificam a forma de tais objetos. Este produzir e informar se chama 'trabalho'. O resultado se chama 'obra'. (Flusser, 2011, p. 39)

E, ainda,

Instrumentos são prolongações de órgãos do corpo (...) por serem prolongações alcançam mais longe e fundo a natureza, são mais poderosos e eficientes. Os instrumentos simulam o órgão que prolongam: a enxada, o dente; a flecha, o dedo; o martelo, o punho. São 'empíricos'. Graças à revolução industrial, passam a recorrer a teorias científicas no curso da sua simulação de órgãos. Passam a ser 'técnicos' (...). Passam a chamar-se 'máquinas'. (Flusser, 2011, p. 39).

Bem, instrumentos e máquinas, portanto, fazem trabalhos (no sentido marxista, como bem nos avisa o filósofo), mas o fazem sem processar qualquer informação do mundo, um moinho não sabe quando não há mais grão a processar, continua a girar ao sabor do vento. Mesmo a máquina mais avançada, quando não integrada a um computador ou manipulada por um ser humano é apenas um objeto que foi "programado" para executar uma função específica.

O computador já não é uma máquina comum, pois ele pode ser programado diferentemente, e quando a ele são disponibilizados instrumentos adequados, ele pode processar informações coletadas por ele mesmo. Assim, computadores podem ser máquinas e instrumentos metaforicamente, assim como um ser humano também o pode, mas não o são de fato. A ideia de programa então alcança para Flusser uma correlação com o desenvolvimento do Aparelho, uma máquina que não realiza trabalho propriamente dito, mas manipula símbolos.

Assim como as metáforas do instrumento e da máquina podem ser usados para o computador assim como para qualquer ser humano, o conceito de poder ser programado também se insere a qualquer ser humano tanto quanto em qualquer computador. Programa, como já falamos acima, seria uma série de instruções para lidar com determinados inputs e como gerar certos outputs, para Flusser o ser humano é programado culturalmente; a forma de vermos o Mundo que nos cerca, a maneira que escolhemos como reagir a determinadas ações de agentes do Mundo, são incutidas pelo nosso ambiente cultural: viriam então dos seus familiares, da sua religião, da sua educação. O computador então também passa a ser programado por aqueles humanos que por sua vez são programados pela cultura.

Mesmo agora, as operações de computador são confundidas com as funções matemáticas, mas existe um gigantesco abismo entre a operação do computador por um artesão e as funções matemáticas. Muitas carreiras foram devotadas a cruzar este abismo – fazer da computação uma forma de matemática pura ou aplicada – no entanto, nenhuma tradução perfeita entre os dois foi encontrada. Consequentemente, computação continua sendo uma arte e um ofício bastante diferente da matemática e também de toda ciência escrita em funções, apesar da frase do desejo de Lovelace de uma "ciência das operações".

(SACK, 2019, p. 57–58).

Ao analisarmos a História do Computador, percebemos que no seu início ele era tido como uma máquina, mais especificamente de calcular. Porém, com o tempo, suas idiossincrasias ficaram evidentes para os pesquisadores e para os artistas, mostrando que na realidade o computador poderia ser um aparelho que serviria para trabalhar com linguagens, não apenas a matemática, mas todas as linguagens - o Computador poderia ser uma “Máquina Semiótica” (NÖTH, 2008) - um aparelho capaz de catalogar, listar, ligar e até produzir signos e esta possibilidade é uma gigantesca mudança de paradigma.

Como um recorte histórico que inicia esta expressão, podemos citar o *Cybernetic Serendipity*, de Londres em 1968. Desde então, pudemos listar três gerações de artistas de arte computacional (cf.: Tabela 1).

<b>Primeira Geração</b>	1968 a 1973 Entendimento dos processos interdisciplinares Computadores Digitais raros e caros Artistas não-nativos, estes vinham de expressões tradicionais da arte e viam na tecnologia uma forma nova de expressão Tudo era experiência
<b>Segunda Geração</b>	1974 ao início do século XXI Amadurecimento do pensamento interdisciplinar Fundação dos principais centros de difusão, exposição e pesquisa de arte computacional Profícua produção artística
<b>Terceira Geração</b>	Seu início ainda se confunde com o fim da segunda geração; As pesquisas já se aprofundaram e o conhecimento para o trabalho interdisciplinar começa a ser mais complexo; Equipes e coletivos de artistas já conseguem comercializar e museus já realizam compras de obras de artistas com um certo nome; Preocupação com a conservação e o combate a obsolescência das obras de arte computacional

**Tabela 1:** Gerações da Arte Computacional. Fonte: Acervo da Pesquisa.

Como podemos observar na leitura da Tabela 1, a prática da primeira geração foi modificada ao longo dos anos de relacionamento com as tecnologias da computação e da comunicação. Neste ponto histórico em que nos encontramos, já se encontram artistas que no caminho acadêmico ou não, iniciaram suas experiências estéticas utilizando das mídias instáveis.

Abaixo analisaremos o evento *Cybernetic Serendipity* que ditou durante muitos anos a forma como este relacionamento entre artista e computador, deveria ser explorado.

## 2.1. *Cybernetic Serendipity*: o equilíbrio da rigidez das ciências com a fluidez das artes

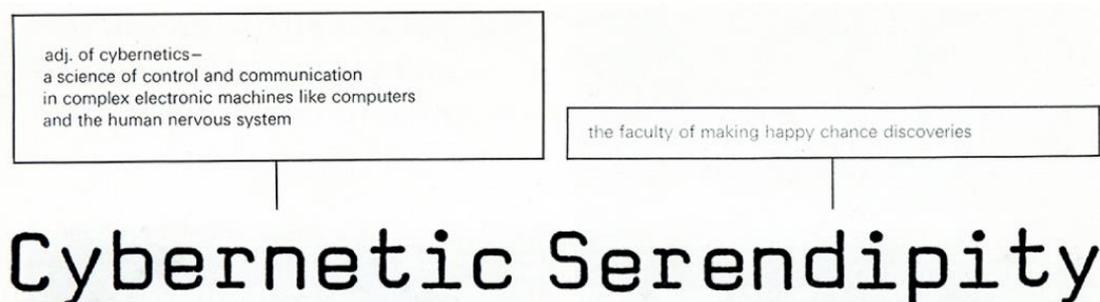
Em 20 de Outubro de 1968, no Instituto de Artes Contemporâneas de Londres, um grupo de artistas e de cientistas se reuniu em uma exposição nomeada “*Cybernetic Serendipity*” (REICHARDT, 1969), onde demonstraram os usos do computador e das teorias da cibernética, inspiradas em Norbert Wiener, com intuito plenamente estético, expondo que os limites da tecnologia podiam ser torcidos e expandidos, e que o computador era mais do que uma máquina de calcular.

A exposição foi a primeira a tentar mostrar a um público totalmente leigo as possibilidades abertas para as artes (Artes Visuais, Música, Poesia, Dança, Escultura e Animação) com o uso do computador. Estima-se que além da equipe de organização, outras 350 pessoas contribuíram, “dentre elas 43 artistas, compositores e poetas, e 87 engenheiros, cientistas da computação e filósofos” (*Cybernetic Serendipity*, 2017).

O evento foi dividido em três áreas de investigação/exposição:

- Obras geradas por computador (computer generated work);
- Aparatos robóticos cibernéticos e máquinas de pintar (cybernetic devices-robots and painting machines);
- Máquinas demonstrando o uso de computadores e da história da cibernética (machines demonstrating the use of computers and the history of cybernetics).

Estas três linhas de atuação e pesquisa compartilhavam do fato de aproximarem a Ciência da Computação e a Engenharia das Artes. O próprio nome escolhido para o evento deixa esta aproximação em evidência:



**Figura 7:** Imagem colocada junto ao sumário do Catálogo da Exposição. Fonte: MASON, 2018.

O nome do evento, portanto, carrega enorme carga simbólica para o momento em que se apresentava. O termo Cibernética foi usado a primeira vez por Platão nos seus diálogos em “As

Leis” e em “A República”, e deriva da palavra grega *kubernetes*, que significa piloto ou timoneiro, apontado por ele como o Governante de um país, mas o termo ficou conhecido por sua definição dada por Norbert Wiener em 1948 no próprio título de seu livro “Cibernética, ou controle e comunicação no animal e na máquina” (Mason, 2018).

De acordo com Weiner, em um nível mais basular, cibernética se refere a um “conjunto de problemas centrados em comunicação, controle e mecânica estatística, seja numa máquina ou em um tecido vivo”. O conceito de Wiener era de que o comportamento de todos os organismos, máquinas e outros sistemas físicos é controlado por suas estruturas de comunicação seja interna ou com seu ambiente. O resultado deste livro foi que a noção do feedback penetrou em quase todos os aspectos da cultura técnica. (op cit. Idem, Ibidem)

O computador começava a receber um destaque como coautor dos trabalhos, mesmo que de maneira ainda insípida, quando continuava a se perguntar quais seriam os limites criativos e de pensamento que um computador poderia executar, mesmas perguntas já efetuadas por Ada Lovelace e Alan Turing em suas épocas.

A segunda parte do título do evento, demonstra a tratativa dos organizadores de explorar o acaso, quebrando com a rigidez metodológica comumente adotada como postura científica. Segundo Venturelli (2017) o *serendipity* é um conceito que representa o acaso nas “invenções e descobertas, científicas, tecnológicas e artísticas”, equiparando o termo com uma ação intuitiva. Os organizadores do evento descreveram o conceito em um *Press Release* sobre a exposição:

“*Serendipity* - foi cunhado por Horace Walpole em 1754. Havia uma lenda de três princesas de Serendipity (antigo nome de Ceilão) que costumavam viajar através do mundo e qualquer que fosse seus objetivos e o que quer que elas estivessem procurando, elas sempre encontravam algo muito melhor. Walpole usou o termo *serendipity* para descrever a habilidade de fazer felizes descobertas ao acaso.” (Cybernetic Serendipity, 2020)

Então, sobre o objetivo do evento, temos nas palavras de Jasia Reichardt, Curadora Chefe da Exposição,

A ideia por detrás deste empreendimento [...] é mostrar as formas criativas engendradas pela tecnologia. O objetivo é apresentar uma área de atividades que manifesta o envolvimento do artista com a ciência e do cientista com as artes: também apresenta as ligações entre os sistemas randômicos empregados pelos artistas, compositores e poetas, e aqueles envolvidos com a feitura e o uso de aparatos cibernéticos. (REICHARDT, 1969, p. 5)

E, comprovando a inovação no pensamento da utilização do computador pelas artes, ela segue:

Cybernetic Serendipity lida com as possibilidades mais do que com as conquistas, e neste sentido, ela é prematuramente otimista. Não há reivindicações heróicas a serem feitas porque os computadores até agora não revolucionaram a música, nem a arte, nem a poesia, da mesma forma que revolucionaram a ciência. (op cit. IDEM, IBIDEM).

No entanto, por outro lado ela evidencia que não havia ainda uma “Estética do Digital” assim, para ela e provavelmente, para os outros organizadores do evento, o uso dos equipamentos eletroeletrônico e dos computadores significava nada mais do que a experimentação de novos suportes para “pintores, produtores audiovisuais, compositores e poetas” (*op cit.* IDEM, IBIDEM). Reichardt ainda reforça que era muito raro que as “novas mídias<sup>14</sup>” e os “novos sistemas” trouxessem “novas pessoas a começarem a se envolver em atividade criativa, seja compondo música, desenhando, construindo ou escrevendo” (*op cit.* IDEM, IBIDEM).

Isto [o envolvimento de novas pessoas na atividade criativa] aconteceu com o advento dos computadores. Os engenheiros para os quais o plotter gráfico controlado por computador representava nada mais do que um meio de resolver certos problemas visualmente, ocasionalmente ficaram interessados nas possibilidades deste output visual, e eles começaram a fazer desenhos que não tinham qualquer aplicação prática, e para os quais os únicos motivos reais era o desejo de explorar, e a busca pelo prazer de ver um desenho se materializar. Então pessoas que nunca colocaram lápis no papel ou pincel na tela, começaram a criar imagens, tanto estáticas quanto animadas, que aproximadamente e muitas vezes se pareciam idênticas com o que chamamos de ‘arte’ e colocamos em galerias públicas. Essa é a mais importante revelação desta exposição. (*op cit.* IDEM, IBIDEM).

Gostaríamos de recorrer à leitura para analisarmos a citação acima e destacar três pontos importantes:

1. Natureza de atividades de experimentação técnica e tecnológica se transformando em experimentações estéticas.
2. Democratização do campo das artes com a entrada de pessoas sem conhecimento técnico tradicional em experimentações estéticas.
3. A ausência de rótulos como Computer Graphic Design, Motion Graphics, Visual Composer, E-music etc. e a utilização de conceitos específicos das artes e da engenharia.

O primeiro item, demarca a dificuldade de determinarmos com exatidão a primeira “obra de arte tecnológica”, até mesmo a primeira “obra de arte computacional”, uma vez que dando continuidade a quebra do academicismo das artes, pessoas de fora deste campo podiam “fazer” arte, mesmo que ainda estivesse dentro de uma especificidade acadêmica que não a das artes (este aspecto vai ser relevante quando analisarmos a atualidade do campo). Essas experimentações continuam existindo e isso tem fornecido muito material visual e sonoro para a Internet, com pessoas de todos os cantos do globo produzindo arte de seus computadores domésticos. O *Cybernetic Serendipity* pode ser considerado, no mínimo, uma vitrine da cultura da produção artística que viria a acontecer nas últimas duas décadas do século XXI.

---

<sup>14</sup> **Nota do autor:** O termo “novas mídias” foi deixado aqui propositalmente para reforçar o argumento temporal.

O segundo item é uma continuação do primeiro, uma vez que pessoas alheias ao fato de estarem “fazendo” arte estão envolvidas com processos de criatividade e suas experiências com os equipamentos é de cunho estético. Isso novamente dificulta um registro histórico, mas caracteriza uma modificação que vinha ocorrendo no campo das artes desde os *Ready Made* de Duchamp que seria a fruição estética além da técnica, mas concentrada na própria significação do objeto artístico. Esta democratização permitiu a popularização do computador e do smartphone, pois foi na capacidade do computador de ser um instrumento de criatividade que os Apple tiveram um crescimento de vendas, e foi a capacidade de fazer fotos e vídeos, editá-los e distribuí-los pela internet que fez com que o smartphone se tornasse desejo de consumo de muitos jovens que ao longo do tempo envelheceram usando esse aparelho e tornando-o parte do seu cotidiano.

O terceiro item, demonstra como os desafios de entender a ontologia das expressões artísticas ligadas ao computador e aos instrumentos eletroeletrônicos foram só recentemente começados a serem enfrentados. Em 1968, momento histórico em que os computadores pessoais ainda não passavam de equipamento de entusiastas e que não faziam mais do que processar simples códigos binários (vide item **1.1.3**), e apenas 53 anos atrás, reforça que o que vemos hoje é ainda o início de um processo, mesmo com a curva de desenvolvimento acentuada que as tecnologias da Comunicação e da Informação apresentam.

Esta conjunção do conhecimento das artes, das ciências e das tecnologias não era novidade já que desde a época dos primeiros computadores e de Babbage que havia uma proximidade entre cientistas e artistas, já que para montar as intrincadas engrenagens e trabalhar no ferro e no fogo, necessitava-se de trabalhos de escultores e outros artesãos. Mas, neste momento não eram os cientistas buscando os artistas para resolver uma questão objetiva, mas sim os artistas procurando os cientistas para explorar os equipamentos em questões estéticas alheias ao objetivo de seus construtores.

A complexidade e raridade dos computadores durante este período significou que toda forma de arte que girava em torno deles estava restrita a um campo especializado de arte, altamente dependente de suporte e fundos. Isso se dava por conta da natureza dispendiosa e de larga-escala dos primeiros equipamentos e na resultante expertise técnica requerida para operá-los. Antes do início de sistemas amigáveis ao usuário, software proprietários e computadores pessoais, os artistas tiveram que construir relacionamentos com cientistas, aprender a escrever o código e frequentemente construir seus próprios hardwares. (Manson, 2018)

Como vimos no item **1.1.3**, a IBM dominava o mercado de computadores, porém dificilmente havia a disponibilidade destes aparelhos de maneira semelhante ao que temos hoje. Todas as grandes empresas e universidades tinham pelo menos um computador IBM, o prestígio era tanto que foi munida de uma carta da IBM USA que Jasia Reichardt, curadora chefe da exposição, teve

acesso a empresas como BOEING e General Motors, conseguindo seus patrocínios, e a artistas e cientistas da década de 1960 do mundo todo. Grande parte destes artistas e cientistas fizeram parte de outras experiências que determinaram a continuidade da arte-ciência (MASON, 2018).

Catherine Mason, pesquisadora e escritora, escreveu o livro “*White Heat Cold Logic: British Computer Art – 1960 – 1980*” que conta da formação dos pesquisadores e artistas britânicos de digital art. Em artigo escrito para o site “Studio International”<sup>15</sup>, intitulado “*Cybernetic Serendipity: History and Lasting Legacy*”(MASON, 2018), a autora descreve que muitas obras e trabalhos experimentais feitos pelos artistas e cientistas enquanto estabeleciam uma linguagem comum e aprendiam o que o computador e os outros equipamentos poderiam fazer, se perderam, já que os equipamentos e componentes eram muito caros e a maioria foi desmontado e direcionado a outros propósitos, e assim tem-se apenas descrições e/ou fotografias.

Encerrado o evento em Londres, parte do acervo foi embarcado para ser exposto no Museu Smithsonian em Washington – EUA, muitas tendo sofrido danos neste transporte, e tendo sido restauradas foram expostas em Washington e no Exploratorium em São Francisco – EUA (Cybernetic Serendipity, 2017). O site Compart, afirma, ainda que as obras que não foram enviadas para os EUA, ou foram vendidas, ou foram devolvidas para seus donos, ou foram transferidos para o Museu da Cidade de Kawasaki – Japão, mas as obras não estão expostas lá, e o Exploratorium comprou parte deste acervo em abril de 1971.

Há uma considerável literatura escrita sobre o *Cybernetic Serendipity*, mas há uma igualmente considerável dificuldade em localizar dados específicos sobre as obras e os seus autores. O próprio catálogo do evento não traz uma Ficha Técnica das obras, restando muitas vezes textos reflexivos sobre a estética envolvida.

Dois outros eventos posteriores são citados como influenciados pelo *Cybernetic Serendipity* e a utilização de computadores nas artes: New Tendecies 4 (Zagreb – Croácia, 1968 -1969) com representação brasileira de Waldemar Cordeiro; e Event One (Março de 1969) da *Computer Arts Society*, sociedade formada após o *Cybernetic Serendipity* na Inglaterra para promover o uso criativo dos computadores nas artes.

## 2.2. Waldemar Cordeiro e Giorgio Moscati: *avant-garde* da arte-ciência no Brasil

Waldemar Cordeiro foi um importante artista abstracionista e concretista ítalo-brasileiro que 1968 inicia a pesquisas de arte em computador e estabelece parceria com o físico Giorgio Moscati,

<sup>15</sup> Cf. <http://www.studiointernational.com>.

dando início à arte-eletrônica no Brasil. No entanto, como descreve Walter Zanini (ZANINI, 1997, p. 233–246) tanto Cordeiro quanto outros artistas tiveram a formação de um background que possibilitou o estabelecimento desta área de atuação artística no Brasil.

O próprio Waldemar Cordeiro tinha em suas preocupações iniciais a pesquisa com o abstracionismo e a arte concreta e para ele o trabalho com arte em computador era um retorno aos seus questionamentos iniciais:

Então volta ao ponto de partida. A arte concreta a que fazia: digitalizava a imagem, números, superfícies com quantidades, relacionava essas quantidades, programava os quadros. A execução era artesanal apenas porque não havia indústria alguma que quisesse fazer isso e as artistas não tinham dinheiro para pagar - nós não tínhamos dinheiro para pagar. Mas intencionalmente os nossos quadros eram programados. Os quadros concretos poderiam ter sido executados por uma tipografia, por uma indústria, por uma máquina, porque eles tinham na sua base um programa numérico - note bem - como a arte digitalizada. Evidentemente que a programação da arte concreta é muito mais elementar do que a programação com computador (..), mas os primórdios estão aí. Essas pesquisas se inserem perfeitamente dentro da arte moderna como se inseriam todas essas tendências chamadas genericamente construtivas (CORDEIRO apud COSTA, 2002, p. 28).

Devemos nos lembrar da dificuldade, já demonstrada nos itens iniciais deste capítulo, de acesso aos computadores. Como nos alerta a professora Helouise Costa:

No final dos anos 60 os computadores eram máquinas de grande porte com um alto custo operacional, existentes apenas em sofisticados centros de pesquisa, e cujo acesso era restrito a cientistas altamente especializados. Por isso, foi somente no Departamento de Física Nuclear da Universidade de São Paulo, que Cordeiro, em parceria com o Prof. Giorgio Moscati, teve condições de iniciar suas pesquisas em arteônica (COSTA, 2002, p. 28).

O professor Giorgio Moscati já estava envolvido em extração de informação de imagens em seu trabalho no Instituto de Física: “Minhas experiências, envolvendo traços de partículas e outros estudos já criavam imagens” (Giorgio Moscati e Waldemar Cordeiro: quando o computador encontrou a Arte, 2012). Segundo Moscati, Waldemar Cordeiro “identificava o computador como um instrumento de armazenamento, reprodução e divulgação da arte” (op cit.) e desta maneira eles iniciaram uma parceria para investigar “como transformariam o computador num instrumento de produção artística” (op cit.).

Duas obras saíram desta parceria e estas marcaram o início da Arte Computacional no Brasil: “Beabá” e “Derivadas de uma Imagem”.

- **Beabá**

O BEABÁ era um gerador de pseudo-palavras, para eles era a forma mais direta de gerar “palavras ao acaso”, pois envolvia o sorteio de conjuntos de letras de vários comprimentos (Giorgio Moscati e Waldemar Cordeiro: quando o computador encontrou a Arte, 2012). Moscati

afirma na entrevista que buscou medir com uma régua o comprimento de palavras no dicionário para programar o acaso. Eles imaginavam que teriam como resultado conjuntos gerados aleatoriamente, considerando as combinações que alternariam vogais e consoantes, e por isso teriam semelhança com palavras de uma língua, no entanto, por acaso algumas das tais palavras geradas poderiam existir.

O conteúdo informativo de três consoantes e três vogais era tratado por um computador (IBM 360/44) e um software gerava palavras ao acaso e calculava um índice de significação; quanto mais elevado esse índice, maior a probabilidade de as palavras existirem na língua portuguesa, ou seja, os autores descreviam que a frequência de aparecimento de uma letra é aproximadamente proporcional àquela com que figura no dicionário da língua portuguesa. No processo tecnológico, uma vez criado o software que combinava pares de vogais e consoantes, era impresso um cartão, não muito diferente dos cartões perfurados que nos anos 1960 eram o meio de incluir dados e comandos nas primeiras máquinas computadorizadas.

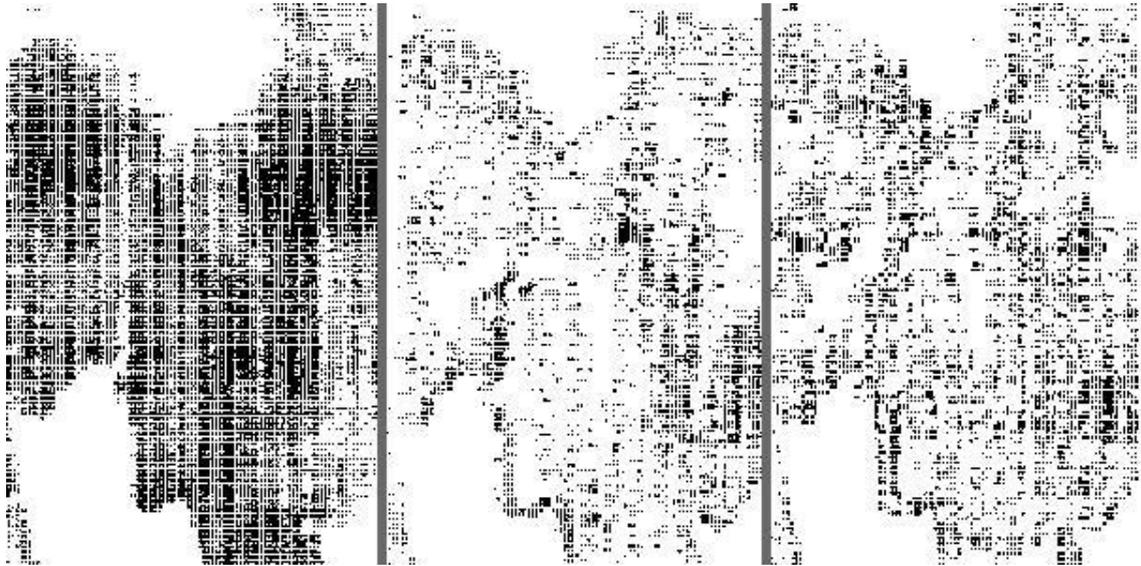
- **Derivadas de uma imagem**

Tinha como objetivo “a geração de imagens derivadas a partir de uma fotografia tomada como matriz” (Costa, 2002, p. 29), “na prática, o programa que ‘deriva’ a imagem faz simplesmente uma operação aritmética que compara os graus de preto de um pixel com o grau de preto de seus vizinhos” (Giorgio Moscati e Waldemar Cordeiro: quando o computador encontrou a Arte, 2012).

Para esta obra (Figura 8) foi escolhida uma imagem de uma propaganda do Dia dos Namorados e em um processo visual-manual Cordeiro começou a atribuir números de 1 a 7 aos 10.976 pontos em que a imagem foi dividida, e estes números faziam menção as variações entre branco e preto e todos os tons de cinza entre eles (Costa, 2002, p. 29).



**Figura 8:** CORDEIRO, Waldemar; MOSCATI, Giorgio, BEABÁ, 1968. Software, cartões perfurados. (Kassab, 2009).



**Figura 9:** Waldemar Cordeiro e Giorgio Moscati. Da esquerda para a direita, 'Transformação em grau zero', 'Transformação em grau 1' e 'Transformação em grau 2', 1969 | Foto: Reprodução (Giorgio Moscati e Waldemar Cordeiro: quando o computador encontrou a Arte, 2011).

E, então nos alerta Costa que:

Devemos considerar que o equipamento utilizado não possuía escâner nem monitor, periféricos raros na época. A entrada dos dados se fazia por meio de cartões perfurados e a saída por uma impressora de linhas. Portanto, trabalhar com imagens nesse tipo de computador significava, em certo sentido, subverter os princípios da máquina, obrigá-la, ou melhor dizendo, programá-la de modo a realizar algo não previsto em seu sistema. (Costa, 2002, p. 29)

O papel de Moscati foi o de elaborar um programa que baseado na função matemática da derivada iria realizar transformações sucessivas na imagem digitalizada por meio dos cartões perfurados. E, “[p]ara a saída dos resultados, ele programou a impressora para utilizar um artifício de sobreposição de letras e sinais gráficos diversos, de modo a dar materialidade às imagens derivadas” (COSTA, 2002, p. 29).

Foram várias as composições resultantes. Tornou-se importante notar como as bordas dos resultados geravam diferentes contrastes e profundidades. “Só muito mais tarde nosso trabalho foi considerado inovador no processamento de imagem e detecção de bordas”, afirma o professor. (Giorgio Moscati e Waldemar Cordeiro: quando o computador encontrou a Arte, 2012)

Cordeiro então criou o termo “Arteônica” (fusão entre arte e eletrônica) para categorizar as suas obras de arte em computador e em 1971, ele organiza a exposição com o mesmo nome na Fundação Armando Álvares Penteado (São Paulo). Em 1972 ele passa a dirigir o Centro de Processamento de Imagens no recém criado Instituto de Artes da Universidade Estadual de Campinas (Unicamp), e em junho de 1973 ele faleceu aos 48 anos de idade.

Em setembro de 1993, em Recife durante o VI Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens (Sibgrapi), foi feita uma nova exposição com o nome de “Arteônica”<sup>16</sup> e que comemorava o pioneirismo de Waldemar Cordeiro e Giorgio Moscati apresentando as obras: Beabá, Derivados de uma imagem, Retrato de Fabiano, A mulher que não é B.B., Gente e Pirambu; todas de autoria de Cordeiro e algumas com participação de Moscati e outros artistas/cientistas.

Apesar de curta a passagem pelo campo da Arte Computacional, principalmente pela sua morte tão prematura, essa ação de Waldemar Cordeiro permitiu, no mínimo, que outros artistas presenciassem as possibilidades do uso criativo do computador e acabassem por influenciar uma geração.

## 2.3. Desdobramentos

Antes de seguirmos em frente cabe reforçar um fato importante para aqueles que tentam estabelecer os pontos do passado da arte computacional: durante os últimos sessenta anos trabalhamos com o rótulo das Novas Mídias<sup>17</sup> e, talvez, impressionados por toda a aura do objeto tecnológico e da esperança e otimismo que eles trazem emanante de um futuro mais justo e balanceado, acabamos por muitas vezes negligenciar o passado. Esta tese se une a tantos outros esforços, que em parte serão abordados no capítulo 3, mas como cita Margit Rosen em texto Editorial de seu livro:

A história do uso do computador na *fine art* [...] começou aproximadamente sessenta anos atrás. No entanto, até cerca de dez anos atrás, uma espécie de eterno presente reinava no mundo da digital art. Arte computacional tinha um passado, mas faltava uma memória. A história não parou, como em 1984, porque alguém apagou todos os traços do passado, mas porque faltou uma memória. (ROSEN, 2011, p. 9)

Assim, conforme aprofundamos na leitura e na investigação, linhas surgem que unem eventos, artistas, teóricos, institutos de pesquisa e universidades de modo que formam o que hoje temos como o panorama da produção artística em Arte Computacional.

Os primeiros desdobramentos de eventos e ligações que conseguimos rastrear abordam os anos de 1968 a 1973 aproximadamente e iniciam um ano após a *Cybernetic Serendipity*, quando aconteceu em Zagreb (Croácia), o evento “*Tendencies 4*” que tinha como tema: Computadores e Pesquisa Visual (*Computer and Visual Research*), abria para uma exposição que acompanhava a numeração do evento e se chamava “*New Tendencies 4*”. Vários artistas apresentaram suas obras, entre eles Waldemar Cordeiro que expôs a obra “Beabá”. Os trabalhos tinham sido feitos em

<sup>16</sup> Cf. Arteônica: <https://www.visgrafimpa.br/Gallery/waldemar/waldemar.htm>. Visto em: 05 Janeiro 2021.

<sup>17</sup> **Nota do autor:** O termo “novas mídias” foi deixado aqui propositalmente para reforçar o argumento temporal.

computadores digitais (comerciais como os IBMs e os Gradientes, e computadores montados por entusiastas), aparelhos analógicos como Osciloscópios, estruturas de instalação com motores e partes elétricas e mesmo obras de arte computacionais feitas à mão (BASICEVIC *et al.*, 1970).

De acordo com Jerko Denegri (Denegri, J. in ROSEN, 2011, p. 19), o movimento artístico que deu origem ao New Tendencies (este sendo um nome guarda-chuva para representar outros nomes do movimento que se desdobrava em outros países), já vinha se formando com a atuação do grupo Exat 51 (de “Experimental Atelier 1951”) ativo de 1951 a 1956 em Zagreb, e tinham como ponto de convergência a:

Uma “aproximação construtiva para a arte” indo das primeiras abstrações geométricas do pós-guerra no caso do grupo Exat 51 e os neo-construtivistas ópticos e os Gestaltistas, Cinética e arte programada (arte programmata) no caso do New Tendencies (Denegri, J. in ROSEN, 2011, p. 19)

O autor ainda aponta o fato de estes grupos tinham em comum vários artistas protagonistas na cena artística croata, ou seja, artistas que vieram a participar ativamente de ambos. No Brasil, fato semelhante ocorre com a inauguração da pesquisa visual em Arteônica de Cordeiro, afinal, ele mesmo já era um fenômeno no construtivismo brasileiro e ao longo dos anos em exposições dentro do Brasil tivemos obras que relembram trabalhos dadaístas, construtivistas, op e motion art, surrealismo, entre outros movimentos modernos.

Evidência disso está em parte na escolha de uma fotografia como matriz para os trabalhos de “Derivadas de uma imagem”, quando para ele a retícula fotográfica era semelhante ao padrão numérico da imagem computadorizada, e com isso retornamos à citação trazida na página 69 destacando especialmente a seguinte parte:

Os quadros concretos poderiam ter sido executados por uma tipografia, por uma indústria, por uma máquina, porque eles tinham na sua base um programa numérico - note bem - como a arte digitalizada. Evidentemente que a programação da arte concreta é muito mais elementar do que a programação com computador (...), mas os primórdios estão aí. (Cordeiro, Waldemar apud Costa, 2002, p. 28).

Podemos destacar que o intercâmbio de informações entre artistas no mundo começou a aumentar conforme as tecnologias de informação e de comunicação começaram a diminuir distâncias físicas tornando-as virtuais (mesmo que para isso também tenha evidenciado até hoje uma distância econômica entre as pessoas). A participação brasileira neste cenário mundial demorou um pouco mais para acontecer porque as tecnologias computacionais não eram produzidas no país, todas eram importadas e com altas taxas de impostos, o que permitia que apenas grandes centros de pesquisa tivessem acesso à tecnologias, mesmo assim bem atrasadas em relação ao resto do mundo ocidental. Este talvez tenha sido o principal motivo da primeira e

segunda geração de artistas computacionais brasileiros tenham sido formados exclusivamente dentro da academia (iremos abordar esse ponto em breve).

Apesar das edições iniciais do *New Tendencies* não terem sido sobre arte computacional, elas representaram um importante fator agregador e difusor de informações uma vez que em suas edições cada vez mais países estavam representados e artistas e teóricos que só se conheciam de nome passavam a se conhecer pessoalmente (Denegri, J. in Rosen, 2011, p. 19 - 21).

Quase em paralelo, outro evento representou a estabilização da base da arte em computador na Inglaterra. Como consequência imediata do *Cybernetic Serendipity* houve a formação do grupo Computer Arts Society (CAS) que tem seu debut em 1969 no “*Royal College of Art*” em Londres com o evento propriamente nominado: *Event One*.

A CAS tinha como objetivo primeiro “promover o uso criativo dos computadores nas artes, e encorajar o intercâmbio de informações nesta área” (MASON, 2009). Mantendo o mesmo princípio da interdisciplinaridade, o *Event One* teve a participação de arquitetos(as), artistas, programadores de computadores e cineastas entre outros; e, segundo Mason (2009), ofereceram uma gama de experiências ao visitante da escultura e imagens à performances ao vivo e workshops. Esta interdisciplinaridade é importante características dos grupos de pesquisa e coletivos que atuam com arte computacional e eletrônica nos dias atuais.

Assim como em Zagreb, os integrantes do CAS tinham interesse em manter a rede de colaboração e informação que foi possível por meio do *Cybernetic Serendipity* que fez com que as barreiras entre as diversas esferas do conhecimento fossem abertas. Estes indivíduos (tanto em Londres como em Zagreb como em todos os países do mundo que tiveram um mínimo contato com o *Cybernetic Serendipity*) passavam a se verem como uma parte de um movimento maior “compartilhando o interesse no poder das novas tecnologias e o que isso poderia significar para as artes” (MASON, 2009).

No momento em que o *Cybernetic Serendipity* tornou possível o pensamento de uma linguagem comum entre as Artes e a Computação e a Engenharia, iniciou-se um processo de integração, agregando outras áreas da ciência e das Artes, pois se no *Cybernetic Serendipity* a maior parte dos trabalhos envolviam imagens e sons, no *Event One* já se deslumbrava a inserção das outras expressões.

Era reconhecido que esta era uma área que tinha tido uma crescente atividade, mas com poucas publicações formais de métodos e resultados e poucas comunicações entre artistas de diferentes campos de atuação (música, artes visuais, artes performáticas e por ai vai). CAS e as variadas atividades de seus membros demonstra que no início era compreendido que as inovações da computação iriam afetar fundamentalmente a humanidade. Um dos objetivos definidos pelo CAS era de prover um “fórum no qual artistas e cientistas [poderiam] trabalhar conjuntamente nestas implicações” (MASON, 2009)

Como no evento anterior, os equipamentos eram em sua maioria emprestados e as obras efêmeras, mas da documentação que conseguiu resistir Mason conseguiu identificar pelo menos 700 visitantes. Entre os trabalhos apresentados estava uma demonstração de aplicações para arquitetura em CAD e com caneta-de-luz em um computador (PDP-7, emprestado da Imperial College), terminais de teletipos com plotters (um tipo de braço com uma caneta para traçar imagens) para gráficos emprestados de companhias de comércio, um link por telefone conseguiu ser feito para um computador PDP-8; o cineasta Malcolm Le Grice colaborou com Alan Sutcliffe (um programador e um dos organizadores do evento) numa performance com o título de Typodrama, um programa gerava instruções e falas para atores performáticos (MASON, 2009, p. 5).

Sutcliffe apresentou um poema gerado pelo computador chamado de SPASMO, usando um computador ICL 1904 e uma impressora em linha ele imprimiu cópias do poema e inseriu nos catálogos do evento. Este poema também foi traduzido para o Servo-Croata e enviado à Zagreb para o Simpósio do *New Tendencias 4* (op. cit.). R John Lansdown produziu uma coreografia gerada por computador que performava uma luta de espadas teatral em que o computador selecionava randomicamente uma série de instruções da Esgrima e transmitia aos atores; e também apresentou um Programa Gerador de Palavras com inspiração na poesia concreta.

Outro desdobramento importante foi a participação de John Lifton que mesmo sendo arquiteto tinha interesse em cibernética e ambientes sinestésicos e chegou a se apresentar no *Cybernetic Serendipity*, e em 1969 ele foi um dos fundadores do “*The Institute for Research in Art and Technology*” (IRAT) que ficou conhecido “como o “*New Arts Lab*” e abrigava um teatro, uma galeria, um cinema e uma série de grupos” (MASON, 2009, p. 7). Um detalhe interessante trazido por Mason (2009) é que Lifton coordenava no IRAT um workshop de Eletrônica e Cibernética que tinha o primeiro computador com terminal para uso livre e exclusivo para artistas.

## **2.4. Segunda geração final de 1970 até o século XXI: formação acadêmica, laboratórios e institutos de pesquisa e conservação**

Os eventos apresentados no item 2.3, representaram uma mudança significativa na aceitação desta tecnologia como parte dos recursos utilizados por um artista. Começou-se a perceber o potencial estético das máquinas e, junto com o próprio mercado econômico, começou-se a vislumbrar novas potencialidades para o universo da computação.

Como já mencionado, os artistas das mídias instáveis passaram a se reconhecer como um grupo e começaram a criar sua própria identidade. Assim começaram a se formar grupos de pesquisa em universidades e institutos de pesquisa voltado para o estudo interdisciplinar, assim como congressos e premiações, antes destinados às artes tradicionais ou à computação, passaram

a ter seções específicas para a arte computacional, a arte gráfica de computador, o poema de computador.

Algumas referências mundiais e brasileiras em eventos, grupos de pesquisa e coletivos de artistas surgidos exatamente neste período forma a segunda geração de artistas computacionais, que não lidavam mais com o total desconhecimento do tema, mas sim com a aceitação deste dentro do campo das artes (cf. VENTURELLI, 2017).

Na questão da tecnologia, como vimos, os computadores pessoais já começaram a serem comercializados e a indústria de software começa a se desenvolver, a cultura hacker passa da montagem eletrônica de computadores digitais para a programação de softwares e games, e neste período de vinte anos, a Internet sai de um projeto de ligação militar para uma rede que conecta o mundo inteiro.

Assim, é importante frisar que no campo artístico, como o computador já havia deixado de ser um objeto estranho e conforme a tecnologia se popularizava, mais os artistas passaram a ter acesso a diversos instrumentos, equipamentos e peças de eletroeletrônica e computacionais para apresentar suas ideias. A ideia da arte performática que surge por volta de 1965, pôde ser aplicada ao computador e suas interfaces, à medida que os mecanismos de *input* e *output* e a capacidade de processamento dos mesmos aumentou, o computador por si só já poderia ser um agente performático, como neste período mostraram as obras de alguns artistas como: Karl Sims com sua contribuição para a Arte Generativa; Christa Sommerer & Laurent Mignonneau e Eduardo Kac com suas Arte Orgânica e vidas artificiais; Suzete Venturelli apresentando obras sobre Vidas Artificiais, Inteligência Artificial, interface humano x computador; Gilberto Prado explorando a Arte Imersiva, a Realidade Virtual, a relação Humano x Cidade; Tânia Fraga com Arte Generativa e Vidas Artificiais; dentre tantos outros.

Este aspecto da capacidade performática do computador, ou sua capacidade de ser pivô da performance fez com que o público visitante das obras passasse de seu caráter passivo de espectador (aquele que observa) para um status de interatividade sendo chamados de *interator* (aquele que interage), sendo este o termo utilizado na bibliografia para as obras que exigem para sua composição simbólica a relação do público com o aparelho.

Para caracterizar melhor este período, segue abaixo um pequeno apanhado de informações sobre os **Laboratórios e Centros de Pesquisa, Eventos e Exposições em destaque** e **Artistas e obras relevantes** que consideramos importantes para caracterizar este período e para a formação da Terceira Geração de artistas da área.

- **Laboratórios e Centros de Pesquisa**

Os principais artistas desta geração estão de alguma forma ligados à Academia. Seu aspecto interdisciplinar inclusive representou, e ainda representa, um desafio para muitas

Instituições de Ensino Superior (IES) que ainda se concentram em modelos lineares e compactados de formação. Em vários momentos, os artistas retratados aqui (também professores) tiveram que construir um espaço em sua universidade ou a própria instituição já conseguia vislumbrar esse futuro interdisciplinar e transdisciplinar e criava institutos ou departamentos de pesquisas próprios para este modelo de ensino e prática.

Em 1979 teve início o primeiro ARS Electronica em Linz na Áustria, onde vinte artistas e cientistas se uniram para criar o “Festival para Arte, Tecnologia e Sociedade”, “*to discuss the Digital Revolution na its possible consequences*” (ARS ELETRONICA, n.d.). ARS Electronica tendo iniciado como um festival hoje é um complexo que engloba o festival<sup>18</sup> que traz uma premiação e uma exibição coletiva; o Ars Electronica Center<sup>19</sup>, um prédio construído em 2009 que abriga um museu interativo de arte eletrônica e serve para abrigar outros projetos e a parte administrativa; e o Ars Electronica Futurelab<sup>20</sup>, o ateliê e laboratório de pesquisa interdisciplinar na área da arte e ciência. Há ainda outras estruturas administrativas e de negócios, provavelmente administradora de patentes e comércio de descobertas e tecnologias. O Ars encontra-se em expansão, já tendo montado uma estrutura no Japão.

No ano de 1981, teve a formação do V2\_, *Lab for Unstable Media*, como centro interdisciplinar de arte e tecnologia da mídia localizado em Rotterdam, Holanda. Segundo o site do grupo na internet a

V2\_ oferece uma plataforma para artistas, designers, cientistas, pesquisadores, teóricos, e desenvolvedores de software e hardware de várias disciplinas para discutir seus trabalhos e compartilhar suas descobertas (About, V2\_, n.d.).

Inicialmente formado como um coletivo de artistas, o centro cujos fundadores (entre eles Alex Adriaansens e Joke Brouwer) denominaram de centro multimídia. No local, houve a apresentação de concertos e performances, experimentos com mídias analógicas (TV, caixas de som e filmes de Super-8), possuía também um espaço dedicado para exibição de pinturas, instalações e arte mecânica. O período que vai de 1981 a 1986 é chamada pelo próprio grupo como época do “Do-It-Yourself” (About, V2\_, n.d.).

Depois disso, o grupo começou a modificar seus interesses conforme o próprio desenvolvimento das tecnologias da comunicação e da informação encontravam seu caminho. Passaram de obras utilizando pequenos computadores e as redes, para instalações interativas, para trabalho em ambientes imersivos e realidade virtual e tecnologia vestíveis.

Em 1985 foi a vez da fundação do Media Lab MIT, em que

<sup>18</sup> <https://ars.electronica.art/keplersgardens/en/>

<sup>19</sup> <https://ars.electronica.art/news/de/>

<sup>20</sup> <https://ars.electronica.art/futurelab/en/>

Sem a restrição de disciplinas tradicionais, os designers, engenheiros, artistas e cientistas do Media Lab se esforçam para criar tecnologias e experiências que permitam às pessoas compreender e transformar suas vidas, comunidade e ambientes. (About the Lab, - Media Lab MIT [Web Page], n.d.)

O Media Lab do MIT apresenta números impressionantes de pesquisadores (mais de 25 pesquisadores seniores, com equipe de pesquisa de mais de 175 indivíduos, além de cientistas e professores visitantes, pesquisadores de pós-doutorado, 150 alunos de doutorado e mestrado, além de 200 alunos de graduação) e pesquisa em andamento (centenas de projetos em áreas como robótica social, próteses físicas e cognitivas, novos modelos e ferramentas para aprendizado, bioengenharia comunitária, e modelos de cidades sustentáveis), fruto de uma política de pesquisa voltada para a produção de produtos e patentes que possam ser comercializadas, ele une a pesquisa, o ensino e a produção voltada para uma esfera mercadológica. Hoje, grandes publicações da área recebem o selo editorial do Media Lab – MIT.

Em 1986 foi fundado o Media Lab/UnB (Media Lab/UnB, n.d), ainda com o nome de “Laboratório de Imagem e Som” (LIS) pela Profa. Dra. Suzete Venturelli, na época recém ingressante no quadro docente da Universidade de Brasília e estando concluindo o seu doutorado em Artes e Ciências da Arte na Universidade Sorbonne Paris I. Até então na pesquisa realizada é o laboratório interdisciplinar voltado para arte e ciência mais antigo em funcionamento (desde a fundação do Centro Arteônica de Processamento de Imagens do Instituto de Artes da Unicamp, que como vimos anteriormente foi fundada por Waldemar Cordeiro). Desde sua fundação, o laboratório que hoje é coordenado pelo Prof. Dr. Antenor Ferreira, já foi amplamente premiado, no Brasil e internacionalmente, e conta com a participação de bolsistas de iniciação científica, alunos de mestrado e doutorado, sendo hoje também o Laboratório de Arte Computacional do Programa de Pós-graduação em Artes Visuais da UnB. E em 1989, o laboratório inicia a organização do Encontro Internacional de Arte e Tecnologia (#.Art, n.d.), o #.Art, que em 2020 completou sua 19ª edição, apresentando a cada ano temáticas diferenciadas para refletir sobre os processo de criação da arte com as tecnologias digitais e pós humanísticas. Na sua segunda edição (1990), o evento iniciou a exposição #EmMeios que ainda hoje apresenta novos artistas brasileiros e da América Latina para o meio.

No ano de 1989 foi a vez da fundação do Center for Art and Media Karlsruhe, ou ZKM Karlsruhe (The ZKM, n.d.), cuja missão foi a de “continuar as artes clássicas para a era digital”, e, segundo o sítio da ZKM na internet, “as vezes chamada de ‘a Bauhaus eletrônica ou digital’”. Denominada por eles como a “Meca da media art” (Peter Weibel in The ZKM, n.d), é um centro de pesquisa, exibição, produção e um museu interativo especializado na coleção e arquivamento e na organização de eventos e mediação. Para esta pesquisa, o ZKM torna-se imprescindível como ponto de referência por ser um dos poucos centros de pesquisa a se preocupar de maneira

prematura com a conservação das obras de arte digitais, virtuais e eletrônicas. Sob coordenação do Prof. Dr. Bernhard Serexhe, o projeto de pesquisa *Digital Art Conservation* foi lançado em 2010 e terminou três anos depois com um livro publicado em três volumes (inglês, alemão e francês).

No ano de 2009 na Universidade Federal de Goiás, o Prof. Dr. Cleomar Rocha funda o Media Lab/UFG, primeiro dos laboratórios a utilizar o nome em inglês (o Media Lab/UnB chamava-se até então MidiaLab) e sede da Rede Media Lab/BR que será estabelecida no ano de 2017 com a entrada do Media Lab/Unifesspa e a mudança de marca do MidiaLab da UnB para Media Lab/UnB. Movido por uma série de projetos e baseado em uma estratégia de crescimento e estabilização dentro do cenário da universidade pública no Brasil, o Media Lab/UFG se tornou um dos laboratórios de maior destaque dentro da UFG (e no país), tem um prédio próprio de mais de 1000 m<sup>2</sup>, onde encontra espaço para nove linhas diferentes de pesquisa. Já se viu com mais de 30 bolsistas (de iniciação científica à pós-doutorado) e chegou a ter incentivo em projetos de dois ministérios diferentes (Ministério da Ciência e Tecnologia e o extinto Ministério da Cultura).

Estes centros e coletivos de artistas dominam o cenário internacional, somado com esforços de outros museus, universidades e institutos que possuem interesse no assunto, mas que não se dedicam integralmente ao mesmo; estes então representam a segunda geração, que está principalmente preocupada em produzir e investigar. A diferença entre as visões dos exemplos da Europa e dos EUA para os da América Latina está basicamente no potencial comercial e no trabalho interdisciplinar. O fácil acesso a tecnologias (que em geral são desenvolvidas nestes mesmos locais) cria uma ligação em círculo que esse fácil acesso a tecnologias promove o descobrimento de novas tecnologias que por sua vez são facilmente acessadas pelas universidades e centros de pesquisa europeus e norte-americanos.

Nos países da América Latina, prevalece a ligação dos artistas de Arte Computacional com o espaço acadêmico. Em sua maioria são professores ou alunos de graduação ao doutorado que se utilizam da estabilidade e da capacidade de criar infraestrutura das universidades (em sua maioria públicas, estaduais e federais), tal como foi na época de Waldemar Cordeiro e Giorgio Moscati. Contudo, a diferença de hoje para aquela época é a facilidade de acesso ao aprendizado de manipulação de imagens computadorizadas, produção de material para web art, vídeo art, pequenas instalações ainda baseadas em *Do-It-Yourself* (DIY). Veremos melhor isso ao descrevermos a terceira geração.

- **Eventos e Exposições em destaque**

Vários eventos e exposições começaram a serem criados e organizados após o Cybernetic Serendipity. Além dos já citados *New Tendencias* e *One Event*, e no Brasil o Arteônica, seguem abaixo alguns outros exemplos de eventos específicos. Além disso, devemos considerar

que as exposições de arte tradicionais passaram a também aceitar as obras de mídias instáveis como a Bienal de São Paulo e Rumos do Itaú Cultural, assim como exposições no Centro de Cultura do Banco do Brasil (CCBB) e no espaço cultural Santander.

- O ARS ELECTRONICA FESTIVAL que já foi citado, teve sua primeira edição em 1979 e desde então tem acontecido anualmente, principalmente na cidade de Linz. No ano de 2020, devido à pandemia de SARS-COV-2, assim como outros eventos, o Festival aconteceu também de maneira online atingindo um total de 120 países (ARS, n.d.).
- ISEA, International Symposium on Electronic Art ou Simpósio Internacional de Arte Eletrônica, inicia em 1988 em Utrecht - Holanda e tem periodicidade anual desde sua terceira edição em Sydney na Austrália, quando então também tornou-se um evento itinerante acontecendo em países e cidades diferentes ao longo dos anos (Past ISEA Symposia, n.d.). Esse fluxo tem contribuído para criar redes de pesquisa e colaboração entre artistas e institutos nos sete continentes.
- FILE, Electronic Language International Festival, ou Festival Internacional de Linguagem Eletrônica, segundo seu site é uma organização sem fins lucrativos que pretende disseminar e a linguagem eletrônica pelo Brasil e América do Sul, e refletir sobre os principais aspectos do universo contemporâneo digital e eletrônico. Ele teve início no ano de 2000 e desde então além do evento principal em São Paulo, ainda tiveram eventos menores no Rio de Janeiro, Vitória, Belo Horizonte, Curitiba, Porto Alegre, São Luís e Brasília.

Os dois próximos eventos já foram citados, mas merecem um aprofundamento uma vez que são os que concentram as principais referências na área no Brasil:

- #.ART, Encontro Internacional de Arte e Tecnologia, criado em 1989 pela Profa. Suzete Venturelli, também fundadora do Media Lab/UnB. O evento ocorre ainda de maneira anual desde seu início e trouxe para Brasília nomes importantes como: Gilberto Prado, Lúcia Santaella, André Parente, Roy Ascott, Edmond Couchot, Priscila Arantes, Sandra Rey, Ettiene Delacroix, Tania Fraga, Cleomar Rocha, Milton Sogabe, Lúcia Leão, Monica Tavares, Chantal Dupont, Maria Beatriz de Medeiros, Yara Guasque, Monica Fleishmann, Martha Gabriel, Iain Mott, Derrick de Kerckhove, François Soulages, Oliver Grau, entre outros.

Ao longo dos anos, forneceu espaço de diálogo, troca de conhecimentos, estabelecimento de parcerias, ensino e exposição de obras de arte que

utilizam a tecnologia digital como uma ferramenta para produzir formas tradicionais, de obras de arte, de outros tipos de produção, que surgiram a partir da colaboração entre arte, ciência e tecnologia e criam sistemas inéditos

indissociáveis de um pensamento que conceitualiza o conhecimento sensível do homo aestheticus (#.ART, n.d).

O evento passou a acontecer no Museu Nacional da República de Brasília, onde durante uma semana em geral, comunicações e palestras são apresentadas por artistas, professores e pesquisadores nacionais e internacionais. Entre as falas, a sala anterior abriga as obras selecionadas e já apresentou centenas delas que atendem a diversos tipos de expressão e significação.

- SIIMI, Simpósio Internacional de Mídias Digitais Interativas, é o mais novinho sendo que sua primeira edição ocorreu no ano de 2012. Está aqui listado por ser o evento chave do Media Lab/BR junto com o #.ART, considerando ainda que desde o ano de 2020, pelos laços de pesquisa porém acelerado pelo distanciamento exigido pela pandemia, juntou-se a outros dois eventos, o DAT e o Retina.

Como o próprio nome já aponta, ele se concentra em apresentar projetos artísticos e técnicos, além de investigações em diversas áreas (não necessariamente estéticas), sobre mídias interativas e suas relações com diversas esferas da vida cotidiana. Também ocorre uma exposição de obras, principalmente aquelas apresentadas nas comunicações e palestras, mas não raramente artistas são convidados a apresentar seus trabalhos.

## 2.5. A próxima geração

Na entrada da segunda década do século XXI, as pesquisas e criações artísticas amadureceram e o espaço destinado à esta expressão está se consolidando com pesquisadores e artistas alcançando um certo prestígio internacional. O ambiente computacional já se tornou hegemônico, não há um país no mundo que não tenha acesso à Internet, mesmo que esteja em nível deficitário.

No início dos anos 2000, as preocupações de conservação para as obras de mídias instáveis tornaram-se ponto de importante interesse para pesquisas de universidades e institutos museológicos. Em 2013, o Prof. Bernhard Serexhe, coordenou “*The Project digital art conservation*”<sup>21</sup> (SEREXHE, 2013) e nas suas palavras de abertura ele escreve:

O problema geral abordado aqui, ou seja, a preservação e transmissão da cultura digital genuína de nosso tempo para as gerações futuras, está destinado a se tornar uma questão decisiva para todas as instituições culturais no século XXI, independentemente de apoiarem os padrões de valores tradicionais, estruturas

<sup>21</sup> Segundo Serexhe, o “Digital art conservation was chosen as the short, internationally understandable title. The project’s official title is Digital Media Art in the Upper Rhine Valley. Conservation – Restoration – Sustainability,” (SEREXHE, 2013, p. 13).

de funcionamento e objetivos, ou eles próprios procuram fazer parte da vanguarda do progresso tecnológico. (SEREXHE, 2013)

O impacto da obsolescência dos equipamentos digitais, eletrônicos e computacionais, criada muitas vezes apenas por uma questão comercial, tem preocupado os museólogos e historiadores da arte (cf. MORALES, 2019; e VELLOSILO, 2015) e aos poucos tem começado a despertar a preocupação dos artistas.

A principal característica desta geração é que ela é a primeira geração derivada de uma geração cujos agentes eram artistas nativos da expressão digital (a primeira geração era composta de cientistas e artistas curiosos com as possibilidades da computação e da eletrônica). Desta maneira, estes indivíduos pertencentes à terceira geração teriam muito a ganhar ao preservar as obras de seus professores e mestres - e as suas próprias - além de buscar escrever a história dos mesmos.

A inserção de outras ciências como as ciências biológicas (fisiologia de insetos, botânica, genética), geologia e cartografia (mapas, deslocamento, química de solos e rochas), física (óptica, lentes, luzes), entre outras, junto ao trabalho artístico fez com que a interdisciplinaridade e colaboração fosse condição *sine qua non* para o aprofundamento da estética das mídias instáveis, e dificultasse um pouco mais o processo de conservação, por exigir, entre outras coisas, uma linguagem ubíqua entre estes setores da ciência.

No aspecto da interdisciplinaridade, os grandes institutos, como o Media Lab – MIT, o ZKM e o ARS, acabam por ganhar mais força e notoriedade, pois apoiado em poderio econômico e sustentados por parcerias com universidades eles encontram inserção em qualquer linha paralela de criação<sup>22</sup>, como podemos ver no Media Lab – MIT com seus 22 laboratórios (Figura 10) com nomes tão poéticos como: “Future Sketches” (responsável por investigar a essência do código como mídia criativa) ou o “Nano-cybernetic Biotrek” (que busca inventar tecnologias de dispositivos nanoeletrônicos para criar novos paradigmas da simbiose vida-máquina) (Media Lab – MIT, n.d.), ou as obras de vida artificial criadas por Christa Sommerer e Laurent Mignonneau (artistas residentes na ARS Electronica de Linz).

---

<sup>22</sup> Inclusive, foi para garantir esta força de produção que os laboratórios Media Lab da UFG, UnB e Unifesspa (depois entrariam o Media Lab/UAM e o Media Lab/ PUC-Campinas) se uniram em uma rede, a Rede Media Lab/BR.

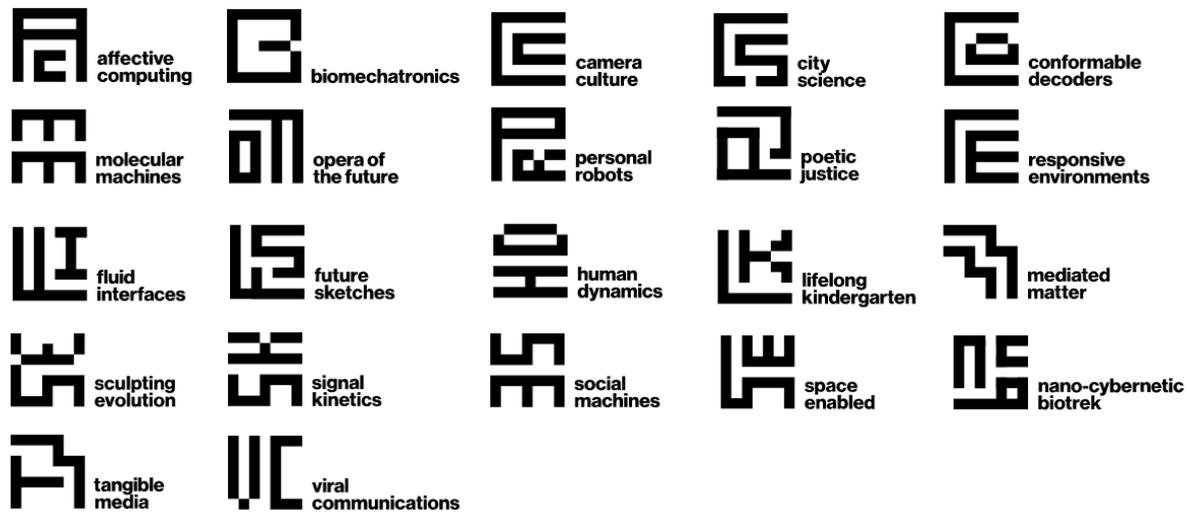


Figura 10: Laboratórios do Media Lab - MIT. Fonte: Media Lab MIT (n.d.)

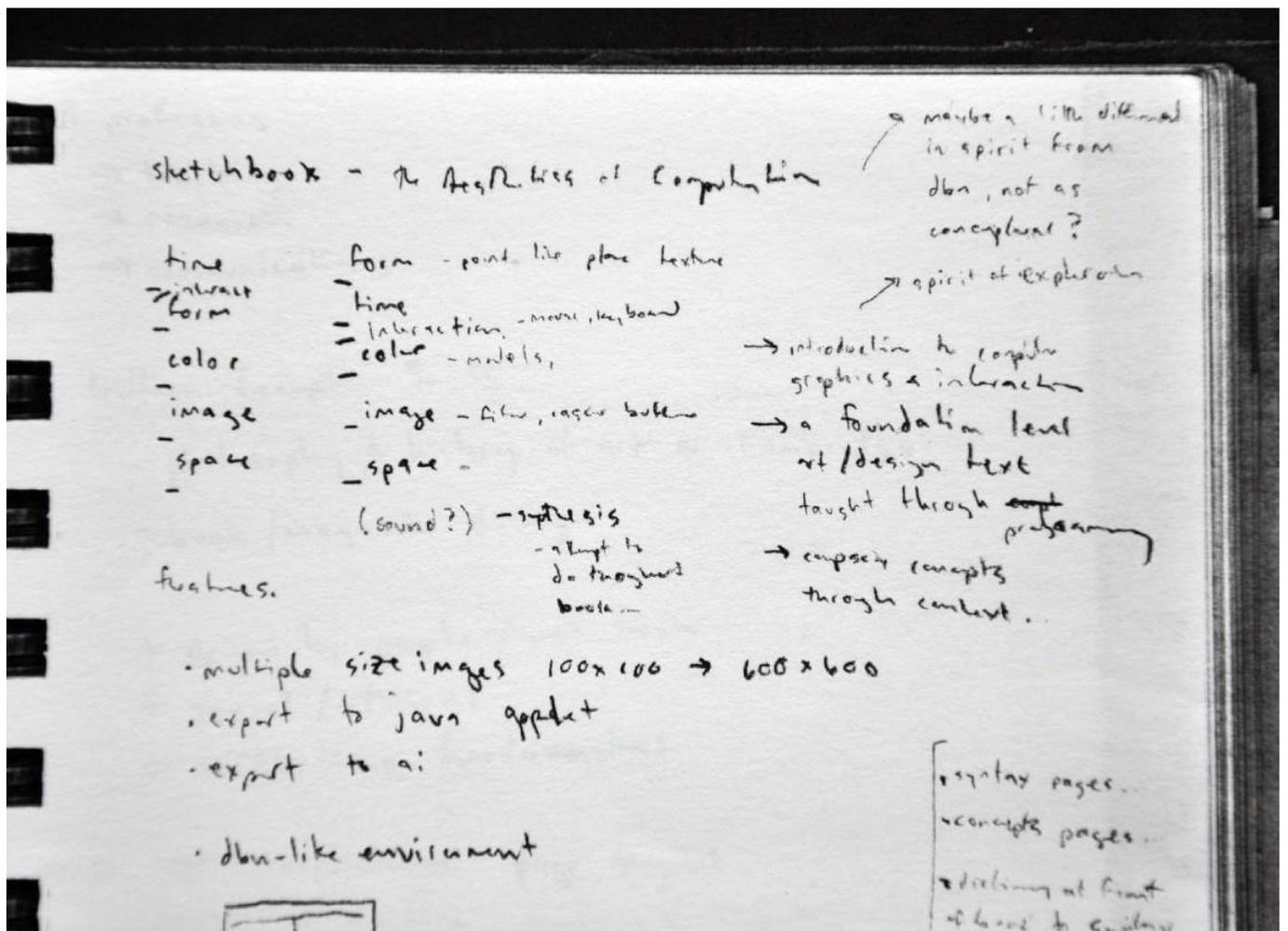


Figura 11: Notas no caderno de Reas. Imagem disponibilizada no artigo "A Moderne Prometheus". (2018)

- **Ferramentas para artistas-programadores**

Outro aspecto importante a se considerar nesta fase foi a invenção de Frameworks e softwares específicos para a criação artística. O Processing (Processing Copyright, n.d.), um dos principais Frameworks utilizados hoje na arte computacional e também uma linguagem de programação derivada do Java, foi um projeto em 2001 de Ben Fry e Casey Reas. Ben Fry na época era candidato ao PhD do MIT e Casey Reas estava trabalhando como Professor Associado no Interaction Design Institute Ivrea.

Processing é uma linguagem que surgiu da evolução do projeto “Design By Numbers” (DBN) de John Maeda (artista visual, programador, foi Coordenador Geral do Media Lab MIT, tendo sido professor por 12 anos à frente do grupo chamado “Aesthetics + Computation”). A DBN tinha a intenção de ser um software para ensinar artistas e designers a programar, Maeda também ajudou a desenvolver o Scratch, um programa feito para ensinar a lógica de programação para crianças e adolescentes.

Todavia, Reas e Fry apontam que o DBN tinha uma série de limitações: “DBN was a minimal system, the canvas was Always 100 x 100 pixels and only gray values could be used – there was no color” (REAS, C.; FRY, B., 2018). Assim, após uma tentativa de tentar modificar o DBN, Reas e Fry viram ser necessário desenvolver uma plataforma do início:

Nós queríamos fazer um sistema que era tão fácil de usar quanto o Design By Numbers, mas com uma ponte para realizar trabalhos mais ambiciosos. Nós queríamos permitir as pessoas trabalharem com cores, em tamanhos grandes, criar gráficos 3D, e mais. Rascunhos simples de Processing são quase tão simples quanto os rascunhos do DBN, mas o Processing coloca em uma escala maior – ele tem um “chão mais baixo” e um “teto mais alto”. O teto é similar aos programas C++ que costumávamos escrever com ACU parcialmente porque o resto da linguagem JAVA e suas bibliotecas estavam disponíveis. (REAS, C.; FRY, B., 2018).

Processing então é uma linguagem híbrida que se utiliza de uma base em JAVA, mas com objetos próprios. Os autores inclusive demonstram a diferença trazendo o código costumeiro dos iniciantes em programação - “Hello World” – em JAVA e em Processing:

```
public class HelloWorld {
  public static void main(String[] args) {
    // Prints "Hello, World" to the terminal window.
    System.out.println("Hello, World");
  }
}
```

**Figura 12:** "Hello World" codificado em JAVA. Fonte: REAS; FRY, 2018.

E,

```
print("Hello World!");
```

**Figura 13:** "Hello World" codificado em Processing Fonte: REAS; FRY, 2018.

O intuito com isso é tornar a programação um exercício mais acessível, permitindo que obras bidimensionais, tridimensionais, animações e outras<sup>23</sup> pudessem ser realizadas por artistas com aprendizado mais focado na realização de elementos visuais e menos interessados em uma programação que interagisse com a máquina em todos os níveis estruturais.

Para reforçar a atuação do Processing como linguagem visual, um artista que quisesse criar por exemplo um pequeno retângulo de fundo cinza, com linhas pretas sobre um cenário totalmente branco escreveria as seguintes linhas de código:

```
1. background(255);
2. stroke(0);
3. fill(150);
4. rect(50,50,75,100);
```

**Figura 14:** Retângulo cinza sobre fundo branco. Fonte: Acervo próprio (2020).

Os termos em inglês seguem um padrão internacional quando se trata de linguagem de programação (mesmo existindo linguagens que se utilizam de outros idiomas, inclusive fictícios, o mais comum são as Classes, um rótulo que caracteriza toda uma série de características de um determinado objeto, serem apresentadas em inglês), mas são simples:

**background:** significa o fundo do cenário, nele pode ser inserida uma cor, um gradiente, uma definição de cores. No caso, o código indica um padrão de cores, que por se tratar de um único número de três dígitos o programa já reconhece como sendo em tons de cinza. Caso fosse em padrão RGB (Red, Green e Blue), teríamos algo como (255, 255, 255) e teríamos o branco.

**stroke:** são as características da linha, neste caso apenas a cor da linha está definida, e como na situação do Background apenas um número está aparente e o 0 aponta no ponto diretamente inverso do 255 em uma gradação de 256 posições, ou seja, estamos falando do preto total.

**fill:** características do preenchimento de todos os objetos que forem listados abaixo. Segue o mesmo princípio do **background** e do **stroke**, e o fato do número apresentado ser 150, coloca-o mais ou menos no meio da gradação, o que podemos intuir ser um cinza mais escuro.

<sup>23</sup> Ao longo desses vinte anos, o Processing recebeu uma série de add-ons (bibliotecas de códigos que podem ser lido por um programa base adicionando outras funcionalidades ao mesmo) da comunidade que permitem que diversos periféricos sejam explorados como instrumentos de entrada e saída de interação com o Processing. A exemplo: "Arduino (Firmata)" de David A. Mellis que permite a comunicação direta com uma placa Arduino; ou a "Ketai" que é uma biblioteca para Android para que na programação você possa adicionar funcionalidades envolvendo os sensores, as câmeras, o multi-touch, o sistema de rede, o bluetooth entre outros e que foi desenvolvido por Daniel Sauter e J.Duran.

**rect:** aqui está uma das coisas mais interessantes do Processing: em outras linguagens de programação, para desenhar um simples retângulo uma série de códigos seria necessário para que o computador entendesse que a única função que ele precisaria executar é mostrar um retângulo cinza, o Processing reduz isso a um simples nome, `rect()` para retângulo, `ellipse()` para elipse, `line()` para linha, `arc()` para um arco, `circle()` para círculo, `quad()` para um quadrilátero, `square()` para um quadrado e `triangle()` para um triângulo formando as figuras Primitivas 2D.

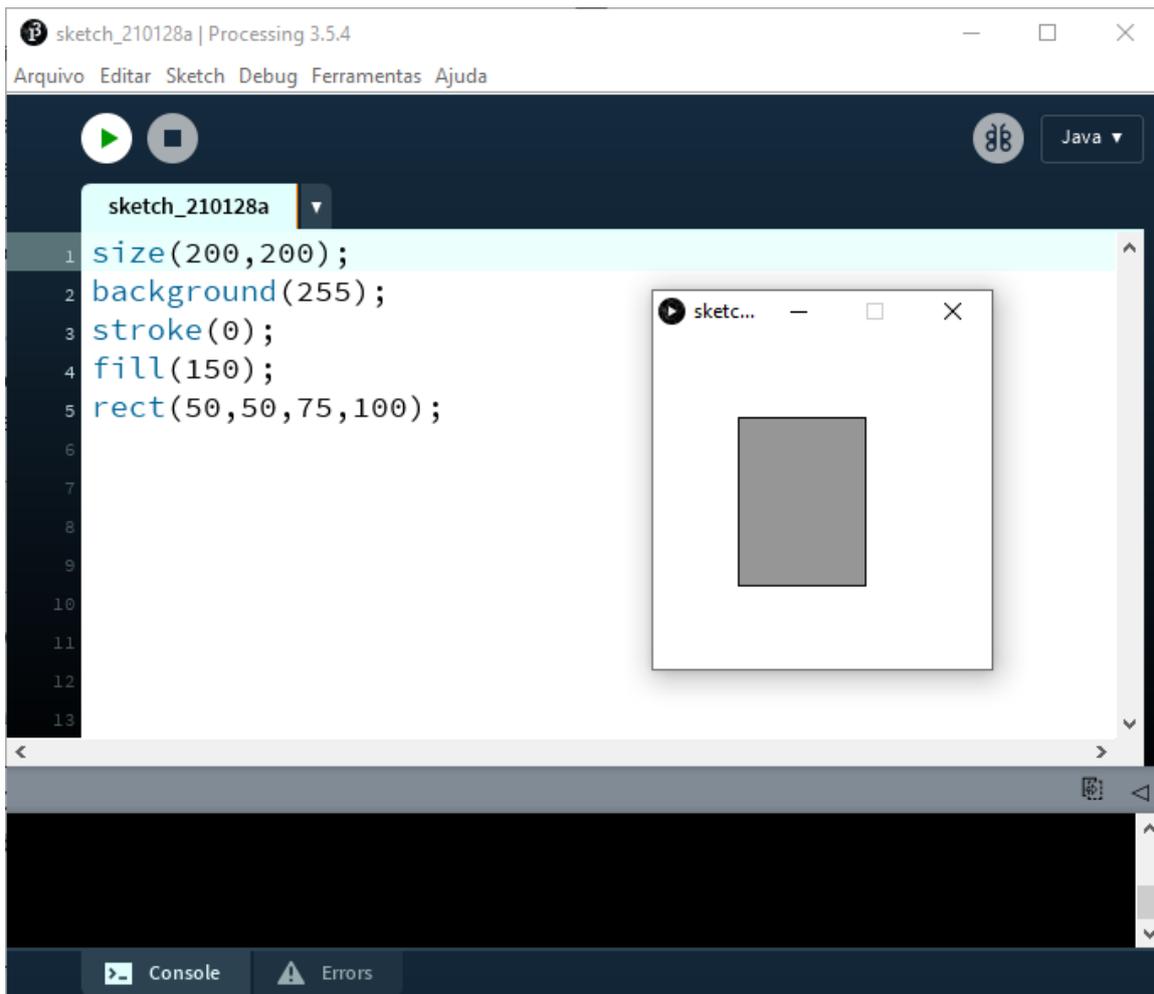
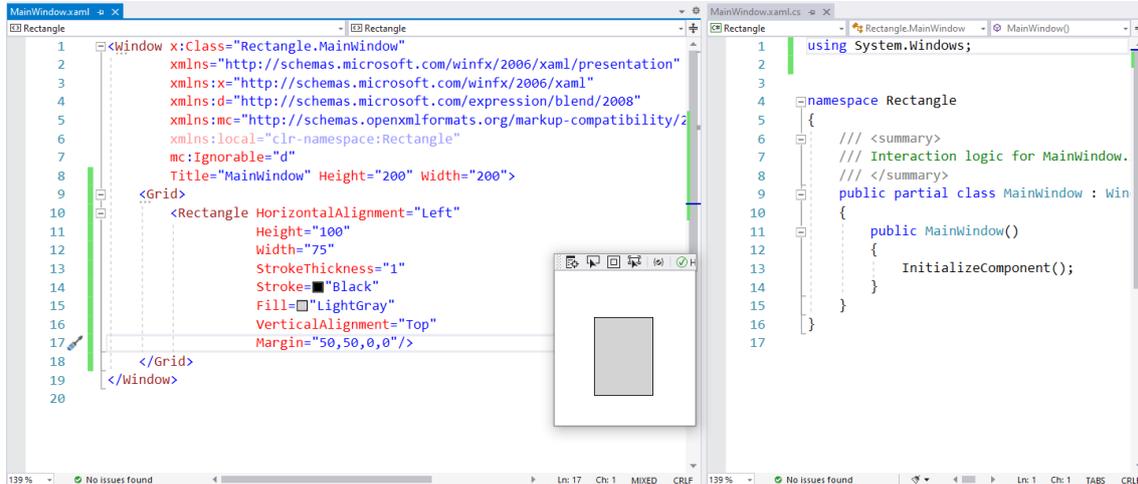


Figura 15: Captura de tela do framework Processing com o código da figura anterior e seu resultado.

Como qualquer um pode imaginar, os números dentro dos parênteses após o **rect** são relativos aos quatro pontos do retângulo, e são as posições em Pixels absolutos como coordenadas, no caso, a imagem tendo 200 x 200 pixels, o primeiro ponto fica a exatos 50 pixels do eixo X e 50 pixels do eixo Y e o ponto em sua diagonal fica no ponto 75 no eixo X e 100 no eixo Y, o resto das linhas o Processing se encarrega de preencher.

A título de comparação, temos a mesma figura programada em uma linguagem relativamente mais simples e com uma interface mais intuitiva se comparado com outras frameworks, mas ainda assim visivelmente mais complexa do que o processo em Processing:



**Figura 16:** Aplicação em WPF do Visual Studio em C# e XAML. Fonte: Acervo próprio. (2020).

```

1. <Window x:Class="Rectangle.MainWindow"
2.     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3.     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4.     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5.     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6.     xmlns:local="clr-namespace:Rectangle"
7.     mc:Ignorable="d"
8.     Title="MainWindow" Height="200" Width="200">
9. <Grid>
10.    <Rectangle HorizontalAlignment="Left"
11.        Height="100"
12.        Width="75"
13.        StrokeThickness="1"
14.        Stroke="Black"
15.        Fill="LightGray"
16.        VerticalAlignment="Top"
17.        Margin="50,50,0,0"/>
18. </Grid>
19. </Window>

```

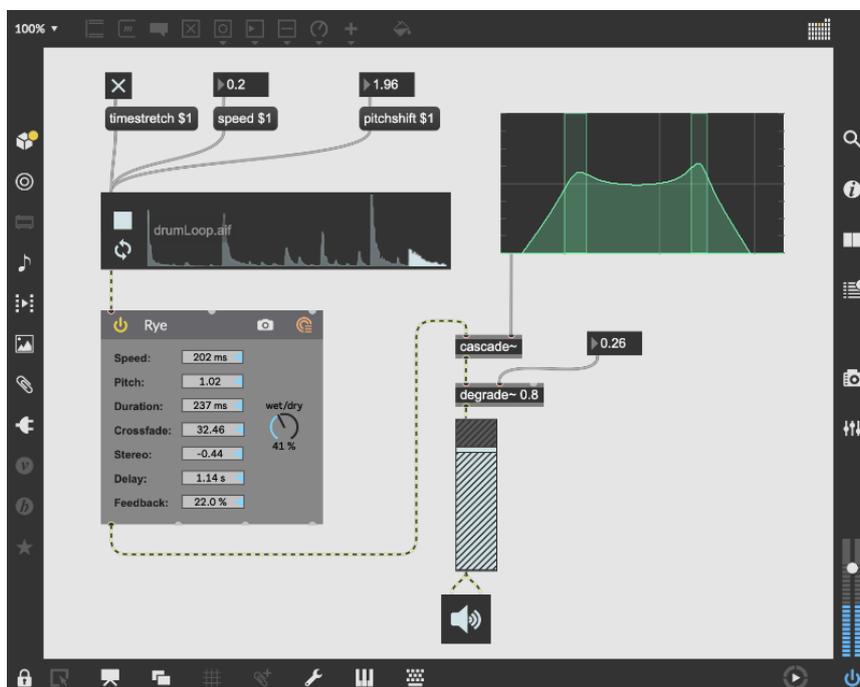
**Figura 17:** Código do XAML descrevendo o retângulo e suas características. Fonte: Acervo próprio (2020).

Mesmo que sejam executados códigos simples, estas linguagens e frameworks estão preparadas para executar todo tipo de trabalho de programação, assim, é como colocar um carro de corrida para ir buscar o filho na escola. Aqui cabe uma ressalva: códigos complexos também podem ser incorporados ao Processing.

Bem diferente dos códigos feitos em cartões perfurados do início da computação, a busca pela aproximação dos chamados “Usuários Finais” do uso de computadores domésticos - e

até para facilitar o ensino da programação nas escolas e universidades – algumas linguagens adotaram uma forma mais gráfica de elaborar um código. Os programas construídos para auxiliar a criar outros programas são chamados de IDE de “*Integrated Development Environment*” no original em inglês ou “Ambiente de Desenvolvimento Integrado” em português e simulam o fluxo de um algoritmo por disposições de linhas e quadros.

No ambiente artístico podemos traçar um início no software MAX desenvolvido por Miller Puckette em 1985, ainda com o nome de Patcher, sendo uma linguagem de programação visual aplicada para música e multimídia. Puckette também vai desenvolver outro software em 1996 uma ferramenta de composição open-source chamada de Pure Data (Pd).



**Figura 18:** Exemplo de um programa feito no MAX. Fonte: Cycling74, n.d.

Este tipo de programação “*no-code*” possui seus desafios técnicos (entender a função de cada caixa, alterar os valores, realizar os inputs e outputs dos dados), mas não é tão hermético quanto à sintaxe das linguagens de programação. Desta forma, vários programas utilizados na produção artística de mídias instáveis utilizam deste sistema de “Nós”. Além do MAX e do Pd cujo foco é na produção em áudio, temos entre os comumente usados: Houdini (programa muito usado pelas produtoras hollywoodianas para criação de efeitos especiais e animações); Quartz Composer (especializado em processamento e renderização de dados gráficos e só disponível para MacOS); o Reaktor (uma linguagem de DSP e processamento MIDI); o TouchDesigner (uma linguagem visual para conteúdo multimídia real-time); e o Blender que por sua versatilidade, longevidade no mercado

e por ser um dos programas que mais utilizo para minhas criações, requer um pouco mais de descrição.

O software Blender teve sua primeira versão lançada em janeiro de 1994, agora em junho de 2022 está na sua versão 3.1.2, que de longe não representa o tempo que este software está no mercado, fez com que houvesse a necessidade de mais habilidade e menos programação no trabalho de criar objetos e animações tridimensionais. Apesar de não ser o primeiro software a apresentar um ambiente de criação para imagens e animações tridimensionais este é um software Livre e Open- Source que alcançou um nível de qualidade e confiança que passou a ser adotado até mesmo por empresas grandes da indústria de games como a Epic Games, que passou a patrocinar a Blender Foundation, e desde 2006 com “*Elephants Dreams*” é utilizado na indústria cinematográfica<sup>24</sup>.

Em sua tese de doutorado em Artes Visuais pela Universidade de Brasília em 2019<sup>25</sup>, o artista e pesquisador Alexandre Galvão de Queiroz Rangel (RANGEL, 2019, p. 45) faz uma pequena lista dos principais ambientes de criação multimídia/tridimensional que encontraram um ápice de aceitação do mercado, e depois tiveram quedas vertiginosas. Rangel cita que programas como Macromedia Director ou o Maya (atualmente Autodesk) que ao serem comprados por outras empresas foram absorvidos em outros softwares ou mudaram tanto sua estética e filosofia que ficaram irreconhecíveis (Rangel, 2019, p. 44).

Diante do fato de que softwares comerciais costumam ser descontinuados ou serem transformados em interfaces totalmente diferentes diante dos fluxos de compra, venda, fusão e falência das empresas produtoras, os artistas das mídias instáveis migraram paulatinamente para softwares livres conforme estes evoluíam ao nível dos softwares comerciais. Nos últimos trinta anos os softwares livres conseguiram formar uma comunidade em torno deles que conseguiu em um primeiro momento fazer frente e depois até superar a eficiência das equipes contratadas pelas grandes empresas de software.

Este capítulo buscou demonstrar a forma como o artista vem utilizando do computador e do código como instrumentos ou suporte de uma expressão artística com semântica própria. Esta mescla os termos da ciência da computação e das artes, necessitando a construção de um tipo de linguagem ubíqua, uma forma de que possamos nos fazer entender em todos os processos envolvendo as nossas criações artísticas já que as documentações e especificações de uma obra não servem apenas para a restauração, mas também para a instalação da mesma em ambientes expositivos.

---

<sup>24</sup> Para mais exemplos e projetos cinematográficos (curtas e longas) acessem o site <https://www.blender.org/about/projects/>.

<sup>25</sup> “O artista como desenvolvedor de sistemas computacionais: experiências audiovisuais”.

Todavia, percebe-se que diante da busca da nova geração para aproveitar aquilo que Arianne Vellosillo cita como “humanidades digitais”, ou seja, tecnologias computacionais voltadas para ações humanísticas e culturais<sup>26</sup>, tem exigido um conhecimento mais aprofundado do que realmente é a “arte computacional”, e, a nosso ver, para pensar o que seria uma “Ontologia da Arte Computacional”, é preciso passar por uma análise ontológica do “Computador” e do “Programa” e esta será a abordagem do terceiro capítulo.

---

<sup>26</sup> Segundo a autora: “Existem algumas definições para humanidades digitais que apresentam pequenas variações, como: digital para humanidades, computação para humanidades, informática e cultura digital e informática para humanidades. De qualquer forma, todos apontam a importância do trabalho multidisciplinar e coletivo, e destacam o uso da computação matemática no desenvolvimento de projetos de pesquisa e ensino nas áreas de ciências humanas e da informação. As tecnologias digitais abriram imensas possibilidades para o desenvolvimento de teorias humanísticas em todas as áreas da cultura e do conhecimento. Com o uso de ferramentas computacionais, tem sido incentivada a participação ativa de pesquisadores remotos e em tempo real. Além disso, tem sido incentivada a produção e transmissão de conhecimento de acordo com o espírito do humanismo”(VELLOSILLO, 2015, p. 164)

## 3. Terceiro Capítulo: Uma ontologia da arte computacional

Este capítulo visa estabelecer uma investigação ontológica sobre a Arte Computacional, caracterizando as linguagens de programação, descrevendo suas características e aplicabilidade, bem como o uso do computador tanto como instrumento quanto como suporte e estando ele ocupando um papel na mensagem estética ou não.

### 3.1. Princípios de Ontologia da Arte Computacional

A Arte Computacional é um objeto estético analiticamente muito próximo das Artes Performáticas, já que é o resultado da execução do algoritmo que nos dá a fruição da obra. Mas, vimos em Sack (2019), que a linguagem de programação se aproxima da literatura. Se este é o caso, devemos considerar que a forma de documentação e a análise de sua fruição estética deva ser aproximada aquela da produção literária. Nesse mesmo sentido, é no trabalho do artista, na sua intencionalidade, ao moldar aqueles elementos em torno de uma significância que a torna uma obra de arte.

O código, como pudemos ver ao longo de sua evolução histórica, é uma linguagem de integração entre o Ser Humano e o Computador Aparelho<sup>27</sup>: no começo de seu desenvolvimento era muito simples, como um linguajar adotado por criança em início de aprendizado de fala. Contudo, hoje a complexidade aumentou conforme a linguagem de programação passa a dar conta de administrar muitas interações sensoriais do aparelho com o mundo exterior e traduzi-las para o que ficou conhecido como linguagem de máquina.

Portanto, dois objetos são analisados teoricamente para a investigação da Arte Computacional: o Computador e o Programa. Nos Capítulos 1 e 2, procuramos abordar os dois pelo viés histórico da Ciência e da Arte.

O exercício de reflexão ontológica se tornou fundamental nesta pesquisa, aproximando-a daquilo que faz a Arte Computacional ser a Arte Computacional. Aprofundarmo-nos, assim, na discussão sobre o que deve ser guardado, quando conservarmos a obra de arte computacional.

Como já citado anteriormente, o recorte desta tese são as obras de Arte que existem dependentes de um código de programação, sem excluirmos o fato de que necessariamente haverá um suporte midiático que independentemente de sua função estética deve ser levado em

---

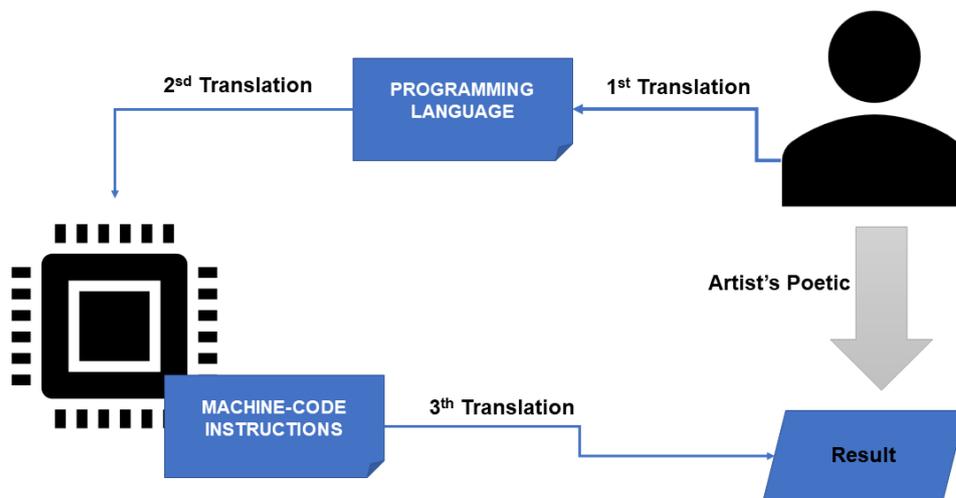
<sup>27</sup> **Nota do autor:** a título de orientação, ao apresentarmos alguns conceitos em caixa-alta estamos realizando uma reflexão sobre o nível mais abstrato ligados a este conceito.

consideração, pois a aplicabilidade da obra de arte computacional também depende da sua possibilidade expositiva.

A ideia em que baseamos as análises ontológicas é a de que a obra de Arte Computacional já existe como fenômeno a ser observado em sua ideia e construção da estrutura de programação, situação em que os esforços para replicar a obra no futuro (com uma das estratégias de restauração apresentadas no próximo capítulo) estarão desobrigados da linguagem de programação original, podendo serem escritas na linguagem que estiver disponível com mesma funcionalidade, apenas se concentrando nos efeitos estéticos buscados pelo artista. Denominaremos de “**Poética do Artista**” as perspectivas estéticas propostas e exemplificadas, daqui por diante no texto.

Tomaremos como base a abordagem de Sack (2019) sobre a programação como uma linguagem para estabelecer a comunicação entre o ser humano e o aparelho. A lógica da programação é um dos elementos que guiam a sintaxe e a semântica da Linguagem de Programação e é importante para considerar como se preservar um código. Textos preservados têm sido atualizados para a escrita contemporânea há muito tempo, caso contrário, textos vertidos em línguas antigas seriam quase que ininteligíveis. E, como já vimos no Capítulo 2, ainda há a questão da tradução, este elemento muito interessante para o computador já que em inglês a palavra para tradução (*Translate*) também é utilizada para um movimento, o que permite interessantes analogias, e que para Sack seria uma forma das Artes Mecânicas entenderem as Artes Liberais (cf. p. 37 desta tese).

A natureza do código de programação para um computador é de um sistema de três tipos de traduções (ou processo de Codificação/Decodificação, como em todos os processos comunicacionais): (1) A poética do artista é a de um resultado de saída (*output*), e é importante salientar que essa poética pode ser desenvolvida ao longo do processo criativo, pois é traduzida em uma linguagem de programação; (2) o código ali criado irá utilizar de processos internos à própria linguagem para traduzir a poética do artista em “linguagem de máquina”; (3) estas instruções em linguagem de máquina devem dar algum tipo de *output* que no caso das artes computacionais seria algum resultado sensorial que o artista irá comparar com aquilo que imaginava como resultado.



**Figura 19:** Diagrama das instruções do computador. Acervo do autor.

Outras traduções podem ser vinculadas ao diagrama apresentado na Figura 19, afinal hoje pouquíssimos usuários chegam a ver sequer os comandos textuais mais simples de um sistema operacional, escusável mencionar sobre um código de programação. Assim, há todo um repertório visual que desde a invenção do mapa de bits por Doug Engelbart (JOHNSON; BORGES; VAZ, 2001, p. 15–35), que levou à interface visual do computador contemporâneo, foi construída para representar os códigos que rodam “por detrás” da interface gráfica.

A aproximação que Sack (2019) faz entre a programação e a produção literária, pode encontrar resistência dentro da academia uma vez que a ideia mais habitual que se tem da computação é a de cálculos matemáticos, tendo o termo computador raízes em uma profissão que era especializada “no uso da régua de cálculo e na ultrapassada divisão de grandes números” (JOHNSON; BORGES; VAZ, 2001, p. 19). Além disso, o termo algoritmo que segundo Raul Wazlawick (2016), tem origem em uma obra denominada “*Algoritmi de numero Indorum*” escrita por Abdullah Muhammad bin Musa Al-Khwarizmi em que descrevia o cálculo utilizando números arábicos ou algarismos indianos que eram muito mais fáceis de lidar do que com os números romanos. Desta maneira o nome latinizado de Al-Khwarizmi passou a significar todos os procedimentos mecânicos para realização de cálculos.

Seguindo a origem no trabalho de Al-Khwarizmi, a abstração das etapas para o cálculo provavelmente seria a origem de outros conceitos relacionados ao termo algoritmo, pois também é mencionado como uma técnica formal para a programação de computadores (WAZLAWICK, 2016); a translação de um problema do mundo real em uma série de passos a serem cumpridos por um certo computador (CAMPBELL-KERRY *et al.*, 2018), e o “manejo formal de símbolos (não interpretados) mediante a aplicação de regras formais” (Alan Turing in GIANNETTI, 2006).

Então, mesmo que exista essa imediata correlação entre algoritmo e cálculos, e computador com números, o conceito de algoritmo pode ser reduzido em uma sequência lógica de instruções e que podem ser traduzidos não apenas na linguagem numérica, mas em qualquer linguagem.

Em um segundo nível de abstração, o código de programação ou programa é escrito<sup>28</sup> seguindo uma lógica de programação que é a mesma para qualquer linguagem de programação. Porém, cada linguagem traz consigo um jargão próprio de funções e classes e toda uma forma de sintaxe que se desobedecida irá apontar para um erro de processamento do código.

Neste ponto é o Computador, como equipamento interpretador do código, que possui restrições quanto a interpretação que dará sobre o código, fruto da natureza do próprio equipamento. Um dos imaginados futuros da “singularidade”<sup>29</sup> seria adicionar ao computador a habilidade de interpretar o código (ou os comandos como fazem crer as principais franquias de ficção científica) de acordo com o Contexto e o Cotexto<sup>30</sup>.

A natureza complexa do computador, da linguagem de programação e da relação destes com o Ser Humano e a sociedade contemporânea mundial, construída em torno destas relações, se tornam elementos muito complexos e profundos que agregados a outros elementos da cultura, do campo das artes e da estética, priorizam a necessidade de uma ontologia da Arte Computacional.

Ontologia é uma palavra resultante de duas outras palavras gregas: *onta* (entes) e *logos* (teoria, discurso, palavra) e sua criação é atribuída à R. Goclenius em que a escreveu em seu *Lexicon Philosophicum* de 1613 (CASTRO, 2008). Ainda de acordo com Castro (2008), ontologia seria o ramo da filosofia cuja pergunta primordial é quais são as características básicas que um objeto ou um fenômeno “É”.

A ontologia dividiu-se conforme os filósofos se apropriavam da mesma para construir suas formas de pensar e analisar os objetos, porém alguns princípios parecem ser comuns, conforme nos explica Castro (2008).

- Princípio de “Ockham’s Razor”: de início, um ramo que analisa os Seres pelo que faz deles serem eles mesmos necessita se ater a um mínimo de características: “constituição básica do mundo e o fornecimento [...] de um inventário econômico dos entes mais primitivos” (CASTRO, 2008)

<sup>28</sup> **Nota do autor:** Neste ponto é interessante observar que o termo é relacionado à produção textual e não ao cálculo matemático. O Código é “escrito” ou “written” em inglês, apenas quando o código é “executado” que passamos a comentar que o computador irá “calcular” as variáveis.

<sup>29</sup> **Nota do autor:** Singularidade é o termo comumente associado ao ponto virtual no futuro em que a tecnologia irá desenvolver-se a um ponto impossível de se prever (Cf. KENSKI, 2016).

<sup>30</sup> **Nota do autor:** Contexto e Cotexto são conceitos utilizados pelo autor Dominique Maingueneau na teoria da Análise do Discurso (MAINGUENEAU, 2011). Consideramos pertinente sua inserção aqui por ilustrar a relação dialógica esperada para a interface com uma “Inteligência Artificial” capaz de corrigir mesmo as imperfeições de codificação em programações específicas. Contexto seria, portanto, o ambiente do discurso, o tempo e espaço em que ele se dá. Já o Cotexto se refere a todos os textos auxiliares na interpretação do Discurso em cena.

- Princípio da não-contradição: nenhum ente pode ser e não ser ao mesmo tempo, com isso há uma coerência a ser obedecida.
- Princípio da individuação: quais critérios devem ser utilizados para diferenciar um indivíduo do outro? Segundo Castro (2008), os teóricos apontam para três maneiras diferentes: “(a) a reflexividade ( $x = x$ ), (b) a simetria ( $x = y$ , quando tudo que é dito de  $x$  também é dito de  $y$ ) e (c) a transitividade ( $x = w$ , se  $x = y$  e  $y = w$ )”. Ela ainda explica que o princípio da simetria é de Leibniz que diz que dois seres são iguais se possuírem a mesma qualidade (o que é apontado um questionamento pela autora já que dificilmente duas qualidades serão realmente iguais, dois lápis de cor vermelho não terão exatamente o mesmo tom). Para São Tomás “dois entes individuais são idênticos segundo a forma específica, mas distintos segundo a matéria de cada um” (São Tomás apud CASTRO, 2008). Já para Aristóteles, o princípio de individuação é a forma específica do ente, sua essência (CASTRO, 2008), vê-se assim possível que em uma espécie biológica, sejamos capazes de enxergar como semelhantes indivíduos que possuem qualidades diferentes (por exemplo cada ser humano).

Os princípios ontológicos são claros como diretrizes para uma Ontologia da Arte Computacional. Nesse sentido, procuramos atingir a essência de uma expressão artística cujos objetos pertencentes a este universo são diversos em formas, tamanhos, componentes, linguagens e, mesmo assim, requerem um mesmo tipo de cuidado básico na sua conservação.

Primeiramente, pode ser interessante uma reflexão quanto ao nome dado a essa expressão artística já que muitas vezes o termo Arte Computacional (*Computational Art*) se apresenta nos textos científicos como sinônimo de Arte Algorítmica (Algorithmic Art) (WEIBEL, 2009) ou Arte Numérica (Numerical Art) (COUCHOT, 2003).

Internacionalmente, o termo foi popularizado pelo *Cybernetic Serendipity* e pela Sociedade de Arte Computacional de Londres. No Brasil o primeiro a utilizar o termo foi Waldemar Cordeiro, que em 1971 organiza o evento *Arteônica*, como já mencionado no primeiro capítulo. No texto de apresentação do catálogo do evento, Cordeiro<sup>31</sup> defende o uso do computador nas artes como uma forma de pensar sobre as mudanças das mídias de massa que já na sua previsão seriam insuficientes para atender a uma massa populacional crescente, e ao buscar defender sua preocupação estética ele menciona o termo “*Computer Art*”.

Arlindo Machado (2015), Suzete Venturelli (2017) e Priscila Arantes (2021) são alguns dos autores que dão destaque ao vanguardismo representado por Cordeiro e por Giorgio Moscati, cientista físico parceiro nas primeiras experiências de arte computacional de Waldemar Cordeiro (cf.: MASCARENHAS; VELHO; HESS, 1993, p. 5–15), no movimento de “*Computer Art*” no Brasil.

---

<sup>31</sup> Cf.: (MASCARENHAS; VELHO; HESS, 1993, p. 21)

A nosso ver, desde Cordeiro e Moscatti o termo “Arte Computacional” tem sido relegado ao obscurantismo com cada vez menos teóricos citando o termo, o motivo tem sido uma relativa falta de perspectiva em relação ao campo de atuação estética desta expressão que tentaremos descrever com maior detalhamento neste capítulo. Grant D. Taylor chega a mencionar que: “O termo ‘computer art’ é raramente utilizado no discurso cultural atualmente. O uso do termo é impor um senso de nostalgia, uma reminiscência sobre uma era de pioneiros e máquinas antiquadas que já passou” (TAYLOR, 2014, p. 1).

Arlindo Machado (2015) elabora uma definição de “Arte Computacional” que pode ser o ponto de partida para nossa análise:

De um modo geral, entende-se por *computer art* um conjunto bastante diversificado de procedimentos, atitudes e estratégias da arte e do artista com relação ao computador. Num primeiro sentido, o computador pode ser encarado como uma ferramenta para a geração e o tratamento das imagens. Uma vez produzidas, modeladas (no caso das imagens tridimensionais) e eventualmente animadas e sonorizadas, as imagens são transferidas para outro suporte (papel, tela, filme, vídeo) e exibida nas formas tradicionais em galerias de arte ou salas de projeção. Na verdade, são raros os casos em que o computador é utilizado estritamente como ferramenta, como se fosse um pincel ou uma paleta mais sofisticados. Muito frequentemente, o trabalho do artista acaba sendo contaminado por alguns processos formadores próprios da informática, de modo que o resultado final não poderia jamais ser obtido de outra forma. (MACHADO, 2015, p. 30)

E, completa,

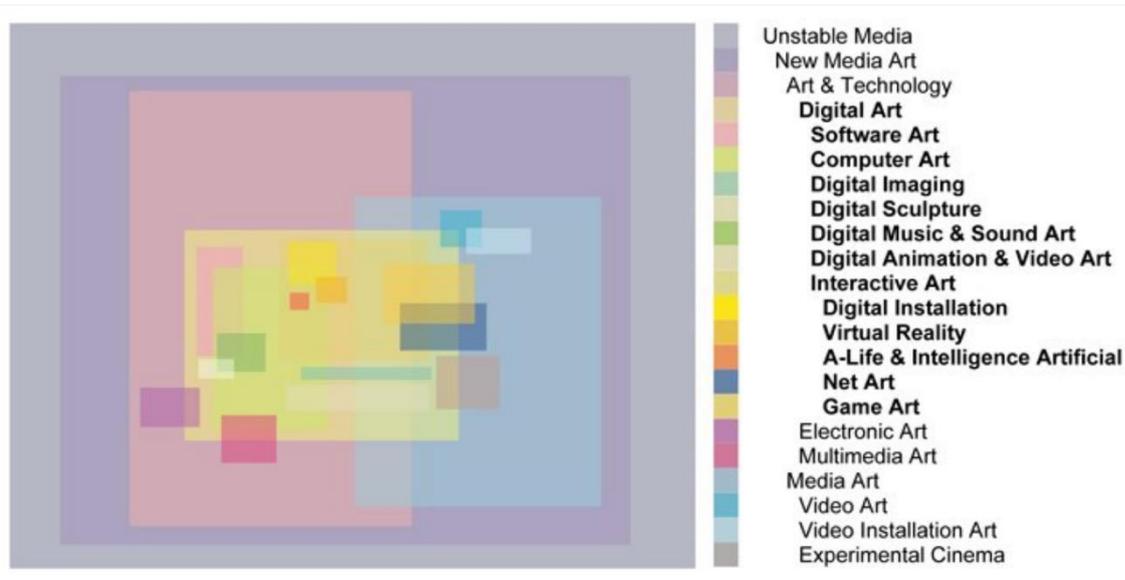
Numa segunda acepção, é o computador que cria a obra, a partir de um programa de criação previamente concebido pelo artista. Neste caso, é possível que a forma final de exibição seja também o circuito tradicional da arte, mas a diferença está no fato das decisões sobre o que fazer e como fazer serem tomadas pelo próprio computador. O artista, neste caso, apenas prevê um conjunto e possibilidades de comportamento do computador, em geral utilizando conceitos de Inteligência Artificial.

Segundo Machado, as obras de Arte Computacional<sup>32</sup> seriam todas as expressões artísticas que utilizassem de alguma forma o computador aparelho. Neste caso, há um padrão que se repete em várias formas de categorização das expressões que se utilizam das tecnologias da comunicação e da computação que é utilizar o suporte ou instrumento como forma de identificar a expressão artística.

Em artigo publicado em 2018 na revista *E-Conservation*, os professores Lino Garcia Morales e Pilar Montero Vilar tinham por objetivo discutir as dificuldades na preservação de Arte Digital e para encaminhar sua argumentação demonstraram como a Arte Digital era observada dentro da classificação das obras midiáticas (Figura 20).

---

<sup>32</sup> **Nota do Autor:** Deste ponto em diante no texto, consideramos que as menções são feitas à Arte Computacional (Computational Art) e não à Arte do Computador (Computer Art), justamente por nossa crítica à classificações determinadas pelo nome do suporte.



**Figura 20:** VILAR, Pilar Montero; GARCÍA, Lino. THE CHALLENGES OF DIGITAL ART PRESERVATION. E-Conservation, n. May, 2018. p. 2.

O diagrama de classificação apresentado na Figura 20 demonstra as imbricações, as hibridizações e as contaminações apresentadas nas obras das artes midiáticas. No caso da Arte Computacional, podemos observar que ela se encontra dentro dos campos da *Digital Art*, *Media Art* e da *Art & Technology*, que por sua vez estão contidas dentro do campo do *New Media Art* e este está contido dentro do campo *Unstable Media*.

No sentido contrário, contido no campo da *Computer Art* estão: *Digital Music & Sound Art* e *A-Life & Intelligence Artificial*; e, além disso, parte de outros campos estão em contato com o campo da Arte Computacional: *Digital Installation*, *Virtual Reality*, *Multimedia Art*, *Digital Sculpture*, *Digital Imaging*, *Digital Animation & Video Art* e *Interactive Art*.

Como foi visto no Capítulo 1, os computadores não são apenas os aparelhos digitais, apesar destes serem os mais comuns após a Segunda Guerra Mundial. Todavia, para muitas atividades os computadores analógicos ainda são um importante instrumento, além disso não há espaço nesta classificação para futuras alterações como as da computação quântica que, por enquanto, só tem agregado rapidez no processamento gráfico como na obra *Ent-* (2021) de Libby Heaney apresentada na exposição “*BioMedien: The Age of Media with Life-like Behavior*” (Figura 21).



**Figura 21:** Um still de Ent- (2021) HEANEY, Libby. Fonte: DAMN magazine. Disponível em: <https://www.damnmagazine.net/calendar/libby-heaneys-ent/>. Acesso em: 09/05/2022.

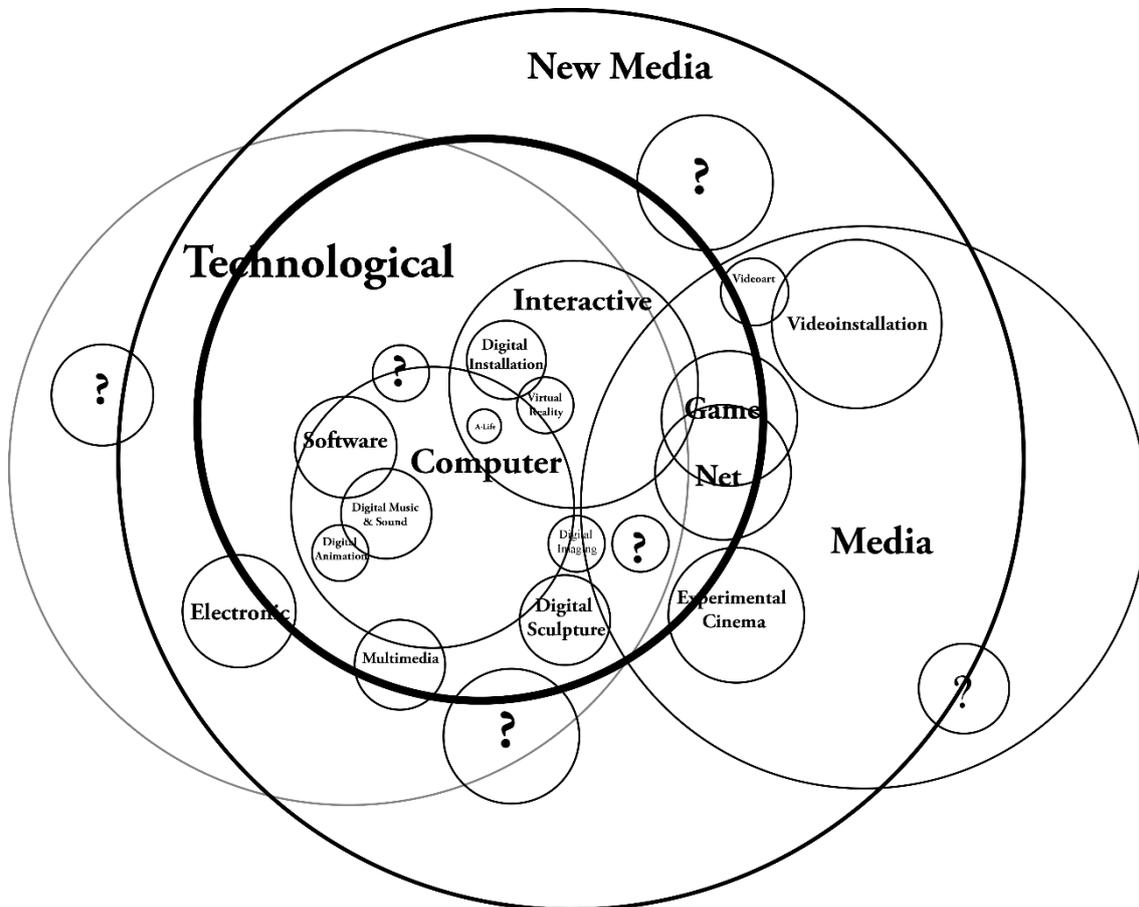
Heaney explora uma recente modificação na tecnologia da computação digital, a computação quântica. A obra é uma instalação de projeção 360° que apresenta seres digitais e camadas de texturas e partículas ao mesmo tempo e pretende discutir o uso da tecnologia quântica para o futuro.

A obra de Heaney demonstra, de maneira audiovisual e bastante lúdica, como o próprio computador quântico lida com as informações que ele gerencia. Em uma determinada cena da projeção, o *iterator* é levado para dentro do buraco de uma árvore que anteriormente estava à sua frente. Os processos e seres virtuais que estavam ativos do lado de fora do buraco continuam a serem reconstruídos e vistos pela entrada da caverna feita pela árvore, enquanto isso vê-se dentro daquela caverna outros seres, enquanto diversas partículas dançam pela tela. Mesmo assim, não há uma diferença estética preciosa<sup>33</sup> apenas uma diferença na velocidade de processamento e a melhoria nos gráficos.

Na obra “*Teoría de la Conservación Evolutiva*” (2019), o Prof. Lino García Morales propõe outra forma de classificação (**Figura 22**). A começar, o conceito de *Unstable Media*, cunhado pelo grupo V2\_Lab da Holanda, é substituído pelo *New Media Art* baseado na conceituação do professor e artista Lev Manovich (2001) em *The Language of the New Media*. A arte por computador continua a ter um espaço central e as mesmas expressões estão classificadas contidas totalmente ou em parte dentro do campo *Computer*. Todavia, este diagrama apresenta uma inovação em relação ao

<sup>33</sup> **Nota do Autor:** Não tentamos descaracterizar o trabalho da artista e pesquisadora, levantar este ponto é importante para constatar que o processo está apenas em seu início.

anterior, pois os círculos com interrogação (?) representam obras que no diagrama anterior não possuíam um lugar definido e, portanto, sem denominação.



**Figura 22:** MORALES, Lino García. Teoría de La Conservación Evolutiva: conservación y restauración del arte de los nuevos medios. 1. ed. Online: BoD - Books on Demand, 2019. Disponível em: [www.bod.com.es](http://www.bod.com.es) p. 28.

Percebe-se que independentemente da classificação realizada acima, nas artes midiáticas, eletrônicas etc. é muito comum o apelo que o instrumento ou o suporte realiza<sup>34</sup>. Isso leva à classificação de uma determinada expressão ser nomeada pelo suporte somente e não pelas diversas intenções poéticas que se encontram em uma exploração estética. Para exemplificar, tomemos uma obra de uma escultura digital que se apresenta em Realidade Aumentada, esta não será apresentada dentro da categoria Arte Escultórica, mas Arte em Realidade Aumentada apenas por causa do suporte em que a obra, evidentemente criada segundo todos os preceitos e metodologias das artes escultóricas, foi apresentada. Se seu suporte fosse outro, como talvez o Mármore, seria correto afirmar que se chamaria Arte do Mármore?

<sup>34</sup> **Nota do autor:** A tese de Arianne Vellosillo (VELLOSILLO, 2015, p. 209) apresenta como parte de uma ontologia, a classificação das obras contemporâneas em que o terceiro e o quarto subnível de classificação observa quais são os suportes da obra. O interessante no diagrama apresentado pela pesquisadora é a possibilidade de hibridismo das obras, mesmo tendo um de seus nós o suporte. (Cf. KENSKI, 2016)

Essa situação ocorre por conta de suporte e imagem da obra computacional não serem ambas do mundo material como eram nas artes tradicionais. Morales (2019) afirma que ontologicamente a obra computacional tem sua essência dividida em duas partes: Imagem (funcionando como Aspecto) e Suporte (funcionando como Estrutura). Os termos Aspecto e Estrutura são conceitos da teoria de restauração de Cesare Brandi (2019) e Morales afirma que essa separação é importante para a Restauração de uma obra: a partir do momento que define o suporte como sistema e a imagem como símbolo, se estabelece uma relação ontológica em que “objeto-símbolo” é como o texto, a mensagem trazida pelo objeto, o conteúdo; por sua vez, o “objeto-sistema” é o continente daquela mensagem, o significante (MORALES, 2019, p. 13).

Morales também busca suporte teórico nas divisões substância-cronológica de José Luis Brea (BREA, 2010) que divide historicamente a recepção da arte em três períodos, a saber: a Imagem-Matéria, a Imagem-Filme e a Imagem-Eletrônica; a primeira explorava a dimensão do espaço e da matéria, a segunda adicionou as dimensões do tempo e do movimento, e a E-imagem (denominação secundária dada por Brea) abstraiu as outras dimensões e tornou-se toda temporal. Imagem-Código é uma obra apenas da dimensão do tempo, assim como Flusser havia colocado a imagem técnica como fruto da zerodimensionalidade (FLUSSER, 2008, p. 13), não há matéria na Imagem, ela só encontra materialidade no suporte, razão pela qual imaginamos que as classificações das “artes das novas mídias”<sup>35</sup> estejam centradas no suporte.

Retornando em nossa reflexão ontológica, em uma obra computacional a visualidade encontra-se “virtualmente armazenada” nos códigos, e qualquer ‘código’ é um sistema de signos (MORALES, 2019):

A Imagem-Código é informação, independentemente do suporte que a contenha, passa de um suporte a outro, se multiplica pelas redes, se copia, transfere, armazena, processa etc., sem perder sua essência ou qualidade. Na Imagem-Código todas as cópias são originais. O código está escrito em uma linguagem formal criado pelos homens, mas para ser processada pelos computadores, não pelo homem.

A Imagem-Código é substância dos “novos meios”. O computador interage com o “mundo real” mediante suas próprias interfaces (denominadas comumente como periféricos); é a interface entre o artista e a realidade, o instrumento capaz de alterar a percepção da realidade como se fosse um plástico moldável. (MORALES, 2019, p. 16).

Essas descrições são importantes para nos concentrarmos na principal hipótese levantada nesta tese, de que uma vez que o código possua instruções e que a visualidade só se dá no suporte e que este suporte não é necessariamente um suporte único, sua essência e sua identidade, estão na ideia e na instrução poética, portanto, qualquer tentativa de conservação ou restauração estará dependente deste fato.

---

<sup>35</sup> Cf. Nota 15.

Citamos novamente Morales (*op. cit.*), que afirma que qualquer processo de conservação ou restauração de uma obra tem como problemática inicial a identidade da obra, sua essência, em suma: “aquelas propriedades ou atributos de uma obra que a fazem diferente do resto e a definem como ‘única’” (MORALES, 2019, p. 16).

Lino Morales continua com a sua interpretação sobre a Metafísica de Aristóteles de que a identidade de um objeto é determinada por sua matéria, forma, eficiência e função, obedecendo os princípios aristotélicos da causa material: “determina de que é feita uma coisa” (MORALES, 2019, p. 17); causa formal: “a disposição ou forma de uma coisa” (*op. cit.*); causa eficiente: “como chega a existir uma coisa” (*op. cit.*); e causa final: “a função ou objeto de uma coisa” (*op. cit.*); o professor então conclui que estas quatro causas permanecendo depois da modificação, o objeto continua o mesmo.

A causa aqui é a razão pela qual o ente (ser ou objeto observado) é. Assim, seria correto afirmar que aquilo que forma um objeto de Arte Computacional é o Computador? Se assim fosse, seria como a fotografia que teria sua causa material na Câmera Fotográfica. O Computador é Suporte e Mídia das obras de Arte Computacional, mas não pode ser sua causa ontológica assim como a tela não o é da pintura.

Dentro da Teoria do Restauro de Cesare Brandi a importância da definição da causa material é o fato de que isso irá representar o “*tempo e o lugar da intervenção de restauro*” (BRANDI, 2019, p. 36)<sup>36</sup> e por isso “a matéria se mostra como ‘aquilo que serve à epifania da imagem’” (*op. cit.*). Ainda, segundo Brandi, a matéria na obra de arte será desdobrada e então definida como “Estrutura” e “Aspecto” (*op. cit.*) que não devem entrar em contradição. Apesar de Brandi estar lidando com obras de arte tradicional, suas afirmações podem nos ajudar a distinguir os traços ontológicos da Arte Computacional.

Primeira crítica à análise de Brandi e que se faz necessária à nossa análise é a de que a Arte Computacional é em geral multissensorial, e diversas obras têm o visual como prioridade, mas são poucas que se apresentam apenas visualmente. Neste ponto é necessário um pensamento interdisciplinar para abarcar todos os aspectos sensoriais como causas materiais.

A nossa segunda crítica corresponde ao fato de que comprovado que o Computador como aparelho não é causa material das obras de Arte Computacional, então para a causa material deverá ser considerado a virtualidade, a abstração da matéria, que um código representa.

Para a primeira crítica é necessário analisar o Computador como objeto estético (item 3.1.1.), já a segunda teremos a análise do Código como objeto estético (item 3.1.2.).

---

<sup>36</sup> **Nota do autor:** A formatação da fonte dos dois conceitos está como no original, demonstrando em nossa interpretação que o leitor deve levar estes termos a um certo nível de abstração.

### 3.1.1. O Computador como objeto estético

A história do uso do computador nas Artes [...] começou aproximadamente sessenta anos atrás. Porém, até cerca de dez anos atrás, um tipo de eterno presente reinou no mundo da arte digital. Arte computacional tem um passado, mas falta a memória. A História não parou em 1984, porque alguém apagou todos os traços do passado, mas porque faltou a memória (Rosen, 2011, p. 9).

Na afirmação de Margit Rosen, acima transcrita, vemos que o uso do aparelho Computador nas Artes é um processo comum desde a década de 1950. Contudo, a natureza deste uso difere-se conforme os artistas, desenvolvedores e engenheiros foram se familiarizando com o equipamento e modificando-o para adicionar funcionalidades, mimetizar processos analógicos ou melhorar a interface (cf. Capítulo 1).

Como vimos anteriormente, tomar o Computador como parte do processo estético da Arte Computacional pode seguir dois caminhos: utilizá-lo como simples ferramenta digital em que o artista toma todas as decisões e movimenta os mecanismos de *input* e depois reproduz a imagem produzida em um suporte analógico ou o artista prevê uma série de variáveis e possibilidades de comportamento do computador e ele autonomamente toma as decisões.

Nestes dois caminhos, o Computador é parte do processo criativo da obra. Ainda, devemos levar em consideração o momento em que o Computador apenas ocupa seu espaço como suporte das obras. Como já mencionamos anteriormente, o Computador não pode ser ele um instrumento como o martelo ou a chave-de-fenda, ele também não é como uma máquina de escrever ou uma tela reprodutora de vídeos, ele pode executar o papel como, mas seria uma utilização muito aquém da sua capacidade. Este, inclusive, foi a apreensão de Moscatti quando Cordeiro Ihe foi apresentado e informou seu objetivo que era utilizar o computador para fazer arte (MOSCATTI, 1993):

Inicialmente fiquei relutante pois temia que Cordeiro tivesse ideias preconcebidas e quisesse usar o computador como um novo conversor gráfico que estava entrando em moda. Rapidamente verifiquei que esta não era sua intenção. Desejava sim estudar as possibilidades deste novo instrumento em Artes e percebera que era necessário conhece-lo em profundidade (...) (MOSCATTI, 1993).

Caberia em uma mesma classificação aquele que se utiliza do Computador como mero instrumento para digitalizar práticas analógicas e, na maioria das vezes, apresentá-la em suporte analógicos? Historicamente, como já vimos no Capítulo 1, isso foi muito comum nos primórdios da inserção do Computador no processo artístico e a pergunta que fica é se isso deve continuar sendo considerado.

Outra questão levantada na nossa avaliação é: qual o papel estético de um Computador quando este não representa um papel na fruição da obra? Tomemos a obra *Legible City* de Jeffrey Shaw (Cf.: SEREXHE, 2013, p. 488–518 e ; SHAW, 1989), as partes visíveis da obra, são a bicicleta imobilizada, a grande tela à frente e ao fundo da sala onde é projetada a imagem da cidade feita de palavras e uma tela menor imediatamente à frente da bicicleta que mostra a visão aérea do mapa

que o ciclista irá percorrer. Esta obra é inegavelmente dependente do Computador para ser exibida, a descrição do computador (um Silicon Graphics IRIS 4D/20) aparece na ficha técnica de *Legible City* que executa um sistema gráfico criado pelo MIT e pelo projeto ARPA com fins militares, o “*The Aspen Movie Map*” e que Shaw é um dos artistas a utilizarem o sistema para fins estéticos (VENTURELLI, 2017, p. 171), mas não se apresenta no palco. Caso refeita a obra ou rerepresentada a chance deste mesmo aparelho ser o suporte para a obra é mínima.

A associação do aparelho Computador com a obra é profunda e muitos consideram sua participação como coautoria: Giorgio Moscati apresentou o Computador como um dos “Atores” da sua narrativa sobre as primeiras experiências de Arte Computacional com Cordeiro (MOSCATTI apud MASCARENHAS; VELHO; HESS, 1993, p. 4). Já no manifesto “Arteônica”(CORDEIRO, 1971 in MASCARENHAS; VELHO; HESS, 1993, p. 21–23), Cordeiro denomina o computador como “*partner*”.

Esse papel é importante para o processo de criação artística, retornando a comparação flusseriana do aparelho com uma agulha (FLUSSER, 2011, p. 39), vemos que no momento de uma criação de uma obra computacional os limites dados pelo Computador e pelo conjunto de softwares nele inserido (sistema operacional, drivers, aplicativos, etc.) são parte da criação, uma agulha diferente feito de material diferente tem um único uso (passar uma linha enquanto penetra em um material); um computador diferente com configurações diferentes possui um repertório diferente de ações e habilidades, afetando assim o trabalho.

Em nossa interpretação, um computador não poderá assumir uma autoria de uma obra por não ser ele a determinar seus parâmetros, pois não cabe ao computador aumentar seu repertório sígnico – talvez apenas se e quando a singularidade for alcançada – mas as manipulações simbólicas feitas na obra quando são determinados os parâmetros, mesmo que dentro de limites físicos, dão oportunidade ao Computador de ser listado como coautor do processo.

Revisitando os conceitos flusserianos (FLUSSER, 2011, p. 89), vemos que a ideia do computador como uma máquina de “cálculos” não é tão distante da real natureza do computador, todavia a perspectiva sobre a forma destes cálculos possa ter sido influenciada pelos primeiros aparelhos computadores que serviam apenas para cálculos matemáticos básicos<sup>37</sup>. Os “cálculos” mencionados por Flusser são na realidade os menores elementos claros e distintos que podem ser usados para analisar e “pensar cartesianamente” (op. cit.) e ele afirma que o Computador é esse aparelho para pensar cartesianamente, já que “pensar é manipular um ábaco de conceitos. Todo conceito claro e distinto significa um ponto lá fora no mundo das coisas extensas (*res extensa*)” (op. cit.).

---

<sup>37</sup> **Nota do autor:** Como vimos no Capítulo 1, o Computador tornou-se a máquina diferencial buscada por Babbage, já que no início apenas fazia os cálculos básicos de maneira automatizada.

Ainda para Flusser (FLUSSER, 2011, p. 90), os computadores criam universos em que seu “pensamento” reduzido a conceitos traduzidos em “0”s e “1”s, os tornam oniscientes e onipotentes. Esse ambiente em que o código é gerado e executado deve ser analisado, pois sendo não apenas Suporte, mas Mídia, este universo, esta forma de pensar, se torna parte da mensagem traduzida pelo código.

Como “objeto-sistema”, o Computador é suporte e geralmente a única parte material da obra:

Na estrutura da arte digital, apenas o suporte é matéria ativa; geralmente um computador ou processador digital em geral, contendo informações na forma de dados e processos (transformadores de dados); tanto intangíveis, imateriais, virtuais. (MORALES, 2019, p. 61)

Quando participa apenas como suporte, a preocupação de conservação deve ser apenas com os componentes de software e o hardware deve ser apenas mantenedor da estabilidade do software, ou seja, mesmo que se mude o hardware, deve se garantir que ele seja capaz de rodar o ambiente completo do software para a obra.

Já quando o computador toma lugar na composição da mensagem da obra, teremos a preocupação de manter também a parte material, podendo mudar, quando possível, o interior dos equipamentos, preservando a parte externa.

Este foi o caso da Recriação de “Colóquio dos Móviles” de Gordon Pask, uma das obras apresentadas no *Cybernetic Serendipity* que na época muitos dos componentes foram criados manualmente por Pask e seus ajudantes.

É um grupo de objetos, os móveis individuais, que entram em discussão, que competem, cooperam e aprendem um sobre o outro. [...] Cada móvel individual tem um conjunto de programas que determinam seus movimentos e seu estado de visibilidade. Cada indivíduo aprende como executar seus programas em ordem de forma a atingir um objetivo; especificamente, para reduzir uma ânsia intrínseca. Seu nível de “satisfação” é refletida parcialmente por seu comportamento e parcialmente por um display visual. (PASK apud REICHARDT, 1969, p. 34–35)

Entre os móveis existem as “Fêmeas” e os “Machos”, e o uso do termo “*whimsy*”<sup>38</sup> por Pask para descrever os dois tipos de móveis, demonstra que o intuito não era criar uma diferença de gênero nem tipificar cada gênero, mas criar uma dicotomia que traduzia entre duas formas complementares da mesma “espécie”. O móvel dito feminino assemelhava-se a um casulo com uma grande abertura oval, já o macho era uma placa retangular com espelhos e uma fonte de luz central.

Em 2018, iniciou-se um projeto para realizar a réplica desta obra<sup>39</sup> que foi “possibilitada” pelo fato de muitos dos sistemas desenhado por Pask em 1968, já existirem para aquisição no mercado,

<sup>38</sup> **Nota do autor:** Whimsy tem tradução difícil para o português, podendo ser traduzido como a expressão “por diversão”.

<sup>39</sup> **Nota do autor:** A obra recriada em 2018 foi apresentada em Paris e depois foi adquirida para a coleção permanente da ZKM onde tivemos a oportunidade de vê-la em funcionamento parcial. Para mais informações, conferir Colloquy of Mobiles in <https://www.colloquyofmobiles.com/>.

assim, circuitos foram substituídos por peças de Arduino, Adafruit e Sparkfun, e lâmpadas incandescentes foram substituídas por LEDs, como pode ser visto no documento de especificações do projeto<sup>40</sup>.

No caso do “Colóquio dos Móviles”, a parte material dos computadores e componentes eletroeletrônicos não representavam nenhuma função estética, na realidade esta função será desempenhada pelos processos computacionais, voltando o hardware a condição de suporte. Assim, cabe analisarmos a condição do código como objeto estético.

### 3.1.2. Sobre o código como objeto estético

A Internet trouxe possibilidades de comunicação e informação multimidiáticas concentradas em uma única fonte (cf. Segundo Capítulo). A sociedade passou pouco a pouco a adotar ferramentas e sistemas informatizados que utilizam a armazenagem “em nuvem”. Este processo de armazenagem que não utiliza o espaço físico de seu computador, mas o espaço físico de um computador de terceiro em servidores<sup>41</sup> e são acessadas por protocolos de comunicação entre computadores (por exemplo FTP), ou por meio de um acesso personalizado mediado por aplicativo de acesso web (os sincronizadores dos principais serviços).

Esse serviço permitiu o aumento da colaboração com equipes descentralizadas e o uso de um mesmo arquivo por pesquisadores de instituições diferentes. Também permitiu a utilização de *softwares* que executam no servidor permitindo a diversos usuários acesso a um mesmo software utilizando um mínimo de espaço físico e de memória de processamento em seu computador pessoal.

Para as Artes esta utilização de softwares em nuvem permitiu a descentralização da produção de algumas obras e a exibição das mesmas, mas, em contrapartida, adicionou uma exigência para a conservação das mesmas: estar sempre online.

A razão por citarmos este fato neste ponto da tese é que diante das categorizações estéticas e da busca ontológica da Arte Computacional, o código como objeto estético se dá em dois momentos: 1. O resultado sensorial é alterado conforme se altera as variáveis de um código escrito no momento da exibição (*Live Coding*); 2. O código possui elementos poéticos pertinentes a uma interpretação sígnica ou literária (*Code Poetry*).

Estes dois caminhos representam obstáculos à conservação poética, pois diferentemente das obras computacionais cujo código permanece salvo em arquivos físicos no computador e que diversos caminhos de programação podem ser tomados para atingir o mesmo resultado pretendido, no *Live Coding* o resultado é obtido conforme o artista vai inserindo as linhas de código e trocando

---

<sup>40</sup> **Nota do autor:** Disponível para download no site do projeto.

<sup>41</sup> **Nota do autor:** Computadores robustos com configurações acima da utilizada em uso doméstico que servem para várias funções, a depender de sua configuração, entre elas: armazenagem de dados, processamento de funções gráficas, geração de criptomoedas, etc.

suas variáveis e no *Code Poetry* o código em si é uma obra de arte literária que pode ou não ser executada, mas que quando é executada cria um outro nível de dificuldade, uma vez que tanto o corpo do código quanto seu resultado são dois momentos de uma obra.

Inicialmente vamos analisar o Live Coding e para isso tomemos como exemplo o sistema denominado Hydra<sup>42</sup> que é um sintetizador de vídeo open-source em live coding. Seu sistema é baseado na linguagem de programação JavaScript e utiliza o WebGL como software de exibição gráfica e sonora, e foi elaborado por Olivia Jack cuja preocupação tem sido a elaboração de uma ferramenta criativa que possibilite que iniciantes tenham uma curva de aprendizado rápida, produzindo obras de qualidade com pouco tempo de estudo, mas que permita que usuários avançados contribuam com obras realmente desafiadoras do processo. Tivemos a oportunidade de conhecê-la pessoalmente em Karlsruhe no início de 2022 quando realizamos uma visita técnica ao *Zentrum Für Kunst und Medien – ZKM*, em que ela estava participando de oficina exatamente com o intuito de incentivar ingressantes e experientes artistas a produzirem diferentes obras com o Hydra.

Segundo as instruções presentes no site do sistema, o Hydra pode ser utilizado para:

- Para misturar e adicionar efeitos para o feed de vídeo, o compartilhamento de telas, os live streams e vídeos.
- Para criar imagens generativas e áudio reativas, e compartilhar com outras pessoas
- Em combinação com outras bibliotecas javascripts como P5.js, Tone.js, THREE.js ou gibber.
- Para adicionar efeitos de vídeo interativo a um website.
- Para experimentar com e aprender sobre feedback de vídeo, fractais e operações com pixel.
- Para fazer stream de vídeo entre navegadores web e live-jam com outras pessoas online (JACK, s.d.)

Poucos comandos são necessários para um resultado visual, na realidade, basta uma linha de código para obtermos um output visual, como podemos observar na figura abaixo (Figura 23).

A linha de código utilizada foi **`osc(20, 0.1, 0.8).out()`**, um oscilador cujos parâmetros definem a direção de ondulação, as cores e a frequência. Caso qualquer destas variáveis seja alterada o resultado é quase que imediatamente alterado (no caso do Hydra é necessário o comando Ctrl+Enter para recompilar a obra). Na Figura 24 podemos ver a alteração causada quando o primeiro parâmetro (20) é alterado para (90).

---

<sup>42</sup> Hydra: Live coding vídeo synth. JACK, Olivia. Disponível em: << <https://hydra.ojack.xyz/>>>. Acesso em: 10/05/2022.



**Figura 23:** Still da saída visual do Hydra após a inserção de apenas uma única linha de código. Fonte: Site do Hydra. Código utilizado como exemplo na documentação.

As variações seguem processos criativos de experimentações empíricas sendo que o artista pode realizar dois tipos de exibição: 1. Exibição “ao vivo” em que seguindo determinada performance o(a) artista acrescenta linhas de código, retira-as, modifica variáveis ou aplica códigos pré-elaborados. A Poética do Artista neste ponto se dá pela interpretação de determinada temática ou para explorar alguma funcionalidade. Neste modelo de exibição, dificilmente o(a) artista promove a documentação de suas peças por ser uma variação de performance. 2. A exibição de uma obra “finalizada” em que o código executa em *loop* obedecendo os padrões definidos como ideais pela Poética do Artista.

O artista e pesquisador, Alexandre Rangel, em sua tese, ao apresentar a forma como as diversas linguagens de programação interferem em seu processo criativo, apresenta a sua experiência no uso do *Live Coding* que segundo ele “um dos aspectos mais importantes da criação em live coding é a improvisação” (RANGEL, 2019, p. 35). Este aspecto de improvisação também cria dificuldades para a conservação, já que linguagens e frameworks que trabalham com Live Coding também são experimentais.

Rangel ainda alerta que diante da curta existência do *Live Coding* como experiência estética “as linguagens que forma inventadas e funcionavam em sistemas de 5 anos atrás podem já não funcionar tão facilmente” (op. cit.).



**Figura 24:** Still da saída visual do Hydra após a modificação dos parâmetros da linha de código. Fonte: Site do Hydra. Código utilizado como exemplo na documentação.

As obras denominadas de Code Poetry são aquelas cujo código é construído como um texto poético e elas podem ou não serem compiladas e executadas gerando um resultado computacional. Nesta classificação também tomamos o cuidado de inserir obras computacionais cujo artista-desenvolvedor procurou controlar os elementos do código como variáveis e funções para que estas também tivessem significação quando analisadas junto à intenção estética da obra como um todo. Neste último caso, difere-se do que comumente é listado como *Code Poetry*, pois raramente o *iterator* irá ter acesso ao código, conhecendo esse simbolismo apresentado.

O projeto “**.code –poetry**” de Daniel Holden e Chris Kerr é um livro impresso, e-book e uma versão online auto atualizável de doze poemas-códigos cada um escrito em diferentes códigos de programação. As obras variam sua poética em três estágios: Na parte semântica do código, apresentando resultados poéticos como texto; na configuração física-visual do código, lembrando as obras da poesia concreta brasileira e no resultado da execução do código. Na obra **Flocking**, o poema conta sobre um floco de neve que caiu durante a cerimônia de abertura das Olimpíadas de Seul. Nesta obra, a história é contada em comentários aplicados no início do código (parte superior direita da Figura 26), o mesmo formato é visto em figuras de código ASCII que dançam pelo lado esquerdo da tela (Figura 25), enquanto o código propriamente dito é apresentado nas cores descritas na parte comentada.

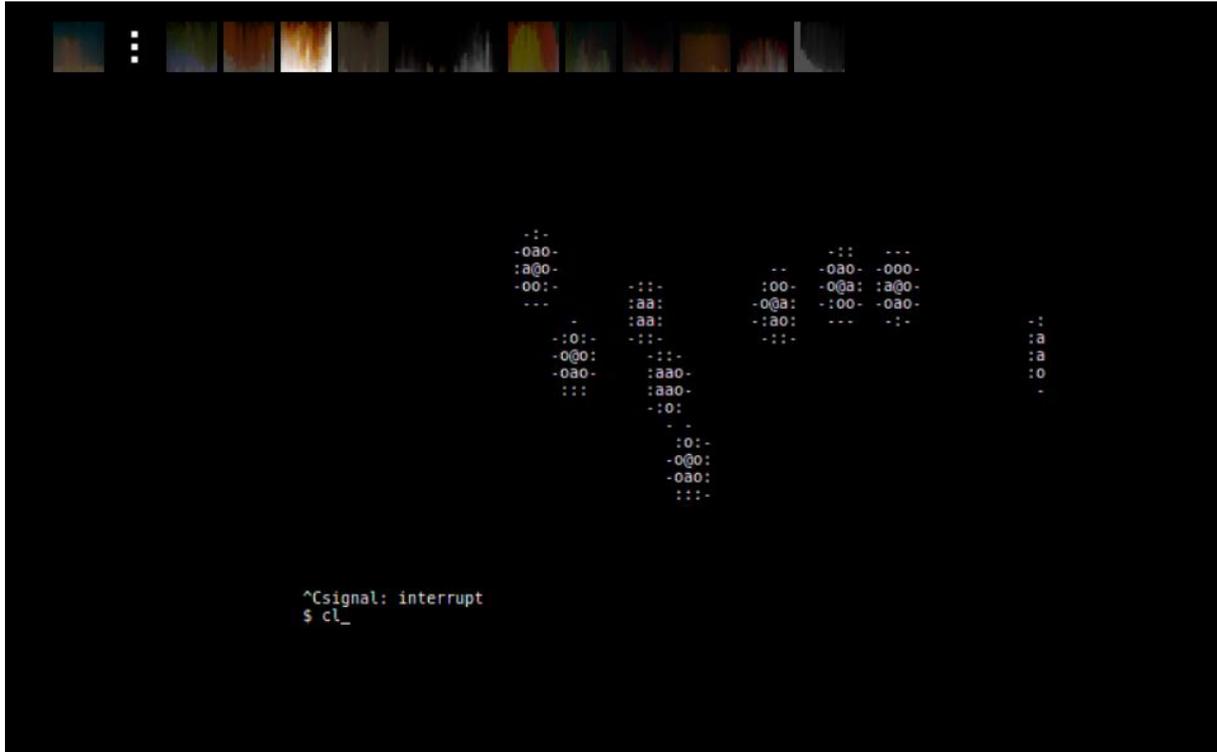


Figura 25: Still da obra Flocking (lado esquerdo) no site code-poetry.com. HOLDEN & KERR (s.d.).

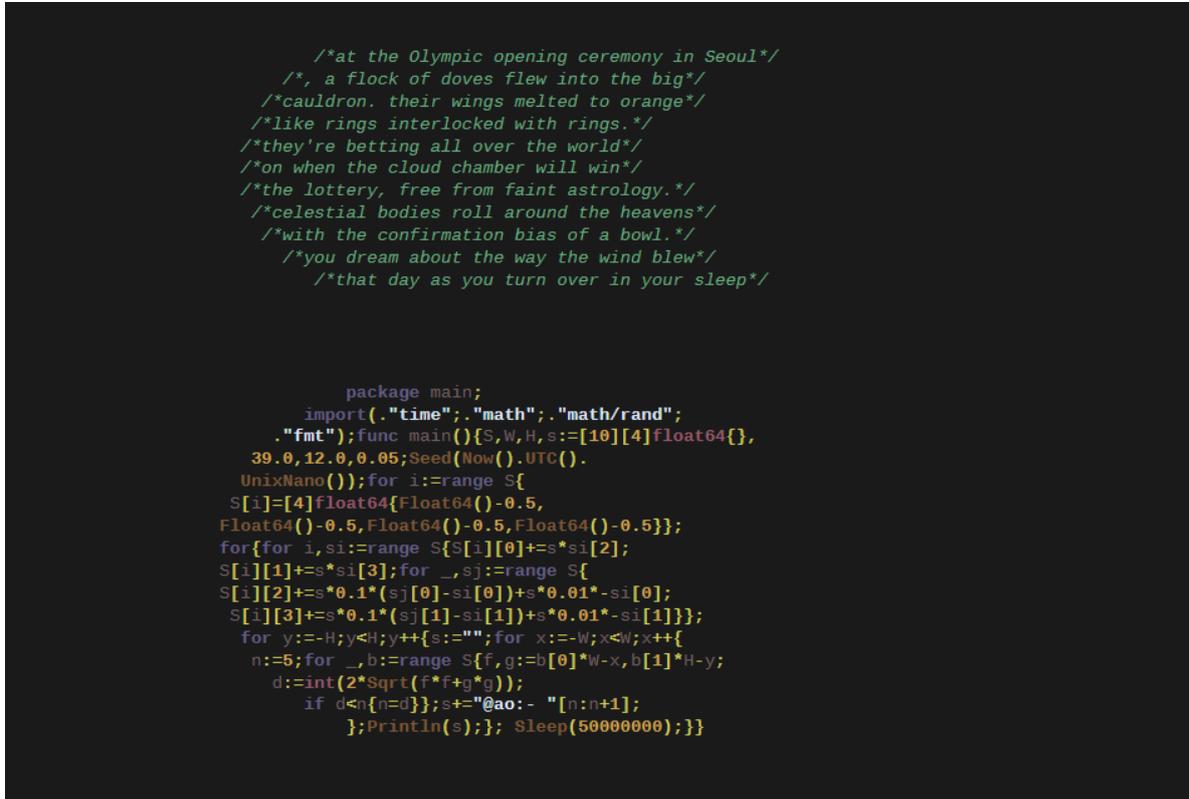


Figura 26: Still da obra Flocking (lado direito) no site code-poetry.com. HOLDEN & KERR (s.d.).

Além do *Live Coding* e do *Code Poetry*, uma situação pode ser colocada como exceção já que, fora o(a)s artista(s) e as pessoas responsáveis pela restauração ou recriação de uma obra computacional, nenhum *iterator* irá vislumbrar um código-fonte de uma obra, é a da obra em que o(a)s artista(s) elaboraram segundo uma semântica especial em que nomes de funções e variáveis são extremamente específicas. Este tipo de trabalho único deve ser documentado e preservado mesmo que a linguagem de programação seja substituída – o que hoje seria cada vez menos complicado já que as linguagens estão se apresentando cada vez em um nível mais próximo da linguagem humana.

Após apresentar diversos de seus trabalhos executados nos últimos vinte anos, Alexandre Rangel, retoma na sua tese a questão da criatividade desta vez apresentada na própria criação da linguagem de programação (RANGEL, 2019, p. 37). Para tanto apresenta alguns exemplos de linguagem que não apresentam uma semântica e uma sintaxe não tradicional. As linguagens citadas foram: (1) Piet, que a composição visual à estilo de Piet Modrian não é o output de um programa, mas o próprio código; (2) Gertrud, cujo código é escrito com frases comuns escritas em qualquer língua e para cada frase é calculada a média do tamanho das palavras e então:

As palavras que não se encaixam na média de tamanho de cada frase são usadas como índices relacionados às funções de processamento do programa (cálculos matemáticos, operações de fluxo do programa, criação de variáveis, etc). (RANGEL, 2019, p. 39)

Ao final, o código apresenta-se em uma forma que lembra um texto em prosa da escritora Gertrud Stein. (3) a linguagem Cow cujo código é composto por onomatopéias do som de vaca como é comumente reproduzida em inglês (Moo). (4) Arnold cujas frases faladas por Arnold Schwarzenegger em seus filmes são utilizadas como linhas de código; (5) Shakespear, que utiliza as famosas frases das peças de William Shakespear; e a última das linguagens citadas (6) Pikachu, que utiliza os sons do personagem de desenho animado como parte do código. A estas linguagens Rangel denomina, segundo ele mesmo baseado na forma como a “comunidade de programação” denomina, “linguagens esotéricas”(RANGEL, 2019, p. 39).

Estas linguagens são exemplos de como o código pode ser tomado como objeto estético, em que há uma real aproximação com o conceito de Sack, citado no início do capítulo, sobre a aproximação entre código e as artes do texto. De toda a maneira, obras desenvolvidas em linguagens assim são extremamente difíceis de serem conservadas, já que a Poética do Artista e a própria obra estão hibridizadas com a linguagem.

Com isso retomamos a discussão levantada acima sobre em que “parte” da obra computacional é aquela que possui “aquilo” que a faz ser única. Para exemplificar nossa pergunta, segundo Morales (2019), no final de 2012, o MoMA anunciou a compra do código de 14 jogos de fliperamas clássicos. Adquiriram o hardware e o software dos jogos mesmo obsoletos, mas a ação

de comprar o código, não foi apenas as linhas escritas, mas o direito de fazer dele o que quiser (como se uma instituição comprasse o design de um veículo e a partir dali pudesse mudá-lo a seu bel prazer. Assim, “o Restaurador pode atuar sobre a informação, já não como código concreto, mas como algoritmo, partitura e conservar a imagem; pode aplicar através da recriação o paradigma da conservação evolutiva” (MORALES, 2019, p. 61).

A mensagem carregada pelo código, e este sob a mediação do suporte, é o fator que realmente importa para a fruição da obra, já que os materiais não são mais controláveis (a tinta azul de determinada tonalidade continuará para sempre a possuir uma tonalidade muito próxima, mas um monitor de computador atual não tem a mesma definição de um monitor de cinco anos passados) e mesmo que a demanda do tempo nos limites em nossa criação, a intenção do artista é condição *sine qua non* para a conservação das obras computacionais.

Neste capítulo damos uma contribuição para o pensamento ontológico da arte computacional, questionando as classificações dadas mediante apenas ao suporte, descrevendo a essência da arte computacional não no aparelho computador, mas nas ideias que levam a escrita do código e na poética intentada pelo(a)(s) artista(s).

## **Parte 2: Como reter? Como guardar?**

Esta parte traz os capítulos “Experiências em Conservação de Obras em Mídias Instáveis” e “Protocolo de Conservação de Obra de Arte Computacional” que procuram responder quais tem sido as estratégias utilizadas até então na conservação destas obras; e apresentar o Protocolo de Conservação Estética de Artes Computacionais, principal objetivo desta tese.

## 4. Quarto Capítulo: Experiências em Conservação de Obras em Mídias Instáveis

Este capítulo tem por objetivo apresentar a “Teoria da Evolução Conservativa” e as estratégias e instrumentos envolvidos hoje na conservação das Artes Computacionais em que nosso protocolo se junta ao esforço para a criação de metodologias que ajudem na conservação e restauros das obras desta expressão artística.

### 4.1. Teoria da Evolução Conservativa

Morales (2019, p. 21) conceitua a Conservação Evolutiva como: “um paradigma da Restauração baseado na permanência através da modificação”, o que parece ser o caminho contrário das metodologias de conservação em artes tradicionais (Cf. BRANDI, 2019) que buscam retornar a obra a seu estado original<sup>43</sup>.

Continuando, o professor Lino Morales afirma que a Restauração da:

[...] arte ‘contemporânea’ deve superar, sem negá-la, a da arte ‘tradicional’ e deve resolver as crises e contradições que este *corpus* impõe. A Restauração da arte das ‘novas mídias’ deve superar, sem negar, a Restauração da arte ‘contemporânea’ (MORALES, 2019, p. 25)

Esta afirmação vai ao encontro do pensamento flusseriano que entende que ainda não haviam categorias adequadas para definir os aparelhos pós-industriais (FLUSSER, 2011, p. 40–41), do final da década de 1980 (quando o livro *Filosofia da Caixa Preta* foi escrito). Entretanto, muito se avançou na procura por descrever adequadamente as categorias pós-industriais, e o argumento de Morales acima apresentado é um destes esforços.

Já foi abordada a questão da transitoriedade das tecnologias da computação e da comunicação, em que sua evolução se dá pela obsolescência das versões anteriores, sendo assim, inevitavelmente as obras criadas com qualquer tecnologia computacional ou

---

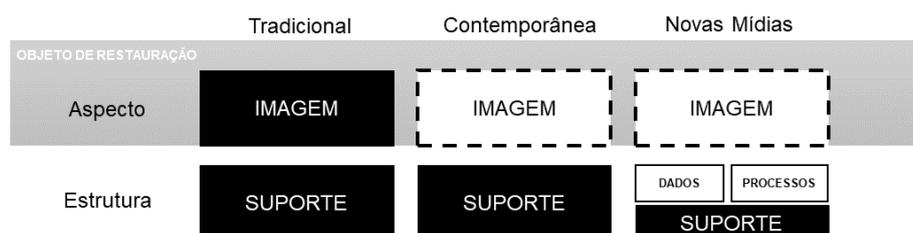
<sup>43</sup> **Nota do autor:** Cabe aqui uma pequena ressalva de que o estado dito “original” de uma obra é parte de um estudo histórico-estético, assim como esclarece o próprio Cesare Brandi (2019). Uma obra pode ter recebido alterações que podem ser marcos históricos importantes (ninguém pensaria em resgatar as pinturas anteriores feitas abaixo da Gioconda), mas que tem a obrigação de ser descartadas quando inoportunas (a folha de parreira colocada por cima das partes pudicas de Davi no Vaticano).

eletroeletrônica em algum momento terão peças que irão se quebrar e não se acharão mais a mesma peça, do mesmo modelo, para repor.

A preocupação que se passa a enfrentar é a de estar sendo alterada a obra a ponto de criar uma cópia ilegal, uma falsificação (MORALES, 2019, p. 30), já que “em um processo de Restauração não se produz uma cópia mas sim uma alteração do original” (*op. cit.*).

Para não ocorrer em problemas deste tipo, a “Teoria da Conservação Evolutiva” “propõem, como *prova de autenticidade*, um sistema de notação, um sistema simbólico de representação da informação [...] que garantisse as diferenças discerníveis do objeto de Restauração” (MORALES, 2019, p. 49).

Ao longo do livro, professor Lino procura descrever as estratégias de conservação de obras de mídias instáveis, mas o faz adaptando a ótica aplicada da teoria da restauração brandiana em uma morfologia própria.



**Figura 27:** Diagrama das Morfologias Comparativas entre obras. Fonte: Morales, Lino. (2019), p. 56

As caixas pretas simbolizam a “matéria” a qual as teorias tradicionais da Restauração tem como objeto principal de atenção (MORALES, 2019, p. 56). Apesar da Imagem ser a consequência final da obra de arte e por isso ser ela a atenção das novas teorias de conservação artística (incluindo a nossa aqui descrita), o Suporte como objeto material é também mídia e assim como nos apontou McLuhan é parte da mensagem (MCLUHAN, 1974).

Mais à frente Morales explica sobre a Arte das mídias instáveis e afirma que o objeto se divide e a matéria não constitui estrutura e aspecto, mas apenas uma parte da estrutura: o suporte e apenas ocasionalmente o aspecto (MORALES, 2019, p. 61).

Esta discussão levantada por Morales é importante quando verificada as teorias de conservação e restauração comumente aplicadas às obras de arte contemporâneas. Elas têm por objetivo preservar o suporte para que a imagem não seja alterada, todavia, no caso

das obras de arte computacional o fator da obsolescência atinge o suporte. Assim mesmo, preservando o suporte, o processo e os dados, a imagem pode não ser preservada: “A percepção de uma obra em um monitor de determinada tecnologia, marca, tempo de fabricação, horas de exposição, condições ambientais etc. é única” (MORALES, 2019, p. 62).

Sendo assim, Morales concluí que: “Uma das questões mais ‘comprometidas’ é precisamente adquirir a consciência de que em muitas obras de arte digital, a imagem não é conservável” (op. cit). Ainda, Morales afirma que as obras de arte das mídias instáveis possuem natureza dupla: objeto-símbolo e objeto-sistema.

Como objeto-símbolo, a mensagem, deve ser preservada a todo custo, já que uma vez atacado seu aspecto simbólico a identidade da obra se esvai. Já o objeto-sistema é a funcionalidade, ele interfere no aspecto simbólico, mas sua manutenção é essencial para a continuidade do aspecto sensorial simbólico, que na realidade é a parte em que os *interatores* têm acesso.

Neste ponto é relevante lembrar que na natureza da obra computacional o *hardware* é suporte, salvo quando inserido também no ambiente estético que passa a assumir mais de um papel. Além disso, o código e os dados possuem uma característica de não serem únicos, já que diferentes códigos e diferentes dados podem dar origem ao mesmo resultado estético.

## 4.2. Estratégias utilizadas: análise de casos.

Diante do desafio de lidar com diversas informações para a conservação de cada obra de arte das mídias instáveis, o uso da Internet e de mecanismos de armazenamento em nuvem se destaca como principal estratégia. Contudo, por mais que os sistemas baseados na web tenham uma obsolescência reduzida em questão de adaptabilidade com os equipamentos<sup>44</sup>, a manutenção de um site ou serviço online requer uma constante atualização de sistema, custos operacionais e resolução de problemas tipo help desk. A inabilidade ou impossibilidade (financeira ou secular) de se manter estes “sites-sistemas” é a principal razão de muitas ferramentas deixarem de serem utilizadas.

---

<sup>44</sup> **Nota dos autores:** O primeiro protocolo http é de 1991 e basicamente transmitia apenas texto. Os protocolos http que conhecemos com as funcionalidades abordadas neste argumento são o 2.0 (2015) e o 3.0 (2018) que logicamente mudaram muito da interface, da usabilidade e da segurança de se navegar no ambiente web, mas que para os principais sites e instrumentos on-line não representou uma quebra no uso como num equipamento físico.

Em sua tese de doutorado “Nuevas estrategias para la conservación de colecciones de arte con elementos tecnológicos: propuestas metodológicas de humanidades digitales”, defendida em 2015, a museóloga e curadora do Museu Reina Sofia (Madri), Arianne Vanrell Velloso, apresentou algumas ferramentas utilizadas para a troca de informação e criação de repositório, porém algumas destas ferramentas não podem mais serem acessadas. Tal fato promove uma reflexão sobre a sustentabilidade destas ferramentas para promover a conservação quando elas mesmas não conseguem realizar sua conservação.<sup>45</sup>

Para contornar a problemática do uso de ferramentas terceirizadas algumas instituições (universidades, centros de pesquisa e alguns museus e galerias) que possuem maiores recursos podem investir em equipamentos estilo servidor de dados, em que os componentes (discos rígidos e SSD's) podem ser substituídos ao apresentarem falhas e o sistema possui um sistema de armazenagem em redundância em que mais de um componente armazena os mesmos dados (denomina-se espelhamento).

Este servidor pode ser utilizado mimetizando o uso da Internet (Intranet) que pode ou não liberar acesso externo aos prédios da instituição (esse acesso é monitorado pelos códigos TCP/IP, que é único de cada equipamento, e filtrado por algum protocolo como o endereço MAC ou um servidor proxy). A experiência acadêmica brasileira com esse sistema assemelha-se ao acesso ao sistema CAFE do periódico CAPES que permite acesso ao banco de dados de uma instituição pela autorização do seu equipamento.

Neste ponto da tese é desnecessário informar que esta estratégia possui um custo de implementação extremamente elevado, e como depende de hardware específico, deve-se observar a questão da obsolescência e da quebra de equipamentos. Quando utilizado em nuvem, os arquivos são distribuídos por inúmeros centros de dados como esse no mundo inteiro, ou seja, mesmo que um determinado equipamento tenha que ser descontinuado, outro imediatamente assume a posição e os dados nunca ficam em um único computador. Quando o sistema permanece dentro de um prédio, até mesmo a preocupação com desastres naturais e acidentes devem ser levados em conta.

Todavia, como já informado no início da tese e constantemente lembrado ao longo do texto, nossa intenção é a conservação da Instrução Poética. Estas estratégias citadas acima levam em consideração apenas as características físicas das obras. Nossa análise ontológica permite considerar que a ideia, a motivação, a intenção do(a)s artista(s) é que é o ponto nevrálgico a ser conservado. Diante disso, cabe aos artistas estarem conscientes sobre as metodologias

---

<sup>45</sup> **Nota dos autores:** Um dos exemplos mais claros é o site Yutzu que é declaradamente o ambiente de conservação utilizada pela autora para apresentar seus dados de forma gratuita e pública. Tentativas de acesso ao site, seja na sua raiz seja na área da autora, mostraram-se infundados, apresentando páginas que aparentam ser erros, mas que não possuem qualquer informação sobre o erro (página não localizada e erro de banco de dados, seriam dois dos erros mais comuns e esperados).

comumente utilizadas para a conservação material das obras e as estratégias a serem adicionadas a cada uma destas para a manutenção da Instrução Poética.

As estratégias aqui apresentadas estão listadas no livro “Teoria da Evolução Conservativa” do Prof. Dr. Lino García Morales, todavia ele aponta que cada metodologia foi proposta pelo projeto “Variable Media Network” (DEPOCAS *et al.*, 2003).

O foco do Variable Media é integrar a análise de materiais com a definição de uma obra de arte independentemente de sua mídia, permitindo o trabalho ser traduzido uma vez que sua mídia atual esteja obsoleta. Identificando o comportamento da obra (contido, instalado, performado, reproduzido etc.) e estratégias (armazenamento, emulação, migração e reinterpretação), artistas, conservadores e curadores podem avançar na preservação da arte das novas mídias. (GUGGENHEIM, n.d.).

Entretanto, durante o período de pesquisa bibliográfica desta tese, pudemos encontrar artigos de pesquisadores interessados em cada uma das estratégias, como: Patrícia Falcão do Museu TATE (Inglaterra) que assina o artigo “Introduction to an emulation-based preservation strategy for software-based artworks”, defendendo a emulação como uma importante estratégia, que inclusive é aplicada no TATE. Diego Mellado que no artigo “A System Engineer’s Perspective for the Re-Creation of Media Art *n-Cha(n)-t* by David Rokeby”, estabelece os princípios direcionadores da Recriação segundo conceitos trazidos da engenharia de sistemas.

Além disso, nas análises de caso descritas no livro “Digital Art Conservation” (SEREXHE, 2013), podemos encontrar exemplos de aplicação das estratégias. Inclusive pudemos presenciar grande parte do acervo técnico da ZKM para a estratégia de substituição (cf. item 4.3.1).

Devemos agora realizar uma descrição das estratégias listadas por Morales e pelo VMN, apontando a forma como cada uma pode interferir na preservação da instrução poética.

#### **4.2.1. Substituição**

A substituição é a forma mais básica de conservar uma obra de arte contemporânea (MORALES, 2019, p. 85), não apenas as obras de mídias instáveis. Consiste em criar um acervo técnico com o maior número de peças e componentes capazes de estender a vida útil de um equipamento suporte ou de uma obra dependente de hardware.



**Figura 28:** Um dos diversos espaços de armazenagem para materiais de reposição para as obras do acervo permanente da ZKM. Fonte: Acervo próprio, 31 janeiro 2022.

Esta estratégia exige recursos não apenas para a compra, mas também para armazenamento destas peças e componentes. Em entrevista concedida para nós, durante uma visita técnica realizada no ZKM (cf. item 4.3), Morgane Stricot descreveu a estratégia do centro de pesquisa, que se utiliza primeiramente da técnica de substituição, mas que algumas obras dependem de equipamentos muito dispendiosos, assim ela descreve que esperam até o equipamento ficar obsoleto e compram o máximo possível no Ebay. No período da entrevista (final de Janeiro de 2022) estavam realizando a compra de todos os I pads que estavam disponíveis no Ebay<sup>46</sup>.

Conforme aponta Morales (2019, p. 86), a estratégia de substituição não representa nenhum avanço no combate à obsolescência, mas respeita a integridade do objeto. Acrescentamos que por não alterar nada na questão simbólica, não há nenhum impacto na Instrução Poética.

#### **4.2.2. Migração**

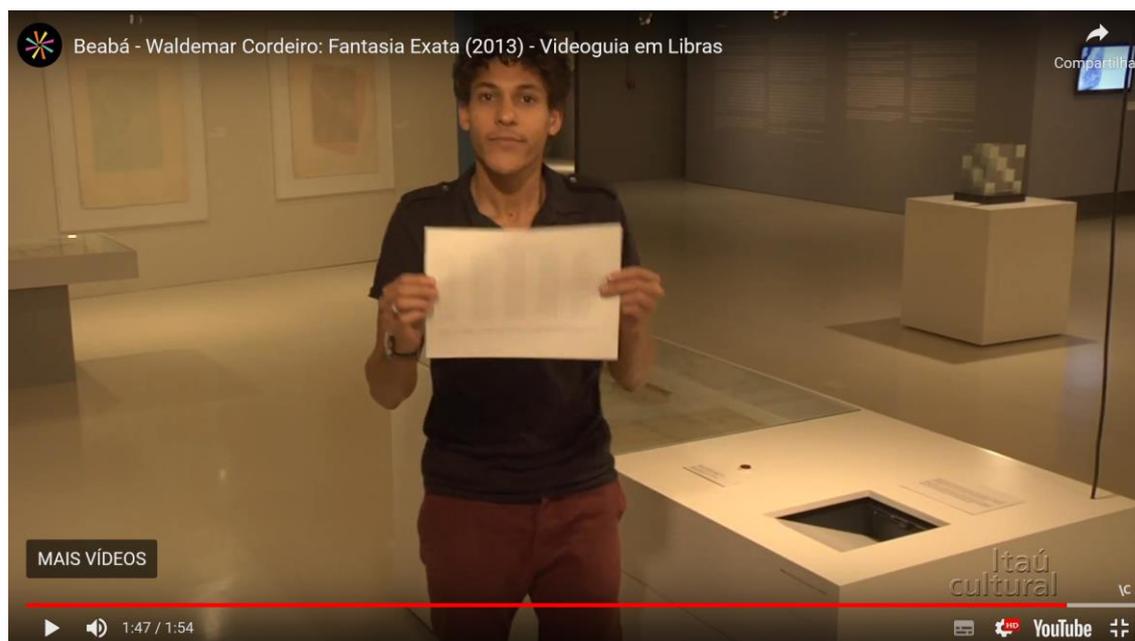
Esta estratégia consiste na atualização do suporte de uma obra de arte, uma transferência de mídias. Por exemplo: obras armazenadas em VHS passaram para CD-ROM e depois foram transferidas para DVD. Nesta estratégia, a perda de informação acontece quando a estética da obra está também na deterioração do suporte. Podemos ilustrar esse fenômeno quando em uma instalação de projeção de cinema, parte da expectativa do(a) artista seja a perda das cores com a

<sup>46</sup> **Nota do autor:** Ainda, Stricot menciona que recebem diversas doações de equipamentos de empresas, escolas e universidades que estão substituindo seus equipamentos.

exposição constante da película fílmica à luz; ou a passagem do material de uma fita VHS com a textura analógica da imagem e do som para um arquivo digital com compactação de áudio e vídeo.

Para obras computacionais essa estratégia possui um impacto reduzido, uma vez que arquivos digitais armazenados em disquetes ou em CD-ROMs possuem a mesma natureza (talvez mude o tamanho dos arquivos). Obras mais antigas podem sofrer algum impacto quando se trata de antigos mecanismos de armazenamento (obras com cartões perfurados) ou obras com rolos de fita.

Na recriação de BEABÁ para a exposição Fantasia Exata (2013) (PRADO; ARANTES, 2016), a obra, que passou por uma total recompilação em outra linguagem de programação, migrou do computador original, um IBM 360, para um desktop comum. A obra original não possuía qualquer mecanismo de input que não a leitora de cartão perfurado e a única saída era uma impressora matricial que imprimia uma série de palavras em seis colunas. Na exposição de 2013, o BEABÁ continuou sendo exposto em um quiosque<sup>47</sup> onde o *interator* apertava um botão, substituindo a leitura do cartão perfurado pela execução de um programa de computador já alocado na memória; na saída uma impressora laser substitui a impressora matricial.



**Figura 29:** Still do videoguia em libras, disponibilizado pelo Itaú Cultural em seu canal do YouTube, sobre a obra BEABÁ. Fonte: Canal Itaú Cultural YouTube. Disponível em: <<<https://www.youtube.com/watch?v=pQ90gVX7cqY&t=107s>>>. Acesso em: 21 maio 2022.

### 4.2.3. Emulação, Virtualização, Simulação e Containerização

<sup>47</sup> Cf. FIGURA 29. Pode-se ver à direita o quiosque com a obra.

A estratégia de emulação visa preservar o código tal como ele foi escrito (MORALES, 2019, p. 87), para tanto cria-se uma “máquina virtual” que emula um ambiente de software de um computador mais antigo para um equipamento com configuração mais nova (Rothenberg, 1998 apud MORALES, 2019, p. 87).

A Emulação absorve um dos problemas de conservação: a do hardware. Além disso, esta estratégia também permite que o museu e o(a)s artista(s) utilizem de mais de uma “instancialização”<sup>48</sup>, já que o hardware se torna relativo. Na visão artística, principalmente nos últimos anos que fomos vitimados pela pandemia do COVID-19, diversas exposições e atividades on-line foram realizadas<sup>49</sup>, dispensando o uso de equipamentos.

Todavia, isso não significa que estes equipamentos não receberão tratamento conservativo, muitas obras possuem uma íntima relação estética com seus hardwares, como foi apontado por nós no Terceiro Capítulo. Além disso, vale citar que os trabalhos adquiridos por museus e galerias normalmente são entregues já com computadores com todo o ambiente de software necessário para executar o programa da obra e todos os acessórios utilizados para a exibição (FALCÃO, Patrícia in GRAU; HOTH; WANDL-VOGT, 2019, p. 276).

Outra informação importante é que para a emulação, assim como para qualquer outra estratégia, é necessária uma documentação muito bem descrita, caso contrário ao virtualizar uma obra pode haver problemas de adequação de drivers ou mesmo plug-in que podem não ser mais capazes de serem resolvidos por não existir mais a possibilidade de retomar o software.

Esta estratégia também não oferece maiores riscos à Poética do Artista desde que observada a completa descrição e documentação da obra.

#### **4.2.4. Reinterpretação**

Esta estratégia é segundo o professor Lino: “[...] a estratégia de Restauração mais potente, assim como a de maior risco e consiste em reinterpretar o objeto-símbolo cada vez que o recria em um novo objeto-sistema” (MORALES, 2019, p. 88). A reinterpretação muitas vezes requer reescrever o código para adaptar a ideia a um novo hardware.

A reinterpretação tem um risco imenso de se tornar uma cópia da obra, uma falsificação, principalmente se não houver nenhuma instrução do(a)s artista(s) quanto à tomada de decisão acerca da estratégia. Morales adverte quanto a isso:

A Teoria Contemporânea da Restauração “defende que o que caracteriza a esses objetos (admite que a Restauração se defina em função de seus objetivos) são

---

<sup>48</sup> **Nota do autor:** Conforme pudemos observar na entrevista com Morgane Stricot na ZKM e no artigo de Patrícia Falcão, o termo “instância” é utilizado para indicar quando mais de uma exibição da mesma obra está sendo executada. Ambas possuem os mesmos sistemas, componentes, materiais e disposição. Porém o termo instância para a computação, pode significar que mais de um objeto representante de uma classe possa estar presente na área de trabalho ou renderização. Por isso, ao invés do termo instanciamento ou instanciação (que guardaremos para utilizar quando nos referirmos diretamente à prática computacional), iremos utilizar instancialização para objetivo de referência à prática museológica.

<sup>49</sup> **Nota do autor:** No ano de 2020, tivemos uma obra (ORI 0.0.6) exposta na exposição “Prospecções Afetivas: em tempos de pandemia” como parte do Encontro Internacional de Arte Computacional (#.ART) daquele ano.

rasgos do tipo subjetivo, estabelecidos pelas pessoas, e não inerentes aos próprios objetos”. A questão é quanta influência tem o respeito a identidade da imagem na transmissão do poder simbólico ou comunicativo. Qual é a margem para que um *resíduo* não se converta em *índice*, nem a obra em um *ícone* de si mesma, senão que permaneça sendo um símbolo do que representa?(MORALES, 2019, p. 89).

Como se trata de uma estratégia lançada quando não há outra opção, esta tem sido utilizada nos últimos anos para tentar recuperar as obras computacionais dos primeiros períodos (das décadas de 1950 a 1970 mais ou menos), muitas destas obras não apresentam documentação e/ou o(a)s artistas (quando vivo) não tem mais acesso ao código-fonte. Assim, podemos citar uma experiência narrada pela conservadora de media-art da ZKM de uma obra que possuía algumas projeções de imagem que conseguiram ser capturadas sem muita perda e foi recriado um programa que se comportasse de igual maneira.

Todavia, mesmo após este período algumas obras passaram por este processo como forma de atualizar a experiência, assim como as qualidades de áudio e vídeo. Citemos como exemplo a obra *Karlsruhe Moviemap* de Michael Naimark, cuja primeira versão de 1991 passou por uma reinterpretação em 2009 pelo ZKM. Interessante notar que antes de decidir pela reinterpretação, foi feita uma migração em 2006, mudando da configuração de hardware original e refazendo o software (inicialmente totalmente autoral) para uma composição realizada em Max/MSP.

A obra se trata de uma instalação em que o *iterator* pode controlar um trem de superfície pela cidade de Karlsruhe enquanto acompanha seu trajeto por um dispositivo à frente. Naimark utilizou o conhecimento adquirido ao ter trabalhado com Nicolas Negroponte no grupo de pesquisa *Architecture Machine Group* responsável pela obra *Aspen moviemap* (1978 – 1980) que havia sido financiada pela DARPA (órgão do governo norte-americano responsável pela defesa do território) para desenvolver seus trabalhos estéticos: “para a área da baía de São Francisco ele criou a simulação do voo de um helicóptero e produziu o moviemap do distrito de Madeleine em Paris para o serviço de transporte da cidade (RATP)” (SEREXHE, 2013, p. 424–425). Após, em 1989/1990, o artista foi contactado pelo ZKM através de seu antigo professor Otto Piene e Naimark criou então a *Karlsruhe Moviemap* para o evento *MultiMediale 2* de 1991, e em 2009 a ZKM encaminhou a sugestão de atualização da obra.

A equipe do ZKM tomou algumas medidas conservativas mesmo após decidir pela reinterpretação: a película de 16mm, que originalmente realizava a parte visual da obra, foi digitalizada, além disso foi conduzida uma entrevista com Naimark para determinar quais seriam as possíveis medidas conservativas a serem tomadas ao futuro e as diretrizes a serem observadas na reinterpretação de 2009.

Na nova versão, em nossa análise, a obra continua com a mesma instrução poética, facilitado pelo fato de o artista ter participado ativamente da reinterpretação. As modificações (captação de imagens duplas em câmeras digitais e o uso de óculos 3D polarizados para criar uma

perspectiva estereoscópica) criaram uma experiência mais densa e mais imersiva, mas de nenhuma forma distante da intenção do artista.

Contudo, retomamos a importância do alerta dado pelo professor Lino Morales sobre a utilização desta estratégia que deve ser utilizada quando já não é mais possível qualquer outra estratégia de conservação.

#### **4.2.5. (Re)Criação**

A estratégia da recriação é, segundo Diego Mellado, um “paradigma técnico [que] é baseado em diversos conceitos da Engenharia de Sistema como a ideia do sistema, a ideia da ‘caixa preta’ e os diagramas de fluxo” (MELLADO in GRAU; HOTH; WANDL-VOGT, 2019, p. 180).

Esta estratégia estaria no ápice da “Teoria da Evolução Conservativa” e Mellado cita o professor Lino ao colocar que esta estratégia prevê “to change so nothing changes” (MELLADO in GRAU; HOTH; WANDL-VOGT, 2019, p. 180) e, continua, é garantir que se obtenha a principal característica da obra de arte em níveis diferentes para que qualquer futura intervenção não altere o seu valor simbólico.

Na recriação nada (ou quase nada) da versão anterior da obra é utilizada, apenas a ideia da obra é preservada. Por isso, a recriação depende de uma documentação que garanta o refazimento de cada um dos passos anteriormente traçados. Este foi o caso do “Colóquio dos Móviles” cuja obra inicial havia sido completamente desmontada e para o projeto de reconstrução de 2018, tudo, desde as partes externas dos “machos” e “fêmeas” até os componentes eletrônicos tiveram de serem refeitos.

O impacto de cada estratégia na Poética do Artista (ou o aspecto simbólico da obra como adotado por Morales e Mellado) é uma preocupação constante a ser observada durante a aplicação. Contudo, por trata-se da preservação da ideia mediante qualquer outra coisa, a Recriação é a estratégia em que a documentação deve ser mais observada e mais minuciosa.

Todas as estratégias acima mencionadas são extremamente técnicas, e não deve ser preocupação do(a) artista executá-las, isso cabe aos museólogos e conservadores. Contudo, o(a) artista deve ter em consideração que suas obras podem vir a ser alvo destas estratégias e por isso deve possuir uma opinião a respeito das mesmas, assim como deve contribuir ao máximo com os detalhes técnicos da construção da mesma.

### **4.3. Bolsa Doutorado-Sanduíche na UdK – Berlin**

Como parte da pesquisa para a escrita e desenvolvimento desta tese, tivemos a oportunidade de realizar parte da mesma na Universität der Künste – Berlin (UdK) com

bolsa ofertada pelo Serviço Alemão de Intercâmbio Acadêmico (*Deutscher Akademischer Austauschdienst - DAAD*) sob orientação do Prof. Dr. Alberto de Campo<sup>50</sup>.

O projeto aprovado se concentraria nas estratégias de conservação das obras de arte computacionais dependentes de hardware. Dois dos acervos que seriam objetos da pesquisa seria o conjunto de obras do Prof. Joachim Sauter, tanto pessoais quanto as feitas pelo ART+COM, estúdio do qual era um dos donos; o outro acervo seria o do Centro de arte e mídia de Karlsruhe (*Zentrum für kunst und medien Karlsruhe – ZKM*), que é por si só o maior acervo do mundo de mídias instáveis<sup>51</sup>.

Por questões de agenda, não foi possível atender a observação do acervo relacionado ao professor Sauter, porém tivemos a oportunidade de realizar uma entrevista informal com o também professor da UdK e sócio de Sauter, Jüssi Ängeslevä. Apesar de informal, a entrevista com o Prof. Ängeslevä permitiu pensar sobre uma questão muito sensível: o Prof. Sauter era, nas próprias palavras de Jüssi, a força motriz da criação da ART+COM nos anos em que ele esteve à frente do estúdio, contudo não era o único responsável, então como seria possível lidar com as intenções do artista falecido enquanto os outros membros e autores das obras estão vivos.

Esta resposta só veio um mês depois durante a visita ao pesquisador e professor Lino García Morales quando ele enfatizou a entrevista com os autores como um dos importantes documentos que devem estar presentes para a conservação da obra.

Retornando aos acervos pesquisados, no final de Janeiro (30 e 31 de janeiro de 2022), o autor desta tese foi acompanhado do Prof. Dr. Alberto de Campo a uma visita técnica ao ZKM para realizar uma entrevista com a conservadora de arte e mídia Morgane Stricot. Este momento foi muito importante para a pesquisa e iremos apresentar os detalhes mais abaixo (cf. item 4.3.1).

Retornando da visita ao acervo da ZKM, embarcamos para Madri, Espanha, para visitarmos dois importantes teóricos base desta tese, a conservadora do Museu Reina Sofia, Arianne Vanrell Velloso, e o professor da Universidade Politécnica de Madri, Prof. Dr. Lino Morales. Também estava planejado realizar uma entrevista presencial com o autor

---

<sup>50</sup> A bolsa foi ofertada pelo PROGRAMA CONJUNTO DE BOLSAS DE DOUTORADO NA REPÚBLICA FEDERAL DA ALEMANHA CAPES/DAAD EDITAL Nº 23/2019. Devido às restrições causadas pela pandemia que assolou o mundo desde o ano de 2019, a ida para a Alemanha para permanência de 12 meses foi adiada por diversas vezes. Ao final, restou apenas realizar a permanência de 3 meses. Contudo, a pesquisa continuou de maneira virtual tendo sido estabelecido um laço colaborativo com a UdK por meio do trabalho do Prof. Dr. Alberto de Campo.

<sup>51</sup> **Nota do autor:** Em 2013, a ZKM possuía mais de quinhentas obras digitais, das quais cem são apenas obras interativas complexas controladas por computador (SEREXHE, 2013, p. 29).

Diego Mellado, todavia por razões de agenda não foi possível. Felizmente, o sr. Mellado dispôs-se a realizar a entrevista em Videochamada por computador.

Ainda, durante o período do sanduíche, também foi possível ter acesso ao acervo do Flusser Archiv na UdK graças a excelente recepção da Profa. Dra. Anita Jóri, que contribuiu com artigos para compreender melhor a visão flusseriana do conceito de programa.

Abaixo realizamos breves relatos sobre as experiências vivenciadas durante o período de dezembro de 2021 a março de 2022.

#### **4.3.1. Relato de Visita Técnica à ZKM**

Em janeiro de 2022, tivemos autorização para visitar o Centro de Arte e Mídia de Karlsruhe com objetivo de verificar como são realizadas as ações de conservação e restauração do centro diante de seu enorme acervo de obras de mídias instáveis.

O Centro de Arte e Mídia de Karlsruhe conhecido mundialmente pelas siglas de seu nome em alemão (ZKM), foi fundado oficialmente em 1997 de um projeto que já existia desde o final da década de 1980, da cidade de Karlsruhe de fundar uma instituição que unisse conceitos estéticos com tecnologias voltadas para pensar o futuro.

Segundo o site do centro, sua coleção é composta por 9 mil obras do século XX e XXI, contendo todos os tipos de expressão de artes visuais, sendo mais de quinhentos obras de arte digitais (SEREXHE, 2013, p. 29)<sup>52</sup> e destas, cem são “instalações digitais complexas, interativas e controladas por computador, um tipo de instalação que não é encontrado em muitas coleções” (op cit). Sendo assim, o acervo da ZKM representa uma grande importância não apenas em quantidade mas em questão da unicidade de muitas das obras adquiridas ao longo dos anos como “Legible City” de Jeffrey Shaw, “Internet Dream” de Nam June Paik, ou “Karlsruhe Moviemap” de Michael Naimark.

A visita de caráter técnico, foi executada em dois dias, sendo o primeiro de iniciativa dos participantes de visualizar as exposições e tomar notas livres sobre as obras e o segundo dia já em caráter oficial. Assim, iniciamos a visita ao ZKM pela exposição BioMedia<sup>53</sup> que estabelecia uma discussão sobre formas de vida e a virtualidade e onde

---

<sup>52</sup> **Nota do autor:** Este dado encontra-se no primeiro capítulo do livro Digital Art Conservation do projeto de mesmo nome executado na segunda década do ano 2000, terminando no ano de 2013. Desde então a instituição já adquiriu outras obras (como o “Colóquio dos Móveis”) e já ultrapassou este número.

<sup>53</sup> **Nota do autor:** A exposição BioMedia (BioMedien) teve início em 18 de dezembro de 2021 e vai até 28 de agosto de 2022. 68 artistas ou coletivos têm obras expostas que propõem uma reflexão sobre os aspectos da vida, especialmente na forma digital.

encontramos a obra de Gordon Pask, e fruto do projeto de recriação chefiado por Paul Pangaro, “Colóquio dos Móviles” que viria a ser objeto de nossa análise.

Neste momento pudemos presenciar alguns detalhes especiais sobre a recepção do visitante, a relação *interator* / obra – principalmente em um contexto de pandemia –, e a conservação das obras expostas, assim como a resposta à falha de algumas das obras ou a quebra de suas peças.

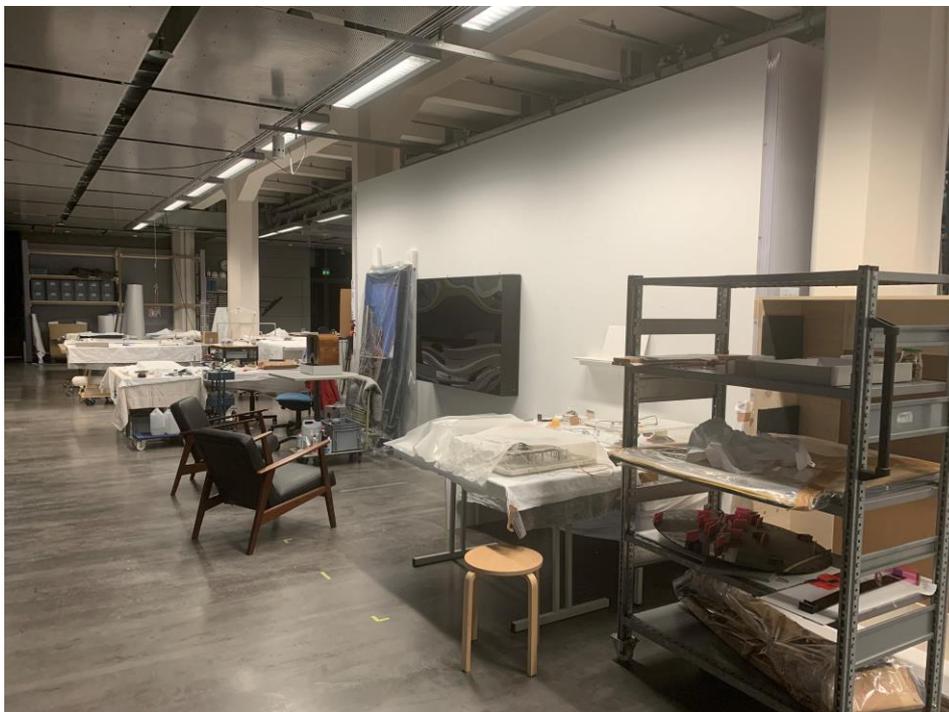
O envolvimento de tecnologias da comunicação e da informação como suporte ou como objeto estético sujeita as exposições a situações que aquelas realizadas apenas com obras de arte tradicional não sofreriam. A experiência com muitas das obras foi limitada pelo mau funcionamento de alguns dos componentes, o que diante da ativação de seis dias por semana por mais de oito horas de funcionamento se justifica que muitas partes estivessem se desgastando, afinal, como já vimos anteriormente, as obras de mídias instáveis não sofrem testes de esforço mecânico ou eletroeletrônico assim como produtos de consumo sofreriam.

Todavia, na grande maioria das vezes quando não havia possibilidade de visualizar a obra funcionando em sua plenitude, outra forma de documentação apresentava como ela “deveria” funcionar, assim, havia sempre um instrumento extra (um vídeo, um conjunto de fotos ou diagrama de funcionamento) para orientar o *interator*.

No dia posterior fomos recebidos por Morgane Stricot, curadora de arte e mídia da instituição, uma das responsáveis pela manutenção, conservação e restauração das obras do acervo e daquelas temporariamente sob os cuidados da ZKM, tais como as expostas nas exposições transitórias.

Inicialmente a visita se deu novamente pela exposição BioMedia, mas desta vez apenas estavam as equipes de manutenção da ZKM e os orientadores culturais que realizavam mais algumas instruções sobre as obras para melhor orientar o público visitante. Stricot foi extremamente solícita de nos apresentar os mecanismos técnicos que atendiam as exposições, como uma sala com um pequeno acervo técnico no mesmo pavilhão onde estava ocorrendo a exposição, que possibilitava no dia a dia uma rápida reposição de alguma das peças ou componentes de uma obra, amenizando qualquer impacto mais profundo.

Outros ambientes que podiam ser observados eram espaços organizados para trabalhos de restauração mais especializados que precisaria deslocar a obra para melhor visualização do problema pelo(a) técnico(a).



**Figura 30:** Ambiente no terceiro andar do prédio com espaços destinados à manutenção das obras que ocorriam no andar térreo. **Fonte:** Acervo próprio (2022).

Após a apresentação do ambiente, fomos recebido no escritório de Conservação de Mídia e Arte Digital que Stricot divide com seu colega Mattieu Vlaminck, nos concedendo duas horas de uma entrevista não estruturada, captada em áudio em que comenta sobre os protocolos de conservação e restauração de obras computacionais aplicadas na ZKM.

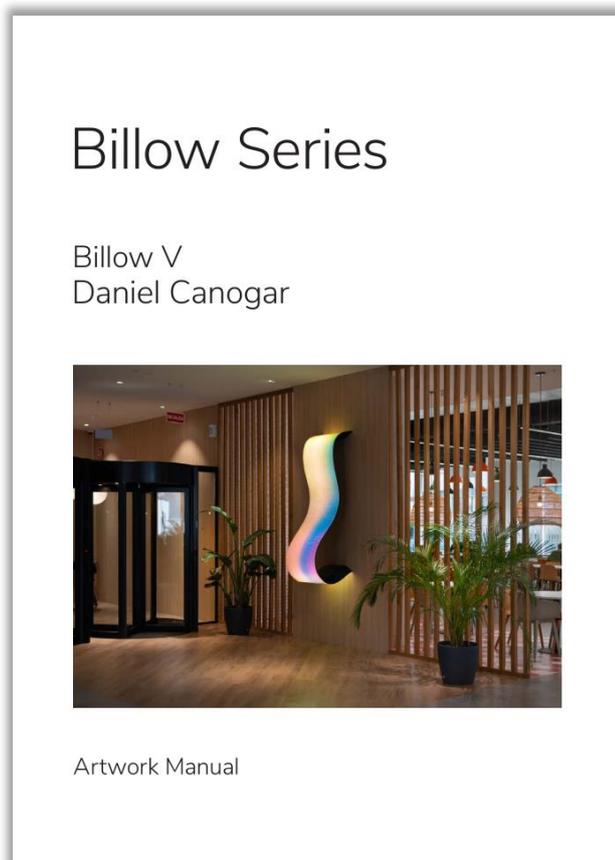
Desde sua entrada no ZKM em 2017 ela e as equipes com as quais trabalha, tem procurado criar um banco de dados sobre as obras cujo processo se inicia logo na aquisição. Com relação às obras adquiridas antes de 2017 ela depende da memória de funcionários mais antigos do instituto, fazendo o trabalho de documentação das mesmas através das montagens das obras durante suas requisições pelos curadores.

O Fluxo de Trabalho para Aquisição de Obras de Arte baseadas em Softwares<sup>54</sup>, tem seu começo com as atividades Pré-aquisição. Nas duas etapas pertencentes a este momento, são definidos os desafios de conservação e restauro das obras de aquisição

<sup>54</sup> **Nota do autor:** a página é de acesso liberado e está disponível em [https://werke.zkm.de/wiki/index.php/Acquisition\\_workflow\\_\(Software-based\\_artworks\)](https://werke.zkm.de/wiki/index.php/Acquisition_workflow_(Software-based_artworks)).

pretendidas, a documentação do artista é muito importante nesta etapa. Inclusive, os conselhos do artista Rafael Lozano-Hemmer publicados em setembro de 2015 no site GitHub com o nome de “Melhores práticas para conservação de mídia arte na perspectiva do(a) artista”<sup>55</sup>, são base para muitas das atividades desta etapa. Dentre alguns citados por Stricot no workflow estão:

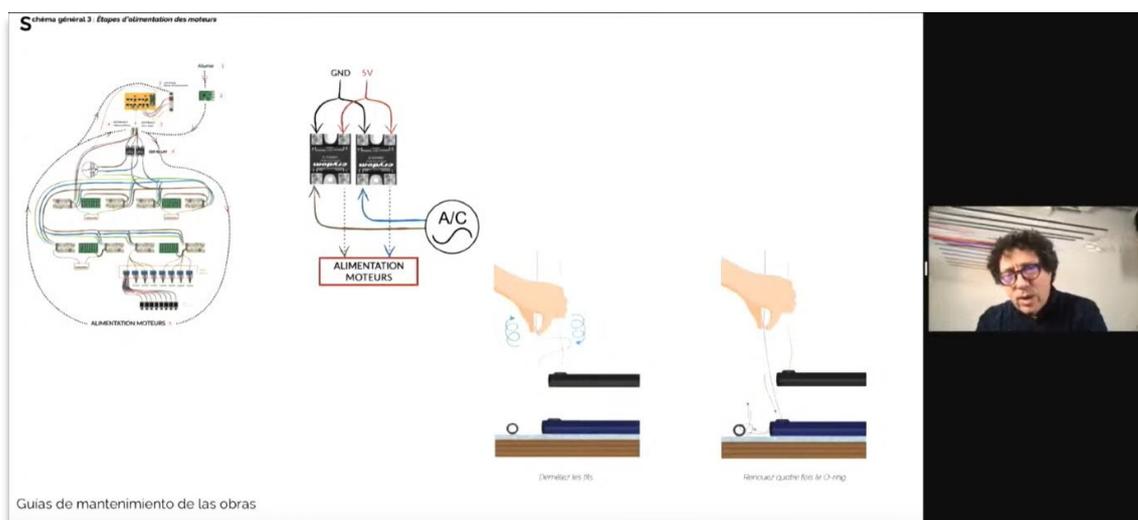
- Fazer um vídeo do projeto, de preferência com você falando sobre e explicando o funcionamento adequado.
- Instalar o projeto em uma variedade de computadores, sistemas operacionais e/ou aparelhos e testar qualquer dependência de hardware ou software.
- Preparar um ou mais cartões de memória com todo o código fonte do projeto.
- Escrever um manual.
- Programar seu computador para executar a obra ininterruptamente por um longo período.



**Figura 31:** Imagem da capa do manual da obra Billow V de Daniel Canogar. Fonte: Fornecido o arquivo digital pelo sr. Diego Mellado.

<sup>55</sup> Cf. Perfil “antimodular” no GitHub. Disponível em: << <https://github.com/antimodular/Best-practices-for-conservation-of-media-art> >>. Acesso em: 20 maio 2022.

Estas ações, tomadas pelo(a)s artista(s) são importantes fontes de informação para futuras tomadas de decisão sobre o futuro das obras. Todavia, conforme menciona na entrevista, a maioria dos artistas somente produz este material após a solicitação pelo instituto. Poucos artistas hoje manifestam a preocupação de produzir um documento mais detalhado, como exemplo disto, teremos o manual produzido por Diego Mellado para o artista Daniel Canogar (Figura 31), entre outros, ou as preocupações apresentadas pelo artista Elias Crispin de em um de seus manuais ter a forma como deve ser inserido um determinado fio em uma conexão (Figura 32).



**Figura 32:** Still do vídeo da 23ª Jornada de Conservação de Arte Contemporânea do Museu Reina Sofia demonstrando o movimento que a pessoa que estiver instalando a obra deve fazer com a mão para conectar a fiação. **Fonte:** YouTube. Disponível em: << <https://www.youtube.com/watch?v=XvpM1OQrA6k> >>. Acesso em: 25 maio 2022. Local do still no vídeo: 2h46'45".

Seguindo o workflow, a segunda etapa do processo pré-aquisitivo é extremamente técnica e refere-se aos valores simbólico e de custo para manutenção da obra, mas, mesmo dentro destas informações técnicas há espaço para o(a)s artista(s) pensarem sobre o futuro de suas obras, afinal, Stricot lista como pontos chaves a serem observados:

- Acesse ao código fonte: O(a) Artista tem o código fonte? Qual a linguagem? Alguém ainda tem o conhecimento, na sua instituição, ou outro lugar?
- Uso de software open source ou comercial: o software tem licença? Você pode fazer um acordo com a empresa do software por suporte e update? Quanto custa a licença?
- Disponibilidade do hardware: O material é raro? Você pode achar peça de reposição facilmente? Pode ser consertada facilmente? Existem outros dispositivos específicos utilizados na obra de arte? Artesanal ou feito sob pedido? O(A) artista tem acesso a todos os esquemas?
- Conhecimento ou habilidades internas disponíveis para montagem e manutenção da obra em exibição: Você possui pessoal qualificado na sua instituição ou você precisa contratar terceirizados para cada montagem?

- Interdependência de Hardware ou Software: O software é compatível com outro computador ou um mais novo? É compatível com outros equipamentos de saída (output)?
- Dependência de Recursos Externos: existe alguma dependência de recursos externos (websites, servers de banco de dados, updates e suporte para software de terceiros etc.) ou tudo é executado localmente no computador? Posso guardar os dados em minha instituição? Quanto tempo esses recursos externos ficarão disponíveis? Isso é muito importante. Se alguma parte da operação do software for localizado em algum outro lugar, então você não pode controlar a disponibilidade do recurso.(ZKM, 2022)

Muitas destas decisões podem ser tomadas pelos artistas durante o período de criação das obras: Será que o programa autoral que resolvi tomar como base é o melhor para a execução da minha obra? O hardware que decidi utilizar como suporte não poderia ser outro que de menor custo ou mais fácil de ser adquirido?

A tentativa do artista de afirmar que sua obra é efêmera já não é tão bem vista pelos museus e centro de arte, a justificativa para a efemeridade da obra deve ser muito bem defendida, uma vez que estas instituições já não pretendem investir seu capital em algo que possui um prazo de validade, ainda mais naqueles que poderiam ser preservados para um futuro muito longo. Como cita Stricot no workflow: “Então a questão é: Vale a pena investir tempo, pessoal e dinheiro na preservação deste trabalho?”(ZKM, 2022).

Durante a etapa de aquisição, o momento que nos interessa se dá no preenchimento da Documentação, que o departamento de conservação disponibiliza um template para preenchimento, em que as informações passadas pelo(a)s artista(s) serão transcritas para armazenamento e futura consulta. Na etapa seguinte, é feito o teste destas informações para ver se procedem. Após, na etapa 4, o ZKM inicia a estratégia de conservação dos dados em discos rígidos e outras formas de armazenamento, inclusive contando com um servidor interno e uma cópia espelho em um local externo ao campus do ZKM.

O item 5 também faz parte da estratégia de substituição e migração, são feitas as coletas de todas as possíveis partes sobressalentes para substituição imediata quando necessário. Nesta parte, entra a informação dada acima de que muitas peças são esperadas sua obsolescência para aquisição com custo mais reduzido.

As etapas finais de aquisição, são o testamento do artista sobre os dados coletados, inclusive, é feita uma entrevista com o(a)s artista(s) para dirimir todas as possíveis dúvidas restantes.

A entrevista com Stricot e eventual participação de Vlaminck seguiu narrando as experiências de ambos com a restauração e recriação de obras, incluindo uma que sua

versão original era em Assembler<sup>56</sup>, e a ajuda que tiveram com grupos de entusiastas em fóruns de internet para conseguir informações sobre a semântica da linguagem ou peças de obras que já não se encontravam com facilidade.

Outra dificuldade manifestada foi com a presença de diversas mídias antigas como: disquetes, fitas cassete, fitas VHS, ou outros formatos, que ainda não foram definidos protocolos específicos para este tipo de preservação de material, principalmente pelo volume destes e a forma de leitura dos mesmos.

A visita encerrou-se com uma experiência única, quando Morgane retornou ao átrio da exibição BioMedia para tentar descobrir o que havia acontecido com a obra “Colóquio dos Móviles” já que apenas a Fêmea<sup>57</sup> estava funcionando adequadamente.

#### **4.3.2. Relato de entrevista com pesquisadores**

Como citado anteriormente, o período de permanência em Berlim na Alemanha permitiu acesso a diversos teóricos, acervos e, no mínimo, era possível aproveitar o fato de estar em um mesmo fuso-horário para realizar videochamadas que de outra forma seriam executadas bem cedo na manhã ou bem tarde na noite.

Muitas pessoas foram importantes aqui, mas alguns pesquisadores e professores foram direcionadores para a pesquisa tomar corpo. Como mencionamos anteriormente, o professor, artista e designer Jussi Ängeslevä, que de certa maneira tornou-se um herdeiro do legado artístico do professor Joachim Sauter, contribuiu para que pudéssemos pensar sobre a forma de documentação que levasse em consideração as contribuições múltiplas dadas por equipes inter e transdisciplinares. Além disso, uma de suas preocupações ficou sem ser respondida por não termos tido oportunidade de continuar com nossas digressões; esta seria o fato de o que fazer com memórias de obras acumuladas em mais de 30 anos de trabalho contínuo sem ser executada uma documentação específica por parte do artista.

Além da visita a Karlsruhe, tivemos a oportunidade de nos deslocarmos para Madri e ter acesso a dois teóricos considerados basilares para a elaboração da problemática desta tese: Arianne Vanrell Velloso e Lino García Moralles. A primeira é autora da tese *“Nuevas estrategias para la conservación de colecciones de arte con elementos tecnológicos: propuestas metodológicas de humanidades digitales”* de 2015, que aborda o

---

<sup>56</sup> **Nota do autor:** Assembler é uma linguagem muito mais próxima da linguagem do computador e que não possui a capacidade de se colocar comentários no código.

<sup>57</sup> Cf. Item 2.1.1.

uso das humanidades digitais como ferramentas para a conservação de obras em mídias instáveis.

Vellosillo nos recebeu no Museu Reina Sofia e concedeu uma entrevista além de um passeio para visualizar as principais obras com elementos tecnológicos.

Ponto importante abordado na entrevista com a sra Vellosillo foi a de que diversas dos instrumentos citados por ela na tese já não existiam mais no ano seguinte. Como resposta, ela reforçou o fato de que é importante que independentemente do sistema a que o artista se propõe a utilizar, ele deve possuir uma documentação o mais detalhada possível sobre sua obra. Foi por meio dela que ficamos conhecendo o artista Elias Crispin que inclusive concordou com uma entrevista que não ocorreu devido a problemáticas causadas pelas restrições pandêmicas, porém pretendemos retomar esse contato assim que possível.

A pesquisadora também mencionou um grande projeto de documentação da obra Guernica de Picasso<sup>58</sup>, que sem dúvida é o chamariz de todo o acervo principal do museu. Neste projeto, foram utilizados dois robôs com câmeras fotográficas especiais que capturaram diversas fotos com luz visível, luz ultravioleta, infravermelho e radiografias, compondo uma imagem chamada pelo projeto de Gigapixel. Esse projeto permite que possamos perscrutar o quadro como se estivéssemos a centímetros de distância do mesmo, observando as ranhuras, as pinceladas, o uso de materiais diferentes na criação da imagem e inclusive restos da tinta vermelha que foi utilizada em um protesto contra a obra.

Este tipo de projeto é dispendioso e não pode ser executado em todas as obras, mas demonstra que mesmo no caso de obras de arte tradicionais, o uso das ferramentas tecnológicas pode contribuir com um melhor entendimento do trabalho dos autores.

No dia seguinte à entrevista com Vellosillo, fomos recebidos pelo Prof. Dr. Lino García Morales, na Universidade Politécnica de Madri no Campus de Montegancedo. Tivemos conhecimento do trabalho de pesquisa do professor por meio de artigo publicado em 2018 em que abordava os obstáculos da conservação de obras de arte em mídias instáveis. Neste artigo, ele e a Professora Pilar, faziam uso de um diagrama de classificação das expressões artísticas que contribuiu para repensar o papel ontológico da arte computacional diante deste universo de suportes variáveis.

---

<sup>58</sup> **Nota do autor:** RepensarGuernica. Disponível em: << <https://guernica.museoreinasofia.es/> >>. Acesso em: 25 maio 2022.

Ao entrar em contato por email o professor mencionou o livro que tinha escrito sobre a “Teoria da Evolução Conservativa” descrita no início deste capítulo que apontou para a necessidade de uma documentação realizada pelo artista e que constasse na mesma as informações pertinentes à sua poética.

Todavia, foi um de seus alunos egressos, Diego Mellado, que serviu de inspiração para buscarmos uma metodologia descritiva em níveis de compreensão das camadas de uma obra de arte em mídias instáveis. Mellado é um engenheiro que se especializou em compor as informações técnicas necessárias para a confecção de montagem e apresentação de obras de artistas das mídias instáveis.

O artigo que tivemos acesso, de título “*A System Engineer’s Perspective for the Re-Creation of Media Art n-Cha(n)-t by David Rokeby*”, apresentava uma representação da obra n-Cha(n)-t do artista David Rokeby em níveis de abstração e generalização. Durante nossa entrevista foi possível apresentar a ele as ideias que havíamos desenvolvido até então e uma de suas críticas foi importante para a melhor caracterização de nossos objetivos: no caso, ele citou que o trabalho de diagramação dos níveis da obra EVO\_CIRCUITO estava ficando muito clara, mas muito mais caracterizada como trabalho de um engenheiro de sistema, o que não era pertinente ao intuito do trabalho de pesquisa. Assim, que o protocolo criado aqui tornou-se voltado para que o artista pudesse deixar clara sua intenção poética com a obra.

#### **4.3.3. Participação na disciplina de Arte Generativa**

Uma das ofertas generosas que tivemos durante o período do doutorado-sanduíche foi a participação como convidado na disciplina de Arte Generativa Computacional do Prof. Dr. Alberto de Campo.

O professor é titular da cadeira de arte generativa e arte computacional no Instituto de Mídia baseada no Tempo da Universität der Künsten – Berlin desde 2009. Com doutorado em música computacional, seu trabalho artístico é voltado para a composição de peças sonoras computacionais e a criação de instalações que são também suporte para obras sonoras.

Na aula foi possível observar o processo criativo dos alunos, bem como receber feedback de ideias para obras e da tese. Foi também como resultado das discussões desta classe, em ano anterior à minha participação, que foi concebida e desenvolvida a obra “Future Voices” cuja documentação foi assumida por nós como parte do desenvolvimento da pesquisa desta tese.

## 5. Quinto Capítulo: Protocolo de Conservação de Obra de Arte Computacional.

Este capítulo apresenta as bases técnicas que deram origem ao modelo C7Art, assim como, descreve o passo a passo de implementação e os requisitos e especificações necessárias. Posteriormente são apresentados dois estudos de caso: EVO\_CIRCUITO, obra computacional em instalação pensada e criada em conjunto com o Media Lab/UnB; e Future Voices, obra sonora generativa, produzida pela “Sociedade pelas buscas não triviais” (S4NTP), cujos componentes são os professores Alberto de Campo e Hannes Hoelzl e os alunos da disciplina de Arte Generativa da Udk.

### 5.1. Modelo C7Art

Para este protocolo baseamo-nos em estratégias utilizadas comumente em arquitetura de software, engenharia de sistemas e métodos de documentação de desenvolvimento de software, em uma hibridização com as documentações feitas por artistas. O objetivo é criar uma comunicação clara da intenção da mensagem estética do(a)s artista(s) (Poética do Artista) e possibilitar que qualquer desenvolvedor de software possa, considerando a ausência do(a)s artista(s), recuperar uma obra até onde seja possível, ou recriá-la em último caso.

#### 5.1.1. Princípios direcionadores

Desde a substituição de parte das instruções das funções de máquina por símbolos (cf. item 1.2.2), iniciou-se o pensamento estruturado do código, afinal, mais do que hoje em dia, o gasto de tempo de máquina era muito caro, e como se dava pela leitura de uma fita ou de cartões, se não houvesse um planejamento, muito tempo poderia ser gasto tanto no desenvolvimento dos cálculos quanto para tentar encontrar os "bugs".

Foi assim que, nas últimas décadas do século XX, algumas metodologias para criar códigos "limpos" (códigos otimizados de escrita e leitura facilitada), entre elas: a Structured Systems Analysis And Design Method (SSADM), criada em 1980; a Rational Unified Process (RUP), criada em 1997 e a Unified Modelling Language (UML), criada entre 1994 e 1995. Dentre as propostas, selecionamos para exemplificar a metodologia, a UML por tratar-se de uma linguagem de estruturação de arquitetura de software e que é componente essencial dos últimos níveis tanto do modelo C4 quanto do C7Art.

UML é uma "notação padronizada para comunicação do design de um sistema de software" (BROWN, 2021, p. 3) desenvolvida para criar "padrões" e "diagramas" que especifique, visualize, construa e documente um determinado software, um modelo de negócios e outros sistemas que não sejam softwares (EVITTS, 1999, p. 23 e 33). Considerada uma linguagem pelo *Object Management Group* (OMG), a UML é bastante rica em detalhamento, porém sua série de símbolos e nomenclaturas obriga a um letramento por parte do usuário e dos leitores do projeto, mesmo que facilitado por diagramas, mas suas conexões e elementos acabam por criar dificuldades para qualquer um de fora da área.

Como já mencionado anteriormente, o objetivo de nosso protocolo é o uso indiscriminado e simplificado, criando uma linguagem mais universal para descrição de obras computacionais de modo a serem preservadas as intenções estéticas do(a) artista. Portanto, por mais que haja um material descritivo e documental muito rico a ser explorado na UML, para um artista-desenvolvedor ou para qualquer interessado em conservar ou re-criar uma obra de arte computacional seria uma linguagem a mais a ser aprendida e explorada.

Em "A System Engineer's Perspective for the Re-Creation of Media Art n-Cha(n)-t by David Rokeby"(MELLADO, 2019), Diego Mellado utilizou-se de uma série de metodologias e instrumentos da Engenharia de Sistemas para realizar um passo além da conservação, a Re-Criação de uma obra de arte, no caso a n-Cha(n)-t do artista David Rokeby. Para isso Diego teve a ajuda do artista e de vários documentos, contudo, o código fonte da obra havia se perdido já que fora desenvolvido em uma IDE que não é mais acessível a nenhum computador. Apenas o código de uma obra anterior a qual a n-Cha(n)-t foi um desdobramento, estava disponível, apesar da constatação de que era um código extremamente complexo:

Diversas dezenas de milhares de linhas de código, que são um resultado – como Rokeby mesmo disse – de mais de 10 anos de mudanças e adições, escritos de maneira que pudessem ser compilados (...) em diferentes sistemas modernos. (MELLADO, 2019, p. 183–184)

O que se destaca no processo descrito por Diego Mellado é, além da defesa da "re-criação" como alternativa para salvar uma obra de arte, a organização descritiva da obra: ela caminha de níveis diferentes alternando abstrações e descrições detalhadas conforme vai aprofundando os níveis. Assim, um L0 seria o nível mais superficial e conteria uma descrição genérica da obra, com especial atenção para o fato de que para o autor o método descritivo de preservação deve representar a obra de duas maneiras (MELLADO, p. 182): (1) descrever a obra na maneira como ela é percebida, focando assim no seu valor simbólico; e (2) uma descrição de suas partes, estrutura e a relação entre esses elementos, fazendo assim, segundo o próprio autor, uma ligação direta com a Teoria da Restauração de Cesare Brandi (BRANDI, 2019). Convém ressaltar o fato de que para

exemplificar a descrição da L0, Mellado utilizou um diagrama com as diversas partes da obra (individuals) e suas relações.

O L1 já seria uma descrição de cada uma das partes da obra, uma lista das características de cada um dos componentes estruturais, descrevendo como cada elemento deve se comportar e como estes estão conectados. Esta descrição está condizente com a ideia de "Caixa-Preta" que o autor relaciona com um processo de abstração utilizado dentro da Engenharia de Sistema, que permite que determinado elemento possa ser substituído por outro com as mesmas características e comportamento, por exemplo um computador ou um monitor, as tecnologias mudam, mas se a função for apenas de apresentação, qualquer modelo que atenda ao exigido pode ser usado, descartadas as referências estéticas.

Nos níveis mais abaixo, cada subelemento pode ser descrito e tornado uma "Caixa-Preta" passível de uma substituição em caso de perda, descontinuidade, obsolescência. L2 então seria uma mesma atividade dos elementos formadores da L1 e por assim adiante.

O segundo preceito seguido para o desenvolvimento de nosso protocolo, foi a procura por uma linguagem ubíqua. Nossa pesquisa nos direcionou para a metodologia Domain Driven-Design (DDD), uma metodologia desenvolvida por Eric Evans (EVANS, 2016) para criar modelos de diagrama pré-código que serão utilizados como referência por um desenvolvedor de software (especialmente aos programadores de linguagens voltadas ao objeto) na arquitetura de um software. Uma das premissas mais basilares desta metodologia está na palavra "Domínio" (Domain) que se refere ao campo de saber ou de negócio para qual o software está sendo estruturado: "Contudo a complexidade mais significativa de muitos aplicativos não é a técnica. Ela está no próprio domínio, a atividade ou negócio do usuário" (EVANS, 2016, p. XXI).

Entretanto, no uso deste conceito, que Evans faz ao longo do livro, entendemos "Domínio" como assemelhado ao conceito de "Campo" utilizado por Pierre Bourdieu em "Economia das Trocas Simbólicas" (BOURDIEU, 1999) e em suas obras posteriores. Foi em tal paralelo que encontramos apoio para aplicar a mesma metodologia criada principalmente para softwares que atendessem a um público empresarial/comercial/institucional, como uma metodologia propícia a ser utilizada para a conservação de obras de arte computacional, pois uma vez que ela deve compreender o domínio em que se insere, o DDD pode ser adequado à essência da obra de arte computacional.

Deve-se salientar que para o DDD a documentação é material complementar ao código sendo o código parte da própria documentação mediante a utilização de comentários (EVANS, 2016, p. 35). Todavia, o uso de cada documentação deve focar nas características de cada método, se afastando de um jargão enclausurado à própria equipe de desenvolvimento, sempre considerando que o papel de cada documento é preencher lacunas de entendimento.

O terceiro preceito utilizado no C7Art foi o modelo C4, este ocupa um espaço muito específico na composição do protocolo, por esta razão é apresentada em um subnível nesta tese.

### 5.1.2. Modelo C4

Assim chegamos ao Modelo C4, desenvolvido por Simon Brown, preocupado com o contínuo abandono do UML (por ser ele muito denso e de difícil aprendizagem), e de qualquer tipo de organização visual da arquitetura de software (BROWN, 2021, p. 2). Segundo Brown, sua intenção é

Isso se trata de efetivamente e eficientemente comunicar a arquitetura de software que você está construindo levando em consideração:

- Ajudar a todos a entender o cenário maior do que está sendo construído, e como isso se encaixa no cenário.
- Criar uma visão compartilhada do que você está construindo com o seu time de desenvolvimento.
- Providenciar um ponto de foco para a concentração de seu time de desenvolvimento sobre o que é o software e como ele está sendo construído.
- Providenciar um ponto de foco para as técnicas de conversação sobre novas funcionalidades a serem implementadas.
- Providenciar um mapa que pode ser usado pelos desenvolvedores de software para navegar pelo código fonte.
- Ajudar a explicar o que você está construindo para pessoas de fora do time de desenvolvimento, sendo técnicos ou não.
- Rápida inserção de novos membros no time de desenvolvimento do software. (BROWN, 2021, p. 2)

Sua metodologia divide o software em 4 estágios ou níveis:

**(1) Sistema do Software:** Assim como em Mellado (in Grau *et al.*, 2019), o primeiro nível é o mais abstrato e generalizante, representa para Brown os valores encontrados ou buscados pelo desenvolvedor como "ponto de partida" (BROWN, 2018 p.46).

**(2) Contêineres:** Para Brown, os contêineres são representações de parte da aplicação que detém dados ou códigos, podendo ser bancos de dados, sub-rotinas, softwares de atuação paralela, autenticadores de login etc. Aqui o jargão aplicado por Mellado é o de "atores" da obra ou os "individuals". Apesar dos termos utilizados por Mellado serem mais propícios a identificar a interatividade das obras, a ideia de contêiner é mais genérica e pode ser utilizada para maior número de atividades. De qualquer maneira, esta é a parte em que "mostra os blocos de construção de nível mais elevado (contêiner) e como eles interagem" (BROWN, 2018 p.46).

**(3) Componentes:** Componentes são as partes compositoras dos contêineres e estas não são partes autônomas, ou seja, não funcionam se não em sua relação com outras partes. Um exemplo a ser dado para ilustrar, podemos considerar um Desktop como um

Contêiner de uma obra, a Placa de Vídeo seria um Componente, assim, você pode retirar a Placa de Vídeo porém ela não terá nenhuma utilidade isolada.

**(4) Código:** Este é o nível mais profundo e é relativo diretamente ao código dentro da linguagem de programação escolhida. Nela está a semântica e a sintaxe da linguagem com suas classes, interfaces, funções, objetos, etc.

### 5.1.3. Adaptando os modelos

O modelo C4 foi desenvolvido pensando em uma arquitetura de software, principalmente os sistemas comerciais com funções utilitárias diferentes da função estética. Assim, fora os requisitos básicos de sistema colocados no primeiro nível (C1), não há uma preocupação com o hardware no mesmo nível que em uma obra de arte computacional.

No início da pesquisa, imaginávamos que o protocolo serviria apenas para a conservação de um código, porém, mesmo o hardware sendo para determinadas obras inexpressivo na questão visual, a sua funcionalidade deve ser preservada dentro do entendimento da composição da obra. Desta forma, propomos uma modificação no modelo C4 em casos em que a obra de arte computacional for composta de parte hardware e parte software.

O modelo seguiria a descrição estrutural da obra partindo da intenção do artista (C1), seguindo para a descrição de cada parte da estrutura da obra (C2), seguindo para os componentes destas estruturas (C3) e no momento que entra na descrição do C4, abre-se um novo diagrama do modelo de 3 níveis, desta vez se referindo a forma como o código foi elaborado.

Contudo, percebemos que nas obras de arte computacional não se tratava apenas de demonstrar o funcionamento do hardware e o do software, mas se quiséssemos relacionar com a intenção poética, deveríamos possuir determinados níveis de correlação entre o hardware e o software e a obra como sistema e o *iterator*.

Para desenvolver o modelo de diagramação C7Art entendemos que não havia espaço para tecnicidades excessivas. A primeira versão do modelo foi apresentada a algumas pessoas, uma delas foi o engenheiro Diego Mellado, que nos concedeu entrevista para falar sobre seu artigo. Ao ser apresentado os diagramas, Mellado apontou o fato deles estarem muito próximos de uma visão de engenheiro, e que mesmo que tenhamos conseguido certo êxito neste ponto, não é nossa formação acadêmica ou pretensão para esta tese.

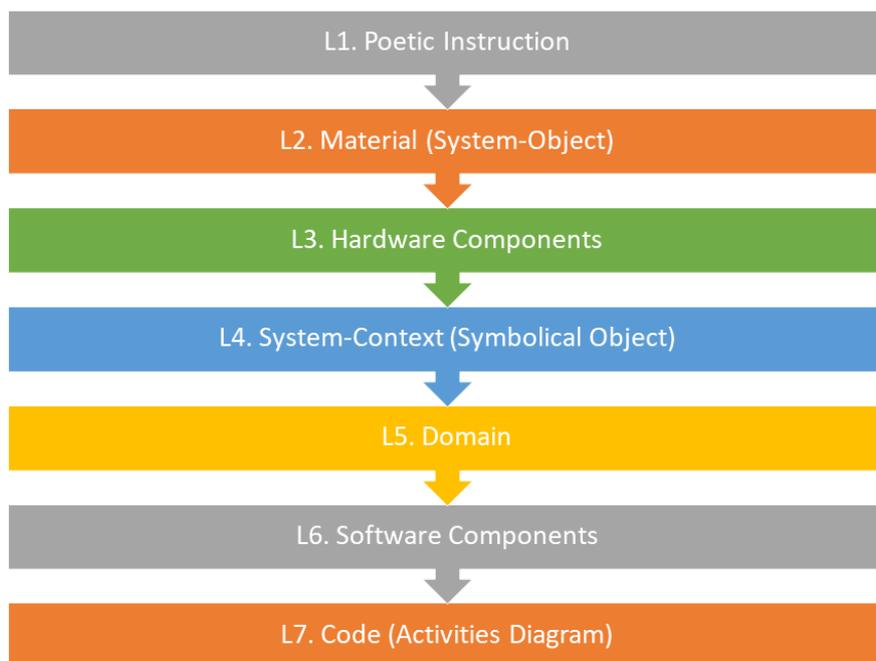
Desta forma, a versão aqui apresentada seguiu algumas orientações básicas: (1) o protocolo deve ser simples de ser desenhado e executado, possuir uma linguagem de entendimento geral (a nomenclatura não pode abrir brechas para interpretação ou, explicar os termos antecipadamente o que não é recomendado por afetar a curva de aprendizado e a aplicação); e (2) deve ser flexível para alterações, uma vez que uma das características da Arte como esfera de pensamento é sua

capacidade de inovação, de exploração dos limites, assim qualquer tentativa de se reduzir os objetos artísticos à categorias muito rígidas certamente irá expor uma série de excepcionalidades.

Assim, após algumas adaptações, chegamos a um modelo de diagrama de fases em 7 níveis.

#### 5.1.4. Diagrama de 7 níveis

Como citado anteriormente, os diagramas do Modelo C4 e do DDD foram base para pensarmos o nosso protocolo. Por entendermos, assim como Morales (2019), que toda arte computacional necessariamente irá ter o computador (ou outro dispositivo que faça esta função) como suporte, sua descrição como parte da obra é importante e deve ser caracterizada, por isso o modelo que imaginamos são de um mínimo de 7 níveis<sup>59</sup>, os 4 primeiros seguem o modelo C4 apenas que são relativos as partes materiais relativas ao suporte e as relações sensoriais, os últimos 3 níveis são relativos ao software e seguem o mesmo modelo culminando na estrutura de um diagrama de atividades da linguagem UML (veja figura abaixo).



**Figura 33:** Os sete níveis básicos do modelo C7Art. Fonte: Acervo próprio.

- a) **L1. Instruções Poéticas:** Neste nível o diagrama é o mais geral possível, demonstrando toda a montagem da obra determinando principalmente os elementos de *input* e *output* e a

<sup>59</sup> **Nota do autor:** Importante aspecto do modelo é que ele atende a 7 níveis virtuais, aspectos específicos da descrição da obra. Se houver necessidade, o responsável pela estruturação do diagrama pode aumentar ou diminuir o número. O protocolo aqui é uma sugestão de uso e aqui são apresentados os dados de seu uso preliminar.

relação entre a mensagem encaminhada (mesmo a mais abstrata mensagem como uma sensação) e o *iterator*.

- b) **L2. Material (objeto-sistema):** Este nível serve para caracterizar o funcionamento da parte material em seu nível mais geral. O importante aqui é frisar a relação da obra com os mecanismos que estão fora do controle do(a)(s) artista(s). Exemplo: fornecimento de energia elétrica, condições ambientais, configuração de Internet, etc.
- c) **L3. Componentes de Hardware:** Este diagrama deve concentrar os componentes internos da parte material. Partes de computador, microcontroladores. Peças compradas devem ser listadas com suas funcionalidades (qual o intuito do uso de cada uma em cada parte da obra). Peças montadas devem ser apresentadas em desdobramentos em subníveis do diagrama.
- d) **L4. Contexto do Sistema (objeto-simbólico):** Neste ponto os diagramas devem se concentrar em descrever a relação entre os componentes de hardware e os componentes do software com a mensagem.
- e) **L5. Domínio:** Este nível contém diagramas que descrevem com mais detalhes o comportamento estético intencionado da obra-software. Contudo, é o nível mais geral que lida com o software, sendo os próximos níveis detalhamentos e abstrações da obra-software.
- f) **L6. Componentes de Software:** Neste nível, o diagrama se aprofunda no funcionamento do software, separando-o por funções.
- g) **L7. Código (Diagrama de Atividades):** O diagrama de atividades é um tipo de representação de comportamento utilizado na linguagem UML. Escolhemos este diagrama como padrão para o protocolo em razão do objetivo principal desta tese que relembramos ser a busca por uma forma de conservar a intenção estética de uma obra. Assim, o diagrama de atividades que descreve o fluxo entre as atividades de um processo é a forma mais ubíqua de se estabelecer uma correlação entre o querer do artista e o trabalho do desenvolvedor do software (mesmo que os dois papéis sejam ocupados pela mesma pessoa).

Os diagramas são parte complementar do protocolo de conservação, e registram ativamente a instrução estética do(a)(s) artista(s). A criação de outras formas de documentação é incentivada e abaixo fazemos uma revisão inicial sobre as formas de documentação encontradas.

#### 5.1.5. Documentação

As restaurações das obras aqui listadas foram facilitadas, pelo menos em parte, por elas terem tido um certo apelo na comunidade artística e acadêmica, foi o caso de “Desertesejo” e “Colóquio dos Móviles”. Em seu site o artista Gilberto Prado conseguiu reunir dezenas de artigos, escrito por

ele e por outros pesquisadores, sobre a obra apresentada em 2000. Mesmo que durante o processo de recriação tenha sido ele o maestro que conduziu os processos, muitas informações foram possíveis de serem verificadas devido aos registros. Em entrevista, o artista também nos informou sobre a existência de alguns arquivos que apesar de não servirem para compor a nova versão, serviam para auxiliar no direcionamento.

O Colóquio dos Móveis não pôde contar com a presença em vida do artista Gordon Pask. Ele e sua equipe em 1967/68 realizaram um rico trabalho de documentação da obra, assim, Paul Pangaro, Thomas McLeish e a equipe do Projeto Colóquio 2018 puderam resgatar muitas informações, mesmo porque diversos componentes eram completamente autorais. Além de contar com um grupo de conselheiros, incluindo as filhas de Gordon Pask, que os orientavam sobre a forma com que o artista pensava.

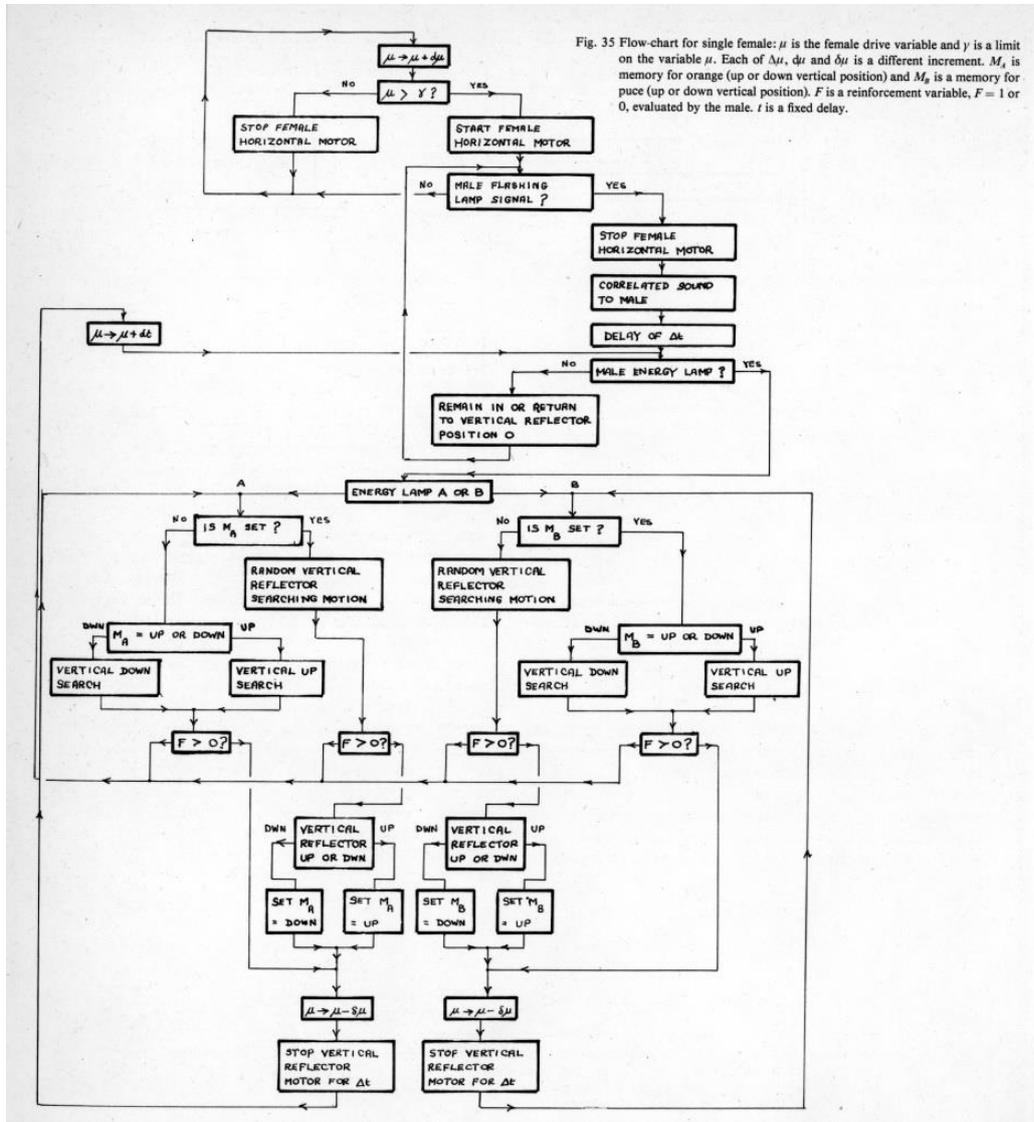


Fig. 35 Flow-chart for single female:  $\mu$  is the female drive variable and  $\gamma$  is a limit on the variable  $\mu$ . Each of  $\Delta\mu$ ,  $\delta\mu$  and  $\delta\mu$  is a different increment.  $M_A$  is memory for orange (up or down vertical position) and  $M_B$  is a memory for puce (up or down vertical position).  $F$  is a reinforcement variable,  $F = 1$  or  $0$ , evaluated by the male.  $t$  is a fixed delay.

Figura 34: Um dos Diagramas de Fluxo originais de Gordon Pask. Fonte: (PASK, 1971).

O diagrama acima (Figura 34) é um dos diversos diagramas apresentados no artigo “A comment, a case history and a plan” escrito por Pask em 1971. Mesmo tendo feito um rigoroso trabalho de documentação, durante o processo de recriação da obra em 2018, a equipe verificou uma série de lacunas nas informações de Pask. Como exemplo, podemos citar que no documento Project Overview disponibilizado pelo projeto, na página 30, é possível ver as anotações ao lado de um diagrama, que diz estarem faltando informações sobre os displays das “fêmeas” assim como a informação sobre a luz das “fêmeas” (PANGARO et al, 2018). Em entrevista concedida a nós em junho de 2022, Paul Pangaro e Thomas John McLeish, narraram suas experiências durante a reconstrução da obra e como lidavam com as informações que faltavam na documentação original. Um ponto interessante comentado foi McLeish narrando que o aprendizado sobre a obra vinha de quando remontavam acabavam por comparar com os diagramas, imagens e vídeos, e muito era construído pela experiência de uso com a obra sendo construída. Inclusive, eles disponibilizaram um conjunto de documentos conseguidos após a recriação, e entre esses documentos há um que apresenta anotações do Pask sobre as modificações feitas à obra após os diagramas. Convém dizer que os diagramas foram feitos anteriormente ao “Colóquio dos Móviles”.

A situação vivenciada pela equipe do Colóquio foi contornada devido a outras fontes de informação, o que representa uma justificativa para criar uma diversidade de documentos até mesmo apresentando redundância de informações.

Abaixo uma lista dos documentos que podem fazer parte da documentação de uma obra:

- Artigos científicos;
- Portifólios ou catálogos de eventos/exposições;
- Vídeos e fotos produzidas pelo próprio artista ou obtidas mediante o uso de redes sociais (atrelando-se o # ao perfil do artista).
- Projetos para financiamento;
- Printscreens de parte de desenvolvimento (principalmente quando as composições são feitas mediante o uso de caixas e suas ligações como obras criadas em Touchdesigner ou Blender);
- Diários de artista;
- Entradas em Sites para acervo de código (GitHub);
- Anotações avulsas.

A lista acima apresenta várias possibilidades de documentação atreladas ao desenvolvimento de uma obra artística, alguns específicos para as obras computacionais, outros genéricos. Quando a presença física do artista já não é possível sua "voz" tem que ser "ouvida" por meio da documentação. Como um fator motivador para a documentação é de enquanto ainda em

vida, o(a) artista pode usufruir desta documentação como memórias das criações e com isso retomar ideias ou desenvolver novas versões de cada obra.

Obviamente é necessário uma disciplina para a documentação e nesse ponto encontra-se uma convergência com o que é exigido de um desenvolvedor de software ao elaborar a arquitetura de seus sistemas. Manuais completos como os apresentados nas Figuras 31 e 32 ainda são raros e dispendiosos. Nas palavras de um desenvolvedor, Eric Evans:

Quando decidimos escrever um software, nunca sabemos o suficiente. O conhecimento sobre o projeto é fragmentado, disperso entre muitas pessoas e documentos e misturado com outras informações de forma que nem mesmo sabemos qual conhecimento realmente se faz necessário. (ERIC EVANS, 2016, p. 15)

Obras como a EVO\_CIRCUITO (Media Lab/UnB, 2018) e Future Voices (S4NTP, 2019) são desenvolvidas por muitos artistas, e nas Artes das Mídias Instáveis essa é uma característica particularmente comum, em que coletivos artísticos e laboratórios desenvolvem e apresentam obras e experimentos artísticos coletivos e interdisciplinares. Neste tipo de arranjo, uma vasta documentação se torna vital para a coesão da intenção estética do grupo, assim como para o andamento do projeto.

#### **5.1.6. Etapas do protocolo para conservação estética**

##### **A) Primeira Etapa**

Percebemos durante a elaboração deste protocolo, que o ideal para o início de uma criação artística em arte computacional é a estruturação da obra a partir da ideia. Sendo assim, todo rascunho deve ser guardado nesta etapa, desde reuniões preliminares gravadas em áudio ou em vídeo, fotos utilizadas para documentação e documentos diários como caderno de viagem; caderno de artista; ou *sketchbook* na trilha das ideias em formação.

Este procedimento não determina em qual etapa a ideia da obra surgirá, já que em um mesmo processo podem surgir diversas ideias ou desdobramentos. Se tomarmos que o acaso possui grande influência sobre as definições estéticas, só iremos saber a importância de um documento no futuro, quando já não houver muitas condições de criá-lo.

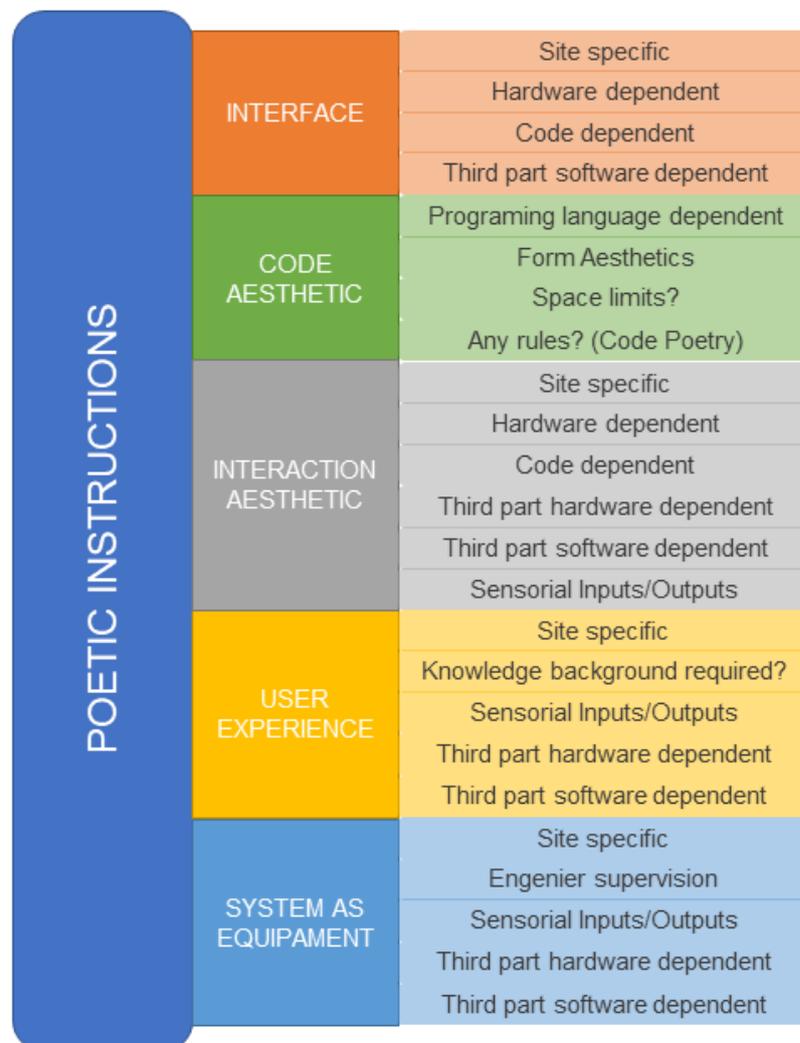
Em entrevista com alguns artistas da tese citados, eles apontaram para a dificuldade de manter uma documentação que não interrompa o fluxo de criação artística. A sugestão do protocolo é que cabe a(o) artista experimentar diversas formas de documentação do seu pensamento criativo para ver qual é mais adequado à suas necessidades, mas realizar a documentação para não perder sua linha de pensamento.

## B) Segunda Etapa

A sugestão nesta parte é a de iniciar a estruturação dos diagramas, começando pelas perguntas de orientação:

- I. Para o(a) artista, onde encontra-se a instrução poética da obra?
- II. Qual a posição do(a) artista com relação aos processos a serem adotados para conservação ou restauração?
- III. O artista aceita que não havendo mais como continuar conservando a obra tal como ela foi finalizada que ela possa ser recriada?

As perguntas II e III são perguntas que requerem a criação de documentação do(a) artista declarando a sua posição sobre as mesmas (cf. Apêndice III). A pergunta I se desdobra em outras observações que devem ser feitas conforme define-se a instrução poética da obra (Figura 35).



**Figura 35:** Instruções poéticas a serem observadas ao início da diagramação. Fonte: Acervo próprio, 2022.

### C) Terceira Etapa

O diagrama pode ser efetuado depois ou durante a criação do código. Ainda, o diagrama pode ser utilizado para estruturar o programa e o programa pode modificar o diagrama conforme o(a)(s) artista(s) modifica(m) o caminho estético construído.

Devem ser definidos os principais elementos de cada um dos níveis, julgando a necessidade de se criar o diagrama para cada um deles (muitas vezes um diagrama anterior já é suficientemente descritivo na avaliação do(a)(s) artista(s)).

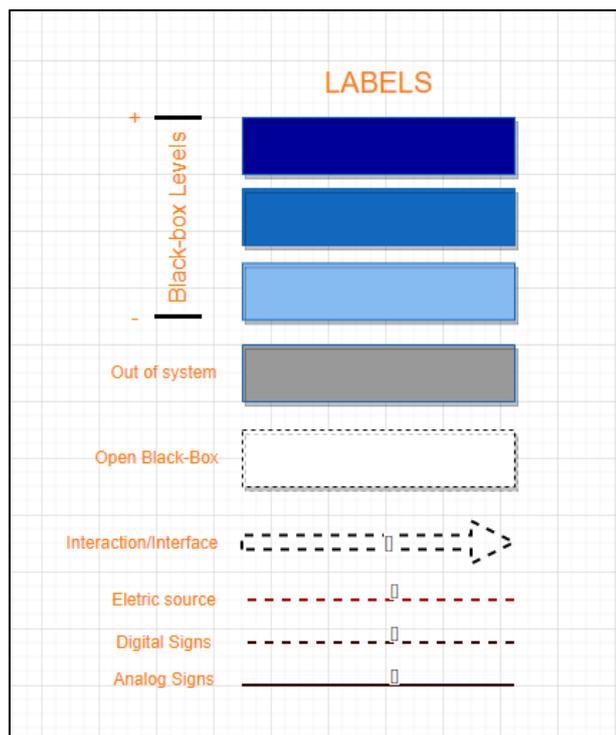
Um bom resultado neste ponto, independentemente se haverá a escrita do código antes ou depois do diagrama, é investir em comentários diretos no código, instruindo sobre as funções e variáveis utilizadas. Rafael Lozano-Hemmer em “Best practices for conservation of media art”, também sugere que:

Trabalhe em qualquer plataforma que você sinta ser a melhor para o projeto e para você, mas se você tiver uma escolha sempre escolha as ferramentas open-source. No meu estúdio nós comumente usamos sistemas comerciais fechados, como o “FaceAPI” para reconhecimento de rosto e “Shout3D” um API 3D online proprietário, apenas caso as companhias entrem em falência ou deixem o software órfão deixando-nos com o trabalho de re-engenharia com componentes equivalentes mais abertos (OpenCV no primeiro caso e Google Earth no outro). (LOZANNO-HEMMER, [s.d.]

Algumas orientações gerais sobre os diagramas:

- O diagrama deve deixar claro toda e qualquer informação pertinente sobre as funções de cada uma das partes da obra;
- Os elementos presentes ao diagrama devem refletir parte da mensagem da obra.
- Os elementos devem ser mencionados e ilustrados, ou utilizados em uma imagem para referência.
- O equilíbrio entre as informações e o tamanho do diagrama deve ser observados. Caso haja a necessidade de se desdobrar mais informações aponte para um subnível.

Abaixo segue uma sugestão para a iconografia dos diagramas (Figura 36).



**Figura 36:** Sugestão de iconografias a serem utilizadas nos diagramas. Fonte: Acervo próprio.

Após a escrita do código e a elaboração dos diagramas, a próxima etapa refere-se às ações de preservação dos arquivos escritos.

#### **D) Quarta Etapa**

Esta etapa deve ser executada depois que há a concordância sobre uma versão de exibição da obra. A reserva do máximo de arquivos abertos (arquivos que ainda podem ser manipulados) documentando-os como arquivos finais deve ser observada. Além disso, caso haja a possibilidade, criar uma cópia (*Ghost Copy*) de todo o ambiente em que a obra é executada, de forma a criar um ambiente de emulação. Importante preservar estes arquivos em mais de um espaço de armazenamento.

É sugerido que haja a listagem de todo o ambiente de hardware e software em que a obra é possível de ser executada, inclusive com versões para múltiplas plataformas (Windows, MacOS, Android, IOs, Linux, entre outras).

Nesta fase, como nos apontou o Prof. Morales, é importante a entrevista gravada, e talvez transcrita, de todos os participantes do desenvolvimento da obra. Esta talvez fosse a solução no caso apresentado por Jussi Ängeslevä quando do falecimento do professor Joachim Sauter e a documentação das obras em que ele não foi o único a contribuir.

## 5.2. Estudo de caso em criação estética e documentação e aplicação do protocolo

A problemática de documentação para conservação das obras computacionais surgiu para nós quando em função da atividade do magistério. Nesta atividade, é necessário apresentar aos alunos o funcionamento e a forma de interação com as obras de arte computacional, tendo como fonte a descrição das mesmas em sites na Internet. Tal fato ocorre em razão da distância da Unifesspa (Marabá-PA) das regiões brasileiras que comumente apresentam exposições de arte das mídias instáveis.

Assim, como opção para nós a apreciação das obras por documentos fornecidos por artistas ou instituições e isso se mostrou um desafio, já que a descrição poética da obra nem sempre nos fazia imergir na mesma.

Ao longo do processo de doutoramento, tivemos a oportunidade de acompanhar criações de obras, discussões sobre criações nossas, experiências de outros artistas conceituados na documentação das suas obras e práticas de documentação de museus e galerias em pelo menos três países diferentes.

A versão do protocolo aqui apresentada, foi modificada pelo feedback dado pelos artistas envolvidos nas criações, evoluindo em sua descrição e criando o modelo de diagramação em 7 níveis. Porém, acreditamos que ele irá evoluir e nosso trabalho como pesquisadores, professores universitários e artistas é contribuir com o que pudermos, assim, o uso deste protocolo deve ser livre e isento de qualquer tipo de cobrança.

Abaixo estão descritos dois dos processos de documentação orientados pelo protocolo e a apresentação dos diagramas de cada uma das obras.

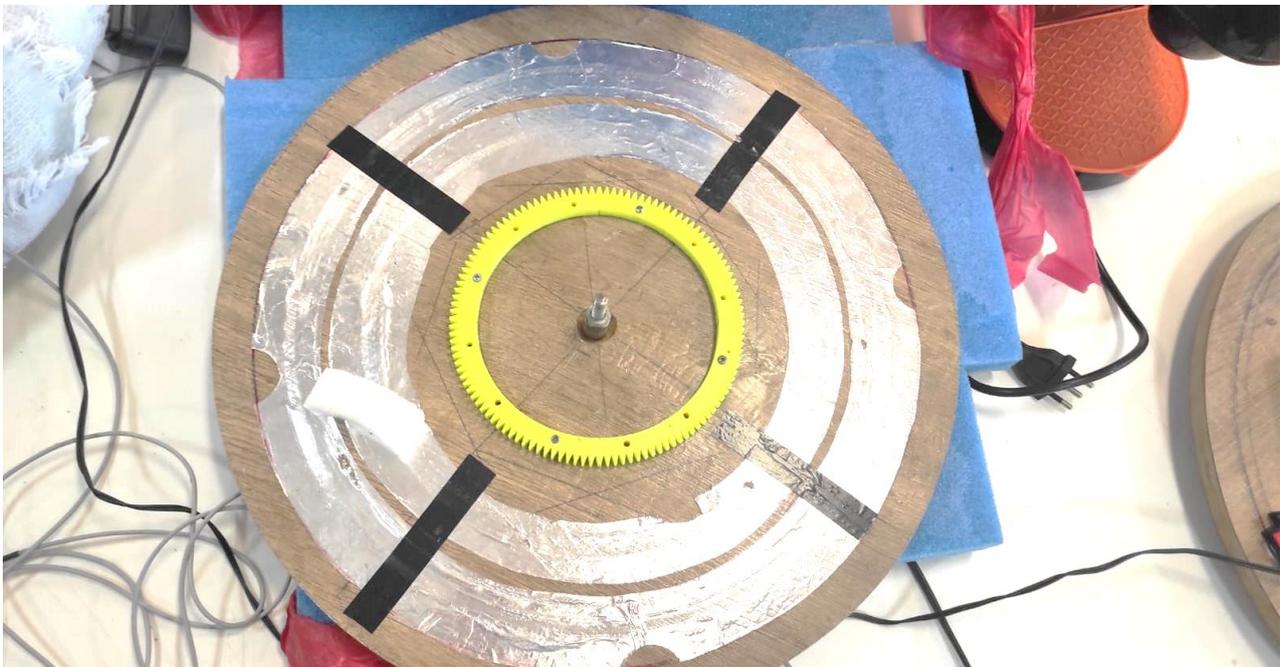
### 5.2.1. EVO\_CIRCUITO (Maio de 2018)

#### A) Histórico

O primeiro projeto executado quando iniciamos nossas atividades como aluno de doutorado e participante do Media Lab/UnB, foi o EVO\_CIRCUITO (**Figuras 37-41**). Este objeto artístico que ocupa uma classificação entre escultura, instalação e obra computacional, tentava explorar a capacidade de aleatoriedade em obras de arte computacionais.



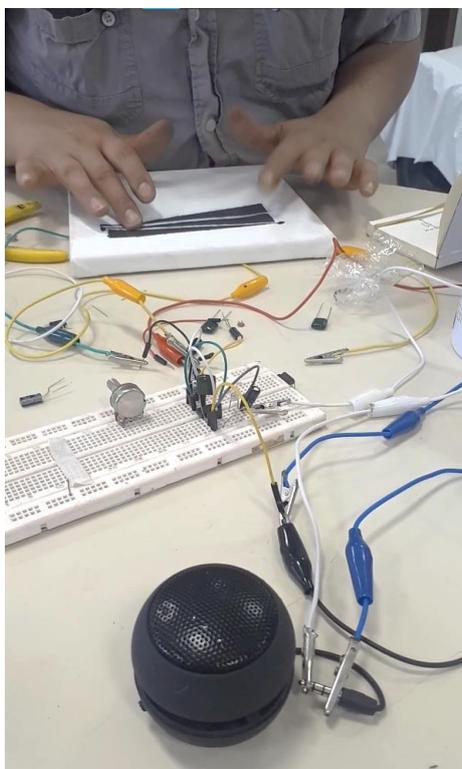
**Figura 37:** EVO\_CIRCUITO montado em Buenos Aires durante o Simpósio Internacional de Mídias Digitais Interativas. Fonte: Acervo próprio (2018).



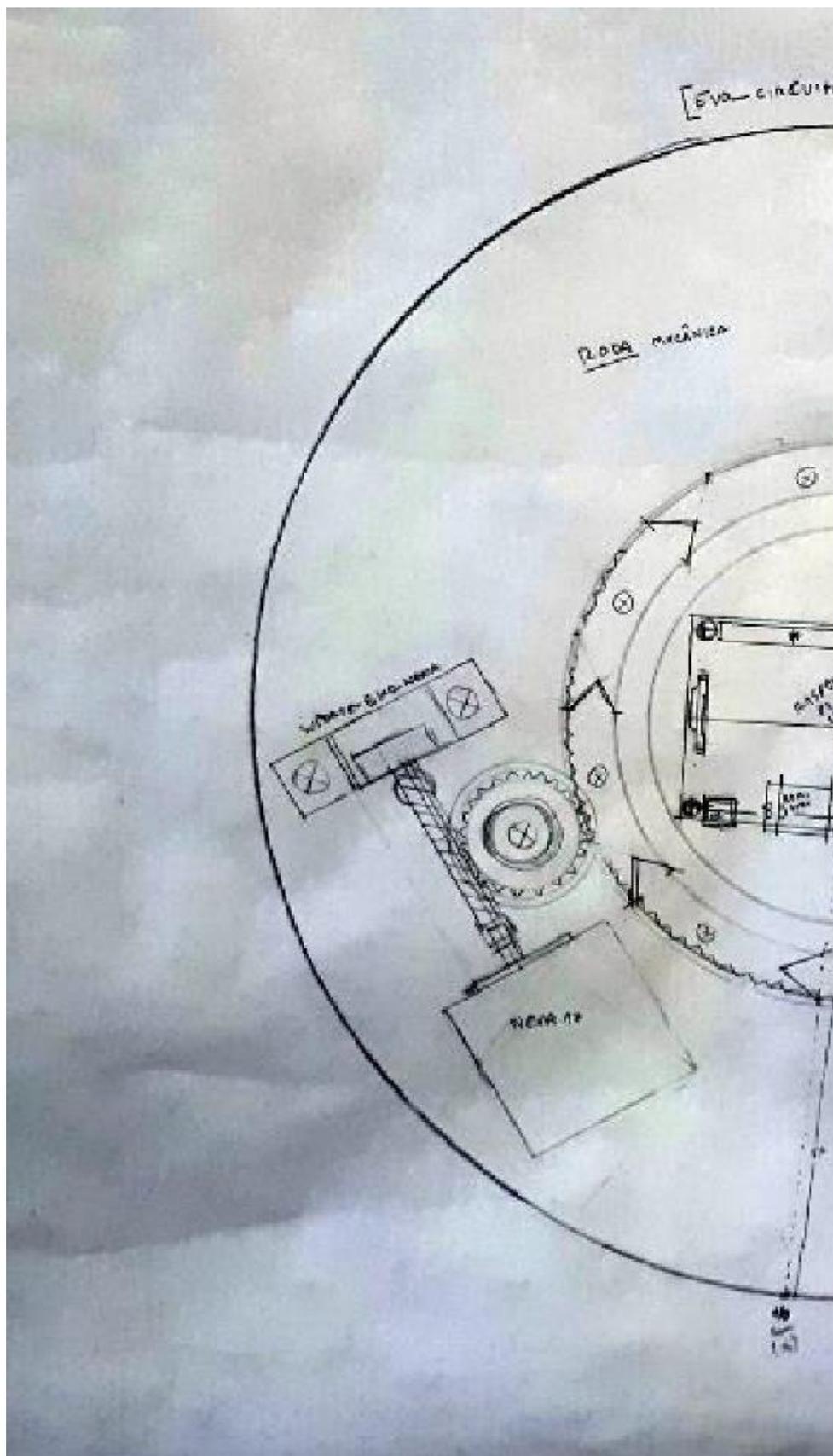
**Figura 38:** Parte interna do corpo do EVO\_CIRCUITO, parte superior. Podemos visualizar as trilhas compostas de papel alumínio. Fonte: Acervo próprio.



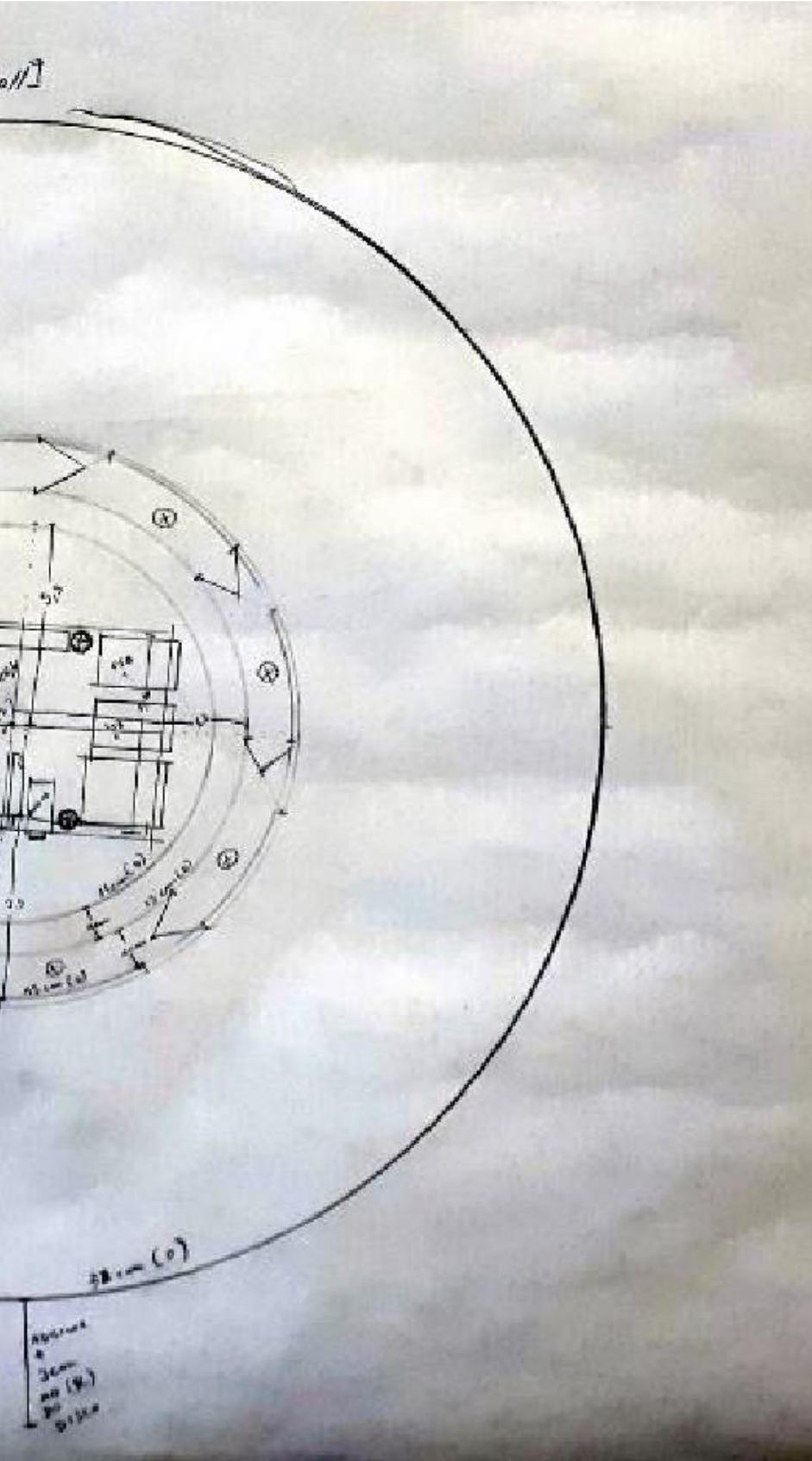
**Figura 39:** Parte interna do corpo do EVO\_CIRCUITO, parte inferior. Esta parte do corpo fica exposta verticalmente em uma superfície lisa. O motor teve que compensar o peso extra da peça pela gravidade. Fonte: Acervo próprio.



**Figura 40:** Prahlada Hargreaves testando as reações das tintas condutivas em tela de pintura. Fonte: Acervo próprio. (2018)

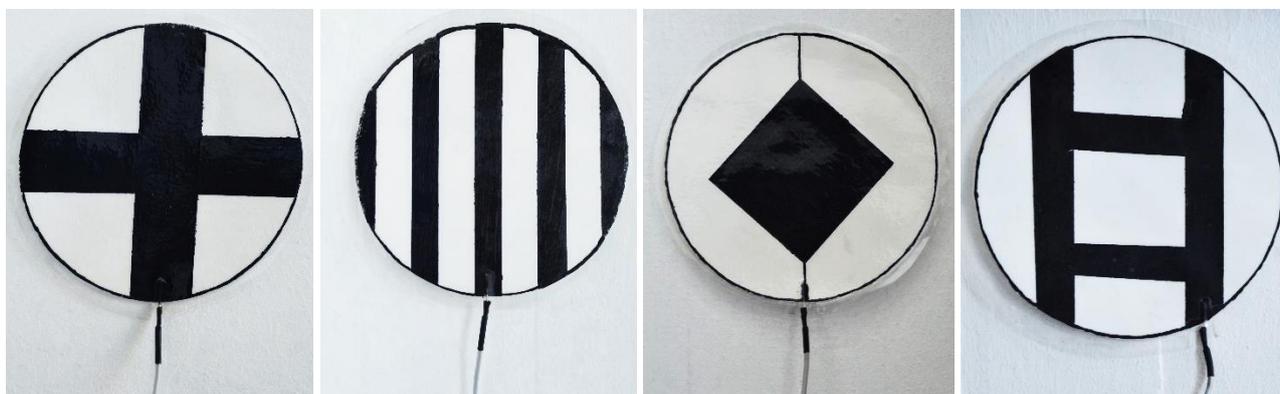


**Figura 41:** Diagrama do inicial do corpo do EVO\_CIRCUITO (HARGERAVES, 2018). Fonte: Acervo próprio.



O primeiro rumo que decidimos explorar foi a experiência com tintas condutivas. Iniciamos a pesquisa com um kit básico da “*Bare Conductive*”<sup>60</sup> adquirido por nós anteriormente. Realizamos o teste com diferentes suportes (papel de gramaturas variadas, papelão, tecido para pintura, etc), o objetivo do teste era determinar os pontos de resistência da tinta e o comprimento máximo da linha sem haver a perda da resistência elétrica.

Após, iniciamos a pesquisa para elaborar uma tinta autoral, chegamos a uma composição utilizável que envolve uma porcentagem de base para unha e pó de grafite. Todavia, o uso desta tinta não permitiu que realizássemos a ideia inicial fazer com que cada traço fosse como um *dimmer*, dispositivos que controlam uma variedade de intensidade de uma corrente elétrica. Nesse ponto o conceito dos PAD's como sensores capacitivos ficou evidente para nós e queríamos explorar seu uso.



**Figura 42:** Os quatro PAD's utilizados no EVO\_CIRCUITO. Fonte: Acervo próprio.

Após os resultados dos PAD's, iniciou-se uma série de reuniões para determinar os rumos criativos seguindo a metodologia do serendipitismo (VENTURELLI, 2017). O conceito a ser explorado na obra era a das múltiplas possibilidades de randomização dos resultados de uma obra computacional.

Assim, chegamos a um consenso sobre as formas como a obra iria explorar as variáveis randômicas: uma aleatoriedade mecânica e analógica, a aplicação de um código genético e as variações dos contatos humanos junto aos PAD's. Foi assim que se chegou ao formato final da obra, e como se tratava da primeira experiência dentro do doutorado para a criação de uma obra, priorizamos analisar a forma como cada membro do grupo realizava a documentação, sem nenhum tipo de orientação ou supervisão.

Os diagramas foram confeccionados após a exibição da obra em Buenos Aires – Argentina, que por não ter a documentação técnica necessária, não houve preparo para contornar os

<sup>60</sup> **Nota do autor:** Empresa inglesa que produz as tintas condutivas e microcontroladores baseados em Arduino que utilizam desses contatos sem fio para seus inputs e outputs.

problemas para a exibição e assim, a obra não foi executada. Abaixo segue a ficha técnica da obra com a descrição da mesma. Em seguida, apresentamos os diagramas explicando cada um dos níveis.

## **B) FICHA TÉCNICA**

### **Autoria:**

Teófilo Augusto da Silva, Prahlada Hargreaves, Artur Cabral Reis, Lorena Ferreira e Suzete Venturelli.

### **Técnica/material:**

Técnicas eletromecânica e computacional. Madeira, Plástico ABS, Parafusos, Cabos elétricos, Arduino Nano, Raspberry Pi 3, Tinta Condutiva.

### **Dimensões:**

Obra tridimensional: 15 cm (h) x 45 cm (L) x 45 cm (P) (peça central), espaço a ser utilizado deve ser em uma parede onde será pendurado (necessitará da colocação de um parafuso), e os pads com sensores capacitivos serão ou pintados ou aplicados à parede e os fios estarão abertos. Necessita: saída de energia 220v / 60Hz, Monitor HDMI apenas para configuração da obra e ambiente em que o som possa ser continuamente reproduzido.

### **Descrição da Obra:**

A obra EVO\_CIRCUITO consiste em uma obra híbrida pictórica, escultórica e sonora, composta por elementos como madeira, componentes eletrônicos, fios, fitas e tintas condutivas, e microcontroladores. A parte central da obra contém duas rodas sobrepostas e a parte inferior (base) é composta por um motor com engrenagens. No centro das engrenagens encontra-se um Raspberry Pi 3 ligado a uma placa Arduino Mini. A parte superior (tampa) tem em seu verso (parte interna) uma trilha de alumínio cortada em espaços diferentes que determinam uma duração e uma ativação de uma sequência de programas analógicos randômicos.

O Raspberry Pi é responsável por rodar os programas de interpretação dos sinais recebidos dos contatos feitos pela tinta condutora, o Sonic Pi, que é responsável por emitir o som base, e um software evolutivo que altera a forma como o som se comporta. As fitas, fios e tinta condutiva, são aplicadas de formas diferentes, baseada na textura da parede, do design da obra no espaço e das formas de interação previstas. A tinta condutora sofre desgaste com o uso tátil, e deste modo, todas as fórmulas perdem a resistência conforme a distância entre os pontos fica maior. Por esta

razão, resolvemos os problemas utilizando para os grandes espaços a serem percorridos, o uso das fitas ou fios, enquanto os mecanismos de interação ficam com desenho feitos com a tinta. Sendo assim, o círculo com os microcontroladores e o motor ocupa a parte central da obra e dele partem ramificações como braços de um polvo ou trilhas em uma mata. Assim, seu aspecto orgânico é orgânico. Como cada parede ou superfície é única, o EVO\_CIRCUITO, sempre será diferente a cada exibição.

Baseado na concepção de Couchot (COUCHOT, 2003) sobre a arte permutacional, o EVO\_CIRCUITO produz um som que vai se alterando considerando três fatores: (1) aleatoriedade mecânica: a parte central do objeto era basicamente um grande sanduíche de madeira em que o recheio eram os componentes como um Arduino mini, um Raspberry Pi 3, a fonte e uma série de sensores feitos com fios e parafusos. Esses sensores mais rudimentares eram responsáveis pela aleatoriedade de efeitos causados na música tema da obra. Conforme o disco girava em uma velocidade mínima (usou-se um rotor de impressora 3D para controlar o movimento), a disposição dos parafusos ativava uma das possibilidades de permutação do objeto, de modo que o *iterator*, quando passava para as outras possibilidades de aleatoriedade, nunca pudesse imaginar quais seriam os efeitos que sua atividade faria no som. (2) Os sensores táteis externos: os discos externos, esticados como braços de um polvo, eram sensores capacitivos. Estes reagiam de maneira diferente para cada corpo que o tocasse, além da própria pressão aplicada que podia também variar. (3) O código responsável pela geração da música (produzido em SONIC PI) possui mecanismos de aleatoriedade já inseridos.

Estes três mecanismos tornavam a música aleatória, e ao mesmo tempo, como o código fazia sua própria alteração, ela era a narrativa da própria existência do objeto. A exposição desta obra de maneira funcional no evento não aconteceu, pois, mesmo realizando testes antes da ida e conferindo a integridade dos componentes o Raspberry Pi não dava sinal de *Boot*. Ao retornar para o Brasil e testá-lo, funcionou perfeitamente, restando a dedução de se tratar de um problema de equivalência entre a fonte e a rede energética argentina.

### **C) Documentação**

Como mencionamos anteriormente, esta se tratava de uma primeira experiência pessoal na criação de uma obra conjunta no período do curso de doutorado e por isso adotamos uma postura menos rigorosa na documentação, tentando observar como os outros artistas documentavam sua própria parte na criação.

A grande maioria realizava pequenos esboços que eram feitos de maneira relaxada sem qualquer intuito de tornar aquele rascunho em documento. Isso acontecia tanto com o design da obra, quanto com os códigos de programação que foram utilizados. A Figura 40 representa uma exceção nos documentos criados diante do rascunho.

Esta obra foi essencial para a criação do protocolo, já que sua documentação foi montada posteriormente a criação da mesma, fornecendo assim feedback passo a passo sobre como cada parte da mesma deveria ser documentada. Uma importante atividade foi a de revisar e acrescentar os comentários nos códigos utilizados na obra. Todos os arquivos dos códigos estão armazenados em mais de um local e com mais de um dos membros da equipe.

Abaixo, segue uma descrição sobre como foi implementado o protocolo para o EVO\_CIRCUITO.

#### D) Aplicação Protocolo

<b>1ª Etapa</b>	Como informado anteriormente, os artistas não foram instruídos a guardarem os rascunhos como estratégia de observação. Contudo, após, conseguimos coletar na sala do Media Lab/UnB alguns destes rascunhos. Além disso, algumas partes do processo foram fotografadas, mas nenhuma metodologia foi utilizada para tanto.
<b>2ª Etapa</b>	Esta etapa não fez parte do processo de aplicação do protocolo, uma vez que a criação está entre a produção de um laboratório de uma universidade pública, sendo sua responsabilidade da coordenação do laboratório e as instâncias superiores.
<b>3ª Etapa</b>	A criação dos diagramas ocorreu junto com a adequação do modelo C4 ao modelo C7Art. Foi por meio desta elaboração que percebemos a necessidade de criar uma representação da relação íntima que o hardware e o software estabelecem em uma obra de arte computacional. Abaixo serão apresentados os diagramas com uma breve explicação sobre cada um.
<b>4ª Etapa</b>	O hardware não pôde ser preservado da maneira que estava, uma vez que as peças serviriam a outros propósitos. Todavia toda a descrição dos documentos remete a possibilidade de recriação desta obra. Os arquivos foram armazenados em formato aberto de texto e disponibilizados em repositórios.

*Tabela 2: Etapas de implantação do protocolo. Fonte: Arquivo próprio.*

#### E) Diagramas dos 7 níveis

Abaixo podemos ver cada um dos diagramas construídos para representar o EVO\_CIRCUITO. Entre dois diagramas, disponibilizamos uma breve análise sobre a forma que visualizamos e representamos cada nível, como um possível auxílio para qualquer um que quisesse utilizar este protocolo e seu modelo de 7 níveis.

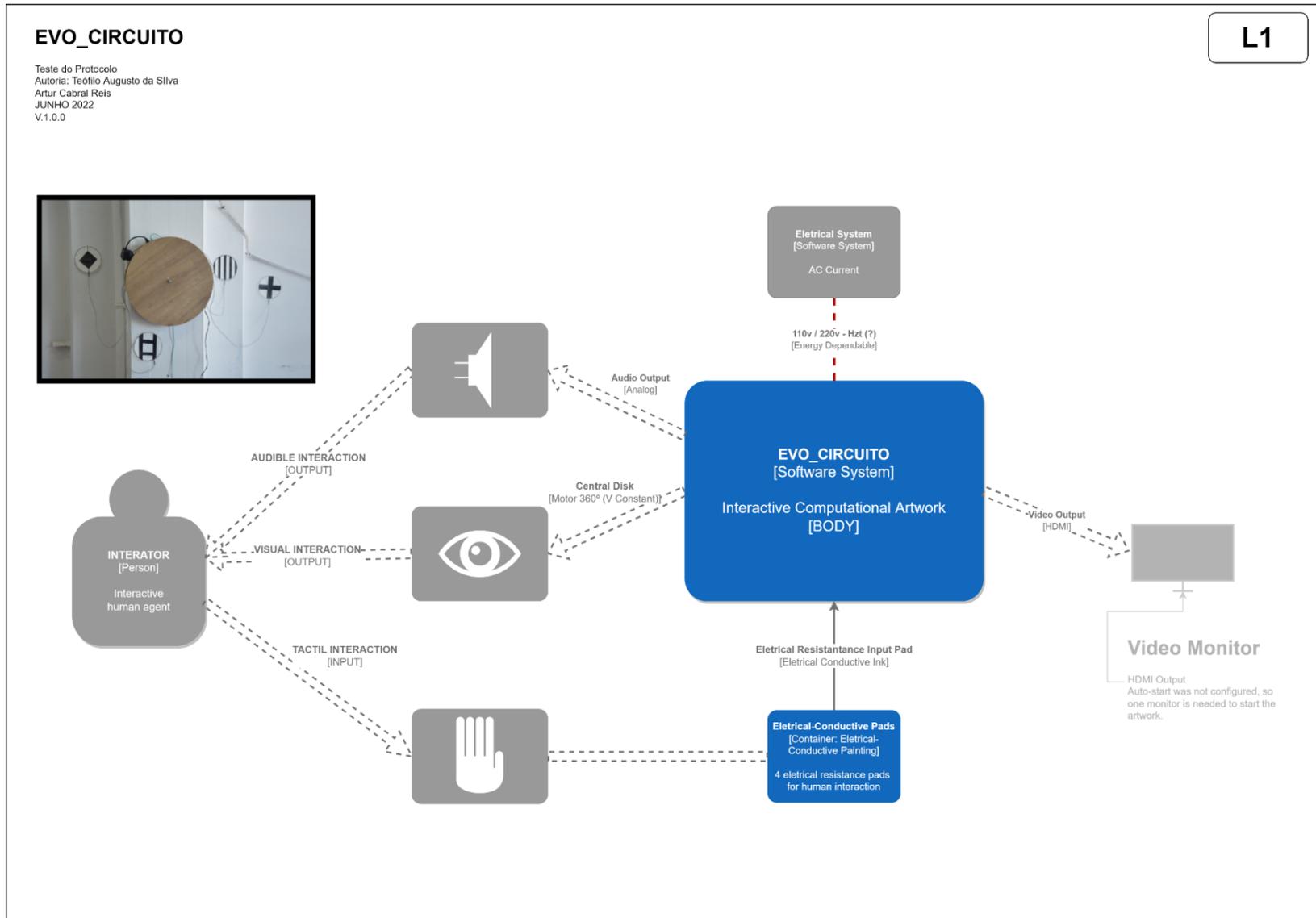


Figura 43: Diagrama EVO\_CIRCUITO nível L1. Fonte: Acervo próprio. (2022)

### A) Diagrama L1 – Instrução Poética (Figura 43)

O diagrama L1 foi elaborado mantendo o corpo do EVO\_CIRCUITO ao centro, caracterizando os PAD's externos ao elemento central da obra. Neste nível do diagrama, fazemos a representação de cada uma das interações sensoriais estabelecidas entre o *interator* e a obra como um systema.

Os componentes cuja atuação independe dos artistas foram representados com a cor cinza, demonstrando igualmente que já se encontram em sua forma mais abstrata, não havendo necessidade de detalhar em subníveis.

Os traços largos e pontilhados serviram para indicar as linhas de input e output da obra. A linha vermelha e tracejada, indica a alimentação elétrica. A linha contínua e preta vinda dos PAD's para o corpo do EVO\_CIRCUITO realiza o input dos dados analógicos.

### B) Diagrama L2 – Objeto-Sistema (Figura 44)

Este diagrama aprofunda-se nos funcionamentos dos componentes de hardware no interior do corpo do EVO\_CIRCUITO. A iconografia continua a mesma, porém neste ponto as interações sensoriais ficam em segundo plano.

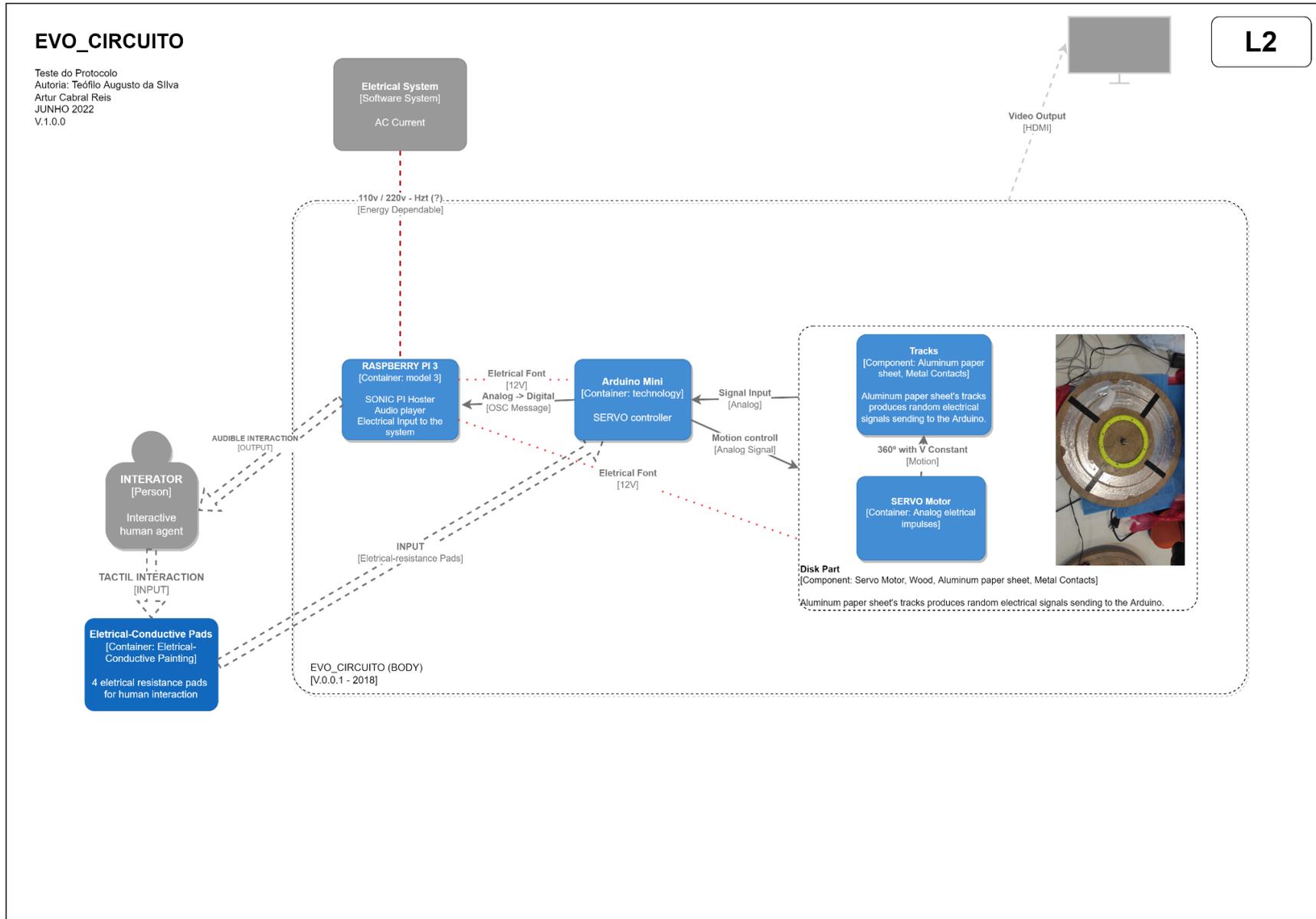


Figura 44: Diagrama EVO\_CIRCUITO nível L2. Fonte: Acervo próprio (2022)

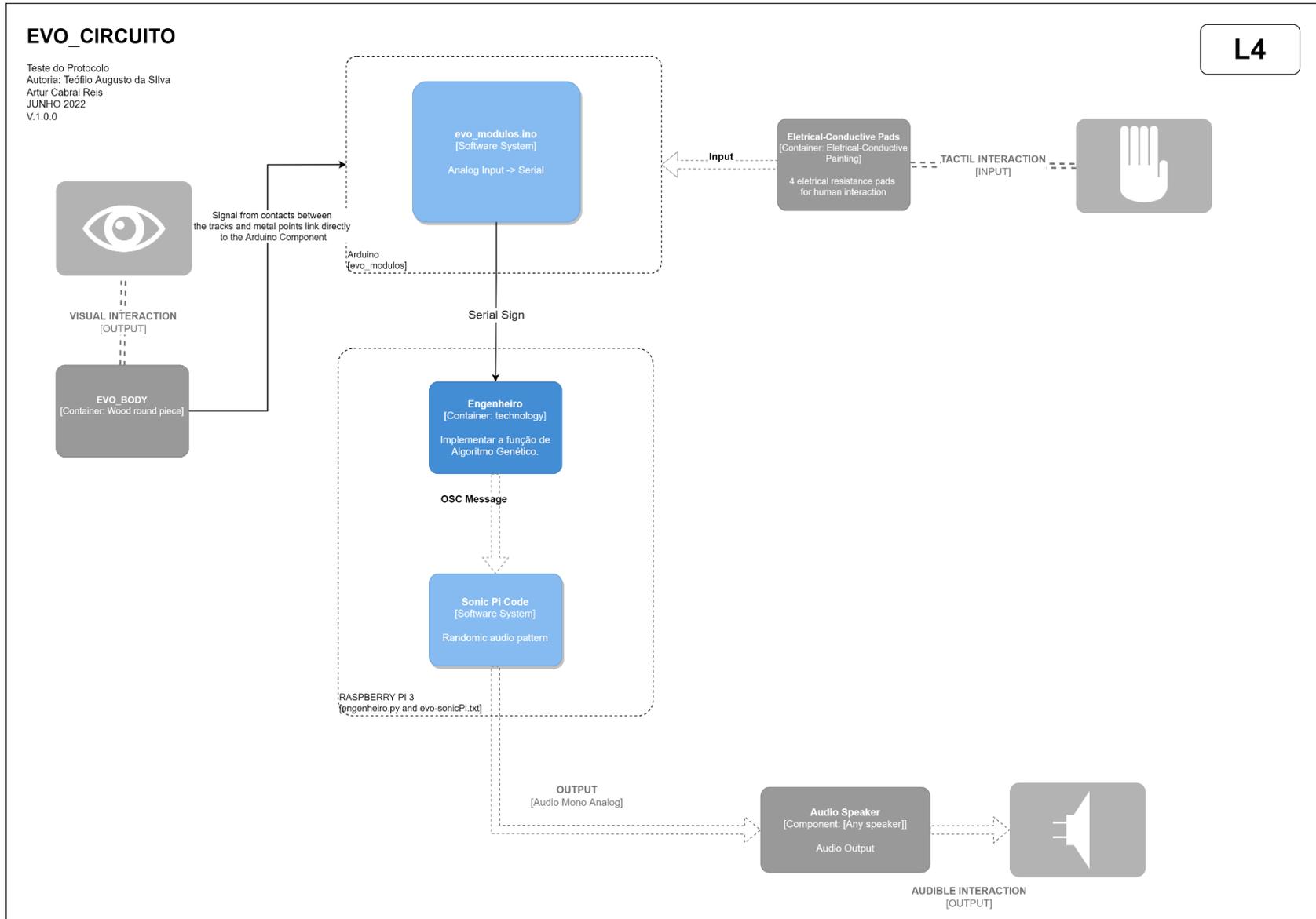


Figura 45: Diagrama EVO\_CIRCUITO nível L4. Fonte: Acervo próprio (2022)

### **C) Diagrama L4 – Contexto do Sistema (objeto-símbolo) (Figura 45)**

A falta do diagrama representando o nível L3 é proposital e se dá pelo fato de não ter sido feita uma documentação sobre a forma como cada um dos componentes foi conectado. Assim, compreendemos que esta é uma perda que terá de ser contornada no futuro.

No L4, o diagrama se concentra em estabelecer a relação entre hardware e software. Nele é possível perceber que os componentes de hardware que foram representados com cor azul mais escuro no L2, reaparecem aqui como um retângulo de linhas tracejadas. E dentro, demonstrando estarem contidos nestes, estão os softwares que estão sendo executados.

No caso, são três os softwares escritos para esta obra e estes aparecerão nos diagramas do L7.

### **D) Diagrama L5 – Domínio (Figura 46)**

O L5 se compromete a descrever a funcionalidade de cada um dos softwares envolvidos na obra. Assim é possível ver que cada um deles aponta para uma função específica do software.

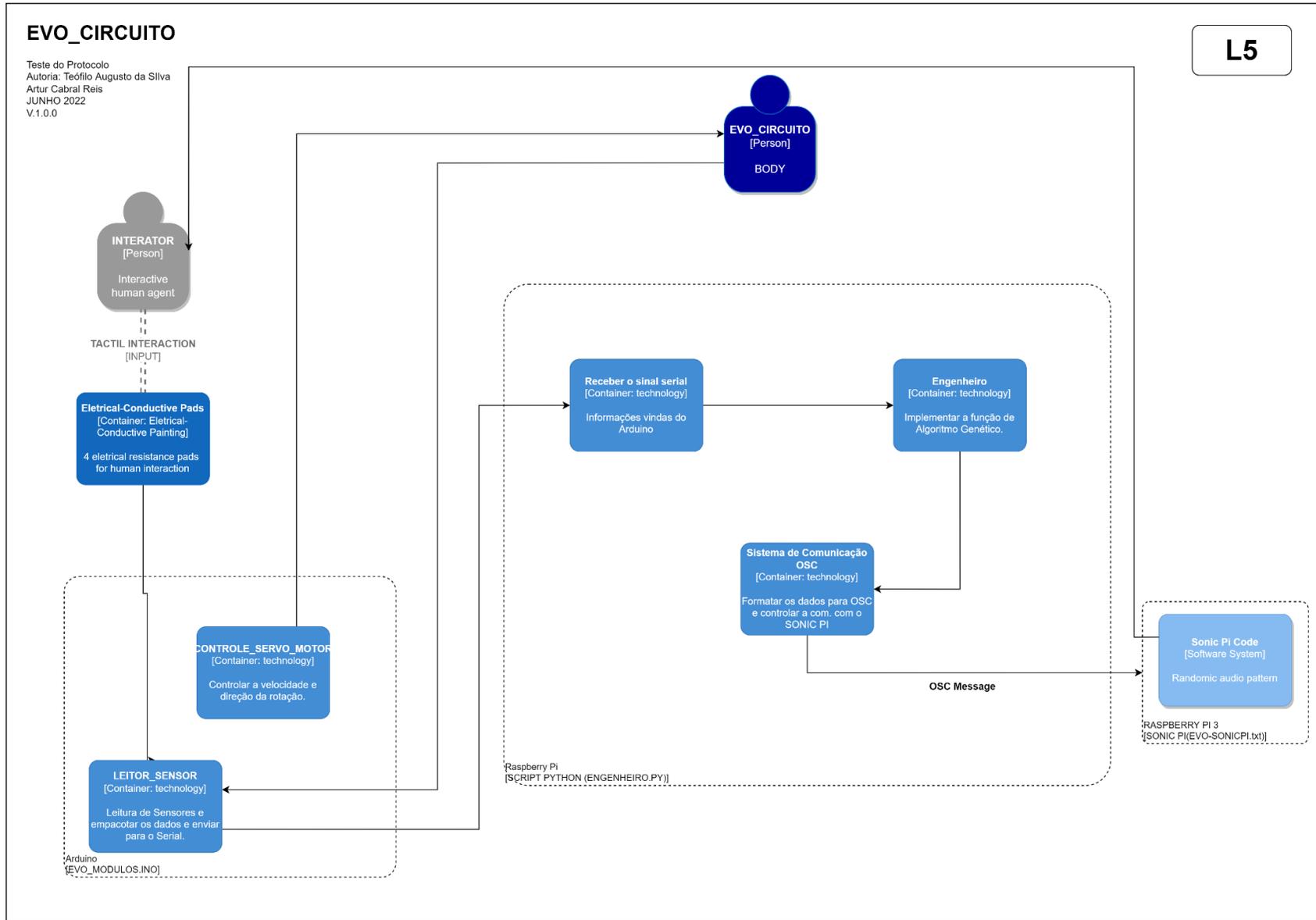
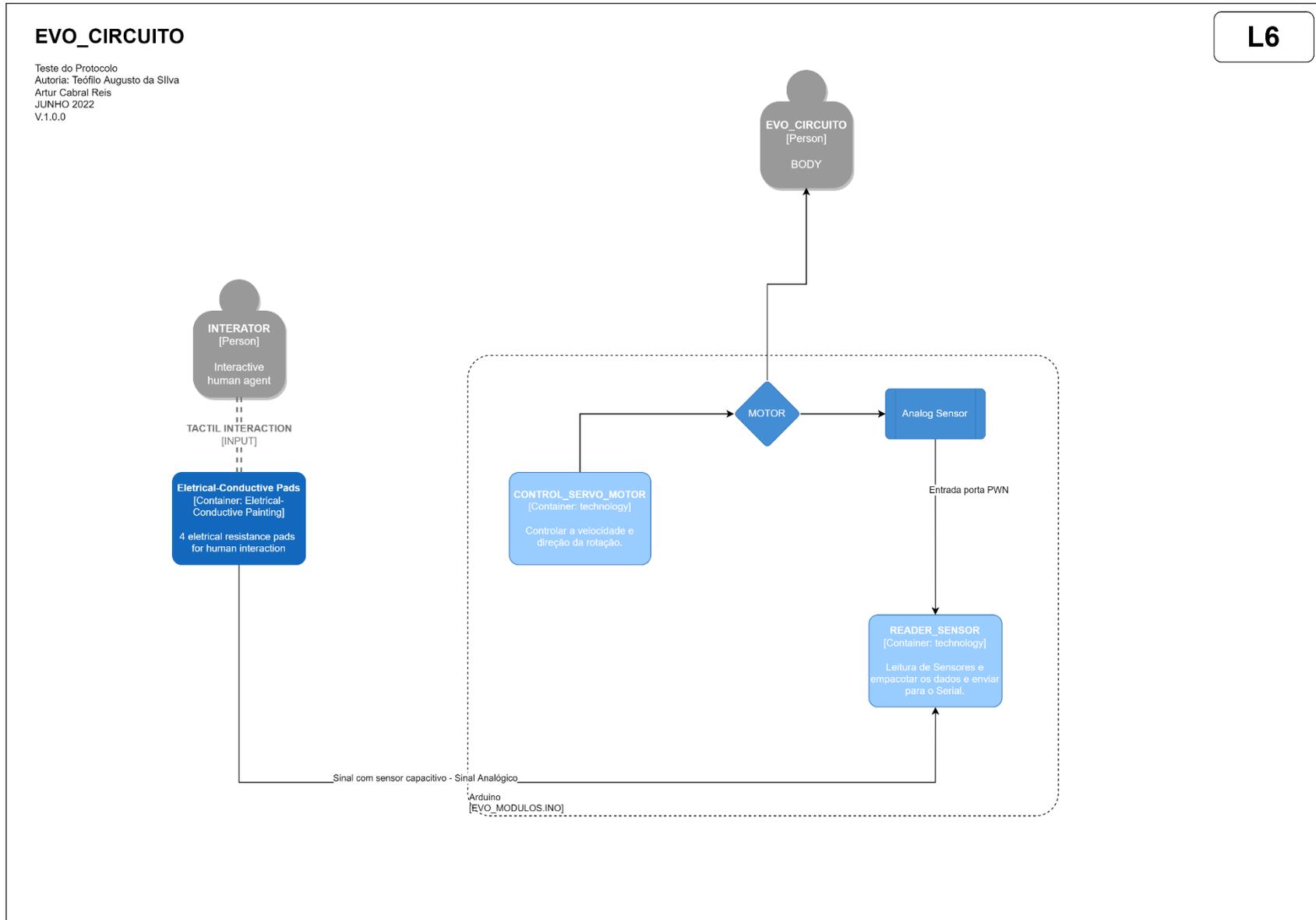


Figura 46: Diagrama EVO\_CIRCUITO nível L5. Fonte: Acervo próprio (2022)



**Figura 47:** Diagrama EVO\_CIRCUITO nível L6. Fonte: Acervo próprio (2022)

**E) Diagrama L6 – Componentes do software (Figura 47)**

Neste nível, ocorre o aprofundamento nas questões arquitetônicas do software. O corpo do EVO\_CIRCUITO aqui foi representado como um indivíduo, tal como o interator, representando-o como agente de atuação já não como sistema (pelo menos neste nível).

Como as ações dos softwares engenheiro.py e evoSonicPi já seriam suficientemente descritas no nível do código, então não haveria necessidade de se elaborar o L5 para os mesmos, ficando apenas o EVO\_MODULOS.ino.

Neste podemos ver as funções relativas ao código inserido no Arduino que compunha o EVO\_CIRCUITO. A relação deste com o motor e com os sensores condutivos.

**F) Diagrama L7a – Código – engenheiro.py (Figura 48)**

Os diagramas do L7 são diagramas de atividades do código. Conjuntamente com o código comentado, eles servem como guia sobre quais as funções imaginadas pelos artistas e suas atuações sensoriais.

Engenheiro foi nomeado assim por ser o programa central do EVO e responsável pela engenharia genética que traz uma camada de imprevisibilidade ao software.

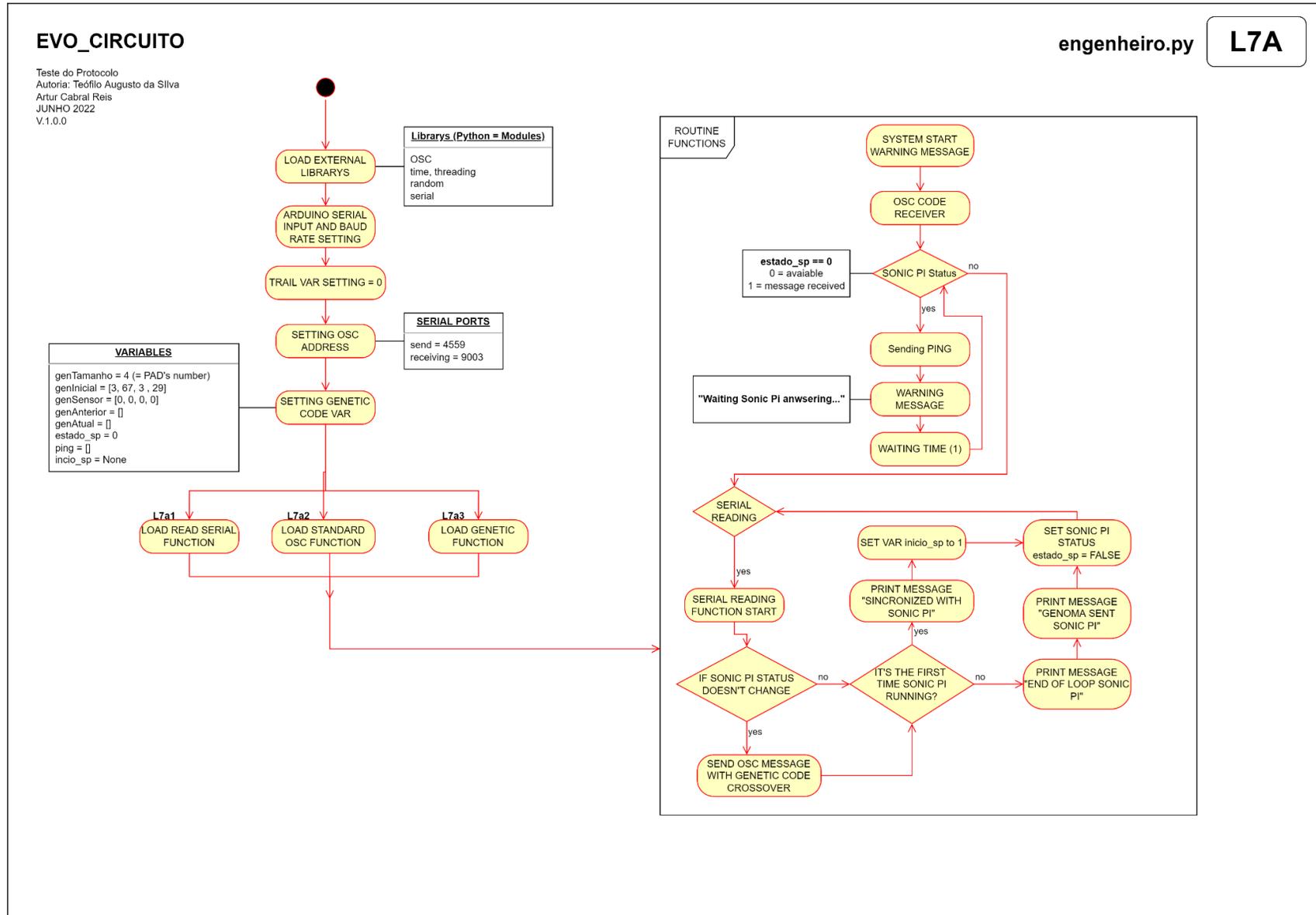


Figura 48: Diagrama EVO\_CIRCUITO nível L7a. Fonte: Acervo próprio (2022)

# EVO\_CIRCUITO

Teste do Protocolo  
 Autoria: Teófilo Augusto da Silva  
 Artur Cabral Reis  
 JUNHO 2022  
 V.1.0.0

engenheiro.py

L7a1

L7a

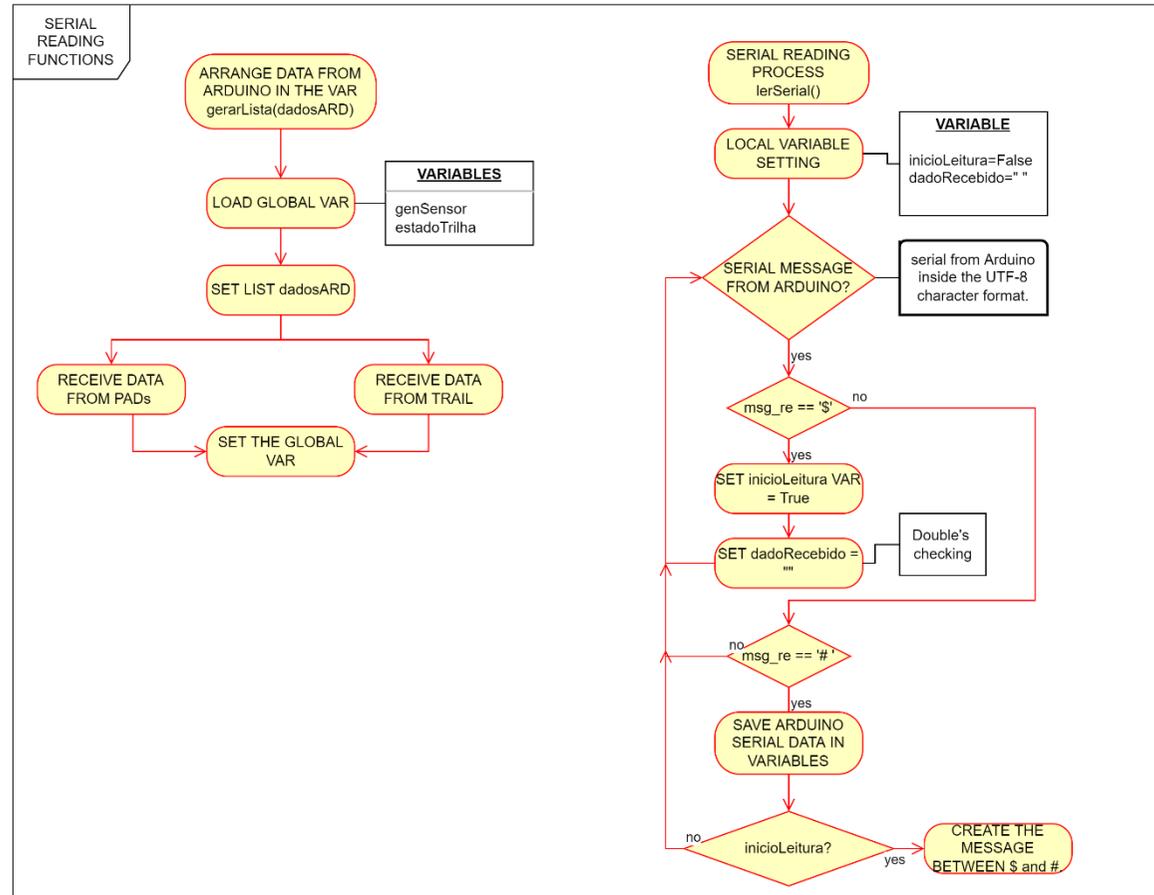


Figura 49: Diagrama EVO\_CIRCUITO nível L7a1. Fonte: Acervo próprio (2022)

**G) Diagrama L7a1 – Código – Função de Leitura Serial do engenheiro.py (Figura 49)**

Este diagrama descreve as atividades executadas na função de leitura serial presentes ao código engenheiro.py.

**H) Diagrama L7a2 – Código – Função OSC padrão do engenheiro.py (Figura 50)**

Este diagrama descreve as atividades executadas na função OSC padrão presente ao código engenheiro.py.

## EVO\_CIRCUITO

Teste do Protocolo  
 Autoria: Teófilo Augusto da Silva  
 Artur Cabral Reis  
 JUNHO 2022  
 V.1.0.0

engenheiro.py

L7a2

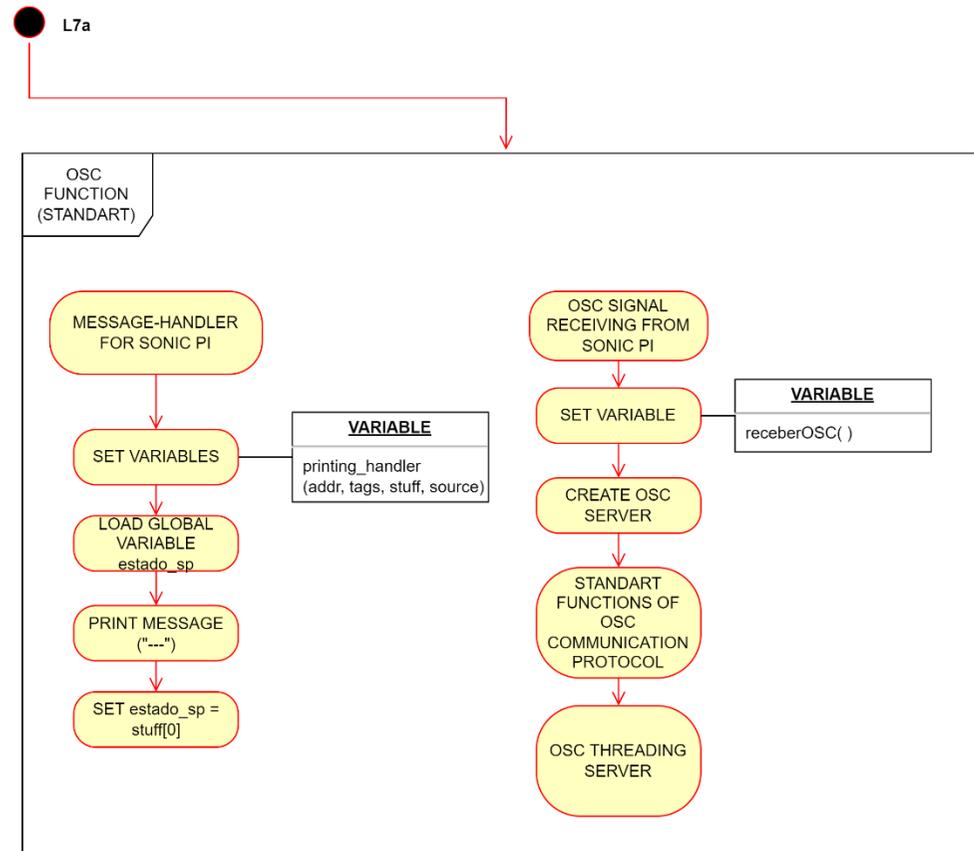


Figura 50: Diagrama EVO\_CIRCUITO nível L7a2. Fonte: Acervo próprio (2022)

EVO\_CIRCUITO

Teste do Protocolo  
 Autoria: Teófilo Augusto da Silva  
 Artur Cabral Reis  
 JUNHO 2022  
 V.1.0.0

engenheiro.py

L7a3

L7a

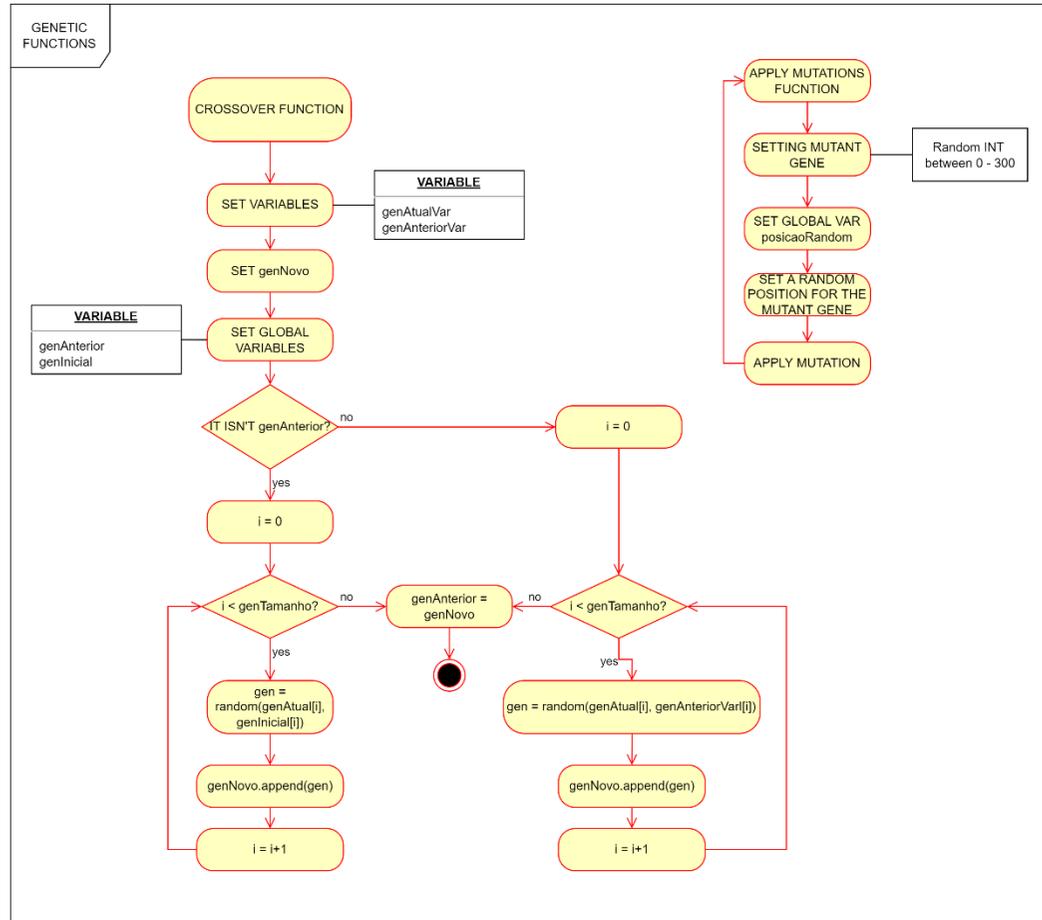


Figura 51: Diagrama EVO\_CIRCUITO nível L7a3. Fonte: Acervo próprio (2022).

**I) Diagrama L7a3 – Código – Função de Genética do engenheiro.py (Figura 51)**

Este diagrama descreve as atividades executadas pela função genética implementada ao código engenheiro.py.

**J) Diagrama L7b, L7b1, L7b2, L7b3 e L7b4 – Código – evo-sonicPI (Figura 52 a 56)**

O diagrama L7b, descreve as atividades realizadas pelo código executado pelo SonicPi, um software de código aberto que também é nativo do Raspberry Pi, um dos microcontroladores que fazem parte do corpo do EVO\_CIRCUITO.

O programa evo-sonicPI possui duas partes distintas: 1) apresentada no diagrama L7b está a parte do programa relativa a comunicação com os outros softwares, recebendo as informações do engenheiro.py e do evo\_modulos.ino. 2) o código relativo a obra sonora que é representado nos subníveis L7b, no diagrama apresentamos as atividades utilizando o jargão técnico específico da arte sonora, podendo ser replicado em qualquer outro software para este fim que não o SonicPi.

As informações recebidas pelos outros softwares são tratadas (são colocadas em uma escala padrão e salva em variáveis internas ao programa) e são utilizadas para a parte da obra sonora (letras a, b, c, d, e, que aparecem ao longo do diagrama).

# EVO\_CIRCUITO

Teste do Protocolo  
 Autoria: Teófilo Augusto da Silva  
 Artur Cabral Reis  
 JUNHO 2022  
 V.1.0.0

evo-sonicPi.txt  
 Language = SonicPi

**L7b**

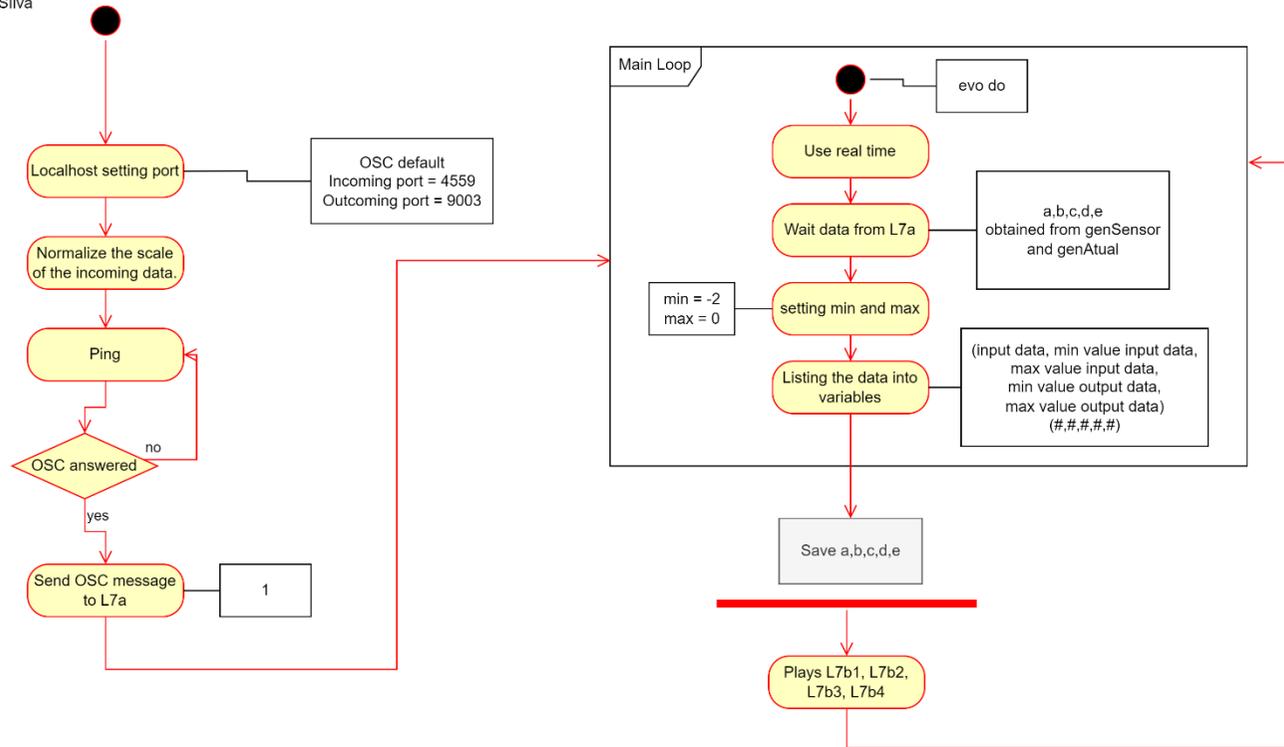


Figura 52: Diagrama EVO\_CIRCUITO nível L7b. Fonte: Acervo próprio (2022)

# EVO\_CIRCUITO

Teste do Protocolo  
 Autoria: Teófilo Augusto da Silva  
 Artur Cabral Reis  
 JUNHO 2022  
 V.1.0.0

**evo-sonicPi.txt**  
 Language = SonicPi

**L7b1**

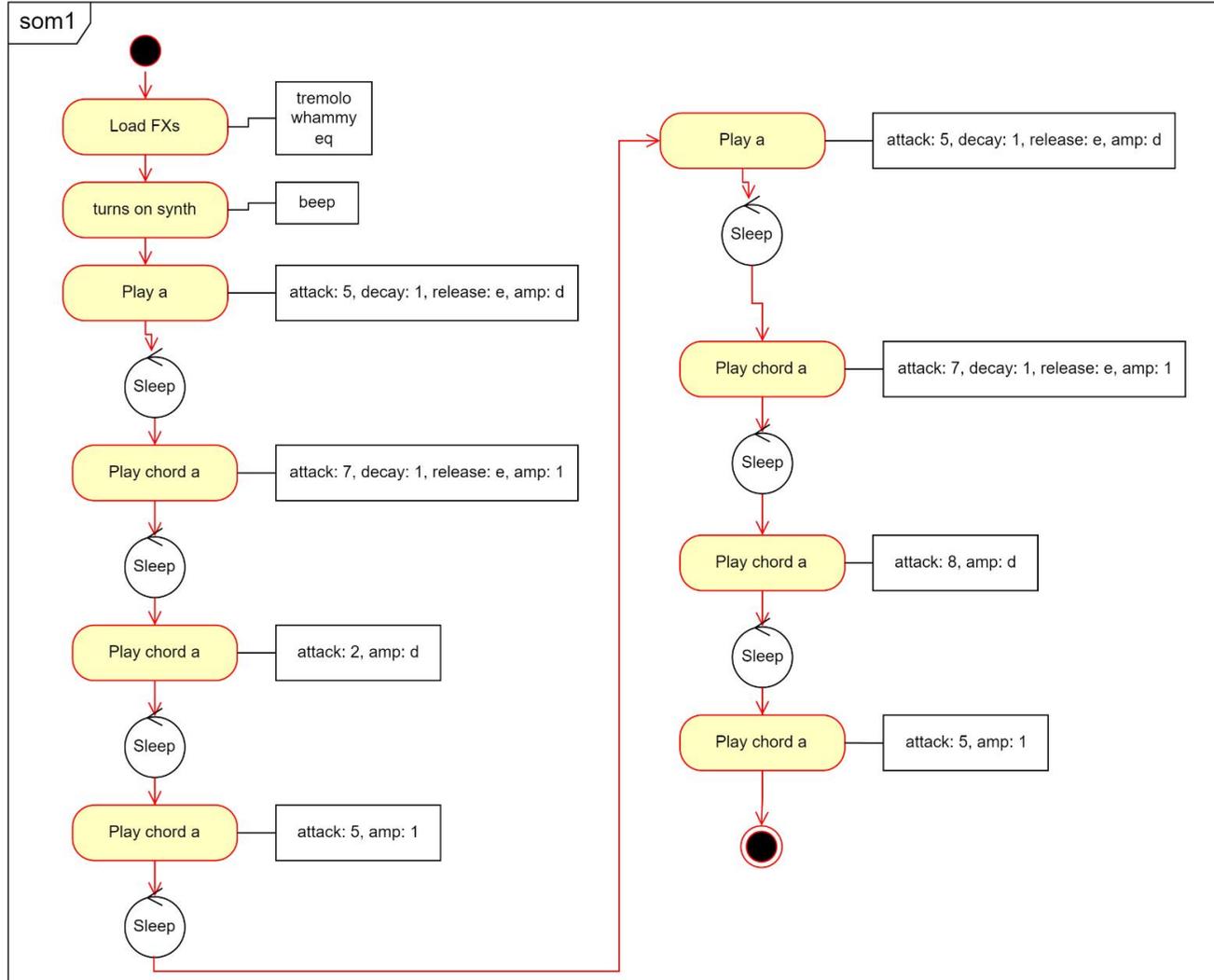


Figura 53: Diagrama EVO\_CIRCUITO nível L7b1. Fonte: Acervo próprio (2022)

# EVO\_CIRCUITO

Teste do Protocolo  
 Autoria: Teófilo Augusto da Silva  
 Artur Cabral Reis  
 JUNHO 2022  
 V.1.0.0

evo-sonicPi.txt  
 Language = SonicPi

L7b2

som2

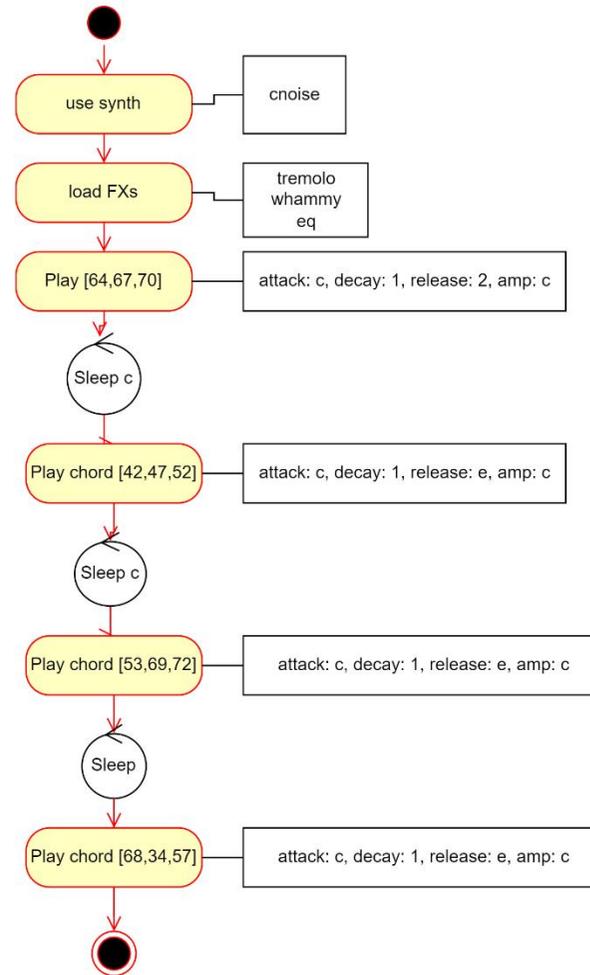
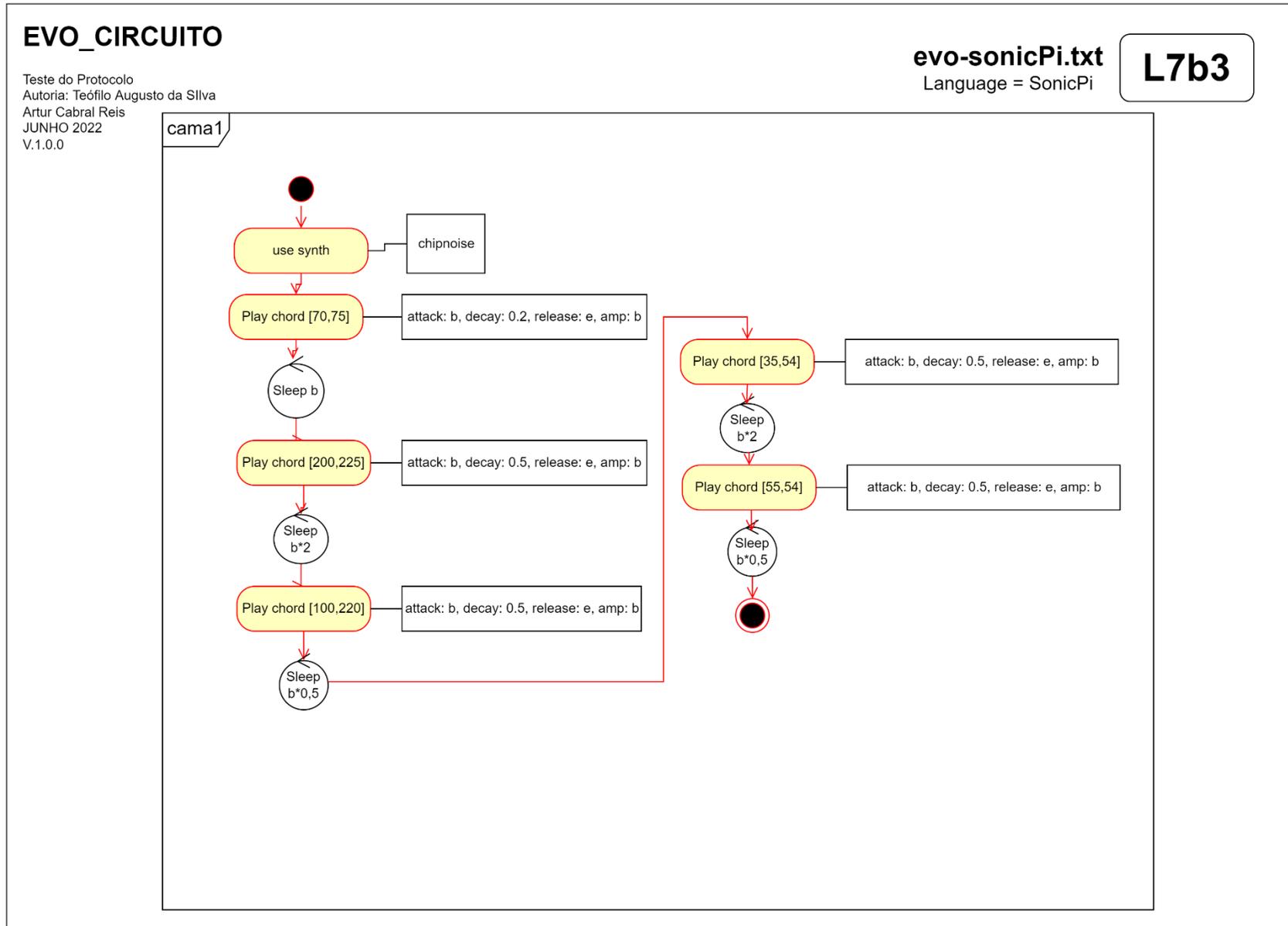


Figura 54: Diagrama EVO\_CIRCUITO nível L7b2. Fonte: Acervo próprio (2022)



**Figura 55:** Diagrama EVO\_CIRCUITO nível L7b3. Fonte: Acervo próprio (2022)

# EVO\_CIRCUITO

Teste do Protocolo  
 Autoria: Teófilo Augusto da Silva  
 Artur Cabral Reis  
 JUNHO 2022  
 V.1.0.0

evo-sonicPi.txt  
 Language = SonicPi

L7b4

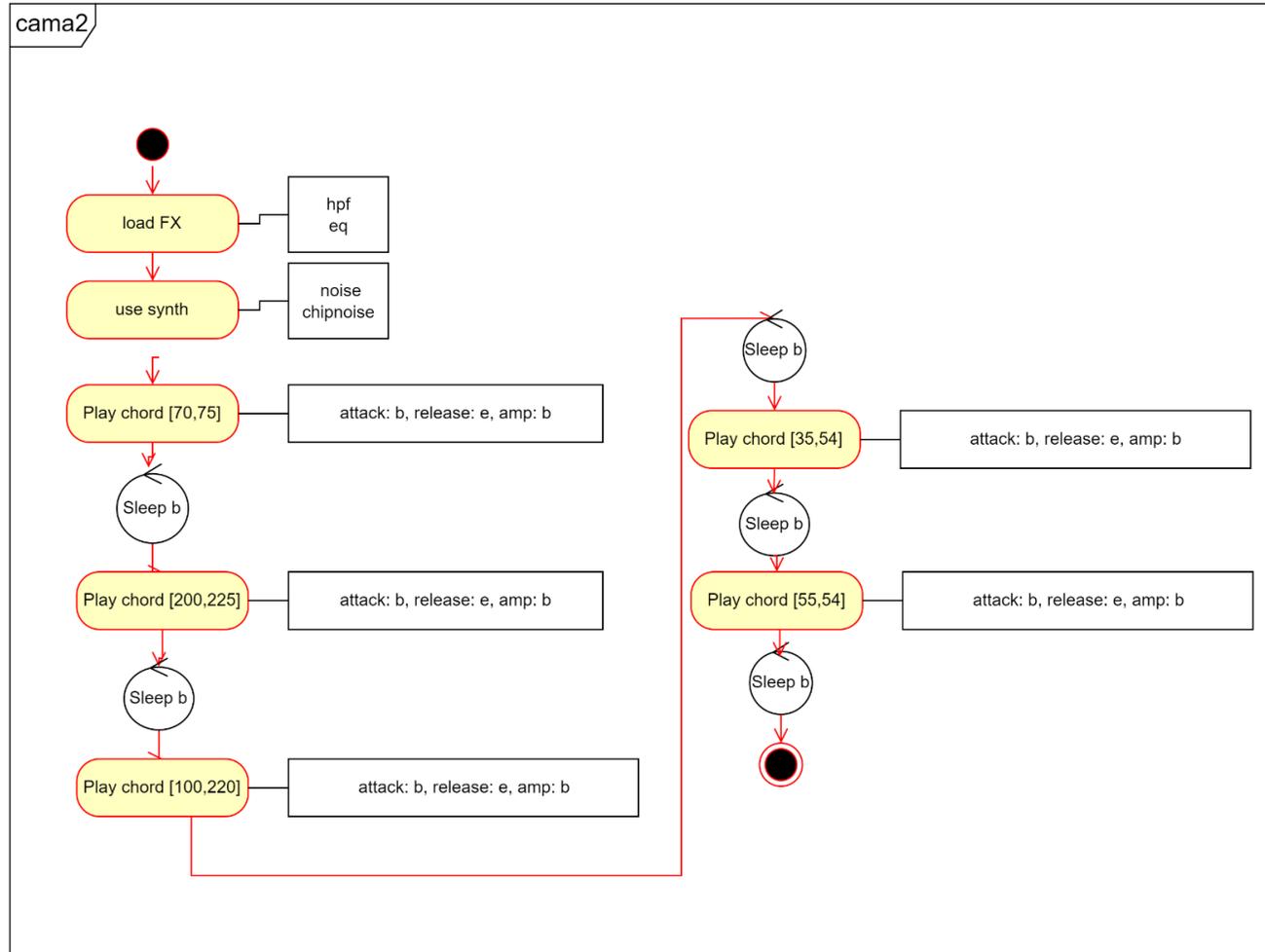


Figura 56: Diagrama EVO\_CIRCUITO nível L7b4. Fonte: Acervo próprio (2022)

## 5.2.2. FUTURE VOICES

### A) Histórico

Future Voices/Zukunftsmusik é uma obra sonora generativa, pensada de maneira colaborativa entre professores e alunos dos cursos de Arte Generativa e Arte Computacional da Universidade de Artes de Berlin (Universität der Künste – Berlin/UdK), formando o coletivo Sociedade para Buscas Não-Triviais (Society for Non-Trivial Pursuits - S4NTP).

No verão de 2020, a CTM Radio Lab publicou um edital de chamada de obras sonoras para ser transmitida. A S4NTP resolveu atender a esta demanda, buscando uma obra participativa e aberta para ser modificada, expandida e criada em novas versões. Partiram então da ideia de uma obra anterior de Hannes Hoelzl, professor e participante do S4NTP, a “*Just a moment of patience*” (jamop) que se trata de uma coleção de falas de indivíduos em línguas diferentes que é apresentada de maneira randômica em um loop infinito.

A proposta Future Voices foi criada no período da pandemia do COVID-19, sendo impossível separar a visão de futuro dos indivíduos e suas insatisfações com as políticas de saúde e controle social estabelecidas até então. Estas narrativas foram aos poucos composto um acervo com mais de 600 contribuições em diversas línguas, narrando diversos aspectos sobre a visão do futuro de cada um dessas vozes que de outra forma “permaneceriam inaudíveis dentro de uma economia de atenção que favorece o barulho, a provocação e teorias da conspiração” (S4NTP, 2020).

Todavia, existe uma crítica na proposta, uma vez que a palavra Zukunftsmusik faz menção a um termo utilizado pelos críticos do compositor e músico Richard Wagner, que desprezavam a composição feitas por ele, colocando que somente no futuro seria possível apreciar esta música. Esse termo é então utilizado na cultura alemã como sinônimo de algo que não é compreensível para aquele momento e pode cair no esquecimento. Esse paradoxo faz da obra Future Voices uma proposta midiática que prospecta o futuro sobre o que nos espera, caso o que desejamos ou acreditamos não fizer parte de um todo.

O projeto também recebeu financiamento da Deutschlandfunk Kultur / [Klangkunst](#), [ORF Kunstradio](#), and [CTM Festival](#).

### B) Ficha Técnica

#### Autoria:

Adel Akram Alameddine, Vinzenz Aubry, Valentina Berthelon, Alberto de Campo, Özcan Ertek, Pedro Ferreira, Juan Pablo Gaviria Bedoya, Bruno Gola, Isak Han, Hannes Hoelzl, HyungJoong Kim, Yun Chu Liang, André Martins, Sujay Mukherjee, Sangwon Nam,

Carlos Andres Ortega, Cagil Ozdemir, Marcel Schwittlick, Ilija Šoršević, Lisa Maria Steppacher, MengXuan Sun, Fang Tsai, ChunLi Wang, Anne Wellmer, Tsingyun Zhang.

**Técnica/material:**

Composição sonora generativa em arte computacional. Software utilizado: SuperColider.

**Dimensões:**

Não compete.

**Descrição da Obra**

A obra Future Voices possui duas partes distintas. A parte principal é um software que reproduz composições que se utilizam de diversas declarações sobre o futuro que foram contribuições de pessoas de diversas culturas e lugares no mundo. A segunda parte são as composições que por si só já são obras únicas. A combinação dos dois programas possibilita a contínua transmissão generativa que já acontece a quase 1 ano e meio.

Em entrevista com o Prof. Dr. Alberto de Campo sobre a obra, ficou evidente que a essência da mesma está na experiência causada pela apreciação da mesma, sem muitas restrições quanto ao suporte. Diferente do EVO\_CIRCUITO, a Future Voices é uma obra totalmente sonora, já houve no mínimo três versões diferentes de suporte, assim, para esta tese iremos considerar o suporte portátil imaginado para exibição da obra em eventos públicos.

**C) Documentação**

A documentação disponível para o Future Voices consiste nos comentários objetivos e descritivos escritos pelo Prof. De Campo nos dois programas que constituem a obra; o projeto escrito para participar do concurso do financiamento; e o website da obra.

## D) Aplicação Protocolo

<b>1ª Etapa</b>	A aplicação deste protocolo se deu como parte da pesquisa para o Doutorado Sanduíche. Não tendo qualquer contato anterior com a obra e com os autores, a obtenção dos documentos partiu de um exercício de pesquisa. Porém, a base das informações aqui listadas são os comentários realizados no código e as entrevistas realizadas com o Prof. Alberto.
<b>2ª Etapa</b>	Em primeira entrevista realizada com o Prof. Alberto, foram feitas as três perguntas iniciais. As respostas quanto a conservação da obra foram positivas e registradas em áudio. A instrução poética foi definida como Experiência de Usuário, sendo assim, o diagrama deve referenciar a alguns itens: se a obra é <i>site specific</i> , se há a necessidade do usuário ter alguma experiência anterior para compreensão da obra ou de como interagir com a mesma, quais os <i>inputs</i> e <i>outputs</i> sensoriais e quais componentes de hardware e software de terceiros foram utilizados.
<b>3ª Etapa</b>	Os diagramas foram criados posterior à escrita do código e beneficiou-se da forma objetiva com que os comentários do código foram redigidos.
<b>4ª Etapa</b>	Os arquivos estão armazenados em mais de um espaço na nuvem e desde sua implementação há quase um ano e meio, é transmitido ininterruptamente de um computador portátil.

**Tabela 3:** Etapas de implantação do protocolo. Fonte: Arquivo próprio

## E) Diagrama dos 7 níveis

Abaixo podemos ver cada um dos diagramas construídos para representar o Future Voices. Entre dois diagramas, disponibilizamos uma breve análise sobre a forma que visualizamos e representamos cada nível, como um possível auxílio para qualquer um que quisesse utilizar este protocolo e seu modelo de 7 níveis.

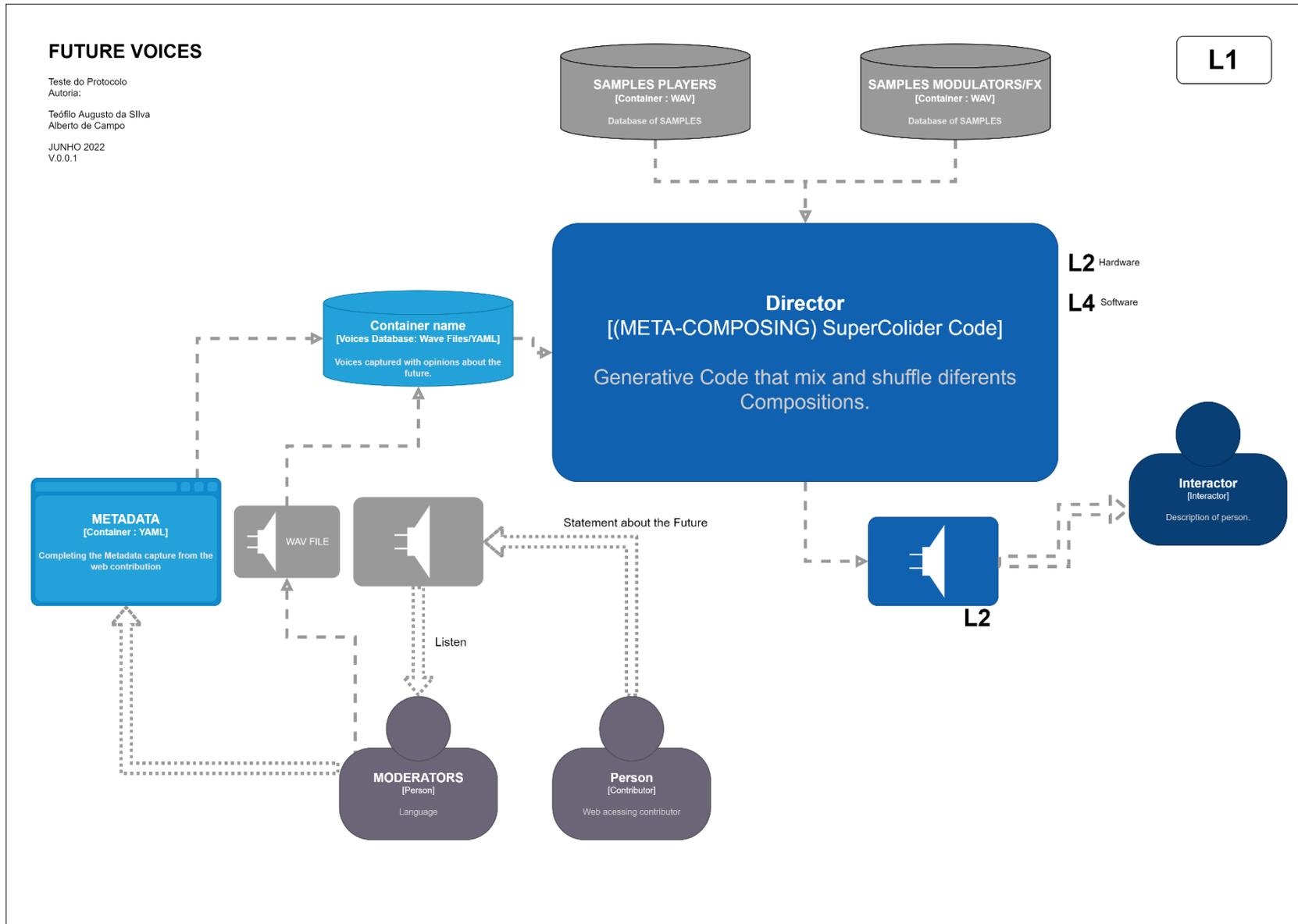


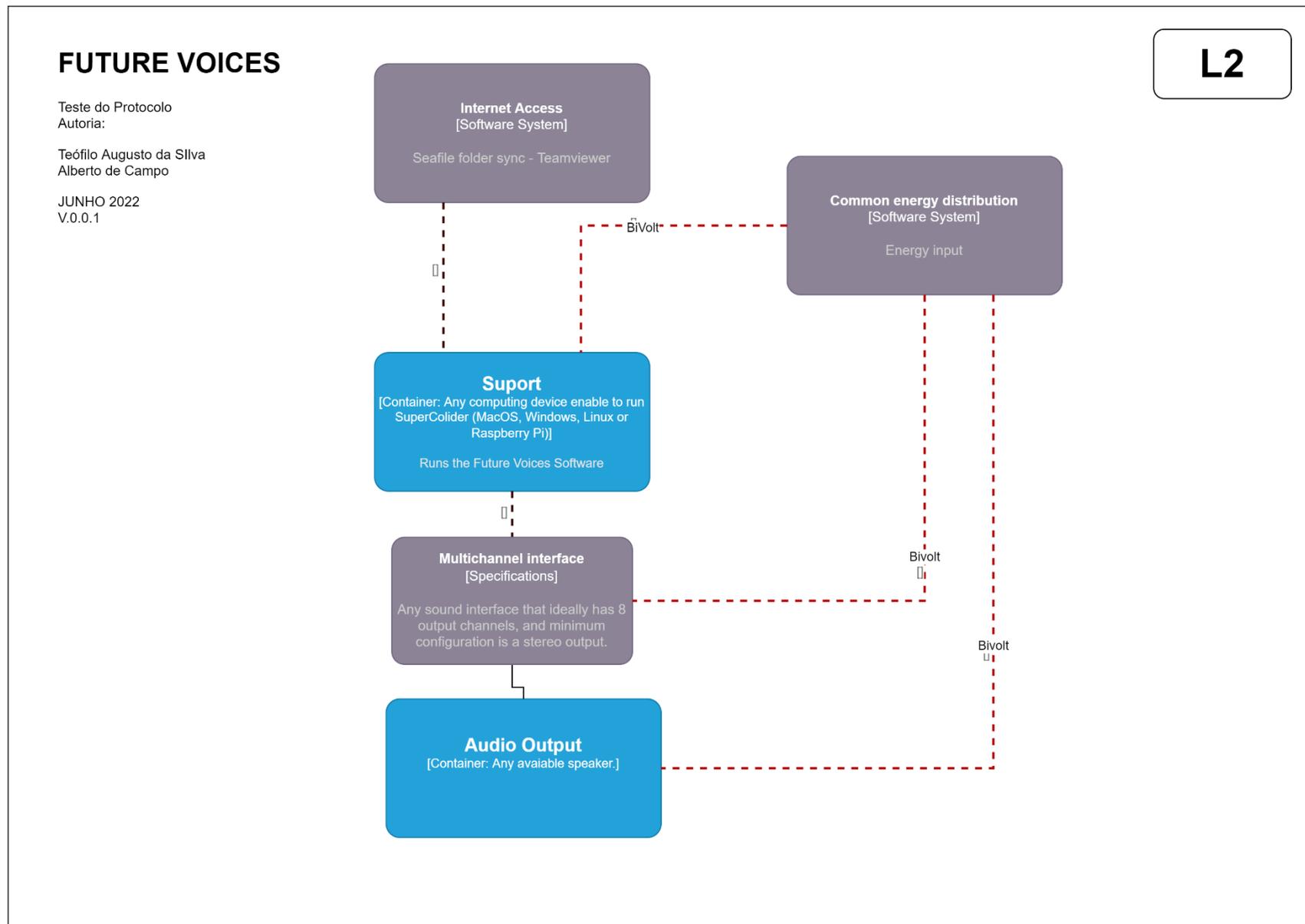
Figura 57: Diagrama Future Voices nível L1. Fonte: Acervo próprio. (2022).

**A) Diagrama L1 – Instrução Poética (Figura 57)**

Conforme entrevista com o Prof. Dr. Alberto de Campo, a instrução poética está concentrada na experiência do usuário, assim, este diagrama se concentra em apresentar como a obra interage com o mesmo sem fazer menção direta à disposição ou qualquer outra qualidade do hardware.

**B) Diagrama L2 – Objeto-Sistema (Figura 58)**

O suporte para apresentação do Future Voices é genérico e sua relação é representada aqui apresentando os componentes aos quais faz uso mas cujo funcionamento independe da vontade do artista.



**Figura 58:** Diagrama Future Voices nível L2. Fonte: Acervo próprio (2022)

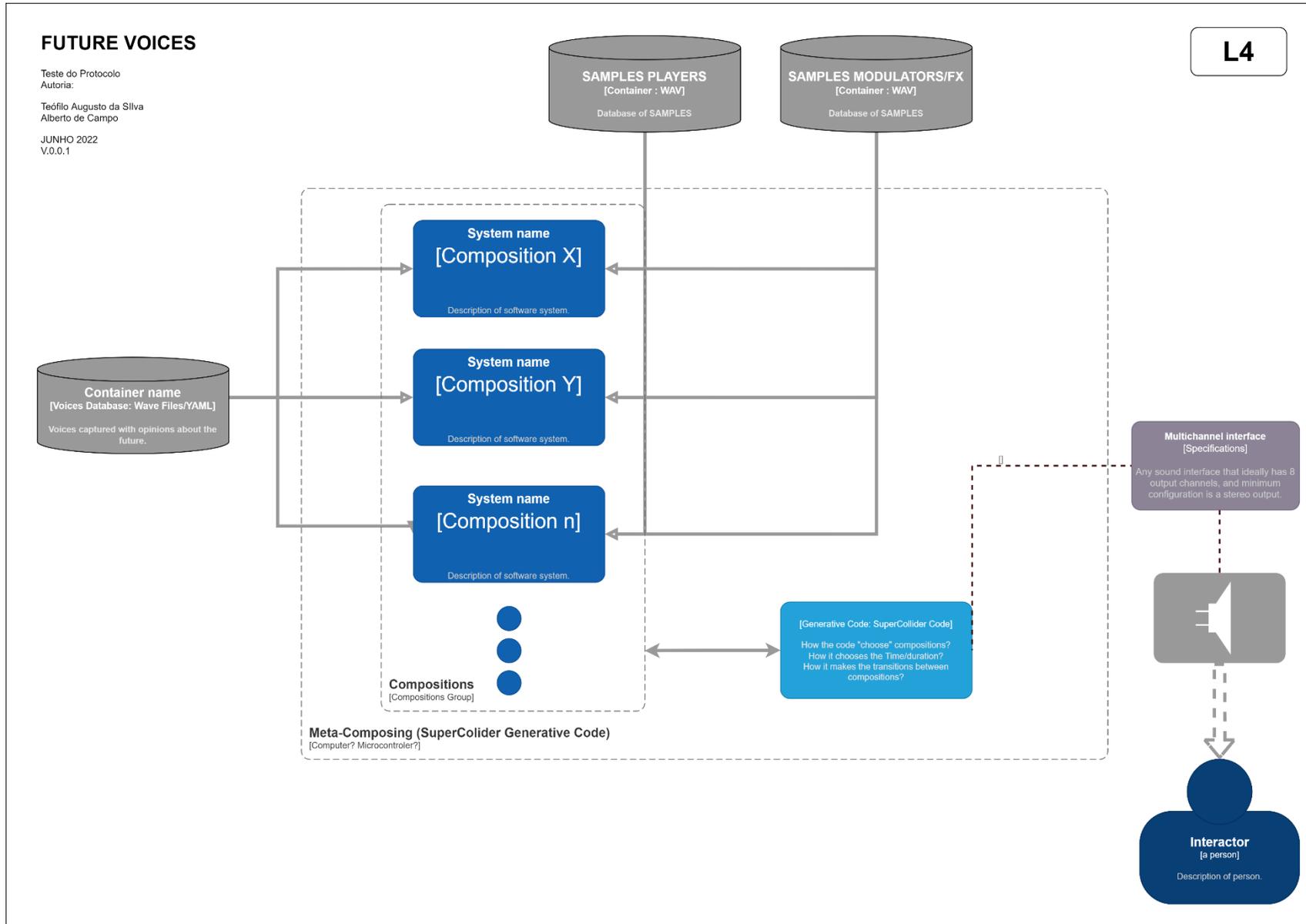


Figura 59: Diagrama Future Voices nível L4. Fonte: Acervo próprio (2022)

**C) Diagrama L4 – Contexto do Sistema (objeto-símbolo)  
(Figura 59)**

Para o componente L3 foi dispensado o diagrama já que não há nenhuma especificidade a ser caracterizada no hardware suporte. O L4 apresenta a relação entre as composições e o código principal da obra.

**D) Diagrama L5 – Domínio Figura 60)**

No diagrama L5, estão representados os dois códigos que compõe a obra e sua relação de funcionamento.

# FUTURE VOICES

Teste do Protocolo  
Autoria:

Teófilo Augusto da Silva  
Alberto de Campo

JUNHO 2022  
V.0.0.1

## L5

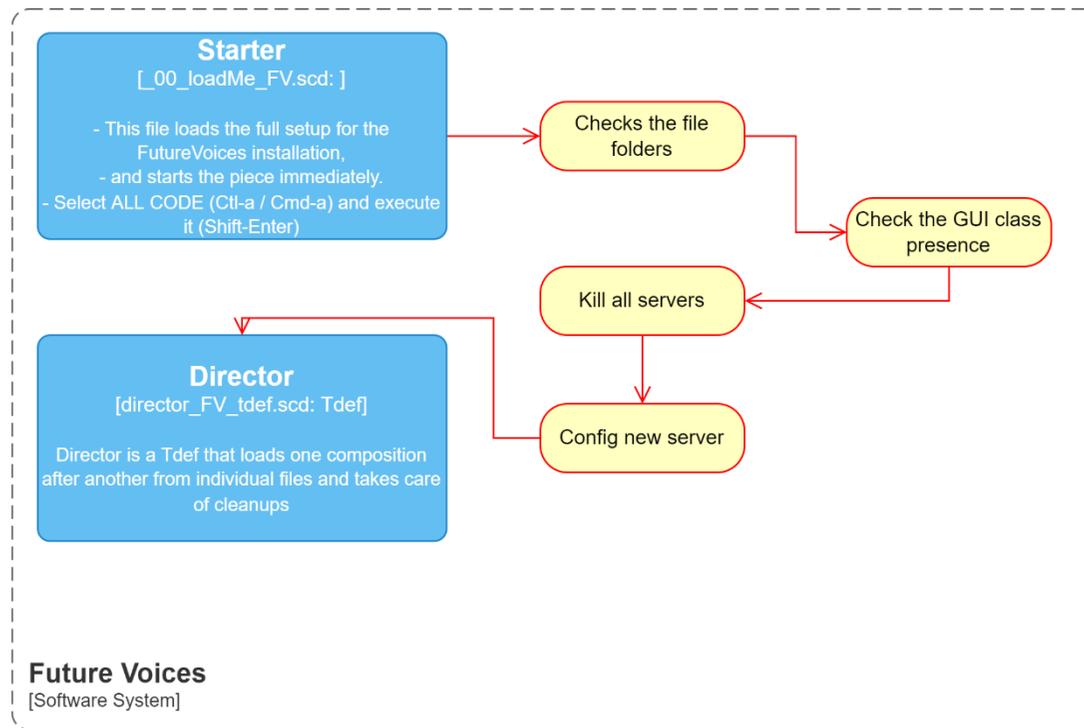


Figura 60: Diagrama Future Voices nível L6. Fonte: Acervo próprio (2022).

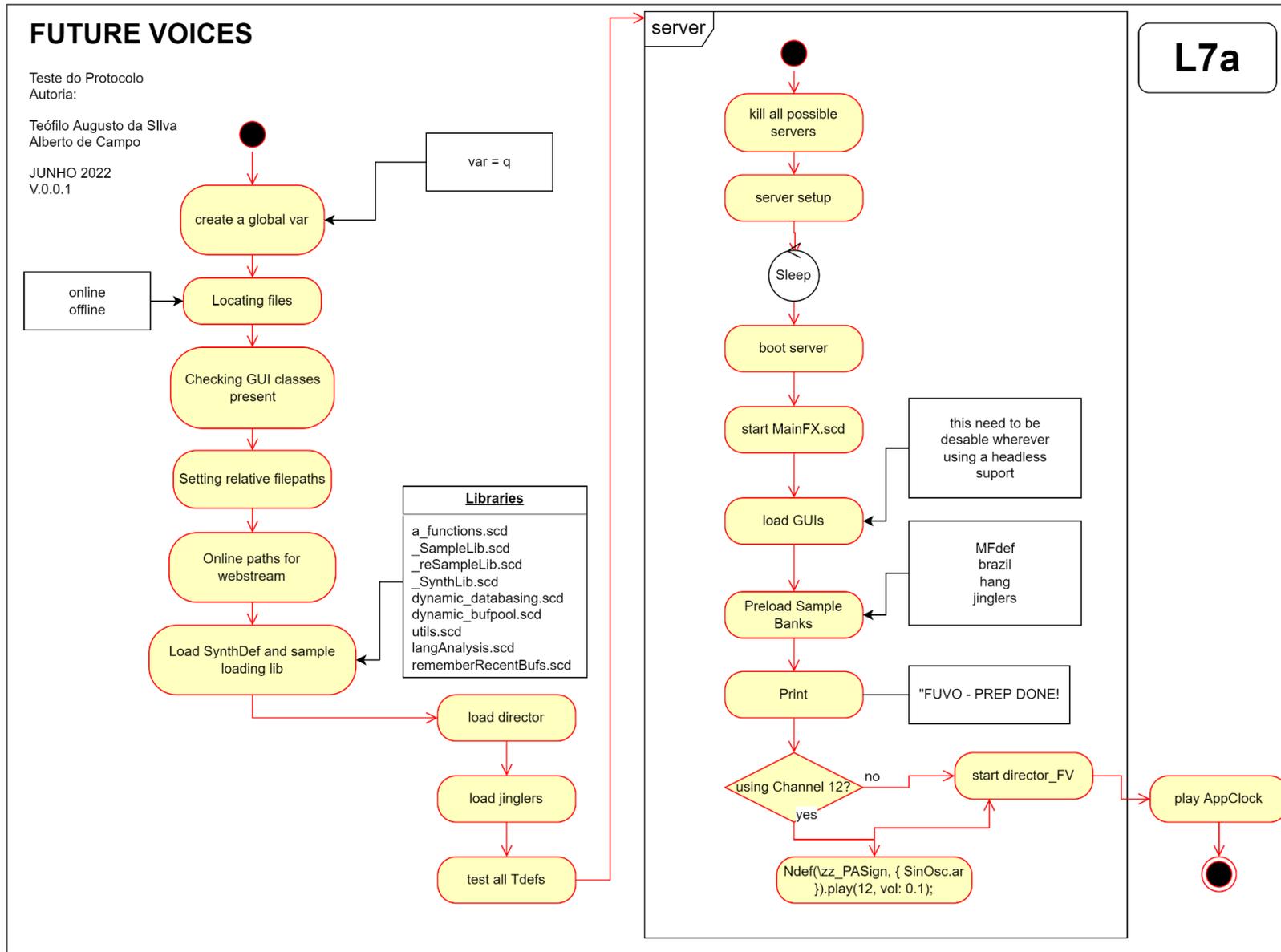


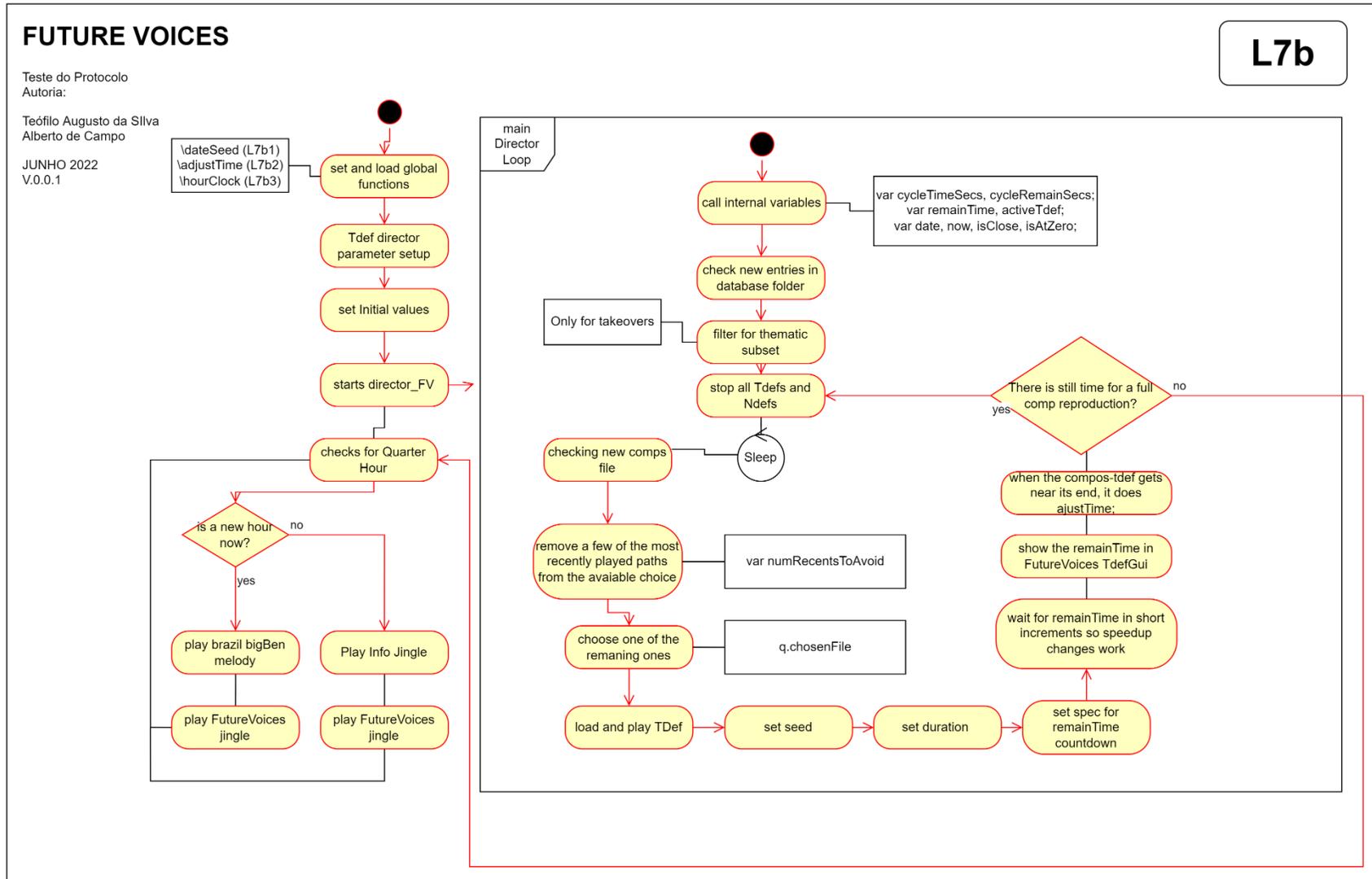
Figura 61: Diagrama Future Voices nível L7a - relativo ao código \_00loadMe\_FV.scd. Fonte: Acervo próprio (2022).

**E) Diagrama L7a – Código \_00loadMe\_FV.scd (Figura 61)**

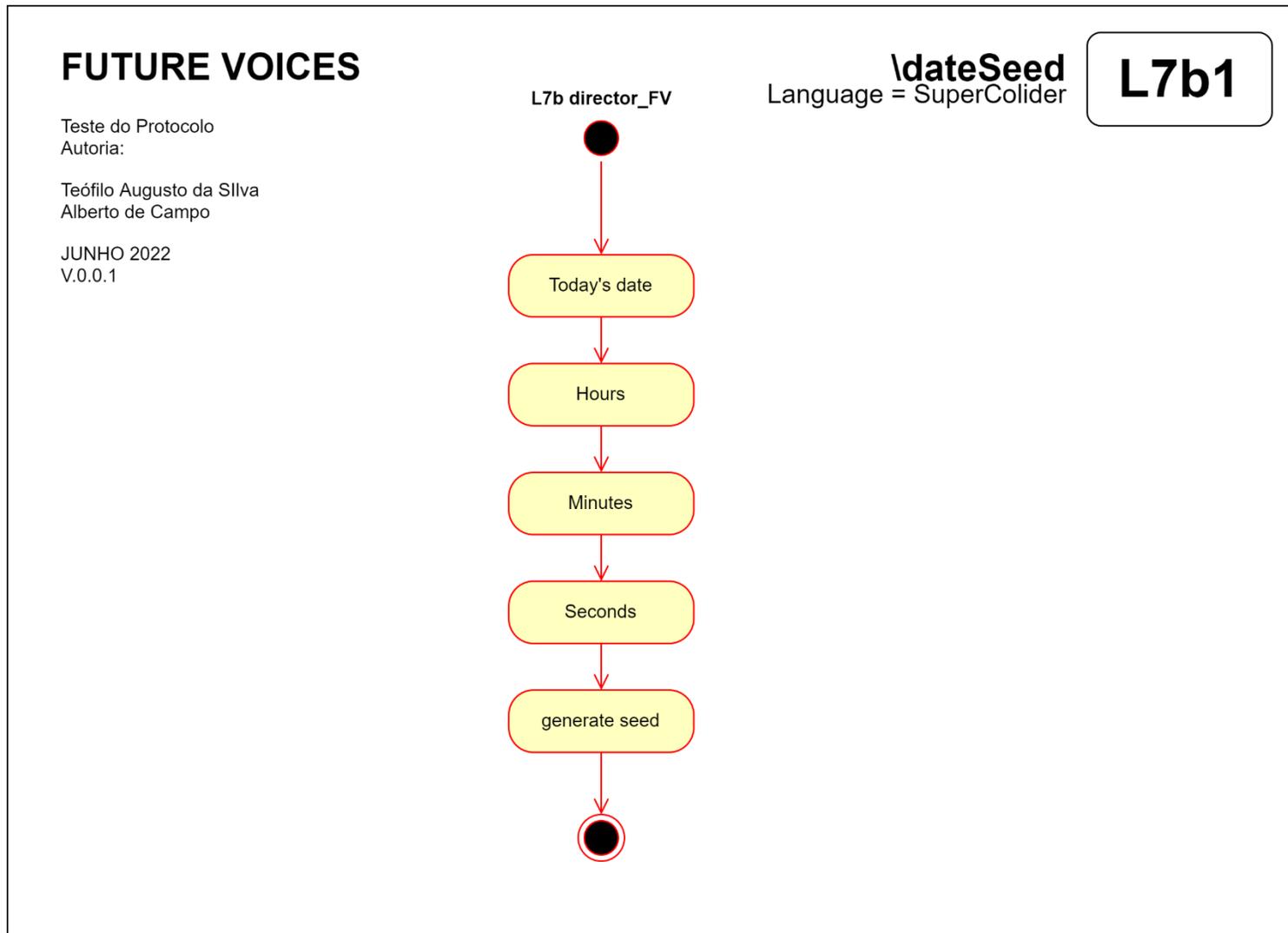
Neste diagrama está representado as atividades realizadas pelo código \_00loadMe\_FV que inicia a obra. Ele prepara os arquivos e monta o servidor do SuperColider para iniciar o código director\_FV.scd (L7b).

**F) Diagrama L7b – Código director\_FV.scd (Figura 46)**

Esse diagrama apresenta o funcionamento do código principal que gerencia as composições. Em destaque está o loop principal que estabelece os parâmetros generativos. No início do diagrama é possível visualizar as referências às funções globais criadas pelo director\_FV.scd e representadas no L7b1, L7b2 e L7b3.



**Figura 62:** Diagrama Future Voices nível L7b - relativo ao código director\_FV.scd. Fonte: Acervo próprio (2022).



**Figura 63:** Diagrama Future Voices nível L7b1 - relativo ao código director\_FV.scd - Global Function \dateSeed. Fonte: Acervo próprio (2022).

# FUTURE VOICES

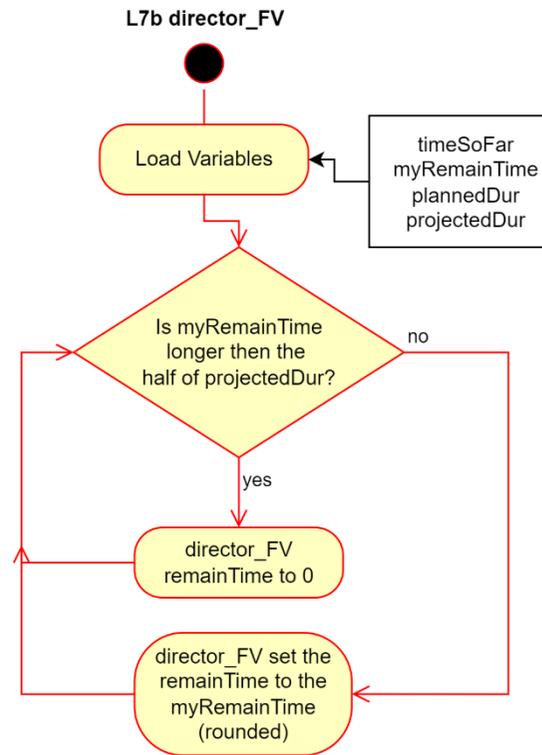
Teste do Protocolo  
 Autoria:

Teófilo Augusto da Silva  
 Alberto de Campo

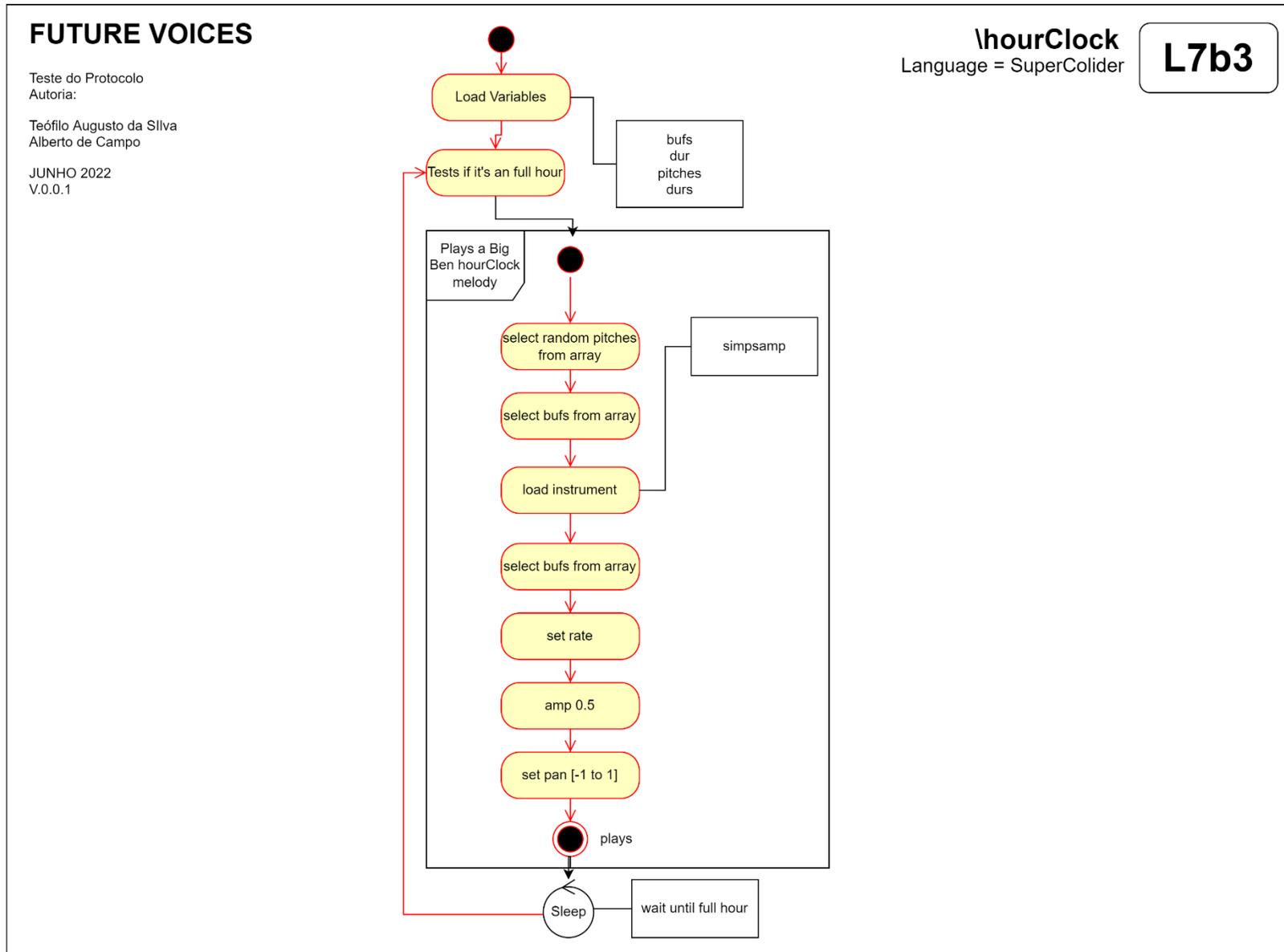
JUNHO 2022  
 V.0.0.1

**\adjustTime**  
 Language = SuperColider

**L7b2**



**Figura 64:** Diagrama Future Voices nível L7b2 - relativo ao código director\_FV.scd - Global Function \adjustTime. Fonte: Acervo próprio (2022).



**Figura 65:** Diagrama Future Voices nível L7b3 - relativo ao código director\_FV.scd - Global Function \hourClock. Fonte: Acervo próprio (2022).

Este último capítulo apresentou nossa contribuição para que os artistas computacionais tenham garantias de que suas intenções poéticas serão registradas. O Protocolo de Conservação Estética de Obras de Arte Computacional busca enfatizar o valor da documentação para um artista deste tipo de expressão artística. Assim, trazemos no quinto capítulo a forma de aplicação deste protocolo, bem como dois estudos de caso aplicados às obras EVO\_CIRCUITO e Future Voices.

## Considerações finais.

Esta tese foi reflexo de um anseio pessoal nosso de obter formas de compreender a fundo as obras computacionais. Como vimos ao longo do texto, obras das mídias instáveis são diferentes em muitos aspectos das obras tradicionais, entre elas o fato de sua poética nem sempre ser captada apenas pela visualização simples do objeto, principalmente por meio de vídeos e fotos disponibilizados na Internet.

O Protocolo de Conservação Estética de Arte Computacional, é a busca derradeira por esse testamento do(a)s artista(s) sobre sua intenção poética. Foi assim, que ao longo da pesquisa, pudemos observar nas universidades que transitamos durante o período do doutorado (Universidade de Brasília, Universidade Federal de Goiás, Universidade Livre de Buenos Aires, Universidade de Artes de Berlim, Universidade Politécnica de Madri, entre outras), as criações de nossos colegas artistas e a apresentação de suas ideias e práticas.

Para tanto, ao longo de cinco capítulos divididos em duas partes, formalizamos uma linha de pensamento que vai do entendimento do nosso objeto de estudo até as formas que já existem para buscar sua conservação, culminando na apresentação de dois estudos de aplicação do protocolo recém-nascido.

No primeiro capítulo, concluímos demonstrando dentro de um recorte histórico, a velocidade com a qual as tecnologias eram suplantadas por versões mais recentes e o processo de obsolescência resultante disto.

No segundo capítulo, abordamos a relação entre os artistas e as tecnologias da computação e da comunicação, seus frutos e consequências, apontando para as diferentes características que cada geração destes artistas.

O terceiro capítulo estabelece uma contribuição para o pensamento ontológico da arte computacional, apontando para o fato de que as classificações dadas atualmente para as artes das mídias instáveis dão muito destaque para o suporte, enquanto, na realidade, a essência da obra computacional, por exemplo, está na ideia e na intenção poética do(a)s artista(s).

O quarto capítulo apresenta as teorias, metodologias e estratégias já adotadas para a conservação e o restauro das obras de arte computacional, além de trazer relatos sobre as visitas técnicas realizadas por nós durante o período de Doutorado-Sanduíche na Universidade de Artes de Berlim – UdK.

O quinto e último capítulo, então, apresenta o protocolo e aponta para as sugestões de aplicação do mesmo, explicando sobre o modelo de diagrama C7Art e demonstrando como este protocolo foi aplicado a duas obras de arte computacional: EVO\_CIRCUITO e Future Voices.

A finalização desta tese não representa a conclusão da pesquisa, apenas entendemos que findou a primeira etapa de uma longa pesquisa que pode contribuir para outros professores, pesquisadores e artistas a documentarem as obras de arte computacional além de abrir espaço para pesquisas que levem este tipo de pensamento às outras categorias de artes das mídias instáveis.

## Referências Bibliográficas.

- ARANTES, P. Waldemar Cordeiro e a Arteônica: reescrituras da arte digital no Brasil e na América Latina. **MODOS: Revista de História da Arte**, v. 5, n. 2, p. 87–98, 15 out. 2021.
- BASICEVIC, D. *et al.* **Tendencias 4: Computers and Visual Research**. . Zagreb - Croácia: [s.n.], 1970. Disponível em: <<http://library1.nida.ac.th/termpaper6/sd/2554/19755.pdf>>.
- BEIGUELMAN, G.; MAGALHÃES, A. G. **Futuros Possíveis: arte, museus e arquivos digitais**. 1. ed. São Paulo: Edusp, 2014.
- BOURDIEU, P. **A economia das trocas simbólicas**. São Paulo, SP: Perspectiva, 1999.
- BRANDI, C. **Teoria da restauração**. 4. ed. Cotia, SP: Atelie, 2019.
- BREA, J. L. **Las tres eras de la imagen: imagen-materia, film, e-image**. Tres Cantos, Madrid: Akal, 2010.
- BROWN, S. **Software Architecture for Developers Vol 2**. 2021-03–22. ed. [s.l: s.n.]. v. 2
- CAMPBELL-KELLY, M. *et al.* **Computer: A history of the information machine, third edition**. [s.l: s.n.].
- CASTELLS, M. **A sociedade em Rede - A Era da Informação: Economia, Sociedade e Cultura**. [s.l: s.n.]. v. 1
- CASTELLS, M. **A sociedade em Rede - A Era da Informação: Economia, Sociedade e Cultura. Textos & Contextos (Porto Alegre)**. [S.l: s.n.]. Disponível em: <<http://revistaseletronicas.pucrs.br/fo/ojs/index.php/fass/article/view/945>>. , 2006
- CASTRO, S. DE. **Ontologia**. 1. ed. Rio de Janeiro: Jorge Zahar Editor, 2008. v. 1
- COSTA, H. **Waldemar Cordeiro e a fotografia**. 1. ed. São Paulo: Cosac & Naify; Centro Universitário Maria Antônia da USP, 2002.
- COUCHOT, E. **A tecnologia na arte: da fotografia à realidade virtual**. Porto Alegre: Editora da UFRGS, 2003.
- CUZZIOL, M.; PRADO, G. Alguns apontamentos sobre a restauração da obra digital Desertesejo. Em: **A memória do digital e outras questões das artes e da museologia**. 1. ed. Belo Horizonte: EdUEMG, 2019. p. 115–125.
- DEPOCAS, A. *et al.* (EDS.). **Permanence through change: the variable media approach = La permanence par le changement: l’approche des médias variable**. New York : Montréal: Guggenheim Museum Publications ; Fondation Daniel Langlois pour l’art, la science et la technologie, 2003.
- ERIC EVANS. **Domain-Driven Design: atacando as complexidades no coração do software**. 3ª ed. Rio de Janeiro: Alta Books, 2016.
- EVITTS, P. **A UML pattern language**. Indianapolis, IN: Macmillan Publishing, 1999.

- FLUSSER, V. **O universo das imagens técnicas: Elogio da superficialidade**. 1. ed. ed. São Paulo: Annablume, 2008.
- FLUSSER, V. **Filosofia da Caixa Preta: Ensaio para uma futura filosofia da fotografia**. 1. ed. São Paulo: Annablume, 2011.
- FRAGA, T. Acertos e desacertos na preservação de acervo em Arte Computacional Interativa Parte 1: imagens de síntese e realidades virtuais. 2016, Brasília: [s.n.], 2016. p. 260–270. Disponível em: <[https://files.cercomp.ufg.br/weby/up/779/o/tania\\_fraga.pdf](https://files.cercomp.ufg.br/weby/up/779/o/tania_fraga.pdf)>.
- GIANNETTI, C. **Estética Digital: Sintopia da arte, a ciência e a tecnologia**. 1. ed. Belo Horizonte: C/Arte, 2006.
- GILBRETH, F.; GILBRETH, L. **Process Charts**. 1. ed. Nova Iorque, EUA: The American Society of Mechanical Engineers, 1921. Disponível em: <<https://archive.org/details/processcharts00gilb/page/n1/mode/2up>>.
- GRAU, O.; HOTH, J.; WANDL-VOGT, E. **Digital art through the looking glass: new strategies for archiving, collecting and preserving in digital humanities**. [s.l.: s.n.].
- GUGGENHEIM. **The Variable Media Initiative**. Museu. Disponível em: <<https://www.guggenheim.org/conservation/the-variable-media-initiative>>. Acesso em: 19 maio. 2022.
- JOHNSON, S.; BORGES, M. L. X. DE A.; VAZ, P. **Cultura da interface: como o computador transforma nossa maneira de criar e comunicar**. Rio de Janeiro: Jorge Zahar, 2001.
- KENSKI, R. **Singularidade: O futuro enloqueceu?** Revista. Disponível em: <<https://super.abril.com.br/ciencia/singularidade/>>. Acesso em: 26 maio. 2022.
- LOZANNO-HEMMER, R. **Best Practices for Conservation of Media Art**. Repositório. Disponível em: <<https://github.com/antimodular/Best-practices-for-conservation-of-media-art>>.
- MACHADO, A. Waldemar Cordeiro: o brasileiro precursor da arte mediada por computadores. **Revista ECO Pós**, v. 18, n. 1, p. 27–35, 2015.
- MAINGUENEAU, D. **Análise de textos de comunicação**. Tradução: Cecília P. De Souza-e-Silva. 6. ed. São Paulo: Cortez, 2011.
- MANOVICH, L. **The language of new media**. 8. print ed. Cambridge, Mass.: MIT Press, 2001.
- MASCARENHAS, N.; VELHO, L.; HESS, L. (EDS.). **Waldemar Cordeiro: Arteônica** Velho, Luiz, , 1993. Disponível em: <<https://www.visgrafimpa.br/Gallery/waldemar/waldemar.htm>>. Acesso em: 20 abr. 2022
- MASON, C. **Cybernetic Serendipity: History and Lasting Legacy**. Disponível em: <<https://www.studiointernational.com/index.php/cybernetic-serendipity-history-and-lasting-legacy>>. Acesso em: 3 jan. 2021.
- \_\_\_\_\_. the Foretieth Anniversary of Event One At the Royal College of Art. **EVA 2009 London**

**Conference**, n. 1, p. 1–15, 2009.

MCLUHAN, M. **Os meios de comunicação como extensões do homem (understanding media)**.

São Paulo, SP: Editora Cultrix, 1974.

MELLADO, D. A System Engineer's Perspective for the Re-Creation of Media Art n-Cha(n)-t by

David Rokeby. Em: **Digital Art through the Looking Glass: New strategies for archiving,**

**collecting and preserving in digital humanities**. Donau - Austria: Donau-Universität, 2019.

p. 179–191.

MENABREA, L. F. et al. **Sketch of the analytical engine invented by charles babbage ... with**

**notes by the translator. Extracted from the “scientific memoirs,” etc. [The translator's**

**notes signed: A.L.L. ie. Augusta ada king, countess lovelace.].** [s.l.] R. & J. E. Taylor,

1843.

MOLES, A. **Arte e Computador**. 375. ed. Porto - Portugal: Edições Afrontamentos, 1990.

MORALES, L. G. **Teoría de La Conservación Evolutiva: conservación y restauración del arte**

**de los nuevos medios**. 1. ed. Online: BoD - Books on Demand, 2019.

MOSCATTI, G. **Discussões com Waldemar Cordeiro**. Catálogo de Exposição. Disponível em:

<[https://www.visgrafimpa.br/Gallery/waldemar/moscati/discuss\\_.htm](https://www.visgrafimpa.br/Gallery/waldemar/moscati/discuss_.htm)>. Acesso em: 2 maio.

2022.

NÖTH, W. Máquinas semióticas. **Galáxia**, v. 1, n. 1, p. 51–73, 2008.

PASK, G. A Comment, A Case History, and a Plan. Em: **Cybernetics, Art and Ideas (Reprinted)**

**from Cybernetic Serendipity**. Londres - GB: J. Reichardt, 1971. p. 76–99.

PRADO, G. Arte digital, diálogos e processos. **Futuros Possíveis arte, museus e Arq. Digit**. 1.

ed. São Paulo: Edusp, 2014. p. 120–134.

PRADO, G.; ARANTES, P. (EDS.). **Diálogos transdisciplinares: arte e pesquisa**. São Paulo, SP:

ECA/USP, 2016.

PRADO, G.; CUZZIOL, M. **Desertesejo (2000/2014): Notes on the Restoration Process**. [s.l.]

Springer International Publishing, 2019. v. 11568 LNCS

RANGEL, A. G. de Q. **O artista como desenvolvedor de sistemas computacionais:**

**experiências audiovisuais**. 2019. 299 f. Universidade de Brasília - UnB, 2019. Disponível

em: <<https://www.alexandrangel.art.br/textos/Tese-AlexandreRangel.pdf>>.

REICHARDT, J. **Cybernetic Serendipity. The Computer and the Arts**. Cybernetic Serendipity.

**Anais...**Nova Iorque, EUA: Frederick A. Praeger Inc. Publishers, 1969. Disponível em:

<[https://monoskop.org/Cybernetic\\_Serendipity](https://monoskop.org/Cybernetic_Serendipity)>

ROSEN, M. **A Little-Known Story about a Movement, a Magazine, and the Computer's Arrival**

**in Art: New Tendencies and Bit International, 1961 - 1973**. 1. ed. Karlsruhe (Germany),

Cambridge (MA - USA), London (England): ZKM (Center for Art and Media Karlsruhe), The

MIT Press, 2011.

SACK, W. **The Software Arts**. 1. ed. Cambridge, MA, EUA: The MIT Press, 2019.

SEREXHE, B. (ED.). **Preservation of digital Art: Theory and Practice: the project digital art conservation**. Karlsruhe: Zentrum für Kunst und Medientechnologie, 2013.

SHAW, J. **Legible City**, 1989. Disponível em: <<https://www.jeffreyshawcompendium.com/portfolio/legible-city/>>. Acesso em: 2 maio. 2022

TAYLOR, G. D. **When the machine made art: the troubled history of computer art**. New York: Bloomsbury, 2014.

TURING, A. On computable numbers with an application to the Entscheidungsproblem. **Proceedings of the London Mathematical Society**, v. S2-42, n. 1, p. 230–265, 1937. Disponível em: <[http://www.cs.virginia.edu/~robins/Turing\\_Paper\\_1936.pdf](http://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf)>.

\_\_\_\_\_. Computing machinery and intelligence. **Mind**, v. 59, n. 236, p. 433–460, 1950. Disponível em: <[papers2://publication/uuid/E74CAAC6-F3DD-47E7-AEA6-5FB511730877](https://papers2://publication/uuid/E74CAAC6-F3DD-47E7-AEA6-5FB511730877)>.

V2\_ORGANISATION. **Manifesto for Unstable Media**, 1987. Disponível em: <<https://v2.nl/archive/articles/manifesto-for-the-unstable-media>>

VELLOSILLO, A. V. *Nuevas estrategias para la conservación de colecciones de arte con elementos tecnológicos: propuestas metodológicas de humanidades digitales*. p. 506, 2015. Disponível em: <<https://eprints.ucm.es/33204/1/T36381.pdf>>.

VELLOSILLO, A. V. Estratégias de conservação e humanidades digitais. **Futur. Possíveis arte, museus e Arq. Digit**. 1. ed. São Paulo: Edusp, 2014. p. 135–146.

VENTURELLI, S. **Arte Computacional**. 1. ed. Brasília: Editora Universidade de Brasília, 2017.

WAZLAWICK, R. S. **História da Computação**. 1. ed. Rio de Janeiro: Elsevier, 2016.

WEIBEL, P. É proibido não tocar: Algumas observações sobre (partes esquecidas da) história da interatividade e da virtualidade. Em: **Arte, Ciência e Tecnologia: Passado, presente e desafios**. 1. Tradução: Cristina Pescador. São Paulo: Editora Unesp : Itaú Cultural, 2009. v. 1p. 91–110.

ZANINI, W. Primeiros tempos da Arte/Tecnologia no Brasil. Em: DOMINGUES, D. (Ed.). **A Arte no Século XXI: a humanização das tecnologias**. 1. ed. São Paulo: Fundação Editora da UNESP, 1997.

ZKM. **Acquisition workflow (Software-based artworks)**. Wiki. Disponível em: <[https://werke.zkm.de/wiki/index.php/Acquisition\\_workflow\\_\(Software-based\\_artworks\)](https://werke.zkm.de/wiki/index.php/Acquisition_workflow_(Software-based_artworks))>. Acesso em: 9 jun. 2022.

## Referências Multimídia.

- #Art. [Web Page]. Disponível em: Media Lab UFG Home Page: <<<https://art.medialab.ufg.br/>>>  
Acessado em 16 jan 2021.
- About the lab. [Web Page]. Disponível em: MIT Media Lab Home Page:  
<<<https://www.media.mit.edu/about/overview/>>>. Acessado em 15 jan 2021.
- Arteônica. [Web Page]. Disponível em: Waldemar Cordeiro Home Page:  
<<<https://www.waldemarcordeiro.com/arteonica>>>. Acessado em 6 jan 2021.
- Ars Eletronica. [Web Page]. Disponível em: Ars Eletronica Home Page:  
<<<https://ars.electronica.art/about/en/>>>. Acessado em 11 jan 2021.
- Cybernetic Serendipity. (23 set 2020). Disponível em: Monoskop: <<[https://monoskop.org/Cybernetic\\_Serendipity](https://monoskop.org/Cybernetic_Serendipity)>>. Acessado em 5 jan 2021.
- Cybernetic Serendipity. (2017). Disponível em: Compart center of excellence digital art:  
<<<http://dada.compart-bremen.de/item/exhibition/3>>>. Acessado em 4 jan 2021.
- Giorgio Moscati e Waldemar Cordeiro: quando o computador encontrou a Arte. (23 out 2012).  
Disponível em: Notícias da USP <<<https://www5.usp.br/uspdestaque/giorgio-moscati-e-waldemar-cordeiro-quando-o-computador-encontrou-a-arte/>>>. Acessado em 5 jan 2021.
- IBGE. (n.d.) Panorama Marabá. [Web Page]. Disponível em: IBGE:  
<<<https://cidades.ibge.gov.br/brasil/pa/maraba/panorama>>>. Acessado em 10 fev 2021.
- JACK, O. **Hydra: live coding video synth**. Sistema de Live Coding. Disponível em:  
<<https://hydra.ojack.xyz>>. Acesso em: 10 maio. 2022.
- Manson, Catherine. (Mars 11, 2018). Cybernetic Serendipity: History and Lasting Legacy. [Web Page] Retrieved January 4, 2021, from studio international:  
<https://www.studiointernational.com/index.php/cybernetic-serendipity-history-and-lasting-legacy>.
- Max. [Web Page]. Disponível em: Cycling74: <https://cycling74.com/products/max>. Acessado em 10 fev 2021.
- Media Lab – MIT. [Web Page]. Disponível em: Media Lab – MIT:  
<<<https://www.media.mit.edu/research/?filter=groups>>>. Acessado em 22 jan 2021.
- Media Lab/UFG. [Web Page]. Disponível em: Media Lab/UFG: <<<https://medialab.ufg.br/>>>. Acessado em 16 jan 2021.

Media Lab/UnB. [Web Page]. Disponível em: Media Lab/UnB: <<<http://medialab.unb.br/index.php/o-medialab>>>. Acessado em 16 jan 2021.

Media Lab/Unifesspa. [Web Page]. Disponível em: Media Lab/Unifesspa: <<<https://medialab.unifesspa.edu.br/>>>. Acessado em 16 jan 2021.

Kassab, Álvaro. Das reflexões ciberestéticas ao robô que pede esmola e retribui com poesia. (7 a 13 set 2009). Disponível em: Jornal da Unicamp <<[https://www.unicamp.br/unicamp\\_hoje/ju/setembro2009/ju440\\_pag0607.php#](https://www.unicamp.br/unicamp_hoje/ju/setembro2009/ju440_pag0607.php#)>>Acessado em 5 jan 2021.

Past ISEA Symposia. [Web Page]. Disponível em: ISEA: <<<http://www.isea-web.org/symposia/past-isea-symposia/>>>. Acessado em 9 fev 2021.

Processing Copyright. [Web Page]. Disponível em: Processing Foundation: <<<https://processing.org/copyright.html>>>. Acessado em 20 jan 2021.

The ZKM. [Web Page]. Disponível em: ZKM: <<<https://zkm.de/en/the-zkm>>>. Acessado em 16 jan 2021.

REAS, Casey; FRY, Benjamin. A Modern Prometheus. (May 29, 2018). Disponível em: Processing Foundation: <<<https://medium.com/processing-foundation/a-modern-prometheus-59aed94abe85>>>. Acessado em 28 jan 2021.

S4NTP. **Future Voices**. Disponível em: <<https://futurevoices.radio/>>. Acesso em: 2 jun. 2022.

*Todos os links foram checados em 10 de junho de 2022.*

## APÊNDICES

---

# Apêndice I. Versonificação de obra de arte computacional.

O apêndice abaixo é baseado no trabalho de Tom Preston-Werner, criador do Gravatars e co-fundador do GitHub. Todavia, esta versão possui sugestões para adaptação do mesmo ao programador-artista.

## Versionamento Semântico 0.0.1

### Sumário

Dado um número de versão MAJOR.MINOR.PATCH, incremente a:

1. versão Maior(MAJOR): quando o artista definir que sua obra alcançou a sua função estética naquele momento,
2. versão Menor(MINOR): quando o artista fizer as primeiras versões apresentadas a um público, mas ainda estiver em fase de teste de compatibilidade de sistemas, e
3. versão de Rascunho(Sketch): quando o software estiver em versões de trabalho, momento em que o artista programador está desenvolvendo os elementos estéticos que deseja.

Rótulos adicionais para pré-lançamento(pre-release) e metadados de construção(build) estão disponíveis como extensão ao formato MAJOR.MINOR.PATCH.

### Introdução

A metodologia de versionificação na obra de arte computacional objetiva direcionar o artista na produção de diferentes arquivos e controlar seu caminho estético. Diferentemente dos softwares utilitários (softwares que possuem uma função específica que faz as vezes de instrumento, ex.: um editor de fotos, um editor de texto ou um editor de vídeo) os softwares-estéticos quase não têm problemas de funcionalidades a serem resolvidos por subversões.

De resto, seguem as instruções do autor original desta metodologia que aos poucos será adaptada e testada com as obras de arte computacional.

“Como uma solução para este problema proponho um conjunto simples de regras e requisitos que ditam como os números das versões são atribuídos e incrementados.

Essas regras são baseadas em, mas não necessariamente limitadas às, bem difundidas práticas comumente em uso tanto em softwares fechados como open-

source. Para que este sistema funcione, primeiro você precisa declarar uma API pública. Isto pode consistir de documentação ou ser determinada pelo próprio código. De qualquer maneira, é importante que esta API seja clara e precisa. Depois de identificada a API pública, você comunica as mudanças com incrementos específicos para o seu número de versão. Considere o formato de versão X.Y.Z (Maior.Menor.Correção). Correção de falhas (bug fixes) que não afetam a API, incrementa a versão de Correção, adições/alterações compatíveis com as versões anteriores da API incrementa a versão Menor, e alterações incompatíveis com as versões anteriores da API incrementa a versão Maior.

Eu chamo esse sistema de “Versionamento Semântico”. Sob este esquema, os números de versão e a forma como eles mudam transmitem o significado do código subjacente e o que foi modificado de uma versão para a próxima.

### **Especificação de Versionamento Semântico (SemVer)**

As palavras-chaves “DEVE”, “NÃO DEVE”, “OBRIGATÓRIO”, “DEVERÁ”, “NÃO DEVERÁ”, “PODEM”, “NÃO PODEM”, “RECOMENDADO”, “PODE” e “OPCIONAL” no presente documento devem ser interpretados como descrito na [RFC 2119](#).

1. Software usando Versionamento Semântico DEVE declarar uma API pública. Esta API poderá ser declarada no próprio código ou existir estritamente na documentação, desde que seja precisa e compreensiva.
2. Um número de versão normal DEVE ter o formato de X.Y.Z, onde X, Y, e Z são inteiros não negativos, e NÃO DEVE conter zeros à esquerda. X é a versão Maior, Y é a versão Menor, e Z é a versão de Correção. Cada elemento DEVE aumentar numericamente. Por exemplo: 1.9.0 -> 1.10.0 -> 1.11.0.
3. Uma vez que um pacote versionado foi lançado(released), o conteúdo desta versão NÃO DEVE ser modificado. Qualquer modificação DEVE ser lançado como uma nova versão.
4. No início do desenvolvimento, a versão Maior DEVE ser zero (0.y.z). Qualquer coisa PODE mudar a qualquer momento. A API pública NÃO DEVE ser considerada estável.
5. Versão 1.0.0 define a API como pública. A maneira como o número de versão é incrementado após este lançamento é dependente da API pública e como ela muda.
6. Versão de Correção Z (x.y.Z | x > 0) DEVE ser incrementado apenas se mantiver compatibilidade e introduzir correção de bugs. Uma correção de bug é definida como uma mudança interna que corrige um comportamento incorreto.
7. Versão Menor Y (x.Y.z | x > 0) DEVE ser incrementada se uma funcionalidade nova e compatível for introduzida na API pública. DEVE ser incrementada se qualquer funcionalidade da API pública for definida como descontinuada. PODE ser incrementada se uma nova funcionalidade ou melhoria substancial for introduzida dentro do código privado. PODE incluir mudanças a nível de correção. A versão de

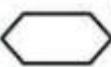
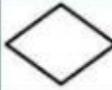
Correção DEVE ser redefinida para 0(zero) quando a versão Menor for incrementada.

8. Versão Maior X (X.y.z | X > 0) DEVE ser incrementada se forem introduzidas mudanças incompatíveis na API pública. PODE incluir alterações a nível de versão Menor e de versão de Correção. Versão de Correção e Versão Menor DEVEM ser redefinidas para 0(zero) quando a versão Maior for incrementada.
9. Uma versão de Pré-Lançamento (pre-release) PODE ser identificada adicionando um hífen (dash) e uma série de identificadores separados por ponto (dot) imediatamente após a versão de Correção. Identificador DEVE incluir apenas caracteres alfanuméricos e hífen [0-9A-Za-z-]. Identificador NÃO DEVE ser vazio. Indicador numérico NÃO DEVE incluir zeros à esquerda. Versão de Pré-Lançamento tem precedência inferior à versão normal a que está associada. Uma versão de Pré-Lançamento (pre-release) indica que a versão é instável e pode não satisfazer os requisitos de compatibilidade pretendidos, como indicado por sua versão normal associada. Exemplos: 1.0.0-alpha, 1.0.0-alpha.1, 1.0.0-0.3.7, 1.0.0-x.7.z.92.
10. Metadados de construção(Build) PODE ser identificada por adicionar um sinal de adição (+) e uma série de identificadores separados por ponto imediatamente após a Correção ou Pré-Lançamento. Identificador DEVE ser composto apenas por caracteres alfanuméricos e hífen [0-9A-Za-z-]. Identificador NÃO DEVE ser vazio. Metadados de construção PODEM ser ignorados quando se determina a versão de precedência. Assim, duas versões que diferem apenas nos metadados de construção, têm a mesma precedência. Exemplos: 1.0.0-alpha+001, 1.0.0+20130313144700, 1.0.0-beta+exp.sha.5114f85.
11. A precedência refere como as versões são comparadas uma com a outra quando solicitado.
12. A precedência DEVE ser calculada separando identificadores de versão em Maior, Menor, Correção e Pré-lançamento, nesta ordem. (Metadados de construção não faz parte da precedência).
  1. A precedência é determinada pela primeira diferença quando se compara cada identificador da esquerda para direita, como se segue: Versões Maior, Menor e Correção são sempre comparadas numericamente.  
Exemplo: 1.0.0 < 2.0.0 < 2.1.0 < 2.1.1.
  2. Quando Maior, Menor e Correção são iguais, a versão de Pré-Lançamento tem precedência menor que a versão normal.  
Exemplo: 1.0.0-alpha < 1.0.0.
  3. A precedência entre duas versões de Pré-lançamento com mesma versão Maior, Menor e Correção DEVE ser determinada comparando cada identificador separado por ponto da esquerda para direita até que seja encontrada diferença da seguinte forma:

1. Identificadores consistindo apenas dígitos são comparados numericamente.
2. Identificadores com letras ou hífen são comparados lexicalmente na ordem de classificação ASCII.
3. Identificadores numéricos sempre têm menor precedência do que os não numéricos.
4. Um conjunto maior de campos de pré-lançamento tem uma precedência maior do que um conjunto menor, se todos os identificadores anteriores são iguais.

Exemplo: 1.0.0-alpha < 1.0.0-alpha.1 < 1.0.0-alpha.beta < 1.0.0-beta < 1.0.0-beta.2 < 1.0.0-beta.11 < 1.0.0-rc.1 < 1.0.0.

## Apêndice II. Legenda de Fluxograma de Processos

 Operação	 Decisão	 Input Output	 conexão de páginas
 Inspeção	 Preparação	 Cartão perfurado	 Preparação
 Demora	 Terminal	 Memória principal	 Decisão
 Transporte	 Junção	 Sub-rotina	 Display
 Armazenamento	 "Ou"	 Tambor magnético	 Extrair
 Ações combinadas	 Disco magnético	 Conector	 Vários documentos
 Processo	 Fita magnética	 Classificar	 Agrupar
 Operação Manual	 Documento	 Fita papel perfurada	 Entrada manual

## Apêndice III. Exemplos de documentos para as respostas I e II do Questionário da Fase I do Protocolo

Conforme relatado no item 5.1.6 desta tese, nas primeiras fases do processo de implantação de um protocolo de conservação para obra de arte computacional, o(a)(s) artista(s) deve estar consciente dos processos e estratégias utilizadas para conservação e restauração de uma obra de arte. Assim, sugerimos que haja entre a documentação de uma obra, documentos com o teor abaixo.

Eu, [Nome do Artista], [Documentos do(a) Artista], [Dados Cívicos], desejo que a obra [Nome da Obra] [Versão se adequado], seja alvo de atividades de conservação e/ou restauros (opções marcadas abaixo), desde que obedecida a intenção poética do(a)(s) autor(a)(es), conforme documentação anexada. Ao artista é dado a possibilidade de retirar este consentimento a qualquer momento futuro desde que tal fato não influencie em termos de contrato de compra e venda já estabelecidos. Assim, estabeleço concordância com as seguintes estratégias ou ações de conservação e/ou restauro (obrigatório a assinatura do mesmo ao lado de cada opção marcada):

Com relação ao hardware:

( ) Substituição de componentes da parte de hardware por outro de mesma fabricação (época e fabricante) Assinatura: \_\_\_\_\_

( ) Substituição de componentes da parte de hardware por outro de fabricação semelhante e mesma função (terceiros) Assinatura: \_\_\_\_\_

( ) Substituição da parte de hardware (completo) por outra de mesma fabricação (época e fabricante) Assinatura: \_\_\_\_\_

( ) Substituição da parte de hardware (completo) por outro de fabricação semelhante e mesma função (terceiros) Assinatura: \_\_\_\_\_

Com relação ao software:

( ) Concordo com a criação de um ambiente de emulação, capaz de servir de suporte para a obra caso não haja mais possibilidade de conservação do hardware; Assinatura: \_\_\_\_\_

( ) Concordo com a Migração da obra para outro ambiente de software capaz de servir de suporte para a mesma; Assinatura: \_\_\_\_\_

( ) Concordo com a tradução da obra em uma outra linguagem de programação; Assinatura: \_\_\_\_\_

( ) Concordo com a recriação total da obra. Assinatura: \_\_\_\_\_

Sendo assim, esclarecido(a) destas possibilidades e com o compromisso que irão respeitar a intenção estética da obra, reafirmo as opções acima e assino este documento.