



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
INSTITUTO DE ARQUITETURA, URBANISMO E DESIGN
CURSO DE DESIGN

GUSTAVO BARRETO HOLZMANN DE VASCONCELOS

**DESIGN COMPUTACIONAL ENQUANTO MEIO DE EXPRESSÃO CRIATIVA:
UMA INVESTIGAÇÃO EXPERIMENTAL**

FORTALEZA

2025

DESIGN COMPUTACIONAL ENQUANTO MEIO DE EXPRESSÃO CRIATIVA:
UMA INVESTIGAÇÃO EXPERIMENTAL

GUSTAVO BARRETO HOLZMANN DE VASCONCELOS

Trabalho de conclusão de curso
apresentado ao Curso de Graduação em
Design do Instituto de Arquitetura,
Urbanismo e Design da Universidade
Federal do Ceará como requisito parcial
para obtenção do título de Bacharel em
Design

Orientadora: Profa. Dra. Aura Celeste
Santana Cunha

FORTALEZA

2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

H716d Holzmann, Gustavo.

DESIGN COMPUTACIONAL ENQUANTO MEIO DE EXPRESSÃO CRIATIVA: UMA
INVESTIGAÇÃO EXPERIMENTAL / Gustavo Holzmann. – 2025.
54 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Instituto de Arquitetura e
Urbanismo e Design, Curso de Design, Fortaleza, 2025.
Orientação: Profa. Dra. Aura Celeste Santana Cunha .

1. Design. 2. Design Computacional. 3. Investigação Experimental. 4. Expressão Criativa. I. Título.
CDD 658.575

GUSTAVO BARRETO HOLZMANN DE VASCONCELOS

DESIGN COMPUTACIONAL ENQUANTO MEIO DE EXPRESSÃO CRIATIVA:
UMA INVESTIGAÇÃO EXPERIMENTAL

Monografia apresentada no curso de Design do
Instituto de Arquitetura, Urbanismo e Design da
Universidade Federal do Ceará como requisito
parcial para obtenção do título de Bacharel em
Design

Aprovada em ____/____/_____.

BANCA EXAMINADORA

Prof.^a Dra. Aura Celeste Santana Cunha (Orientadora)

Universidade Federal do Ceará (UFC)

Prof. Dr. Diego Enéas Peres Ricca

Universidade Federal do Ceará (UFC)

Prof. Dr. Roberto Cesar Cavalcante Vieira

Universidade Federal do Ceará (UFC)

Prof. Me. Glaudiney Moreira Mendonça Junior

Universidade Federal do Ceará (UFC)

Dedico esse trabalho a todos aqueles que me apoiaram ao longo da minha jornada acadêmica, em especial, ao Professor Roberto Vieira, por me ter concedido minha primeira bolsa, que iniciou minha carreira na área da Computação, aos meus amigos, que tornaram o exigente dia a dia universitário mais prazeroso e a minha mãe, que possibilitou a minha graduação e o desenvolvimento deste trabalho.

“Como pode hoje em dia um artista ser inovador e criar algo de novo? Não está já tudo feito? Se um criador pretende estudar as novas tendências da sociedade, deve utilizar os meios e ferramentas mais contemporâneas. Mas que tipo de arte reflete o nosso tempo?” (WOLF LIESER, 2009)

RESUMO

O presente trabalho tem como objetivo explorar os principais paradigmas do Design Computacional enquanto processos e técnicas que podem ser aplicadas com finalidades -além das aplicações projetuais - relacionadas à expressão criativa. Pretende-se, também, demarcar os aspectos únicos deste tipo de manifestação, caracterizada pela utilização do computador, inspirada por categorias criativas tradicionais, como a pintura. Para isso, foi realizada uma investigação bibliográfica para delimitar conceitos centrais a discussão, como algoritmo, aleatoriedade, expressão criativa e a própria área do Design Computacional, além de seus principais paradigmas: Design Paramétrico, Generativo e Algorítmico. Nesse processo, foram descritos e agrupados diferentes tipos de algoritmos, originados a partir dessas subáreas, que podem interagir entre si, gerando combinações únicas. Essas categorias possibilitaram a fase seguinte, na qual, a partir de um estudo sobre a Arte Abstrata e alguns de seus principais expoentes, como Kandinsky, Mondrian e Pollock, foram analisadas obras desses artistas, identificando-se técnicas características em seu processo criativo. Usando essa análise como base, foram desenvolvidos três algoritmos que se enquadram nas categorias conceituais descritas anteriormente, e que traduzem computacionalmente uma técnica específica de cada um dos pintores, criando diferentes produtos gráficos e artísticos a partir desta investigação experimental. Esses resultados serviram de base para fundamentar a discussão empírica com as formas criativas mais tradicionais, demonstrando a singularidade que cada categoria do Design Computacional pode permitir em um contexto de expressão criativa.

Palavras-Chave: Design Computacional. Design. Investigação Experimental. Expressão Criativa.

ABSTRACT

This study aims to explore the main paradigms of Computational Design as processes and techniques that can be applied not only for project-based applications but also for purposes related to creative expression. It also seeks to highlight the unique aspects of this type of manifestation, characterized by the use of computers and inspired by traditional creative categories, such as painting. To achieve this, a bibliographic investigation was conducted to define key concepts for discussion, including algorithm, randomness, creative expression, and the field of Computational Design itself, as well as its main paradigms: Parametric, Generative, and Algorithmic Design. In this process, different types of algorithms originating from these subfields were described and categorized, showcasing their potential for interaction and the generation of unique combinations. These categories enabled the next phase, in which, through a study of Abstract Art and some of its key figures, such as Kandinsky, Mondrian, and Pollock, works by these artists were analyzed to identify distinctive techniques in their creative processes. Based on this analysis, three algorithms were developed, each corresponding to one of the previously described conceptual categories and computationally translating a specific technique from each painter. These algorithms generated different graphic and artistic outputs as part of this experimental investigation. The results served as a foundation for an empirical discussion comparing computational creative approaches with more traditional artistic methods, demonstrating the uniqueness that each category of Computational Design can offer in a creative expression context.

Keywords: Computational Design, Design, Experimental Investigation, Creative Expression.

LISTA DE ILUSTRAÇÕES

Gráfico 1 - Paradigmas do Design Computacional.....	19
Figura 1 - Pintura “Cruz Branca” (1922), de Kandinsky.....	23
Figura 2 - Pintura “Sobre o branco II” (1923), de Kandinsky.....	24
Figura 3 - Pintura “Composição em Vermelho, Azul e Amarelo” (1930), de Mondrian.....	25
Figura 4 - Lavender Mist Number 1 (1950), de Pollock.....	27
Figura 5 - Parâmetros do algoritmo baseado no Design Paramétrico.....	30
Figura 6 - Pseudocódigo do algoritmo baseado no Design Paramétrico.....	31
Figura 7 - Resultado I do algoritmo baseado no Design Paramétrico.....	32
Figura 8 - Resultado II do algoritmo baseado no Design Paramétrico.....	33
Figura 9: Resultado III do algoritmo baseado no Design Paramétrico.....	34
Figura 10: Resultado IV do algoritmo baseado no Design Paramétrico.....	34
Figura 11: Pseudocódigo do algoritmo baseado no Design Algorítmico.....	36
Figura 12: Resultado I do algoritmo baseado no Design Algorítmico.....	37
Figura 13: Resultado II do algoritmo baseado no Design Algorítmico.....	38
Figura 14: Pseudocódigo do algoritmo baseado no Design Generativo.....	39
Figura 15: Resultado I do algoritmo baseado no Design Generativo.....	40
Figura 16: Resultado II do algoritmo baseado no Design Generativo.....	41
Figura 17: Resultado final I do algoritmo baseado no Design Paramétrico.....	43
Figura 18: Resultado final II do algoritmo baseado no Design Paramétrico.....	44
Figura 19: Resultado final III do algoritmo baseado no Design Paramétrico.....	44
Figura 20: Resultado final I do algoritmo baseado no Design Algorítmico.....	45
Figura 21: Resultado final II do algoritmo baseado no Design Algorítmico.....	46
Figura 22: Resultado final III do algoritmo baseado no Design Algorítmico.....	46
Figura 23: Resultado final I do algoritmo baseado no Design Generativo.....	48
Figura 24: Resultado final II do algoritmo baseado no Design Generativo.....	48
Figura 25: Resultado final III do algoritmo baseado no Design Generativo.....	49

SUMÁRIO

1	INTRODUÇÃO	11
2	METODOLOGIA DE PESQUISA	13
3	FUNDAMENTAÇÃO TEÓRICA	14
4	POSSIBILIDADES CRIATIVAS	20
5	PINTURA ABSTRATA: NOVAS POSSIBILIDADES ARTÍSTICAS	22
6	DESENVOLVIMENTO DOS ALGORITMOS	28
6.1	Paramétrico	29
6.2	Algorítmico	35
6.3	Generativo	38
7	RESULTADOS E DISCUSSÕES	42
8	CONSIDERAÇÕES FINAIS	49
	REFERÊNCIAS	51
	APÊNDICE A – LINK DE ACESSO PARA OS CÓDIGOS NO GITHUB	53
	APÊNDICE B - LINK DE ACESSO AOS ALGORITMOS EM FUNCIONAMENTO ..	54

1 INTRODUÇÃO

Com o advento da computação, inúmeras facetas da vida humana foram modificadas de modo permanente, sejam elas de ordem social, econômica e até mesmo política. Dentre essas grandes mudanças de paradigma, se destacam as que ocorreram no mundo da arte, como coloca o autor Wolf Lieser (2009, p. 6): “em comparação com outros meios de expressão artística, o computador tornou-se um instrumento que transformou a nossa sociedade e cultura como nenhum outro o fez”. Nesse cenário complexo, visões contrastantes se expressam: alguns autores defendem que qualquer utilização do computador, para gerar imagens ou sons, por exemplo, constitui uma atividade essencialmente científica e abstrata, portanto, jamais artística (SETZER, 1997). Outros, antagonicamente, possuem uma visão mais inclusiva, propondo que “a arte digital é diversificada e surpreendente. Corresponde à época em que vivemos” (LIESER, 2009, p. 6). O autor não somente reconhece esta nova categoria expressiva, como a percebe como mais contemporânea, isto é, que reflete melhor o nosso próprio *zeitgeist*¹.

O presente trabalho debruça-se sobre essas temáticas, levando em consideração a sua pertinência e atualidade. A partir de uma fundamentação sobre conceitos relevantes para o debate, como expressão criativa, algoritmos, computação e aleatoriedade - além da análise da vanguarda da arte abstrata e de alguns de seus expoentes -, foi argumentado que a incorporação da computação - mais especificamente, do Design Computacional - durante o processo criativo pode gerar resultados únicos, a partir de uma tradução de técnicas oriundas desse movimento artística para a linguagem computacional. Como forma de evidenciar essas particularidades, foram desenvolvidas peças experimentais, produzidas por três algoritmos distintos, criadas utilizando como base processos descritos pelos diferentes paradigmas desta área do Design.

É fundamental estabelecer que em nenhum momento houve a pretensão de favorecer ou hierarquizar esse tipo de expressão criativa em detrimento de outras mais convencionais, mas sim de indicar a possibilidade genuína de novas expressões e configurações da subjetividade humana, intermediada, em maior ou menor grau, por aspectos computacionais.

¹ Zeitgeist é um termo alemão, introduzido inicialmente pelo escritor Johann Gottfried von Herder, para designar o que seria o “espírito do tempo”.

Assim sendo, de que forma os principais paradigmas do Design Computacional – Design Paramétrico, Generativo e Algorítmico – podem servir como um meio para possibilitar uma expressão criativa singular?

A partir dessa pergunta, o objetivo geral deste trabalho é compreender como as técnicas e processos provenientes do Design Computacional se manifestam enquanto possibilidades de produção gráfica e de expressão visual, delimitadas por questões comuns às demais formas de expressão criativa, como intencionalidade e técnica. Para isso, foram buscados os seguintes objetivos específicos:

1. Realizar uma investigação acerca das possibilidades de criação a partir dos paradigmas do Design Computacional;
2. Investigar as possibilidades criativas do Design Computacional em relação a outras formas tradicionais de representação, como a pintura; e
3. Identificar quais as características do Design Computacional que lhe conferem singularidade à medida que proporciona resultados específicos de aplicação de suas técnicas e conceitos.

A escolha do tema se pautou pela contemporaneidade dos debates, especialmente no que diz respeito à área do Design Computacional, que ainda não foi discutida extensivamente, particularmente quando considerada em suas aplicações fora do campo da arquitetura (MINEIRO, 2024). Dessa forma, justifica-se essa decisão pela necessidade de explorar as possibilidades inovadoras que o Design Computacional pode oferecer para a expressão criativa.

Além disso, o avanço tecnológico e o surgimento de novas ferramentas digitais permitiram o desenvolvimento de técnicas e processos originários do Design Paramétrico, Generativo e Algorítmico, que diferem substancialmente dos métodos criativos tradicionais, oferecendo novas formas de expressão e interação entre o criador, a obra e até mesmo, se projetado para tal, com o público.

Por fim, o estudo sobre o Design Computacional como meio de expressão criativa se torna relevante pela sua capacidade de provocar uma reflexão sobre o papel da tecnologia nas práticas artísticas contemporâneas (LIESER, 2009), levantando questões sobre autoria, originalidade, processo criativo e a relação entre arte e ciência, temas muitíssimo relevantes no nosso contexto atual.

2 METODOLOGIA DE PESQUISA

Levando em consideração a natureza dos objetos de estudo, o presente trabalho utiliza como metodologia uma abordagem exploratória. Como coloca Gil (2002), este tipo de pesquisa é pautado por uma maior flexibilidade nos seus processos, e tem como objetivo maior tornar determinado tema mais familiar ao explicitá-lo e construir hipóteses. Em relação ao desenvolvimento da pesquisa, quatro etapas distintas foram realizadas:

- Análise bibliográfica, definição de conceitos-chave e delimitação de categorias de algoritmo, descritas pelos paradigmas do Design Computacional;
- Investigação acerca das qualidades únicas que a arte intermediada pela computação pode apresentar, a partir de uma análise de obras selecionadas;
- Desenvolvimento de algoritmos que evidenciam esses aspectos; e
- Análise e discussão a partir dos resultados gerados.

A primeira etapa serviu como base para definir Design Computacional e seus processos, almejando incorporá-los para a sua posterior utilização. Além disso, realizou-se uma breve investigação etimológica e teórica sobre conceitos centrais da discussão, como computação e criatividade, visando fundamentar teoricamente o debate. A partir disso, foram definidos quais aspectos únicos os computadores permitem em relação a possibilidades criativas, quando comparados com formas artísticas tradicionais. Considerando exatamente esses aspectos, foram desenvolvidos três algoritmos que evidenciam estas características. Eles foram feitos a partir do Processing, uma biblioteca de código *open-source*² que facilita o desenvolvimento de códigos com finalidades gráficas e criativas (REAS; FRY, 2006).

Finalmente, a partir dos resultados obtidos, foi realizada uma análise qualitativa. Como define o autor GIL (2002), este tipo de análise é mais flexível, e pode ser summarizada em quatro etapas:

- i) Redução;
- ii) Categorização;
- iii) Interpretação; e

² O termo em inglês se refere a softwares que possuem o seu código fonte aberto para consulta, exameinação, modificação e redistribuição.

iv) Redação.

A redução pode ser compreendida como o processo de selecionar, simplificar, abstrair e transformar os dados originais provenientes da coleta. A categorização, por sua vez, é o processo de classificar os dados reduzidos pela etapa anterior para possibilitar a interpretação destes. Com base nisso, a interpretação correlaciona, abstrai e descreve estes dados categorizados, buscando acrescentar algo ao questionamento existente sobre o assunto. Por fim, a redação, em que todas as etapas anteriores são descritas e sintetizadas para apresentar os resultados obtidos, dispostos em um relatório (GIL, 2002).

3 FUNDAMENTAÇÃO TEÓRICA

Inicialmente, cabe definir as palavras computação, algoritmo, aleatoriedade e criatividade, termos que serão utilizados diversas vezes ao longo do presente trabalho.

Etimologicamente, a palavra “computação” surge do latim *computatio*, que indica cálculo ou conta. Segundo o dicionário de Cambridge, significa o ato ou o processo de calcular um resultado a partir da utilização de uma máquina (CAMBRIDGE University Press, 2024). Nesses termos, o valor da computação consiste na possibilidade de realizar tais cálculos e contas de um modo sistemático e exponencialmente maior que qualquer indivíduo. É uma máquina digital criada e desenvolvida com tal intuito.

“Algoritmo”, por sua vez, pode ser compreendido, segundo o dicionário de Cambridge (CAMBRIDGE University Press, 2024), como um conjunto de regras ou instruções que, ao serem seguidas na sequência correta, ajudam a solucionar um problema. O CSTA (2017), estabelece que algoritmos são basicamente uma sequência finita de passos para realizar determinada tarefa. A partir destas definições, pode ser inferido que não se restringe ao mundo computacional, uma vez que uma receita de bolo pode ser compreendida enquanto tal. Todavia, é um termo extremamente utilizado na área, por ser traduzido regularmente em programas ou códigos, fornecendo instruções para o funcionamento dos computadores.

Nesse cenário específico, temos ainda, segundo Martinhon (2002), dois tipos de algoritmos: determinísticos ou randômicos (também denominados de probabilísticos). O

primeiro tipo se caracteriza por, dado o mesmo *input*,³ o *output*⁴ obtido será sempre o mesmo. O segundo, paralelamente, difere em relação ao objeto produzido: cada execução do algoritmo vai gerar um resultado diferente, baseado em fatores randômicos descritos no código, independente dos dados utilizados no processo. Apesar de simples, esta conceituação enriquece o entendimento a respeito das possibilidades de criação que um determinado algoritmo pode possuir.

No que se refere a “aleatoriedade”, segundo Verbeeck (2006), existe uma que pode ser tida como natural, que ocorre em diferentes fenômenos, como a formação das nuvens, por exemplo. Apesar de ser possível identificar os elementos que influenciam este processo e certos padrões formais, não é possível discernir uma estrutura interna, que defina uma previsibilidade. Paralelamente, existe o mesmo conceito definido cientificamente. Neste caso, é um conceito abstrato perfeito, no sentido de ser utópico, sem respaldo material, isto é, incapaz de ser reproduzido. Com base nisso, fica evidente a dificuldade de reproduzir este fenômeno no meio computacional, como coloca Verbeeck (2016, p. 16):

During the early emergence of computing and programming, programmers were looking for a way to find what they call *pure random*. The idea of a computer creating randomness may seem preposterous, since computing is based on pure logic and reason, where randomness is about the absence of both logic and reason. Nevertheless, computer scientists wanted to create an algorithm that was completely independent from its context. It would not use any input to generate a random number or sequence of numbers; any number was equally probable to emerge. In the end these independent *pure random* generators generated correlated output: the outcome became predictable after analyzing the previously generated numbers.

Levando isso em consideração, para fins computacionais, são utilizados com frequência algoritmos pseudos-aleatórios, que usam uma *seed*⁵ para determinar os números gerados. Apesar de atingirem um certo nível de aleatoriedade, não correspondem ao estado utópico descrito pelo conceito científico, uma vez que é possível reconhecer padrões - mesmo que para a maioria dos usos práticos essa sequência seja tão massiva que não possa ser identificada - além de poder ser reproduzida com exatidão, com base na utilização da mesma *seed*.

³ O termo em inglês pode ser compreendido como a entrada de dados de determinado algoritmo.

⁴ O termo em inglês pode ser compreendido como a saída de dados de determinado algoritmo.

⁵ Na computação, *seed* ou semente se refere a um número ou um vetor utilizado em algoritmos geradores de valores pseudo-aleatórios. Ao utilizar a mesma *seed*, o resultado será sempre o mesmo.

Apesar disso, foram aplicados no projeto, pois sua previsibilidade não se demonstrou um problema, dado o escopo da sua utilização: para todos os efeitos, gerou resultados imprevisíveis o suficiente para serem tratados como aleatórios. Além disso, são muito mais eficientes em termos computacionais, em comparação aos algoritmos realmente randômicos, desenvolvidos por cientistas da computação, que se utilizam de *inputs* da natureza, como o som atmosférico, para gerar seus valores genuinamente aleatórios.

Finalmente, temos a “criatividade”, que por sua vez, vem do latim *creatus*, que significa criar. O dicionário de Cambridge a define como “a habilidade de produzir ou utilizar ideias originais ou inusitadas” (CAMBRIDGE University Press, 2024). Esse termo se relaciona profundamente com o Design. De acordo com a autora Ellen Kiss (2005), esse campo pode ser compreendido como a integração entre a criatividade e a inovação.

Nesse sentido, expressão criativa na perspectiva do Design pode ser entendida como a aplicação de uma nova ou inusitada ideia, materializada em determinado produto, seja ele material ou não, com certos objetivos ou intencionalidades definidas.

A partir dessa base conceitual, cabe introduzir: o que é o Design Computacional? Esse termo surge na literatura, inicialmente de modo muito associado à arquitetura, a partir da década de 90, e passa a ganhar crescente reconhecimento, com um aumento expressivo a partir de 2009 (CAETANO; SANTOS; LEITÃO, 2020). Apesar desta popularização enquanto área de estudo e paradigma processual para aplicação em projetos, esse campo do design ainda é mal-entendido em relação aos seus principais conceitos, se confundindo, diversas vezes, com áreas análogas ou próximas, mas não semelhantes. Outra questão que complexifica o debate é o fato do campo incorporar conceitos de outras disciplinas, em especial da computação (CAETANO; SANTOS; LEITÃO, 2020).

Diante desta dificuldade, será utilizada a definição de Inês Caetano, Luís Santos e António Leitão (2020) para delimitar não somente o Design computacional, mas também os seus paradigmas mais relevantes para o presente debate: Design Paramétrico, Design Generativo e Design Algorítmico. O estudo desses autores serve como uma taxonomia inicial, mas não se propõe a ser uma definição final, e sim um fundamento para possibilitar um consenso sobre o que caracteriza e compõe o campo, a partir de uma análise histórica e sistemática sobre o segmento e suas aplicações (CAETANO; SANTOS; LEITÃO, 2020).

A partir disso, vale estabelecer o Design Computacional enquanto área. Esse campo do Design se caracteriza por incorporar o uso de sistemas computacionais no desenvolvimento de

designs (CAETANO; SANTOS; LEITÃO, 2020). Ele se distingue do Design Digital, que pode ser compreendido a partir do uso de ferramentas computacionais no processo de design, pela centralidade associada a essas ferramentas. Enquanto no Design Digital as ferramentas computacionais auxiliam no processo, no Design Computacional elas determinam o processo (CAETANO; SANTOS; LEITÃO, 2020).

Logo, fica evidente que, apesar de ambas utilizarem tecnologias no seu desenvolvimento, esses dois segmentos do design se distinguem pela importância concedida à computação. A título exemplificativo pode ser citado um projeto gráfico, que incorpora softwares de edição de imagens vetoriais para o desenvolvimento das peças como Design Digital, em contraste com um projeto que delega ao seu algoritmo o seu principal produto. No Design Digital, é factível obter o mesmo resultado sem a utilização dessas ferramentas digitais – mesmo que isso o descharacterizasse como design digital -, com muito mais trabalho, evidentemente. No Design Computacional, por outro lado, isso não é possível, uma vez que o produto está invariavelmente interligado a um determinado algoritmo, que opera milhares - até mesmo milhões - de operações rapidamente e gera resultados únicos provenientes desses cálculos.

Para compreender o Design Paramétrico, é necessário ter uma definição clara a respeito do que pode vir a ser um parâmetro. De acordo com o Dicionário de Cambridge (CAMBRIDGE University Press, 2024), parâmetro pode ser compreendido como um “conjunto de fatos ou um limite fixo que estabelece ou limita como algo pode ou deve acontecer ou ser feito”. Nesse sentido, podemos definir este termo de um modo ainda mais preciso, ao estabelecer que esse conjunto de fatos podem ser traduzidos numericamente, isto é, ser mensurados. Tal adição é importante, pois nos permite inferir parâmetro como sendo um *input*, isto é, uma entrada de algum tipo de dado, quantificável, em determinado sistema.

A partir desta definição inicial, podemos utilizar a conceituação de Kolarevic para Design Paramétrico como “um processo em que os parâmetros de determinado design são declarados e não sua forma, possibilitando, deste modo, diversas instâncias” (Kolarevic, 2000, p. 4). Outro ponto fundamental é compreender que, inserindo *inputs* com os mesmos valores, os resultados serão sempre idênticos, sendo, portanto, um algoritmo determinístico. Logo, o Design Paramétrico é uma abordagem que descreve um design baseado simbolicamente no uso dos seus parâmetros (CAETANO; SANTOS; LEITÃO, 2020). Se um design é definido por parâmetros finitos e mensuráveis, ele pode ser tido como paramétrico.

Quanto ao Design Generativo, inicialmente, o termo generativo pode ser tido, segundo o Dicionário de Cambridge (CAMBRIDGE University Press, 2024) como a “capacidade de produzir ou criar algo”. A partir disso, podemos definir Design Generativo como um processo que se utiliza de algoritmos – mais autônomos em relação ao Design Paramétrico- para a criação de produtos complexos e distintos, que dificilmente podem ser associados à sua descrição algorítmica (CAETANO; SANTOS; LEITÃO, 2020).

Sob uma perspectiva comparada, enquanto no Design Paramétrico os parâmetros são o ponto central do processo, no Design Generativo a capacidade de criar resultados inesperados, de uma forma imprevisível, se caracteriza como a característica mais fundamental. Mesmo a partir de instruções simples, pode-se obter essas surpresas, a partir da interrelação entre múltiplos componentes do algoritmo. Isso constitui o fator mais importante deste paradigma: a dificuldade de correlacionar o *output* com o algoritmo em si, seja por incorporar a aleatoriedade no seu processo ou por inter relacionar muitos fatores distintos, que ao se modificarem mutuamente, geram resultados que dificilmente poderiam ser previstos em um primeiro momento.

Finalmente, temos o Design Algorítmico. Segundo o autor Terzidis (2004), o Design Algorítmico pode ser tido como uma abordagem baseada na definição de programas de computadores que geram formas e espaços baseados em regras inerentes a softwares de arquitetura, tipologia, código e até mesmo linguagem (TERZIDIS, 2004). Para Queiroz e Vaz (2015), o Design Algorítmico permite criar o design através da manipulação direta do código.

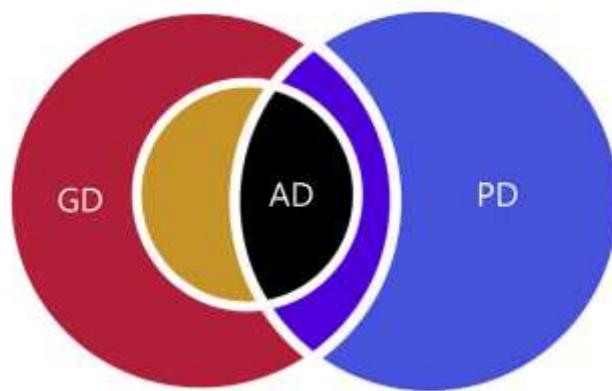
É perceptível que a partir dessas conceituações, muitas similaridades e até mesmo sobreposições teóricas podem ser observadas com os outros paradigmas do design computacional, em especial com o Design Generativo, fator que acaba gerando frequentemente uma incongruência ou sobreposição conceitual no uso destes termos (CAETANO; SANTOS; LEITÃO, 2020).

Caetano, Santos e Leitão (2020) propõem que o Design Algorítmico é um paradigma de Design que também utiliza algoritmos para gerar modelos, sendo, deste modo, generativo. Apesar disso, se distingue do generativo por existir uma clareza entre o *input* dado e o *output* gerado, isto é, ao custo de gerar menos resultados inesperados, este tipo de abordagem oferece um controle muito superior do objeto gerado. Portanto, cabe classificar o Design Algorítmico como um subconjunto do Design Generativo, já que compartilham muitas similaridades –

sejam elas técnicas ou processuais -, porém, com objetivos finais e lógicas algorítmicas distintas.

É possível representar graficamente esses três paradigmas do Design Computacional com o Gráfico 1, retirado do texto “*Computational design in Architecture: Defining parametric, generative and algorithmic Design*” (CAETANO; SANTOS; LEITÃO, 2020).

Gráfico 1 - Paradigmas do Design Computacional



Fonte: Caetano; Santos; Leitão (2020).

No Gráfico 1, GD corresponde a *Generative Design* (Design Generativo), AD a *Algorithmic Design* (Design Algorítmico) e PD a *Parametric Design* (Design Paramétrico). A partir de sua observação, podemos avaliar que, apesar de suas diferenças e similitudes, estes paradigmas do Design Computacional podem, ou não, se sobrepor. Essa possibilidade é fundamental, pois nos permite estabelecer cinco categorias distintas entre esses paradigmas do Design Computacional e suas combinações, para classificar os diferentes tipos de processo que podem ser utilizados para o desenvolvimento de um projeto ou algoritmo baseado no Design Computacional. São elas:

- Exclusivamente paramétrico;
- Exclusivamente generativo;
- Algorítmico, portanto, generativo;
- Simultaneamente paramétrico e generativo; e
- Simultaneamente paramétrico e algorítmico;

4 POSSIBILIDADES CRIATIVAS

A partir dessas classificações, é possível exemplificar cada categoria, bem como os aspectos singulares que as caracterizam. Vale destacar que houve uma grande dificuldade de encontrar a utilização do Design Computacional para fins não projetuais. Essa condição destaca a incipienteza do uso desta área para finalidades criativas. Apesar disso, é possível enquadrar determinados programas como associados a certos paradigmas, por seguirem uma lógica processual e técnica semelhante.

Começando com o Design exclusivamente paramétrico, temos uma ampla gama de aplicações, pautada majoritariamente em projetos arquitetônicos. Nesse contexto, ele se destaca por formas geométricas complexas, bem como pelo uso do algoritmo e da subsequente manipulação dos seus parâmetros para gerar inúmeras variações, invariavelmente associadas entre si, uma vez que seguem diversas regras ou etapas em comum, se diferenciando somente pela variação dos valores associados aos seus *inputs*.

A riqueza deste tipo de Design consiste na inter-relação entre a forma e esses dados. Ao modificar um simples valor, uma nova forma será gerada. Ao multiplicar todas essas possibilidades com todos os parâmetros, se cria uma rica gama de variedades formais que, apesar disso, ainda podem ser associadas ao algoritmo original (CAETANO; SANTOS; LEITÃO, 2020).

Como mencionado anteriormente, um parâmetro pode ser absolutamente qualquer fator mensurável. Nesse sentido, dados valiosos e sensíveis podem ser utilizados enquanto *inputs*, como a intensidade de um som, a frequência de um batimento cardíaco, a captura de um movimento, dentre outros. Tal possibilidade permite criar produtos singulares, que ao interagir com parâmetros, podem sintetizar abstrações formais complexas.

O Design Generativo, por sua vez, surge muito associado a Sistemas de Design Generativo Baseados em Métricas, do inglês *performance-based generative design systems* (PGDS). Esses sistemas são caracterizados por uma definição inicial de métricas ou objetivos, que, a partir dessa definição inicial, geram alternativas diversas, incorporando possibilidades que dificilmente poderiam ser previstas em um primeiro momento. Este tipo de algoritmo se destaca exatamente por essa imprevisibilidade, intensificada pela incorporação da aleatoriedade no processo.

É difícil associar o algoritmo ao produto gerado, uma vez que a inter-relação entre diversos fatores aleatórios com outros não aleatórios gera uma quantidade imensurável de resultado inesperados, a qual os autores denominam de *happy accidents* (acidentes felizes) (CAETANO; SANTOS; LEITÃO, 2020). Nessa lógica, um famoso programa pode ser enquadrado nesta categoria: o Jogo da Vida de Conway. Esse código simula a “vida” ao definir regras simples que simulam a multiplicação e a morte celular. A partir disso, de modo autônomo, acontece a simulação, sendo praticamente impossível prever seu estado um em determinado momento (ADAMATZKY, 2010).

Enquanto subconjunto do Design Generativo, o Design Algorítmico se destaca por possuir possibilidades semelhantes, mas em menor escala, isto é, ao reduzir o número de fatores imprevisíveis no processo, reduz-se o número de possibilidades, mas obtém-se um controle maior do resultado obtido. Nesse sentido, pode ser mais útil em determinadas aplicações, como na reconstrução da emblemática Igreja da Sagrada Família, localizada na Espanha. Por meio de um algoritmo, se permitiu a recriação das complexas estruturas arquitetônicas da obra, possibilitando as obras de reconstrução da mesma (BURRY, 2006).

A partir do momento que esses paradigmas se conjugam, suas características se reestruturam, incorporando aspectos singulares de cada e permitindo novas possibilidades. Nesse sentido, o Design Paramétrico e Generativo surge enquanto paradigma. Ele se destaca pela incorporação de *inputs* nos processos imprevisíveis e potencialmente aleatórios do Design Generativo ou Algorítmico, gerando resultados que podem ser associados (algorítmicos) ou não (generativos) ao código base. A título exemplificativo, pode ser citado uma modificação do Jogo da Vida de Conway, em que o número de espectadores – medido pela captura de movimento – determina a rapidez da multiplicação das células. As possibilidades são diversas, uma vez que é possível alterar praticamente todos os fatores de funcionamento do jogo com uma incorporação de um simples parâmetro. Ao associar vários deles, essa gama de alternativas se multiplica ainda mais.

Com base no que foi apresentado, o presente trabalho utiliza o desenvolvimento de algoritmos para demonstrar o valor da incorporação do Design Computacional no processo criativo, em virtude das características únicas que este tipo de expressão pode apresentar. Deste modo, escolheu-se utilizar uma abordagem exploratória por sua flexibilidade e sua pretensão de construir hipóteses, que dialoga profundamente com a natureza do projeto, uma vez que foi realizado um trabalho experimental, isto é, não se sabia previamente o resultado que seria gerado, considerando não somente a natureza dos algoritmos, mas também o modo

em que foram desenvolvidos para destacar as maiores qualidades que os paradigmas do Design Computacional podem possibilitar.

5 PINTURA ABSTRATA: NOVAS POSSIBILIDADES ARTÍSTICAS

Com o intuito de servir de base para a arte computacional, foi realizada uma breve discussão sobre a arte abstrata, definindo, neste processo, a originalidade deste tipo de representação, bem como seu processo criativo. A partir de um estudo sobre as pinturas dos artistas Kandinsky, Mondrian e Pollock - famosos representantes dessa vanguarda - foi possível compreender a intencionalidade e as técnicas utilizadas no desenvolvimento de suas obras. A partir destas definições, foi possível realizar uma tradução destas técnicas para a linguagem computacional.

Segundo Carvalho e Mansano (2020), a Arte Abstrata pode ser tida como “uma ruptura estética com toda trajetória pregressa da produção pictórica”. Isso ocorreu devido a uma mudança drástica no objeto produzido, como coloca Parente (1993, p. 30):

De fato, nós temos duas maneiras de pensar a imagem: a imagem como uma ilusão que deve ser submetida ao inteligível, que a domestica, a ensina a falar, e a imagem como puro sensível e ser da sensação que afirma o real como novo.

Essa passagem de uma imagem figurada, que representa um objeto específico no mundo real, para uma abstrata, que não possui um paralelo ao mundo material, foi determinante como uma das maiores mudanças de paradigma do cenário artístico no século XX (CARVALHO, MANSANO. 2020).

Os críticos da arte abstrata classificam-na como uma espécie de arte não humana, que é fria e intelectual em demasia. Todavia, como coloca Schapiro (2001), esse tipo de julgamento relega o seu potencial expressivo, uma vez que a arte não figurada ampliou os meios de utilização da representação para fins não figurativos, permitindo novas facetas de sensações e percepções que antes não eram abordadas nas artes.

No começo do século XX, Kandinsky (Figura 1) surge como uma figura central deste movimento. Como coloca Gombrich (1972, p. 451): "mas uma coisa é discorrer sobre tais

possibilidades em termos gerais, e outra é realmente expor uma pintura sem qualquer objeto reconhecível. Parece que o primeiro artista a fazê-lo foi o pintor russo Wassily Kandinsky".

Uma de suas principais características é seu uso específico das cores. Para Kandinsky, não bastava escolher esse elemento com base no seu resultado representativo. Para além disso, existia uma intenção de incorporar diferentes sensações na aplicação das cores, com um desejo sinestésico (KANDINSKY, 1977). Para exemplificar, pode ser citada a coleção *Composition*, em que o artista usou certas cores como equivalentes de determinadas tonalidades musicais.

Figura 1 - Pintura “Cruz Branca” (1922), de Kandinsky.



Fonte: <https://www.todamateria.com.br/kandinsky/>.

A partir dessa nova perspectiva, se iniciou um novo tipo de aplicação, em que não havia a pretensão de criar correlações com o mundo figurado, mas sim de criar texturas,

sensações e até mesmo representar sons. Como coloca Gombrich (1972, p. 451): "Ele sublinhou os efeitos psicológicos da cor pura, o modo como um vermelho-brilhante pode afetar-nos como o toque de um clarim".

Podemos sumarizar esta nova perspectiva a respeito da utilização da cor como uma de suas principais técnicas, que se relaciona profundamente com a sua intencionalidade artística, que pode ser compreendida a partir da pretensão espiritual que o artista tinha com sua arte, isto é, ele almejava atingir o transcendental a partir dos meios artísticos, e, para ele, isto só era possível através de novas formas de representação visual (Figura 2). Nesse sentido, era necessário se livrar das formas de representação da natureza e perspectiva tradicionais para atingir tal objetivo. Como descrito no seu livro *"Concerning the Spiritual in Art"* (1977), isso só poderia ser atingido através do uso da cor, das formas e das composições.

Figura 2 - Pintura "Sobre o branco II" (1923), de Kandinsky.

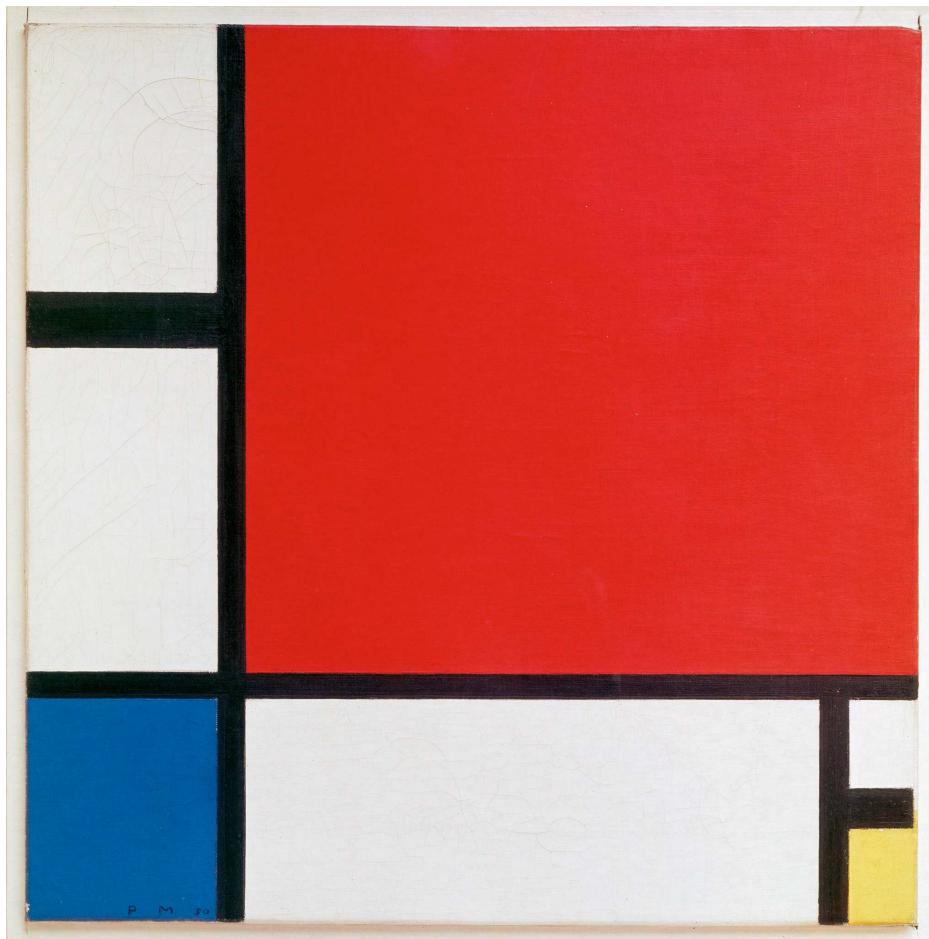


Fonte: <https://www.todamateria.com.br/kandinsky/> .

O artista Piet Mondrian, por sua vez, utilizava somente cores primárias - azul, vermelho e amarelo -, além do preto e do branco, associando o uso delas a uma intensidade formal. Suas obras se destacam, além disso, pela repetição de elementos geométricos - especialmente quadrados e retângulos -, dispostos sempre respeitando uma grade, composta por diversas linhas verticais e horizontais (Figura 3).

Com efeito, Mondrian utilizava na sua produção artística somente estes elementos básicos: linha, plano e cores primárias (ARGAN, 1992). Além disso, regras eram estabelecidas: um de seus quadrados ou retângulos jamais entram em contato direamente, isto é, são sempre intermediados por uma de suas linhas, que constituem o grid da obra, e determina todas as possíveis posições em que os elementos podem ser colocados. Essa regra, além de outras, pode ser evidenciada em inúmeras obras do autor, que inicia sua fase abstrata a partir da década de 1920.

Figura 3 - Pintura “Composição em Vermelho, Azul e Amarelo” (1930), de Mondrian.



Fonte: <http://www.arte.seed.pr.gov.br/modules/galeria/detalhe.php?foto=403&evento=1> .

Pode se ter uma perspectiva que as obras do artista são, de certo modo, dogmáticas, por jamais deixar de utilizar os mesmos elementos formais, além dos ângulos retos e cores primárias. Essa percepção deixa de lado a intenção, até mesmo espiritual, do artista, que buscava encontrar uma arte das “relações puras”, que só poderia ser atingida a partir da utilização dos elementos e regras selecionados pelo artista. Estas tinham sido, segundo ele, omitidas da pintura, como consequência da representação dos traços da natureza na arte (SCHAPIRO, 2001).

Levando isso em consideração, é possível descrever a técnica de Mondrian como uma declaração de regras que determinam a interação entre certos elementos que ele estabelece e que podem ser variados em tamanho, posição e quantidade. Estes objetos interagem com os demais elementos dispostos na tela, criando inúmeras permutações, que apesar de manterem uma consistência gráfica, geram diferentes composições que expressam a sua intencionalidade artística (SCHAPIRO, 2001).

Por fim, cabe citar o artista Jackson Pollock, que se destaca pela sua famosa técnica do *dripping*, que pode ser compreendida como o uso de diferentes ferramentas - incluindo o próprio corpo - para disparar tinta contra a tela, disposta horizontalmente, para obter resultados únicos (Figura 4). Como coloca Gombrich (1972, p.478): “Tornando-se impaciente com os métodos convencionais, colocou suas telas no chão e pingou, derramou ou projetou suas tintas de modo a formarem configurações surpreendentes”. Essa técnica foi revolucionária para a época, considerando que entrava em conflito com séculos de tradição previamente estabelecidos, além de não ser reproduzível, seja por outros ou até mesmo pelo próprio artista.

Figura 4 - Lavender Mist Number 1 (1950), de Pollock.



Fonte: Artchive⁶.

Levando isto em consideração, a aleatoriedade pode ser identificada no seu processo, oriunda de sistemas físicos, como a gravidade, que gerava forças imprevisíveis ao interagir com essas diferentes ferramentas utilizadas pelo artista (VERBEECK, 2006). A aleatoriedade, nesse sentido, não se expressa como uma desordem ou falta de controle, mas sim enquanto uma imprevisibilidade, que ao ser incorporada em um design, permite uma maior flexibilidade nas possibilidades criativas de determinada solução (VERBEECK, 2006). É possível compreender sua técnica como uma experimentação sobre novas possibilidades de disposição da tinta, processo que não somente prevê, mas incorpora a imprevisibilidade e a aleatoriedade do processo físico envolvido.

Fica evidente, a partir do exposto, que essa vanguarda transformou permanentemente o mundo artístico, ao quebrar paradigmas tidos até então como verdades absolutas. Ao

⁶ <https://www.artchive.com/artwork/lavender-mist-number-1-1950-jackson-pollock-1950/>

realizar isso, criou novas técnicas, proporcionou a incorporação da aleatoriedade no processo artístico e, de modo muitíssimo importante, permitiu aos artistas criar imagens não representativas, possibilitando um novo tipo de sensibilidade e intencionalidade, viabilizando novas facetas de expressão da subjetividade humana.

6 DESENVOLVIMENTO DOS ALGORITMOS

A partir do entendimento desses artistas e algumas de suas técnicas, se iniciou o desenvolvimento de três algoritmos, associados respectivamente a cada um dos pintores. Conforme descrito anteriormente, um algoritmo é uma sequência finita de passos para solucionar determinada tarefa. Neste contexto específico, pode ser afirmado que esse conjunto de regras e instruções servem para produzir um artefato gráfico, refletindo, no seu processo, técnicas associadas aos artistas analisados. Os algoritmos foram classificados com base em duas características: a primeira diz respeito a sua natureza *input/output*, isto é, se são determinísticos ou randômicos. A segunda, por sua vez, condiz com qual paradigma do Design Computacional o algoritmo usou como base processual, facilitando o seu planejamento e a sua execução.

Vale esclarecer por que foi utilizado apenas o paradigma do Design Paramétrico, Algorítmico e Generativo. Conforme se evidenciou anteriormente, a partir do Design Computacional, cinco tipos distintos de algoritmos foram identificados, sendo dois destes combinações dos demais. Esse tipo de intersecção traz consigo possibilidades únicas de criação. Todavia, ao mesmo tempo, gera um diluimento das características individuais dos paradigmas que compõem essa combinação, além de possuírem processos mais complexos, uma vez que possuem diferentes lógicas processuais simultaneamente. Levando isso em consideração, e tendo em mente o escopo do presente trabalho, somente as categorias base do Design Computacional foram aplicadas.

Com o intuito de facilitar a compreensão dos códigos a seguir, vale destacar o fluxo de funcionamento da biblioteca *Processing*. Antes da execução do código que desenha efetivamente o produto gerado, existe uma função denominada *setup*, que define variáveis, a velocidade de execução do programa, o tamanho da tela, a cor do fundo da tela, dentre outras configurações. Tudo que serve para ajustar o ambiente de funcionamento da função de

desenho é inserido nessa parte. Por conta disso, é sempre chamada somente uma vez, ao executar o algoritmo. Paralelamente, existe a função *draw*, que é chamada a cada quadro. Enquanto o código estiver em funcionamento, ela vai ser chamada repetidamente.

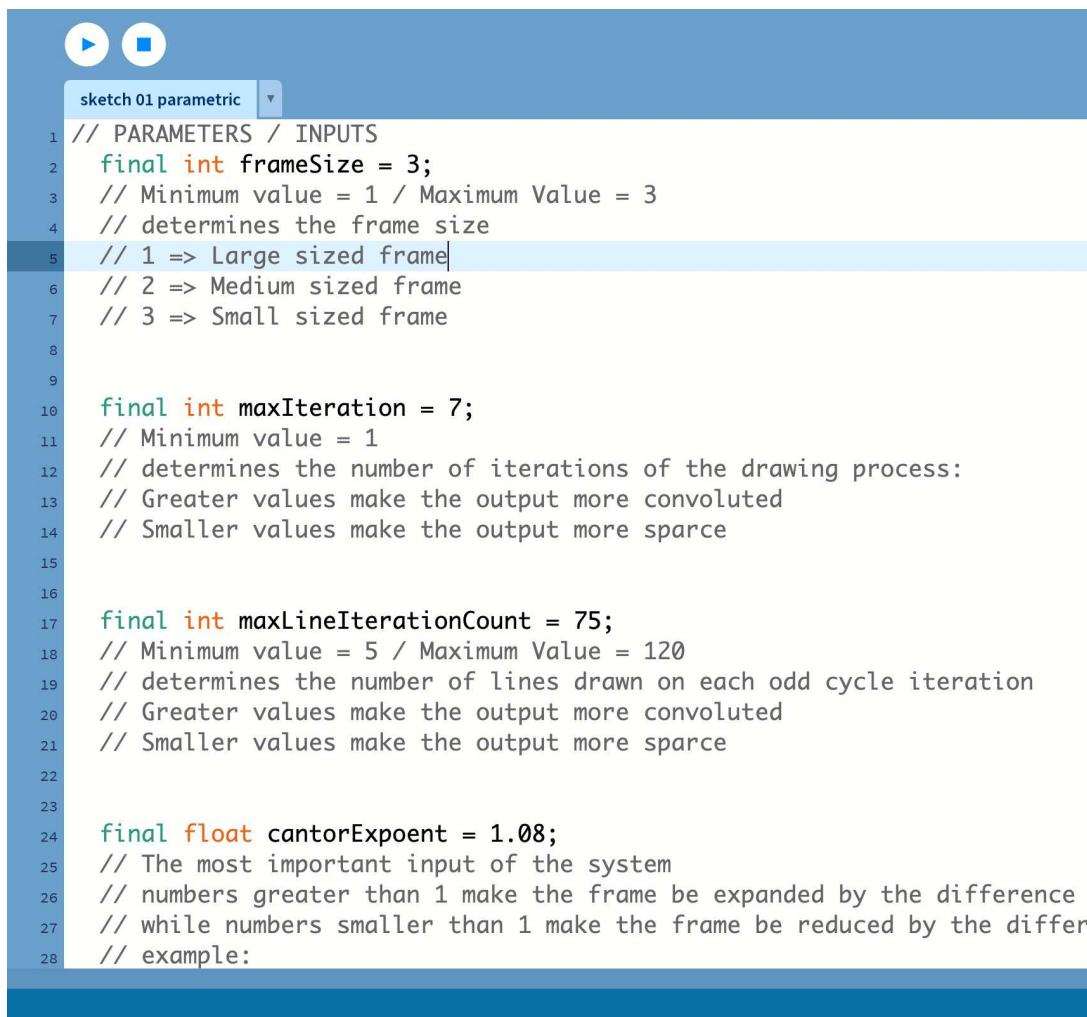
Antes das definições, é preciso deixar evidente: o objetivo do presente trabalho nunca foi recriar nenhuma das obras dos artistas selecionados, nem mesmo produzir algo similar graficamente, mas sim compreender suas técnicas e traduzir isso computacionalmente. O ganho deste processo reside na possibilidade de utilizar processos computacionais, descritos pelos paradigmas do Design Computacional, para gerar resultados gráficos únicos, associados a especificidades que o uso da computação propicia, confirmando seu potencial criativo.

6.1 Paramétrico

O primeiro algoritmo desenvolvido foi baseado no artista Mondrian, associando-o ao paradigma do Design Paramétrico. Essa escolha foi feita por se perceber uma possibilidade de diálogo entre a técnica do artista e os processos descritos por esse campo do Design Computacional, uma vez que é possível atribuir aos parâmetros a responsabilidade de governar a distribuição e a relação entre os elementos. Assim sendo, possui quatro *inputs* (Figura 5), que determinam a área de desenho, o número de iterações, o número de linhas desenhadas e o expoente de cantor⁷. Dados os mesmos valores para esses parâmetros, os resultados serão sempre idênticos, sendo, deste modo, um algoritmo determinístico. A lógica do seu funcionamento pode ser compreendida a partir do seu pseudocódigo (Figura 6).

⁷ Termo que se refere ao valor de multiplicação aplicado ao tamanho do frame a cada ciclo do código, sendo elevado com base no contador de iteração. Adaptado a partir do Conjunto de Cantor, um subconjunto do intervalo $[0,1]$, definido pelo matemático Georg Cantor.

Figura 5 - Parâmetros do algoritmo baseado no Design Paramétrico.



```
sketch 01 parametric

1 // PARAMETERS / INPUTS
2 final int frameSize = 3;
3 // Minimum value = 1 / Maximum Value = 3
4 // determines the frame size
5 // 1 => Large sized frame
6 // 2 => Medium sized frame
7 // 3 => Small sized frame
8
9
10 final int maxIteration = 7;
11 // Minimum value = 1
12 // determines the number of iterations of the drawing process:
13 // Greater values make the output more convoluted
14 // Smaller values make the output more sparse
15
16
17 final int maxLineIterationCount = 75;
18 // Minimum value = 5 / Maximum Value = 120
19 // determines the number of lines drawn on each odd cycle iteration
20 // Greater values make the output more convoluted
21 // Smaller values make the output more sparse
22
23
24 final float cantorExponent = 1.08;
25 // The most important input of the system
26 // numbers greater than 1 make the frame be expanded by the difference
27 // while numbers smaller than 1 make the frame be reduced by the differ
28 // example:
```

Fonte: elaborada pelo autor.

Figura 6 - Pseudocódigo do algoritmo baseado no Design Paramétrico.

```

// CONFIGURAÇÕES INICIAIS
INICIO setup()
    FULLSCREEN() // Tela cheia
    BACKGROUND(#282828) // Cor de fundo
    DEFINIR frameRate COMO 4 // Velocidade da execução
    DEFINIR tamanho do frame
FIM

// LOOP DE DESENHO
INICIO draw()
    SE iterationEnded ENTAO
        ESCREVER "Desenho finalizado"
        NOLOOP() // Finaliza a execução
        RETORNAR
    FIM SE

    DEFINIR tamanhoAtual COMO tamanho do frame * fator de escala(iterationCount)

    SE iterationCount for ÍMPAR ENTAO
        DESENHAR RETÂNGULO NO CANTO ESQUERDO
        DESENHAR LINHAS VERTICais DENTRO DO RETÂNGULO
    SENOaO
        DESENHAR RETÂNGULO NO CANTO DIREITO
        DESENHAR LINHAS VERTICais DENTRO DO RETÂNGULO
    FIM SE

    SE iterationCount >= maxIteration ENTAO
        iterationEnded ← VERDADEIRO
    FIM SE

    iterationCount ← iterationCount + 1
FIM

```

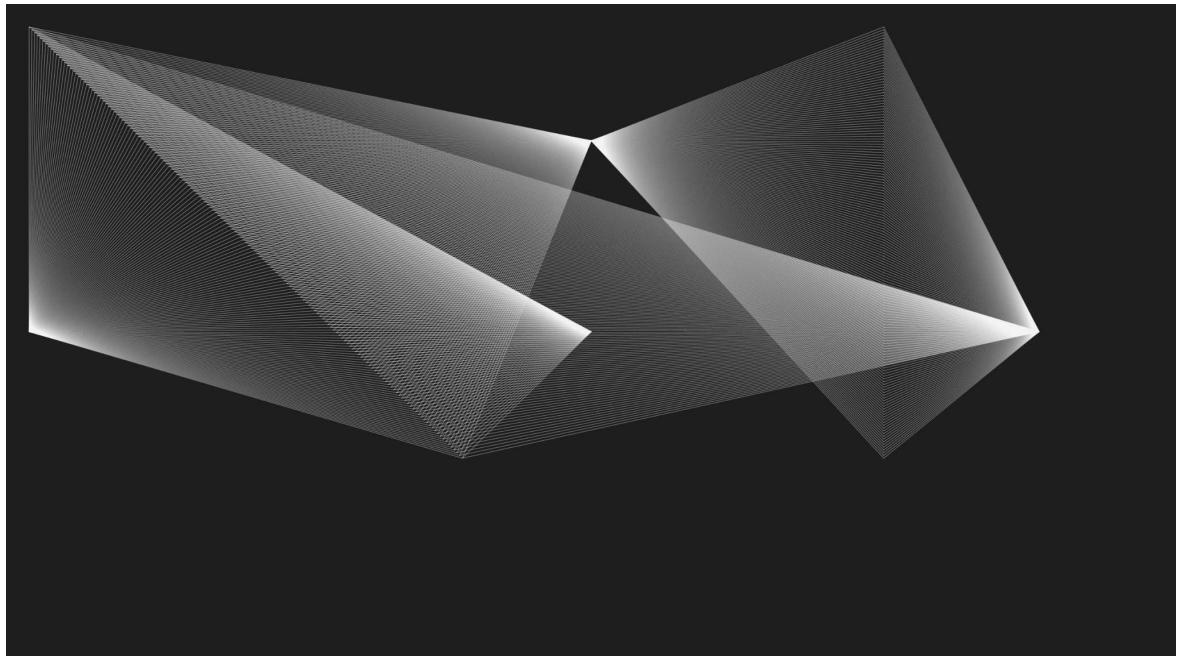
Fonte: elaborada pelo autor.

O algoritmo funciona do seguinte modo: inicialmente, no *setup*, a tela é posta em sua resolução máxima, a cor preta (#282828) é atribuída a tela, a velocidade de execução do código é posta como quatro quadros por segundo e, por fim, define-se o tamanho da moldura, com base no parâmetro *frameSize*. Na função *draw*, por sua vez, verifica-se se a variável *iterationEnded* é verdadeira. Em caso verdadeiro, a execução do código é finalizada. Prossegue-se definindo o tamanho da moldura, com base no expoente de cantor e na variável que define qual a iteração do algoritmo. Na primeira iteração, esse tamanho é o próprio tamanho do molde, definido no *setup*, uma vez que a potência de um número elevado a zero é igual a um. Deste modo, segue-se sucessivamente, desenhando retângulos, círculos e linhas em certas posições, a depender se *iterationCount* é ímpar ou par. No final de cada ciclo,

incrementa-se essa variável, para que ao chegar no valor máximo de repetições, definido no parâmetro *maxIteration*, o código seja encerrado.

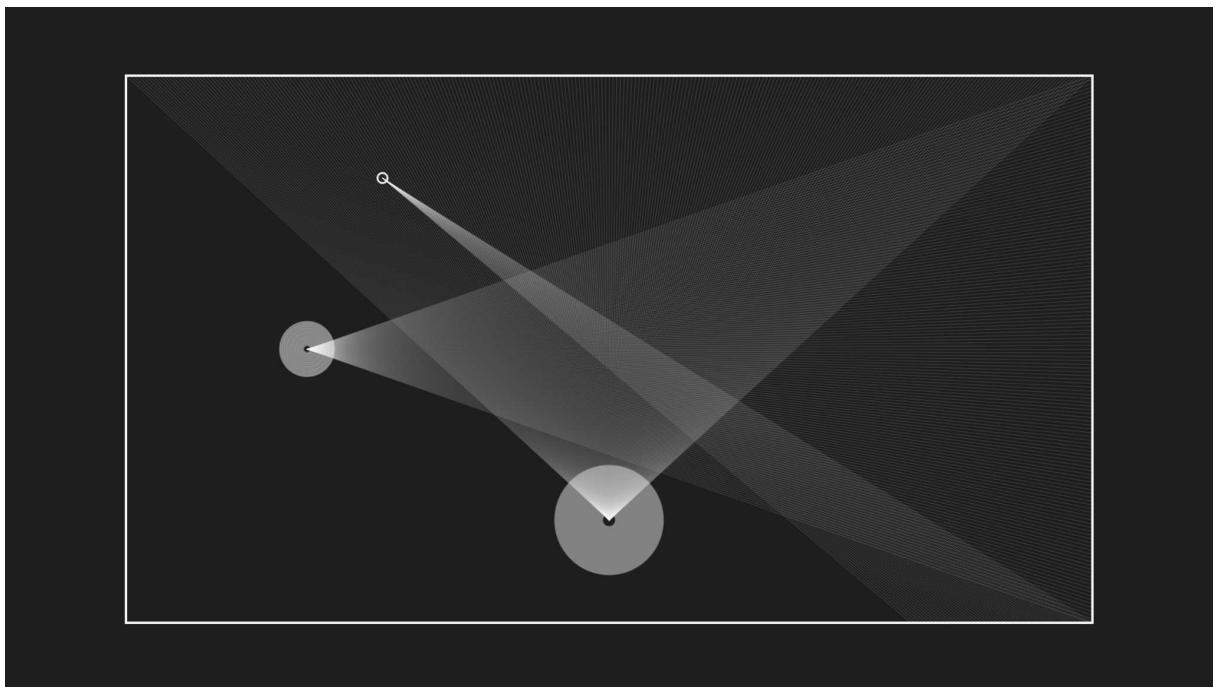
Conforme descrito na seção anterior, a técnica do artista consiste na definição de elementos - formas geométricas, cores, dentre outros - e regras que vão determinar a interação entre essas partes. A tradução desta lógica para a linguagem computacional foi relativamente direta: fez-se necessário determinar quais itens estariam dispostos na tela e definir as regras que iriam gerir a sua distribuição. A seleção destes se pautou em preceitos similares aos de Mondrian: a fim de se obter uma certa organização e clareza visual, somente formas geométricas básicas, como a linha e o retângulo, foram utilizadas. A partir disso, se iniciaram os primeiros experimentos, com linhas (Figura 7) e linhas, retângulos e circunferências (Figura 8).

Figura 7 - Resultado I do algoritmo baseado no Design Paramétrico.



Fonte: elaborada pelo autor.

Figura 8 - Resultado II do algoritmo baseado no Design Paramétrico.



Fonte: elaborada pelo autor.

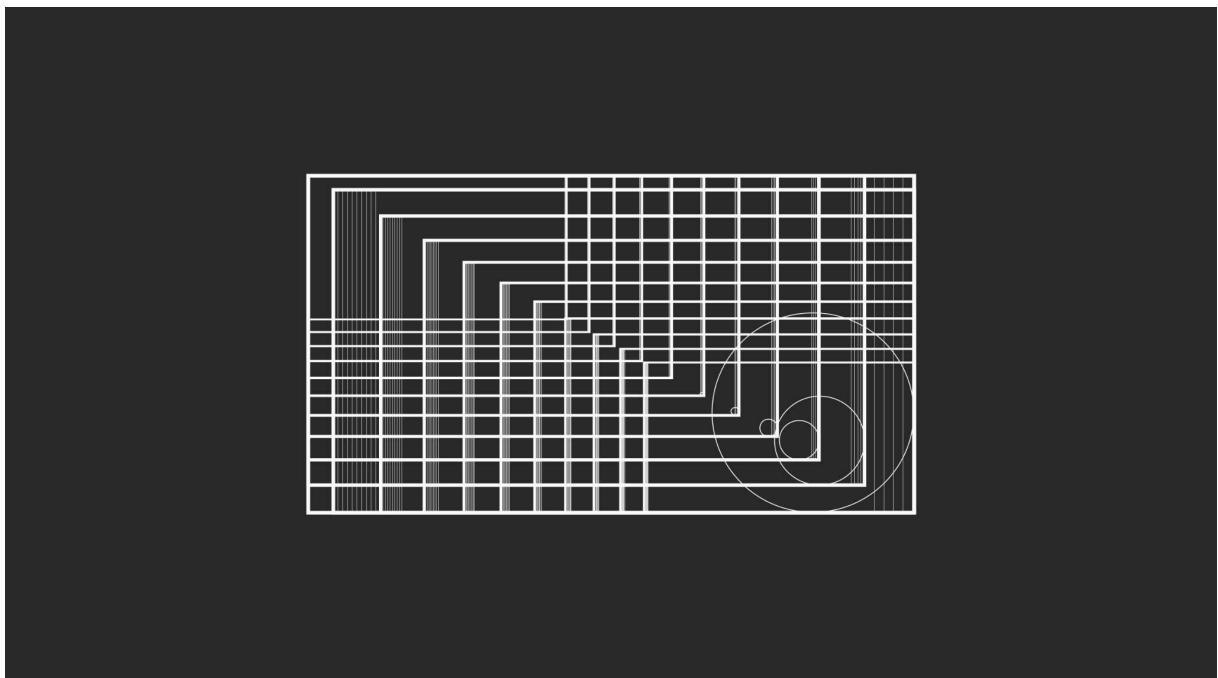
No que diz respeito às regras de distribuição, isso se demonstrou mais complexo do que o esperado: nas primeiras tentativas, a disposição dos elementos foi de fato realizada com facilidade, mas apresentou-se “arbitrária demais”, como pode ser observado nos primeiros *outputs* gerados.

Diante disso, foi preciso estabelecer, antes de tudo, as regras que iriam governar esta distribuição. A resposta para esse dilema surgiu a partir do estudo de uma regra matemática: o Conjunto de Cantor, que estabelece um padrão recursivo⁸ e fractal.⁹ Seguindo o princípio deste conjunto, retira-se um terço de um valor inicial, obtendo um novo resultado, e assim sucessivamente, retirando um terço do próximo valor, com base no número de repetições desejado. A partir da aplicação disso, obteve-se resultados mais satisfatórios (Figura 9), uma vez que é possível identificar um certo padrão matemático nos produtos gerados. Apesar disso, não é tão previsível a ponto de ser repetitivo ou monótono (Figura 10).

⁸ Recursividade se refere a uma propriedade daquilo que pode se repetir indefinidamente.

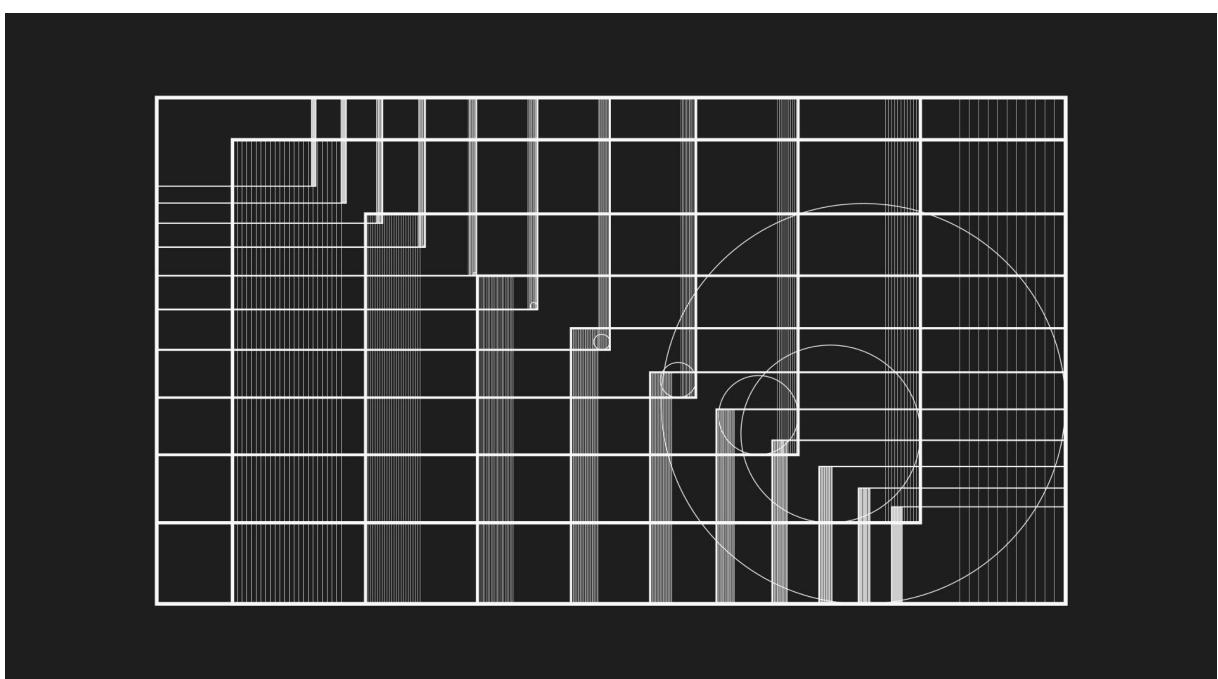
⁹ Estrutura matemática encontrada na natureza, com estruturas auto semelhantes em diferentes escalas.

Figura 9: Resultado III do algoritmo baseado no Design Paramétrico.



Fonte: elaborada pelo autor.

Figura 10: Resultado IV do algoritmo baseado no Design Paramétrico.



Fonte: elaborada pelo autor.

Graficamente, buscou-se a produção de obras que ressaltem a distribuição dos elementos como fator central. Para atingir grande clareza visual, foram utilizadas somente duas cores: a cor preta, levemente rebaixada para seu índice hexadecimal¹⁰ #282828, e a cor branca, no seu índice #FAFAFA. Tal decisão foi tomada com o intuito de não haver uma disputa visual entre os objetos da tela, mas sim uma harmonia entre eles.

6.2 Algorítmico

O segundo algoritmo, baseado em Kandinsky, foi planejado para refletir a técnica do artista de utilizar as cores com base no seu aspecto sensorial e sinestésico. Diante disso, o paradigma do Design Computacional escolhido foi o Design Algorítmico, com base no desejo de evidenciar uma relação entre o *input* sonoro e o resultado gráfico gerado. Assim sendo, o algoritmo precisava ter uma previsibilidade entre a entrada de dados e o *output* gerado. Como consequência disso, era necessário também ser determinístico.

O algoritmo pode ser compreendido a partir do seu principal recurso: a captura de áudio. Esse elemento serve como *input*, modificando o tamanho de linhas, com base na leitura das ondas sonoras, dispostas ao longo de um círculo, em que dimensões maiores representam amplitudes de som mais elevadas e vice-versa. Esse processo foi pensado para dar vazão ao aspecto sinestésico das obras do artista, interligando sons, cores e formas. Assim, uma das questões centrais do algoritmo se demonstrou na representação da relação entre esses elementos distintos, de modo não apenas harmônico, mas também semântico.

Conforme descrito no pseudocódigo (Figura 11), a cada segundo, o *input* sonoro é processado, modificando o comprimento das linhas distribuídas ao longo de um círculo. Esse círculo se completa a cada “x” segundos, sendo “x” determinado pelo parâmetro chamado *steps*. Após a conclusão dessa forma, uma nova é iniciada, com um tamanho reduzido em relação à anterior.

¹⁰ Hexadecimal corresponde a um sistema numérico que representa números em uma base 16, utilizando, portanto, 16 símbolos. Nesse caso, corresponde a uma cor específica.

Figura 11: Pseudocódigo do algoritmo baseado no Design Algorítmico.

```

// VARIÁVEIS GLOBAIS
DEFINIR passos COMO 2400 // Número de passos para formar um círculo
DEFINIR contador COMO 0 // Contador de iterações
DEFINIR contadorDeCírculos COMO 1 // Número de círculos desenhados

// VARIÁVEIS DO CÍRCULO
DEFINIR xCentro COMO 0
DEFINIR yCentro COMO 0
DEFINIR raioCírculo COMO 0

// VARIÁVEIS DE SOM
DEFINIR detectorDeBatida
DEFINIR amplitude
DEFINIR arquivoDeSom

INICIO setup()
    ATIVAR TELA CHEIA
    DEFINIR COR DE FUNDO COMO #F2E9CE
    ATIVAR MODO DE COR HSB
    INICIALIZAR arquivoDeSom, amplitude e detectorDeBatida
    ASSOCIAR entrada de áudio ao detectorDeBatida e amplitude
    REPRODUZIR arquivoDeSom
    DEFINIR xCentro COMO largura / 2
    DEFINIR yCentro COMO altura / 2
    DEFINIR raioCírculo COMO altura / 2 - 200
FIM

INICIO draw()
    SE contador > passos ENTAO
        contadorDeCírculos ← contadorDeCírculos + 1
        contador ← 0
        RETORNAR
    FIM SE

    DEFINIR amplitudeSom COMO AnalisarAmplitude()
    DEFINIR ângulo COMO (2 * PI / passos) * contador
    DEFINIR x COMO xCentro + (raioCírculo * cos(ângulo)) / contadorDeCírculos
    DEFINIR y COMO yCentro + (raioCírculo * sin(ângulo)) / contadorDeCírculos

    SE detectorDeBatida.Detectar() ENTAO
        multiplicador ← 3
    SENO
        multiplicador ← 1
    FIM SE

    DEFINIR tamanhoLinha COMO MapearAmplitude(amplitudeSom) * multiplicador
    DESENHAR LINHA de (x, y) para segundo ponto baseado na posição

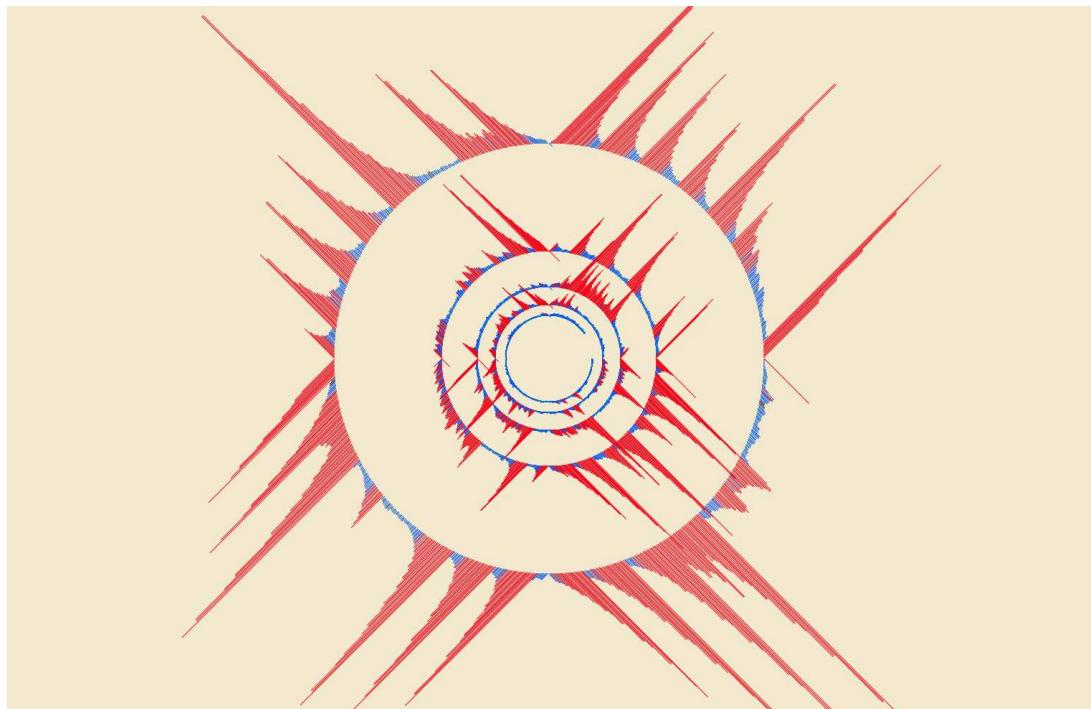
    contador ← contador + 1
FIM

```

Fonte: elaborada pelo autor.

Outro aspecto central do código foi a representação gráfica da passagem do tempo. Com o passar dos segundos, em que cada onda sonora é representada em forma de linha, com base no seu volume, surgiu a questão de como retratar isso. Inicialmente, se pensou em uma representação linear, com uma linha central da esquerda para a direita, com as demais distribuídas verticalmente ao longo de seu comprimento. No final, todavia, optou-se por utilizar um círculo cíclico (Figura 12), em que as linhas são dispostas além do diâmetro da circunferência, com base na região específica na qual elas estão. Essa decisão foi tomada por ser formalmente mais interessante, ao representar essa passagem de modo não linear.

Figura 12: Resultado I do algoritmo baseado no Design Algorítmico.

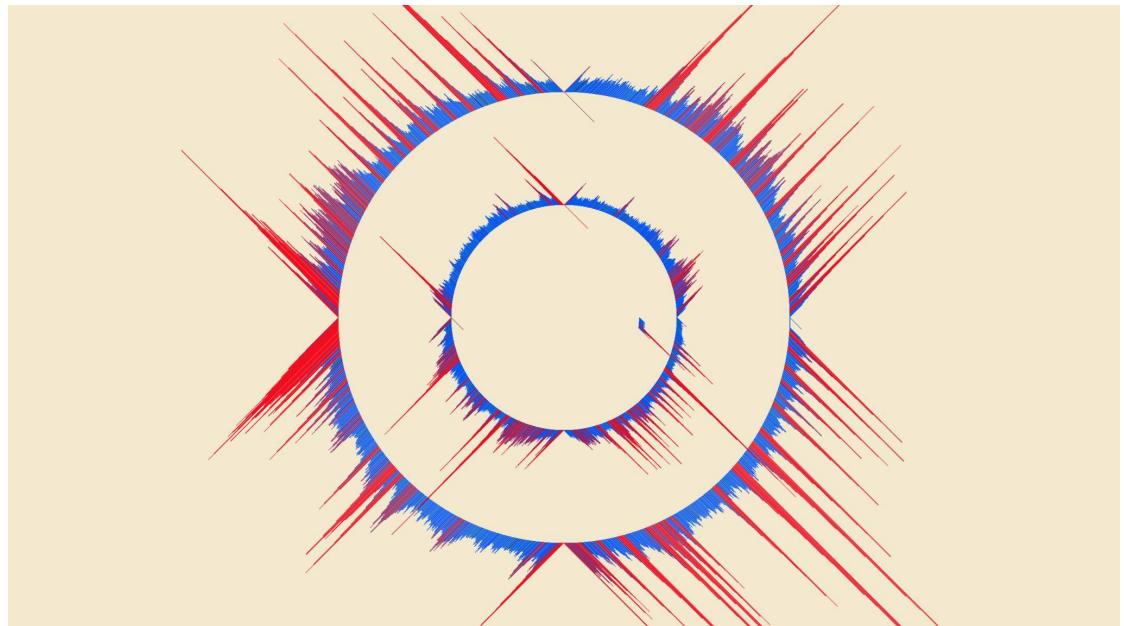


Fonte: elaborada pelo autor.

Graficamente, optou-se por utilizar uma paleta composta de cinco cores complementares, formada por, em ordem não específica: champanhe (#F2E9CE), azul escuro (#0561D9), azul (#0454F2), amora (#8C2668) e vermelho (#F20419). Tal escolha foi feita com base no desejo de utilizar a variação das cores para indicar diferentes amplitudes do som, bem como pensar na sua combinação (Figura 13), possibilitada pela redução da opacidade dos elementos.

De modo sucinto, no *setup* são inicializadas as classes responsáveis pela leitura dos elementos sonoros, além de definir a cor do fundo e inicializar a reprodução da música utilizada como input. Na função *draw* é verificado em qual círculo o código está, ao atingir a condição descrita no pseudocódigo, o próximo círculo é iniciado. A partir disso, realiza-se a leitura dos elementos sonoros. Se a função detectar que houve uma batida percussiva, multiplica-se o tamanho da linha por três. Finalmente, a linha é desenhada, com base nos valores definidos.

Figura 13: Resultado II do algoritmo baseado no Design Algorítmico.



Fonte: elaborada pelo autor.

6.3 Generativo

Por fim, temos o algoritmo baseado em Pollock. Conforme descrito anteriormente, o elemento central identificado no processo do artista foi a incorporação de elementos imprevisíveis para a geração dos seus resultados artísticos. Diante disso, o paradigma que mais se associou às necessidades dessa técnica foi o Design Generativo, uma vez que a imprevisibilidade dos *outputs* obtidos, bem como a possibilidade de se obter produtos inesperados, é uma de suas principais características.

Inicialmente, o algoritmo pode ser compreendido a partir da sua primeira etapa, em que, a partir de uma grade, uma distribuição de vetores ocorre ao longo de toda a superfície da tela, em que a direção destes é determinada aleatoriamente. Consequentemente, é um algoritmo não determinístico, uma vez que ao reproduzir o código com os mesmos valores, os resultados serão sempre distintos. Na segunda etapa, a partir de um sistema de partículas, ocorre uma instanciação de diversos círculos. A posição inicial de cada um deles é determinada aleatoriamente. Nesse cenário, cada partícula é influenciada a se movimentar continuamente com base na direção do vetor mais próximo. O resultado disso é um produto orgânico, que indica diversos caminhos que se assemelham a linhas. O pseudocódigo descreve esse fluxo de modo sintético (Figura 14).

Figura 14: Pseudocódigo do algoritmo baseado no Design Generativo.

```
// VARIÁVEIS GLOBAIS
DEFINIR escala COMO 10
DEFINIR linhas, colunas
DEFINIR campoDeFluxo COMO MATRIZ [linhas][colunas]
DEFINIR tamanhoDoEnxame COMO 6200
DEFINIR enxame COMO LISTA DE Partícula[tamanhoDoEnxame]

INICIO setup()
    ATIVAR TELA CHEIA
    ATIVAR MODO DE COR HSB
    INICIALIZAR enxame COM tamanhoDoEnxame PARTÍCULAS

    DEFINIR linhas COMO altura / escala
    DEFINIR colunas COMO largura / escala
    INICIALIZAR campoDeFluxo COM tamanho [linhas][colunas]

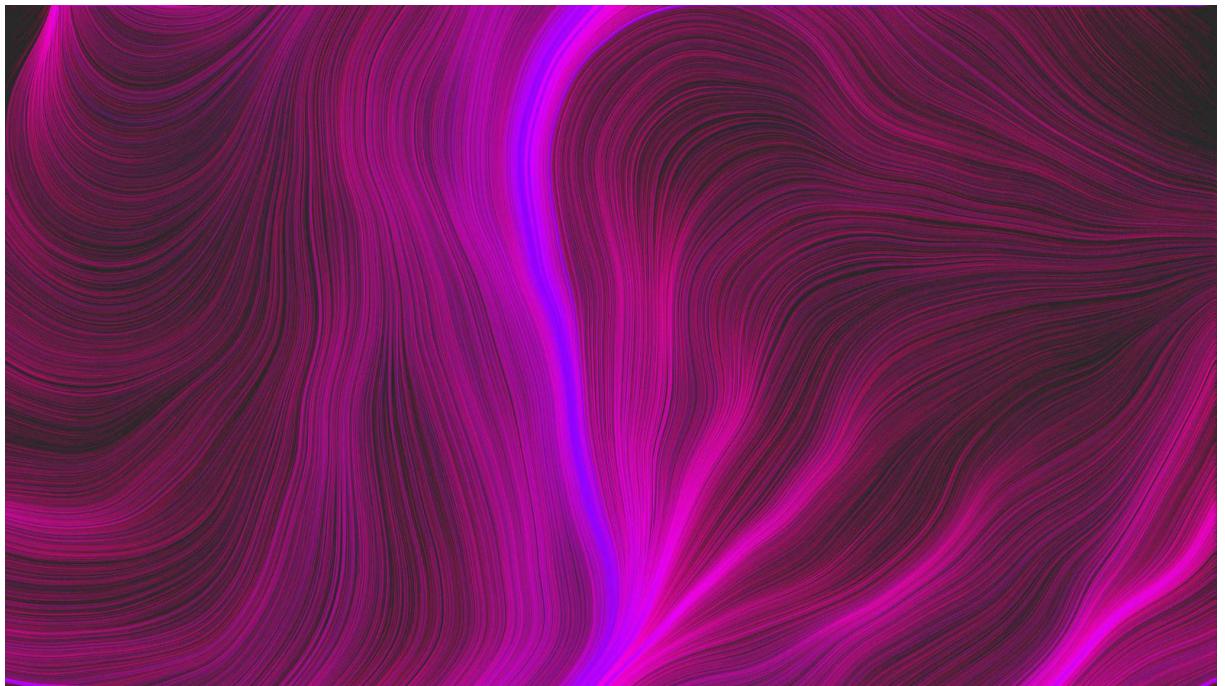
    GERAR campoDeFluxo USANDO Perlin Noise
    DEFINIR FUNDO COMO #282828
FIM

INICIO draw()
    PARA CADA partícula EM enxame FAZER
        EXIBIR partícula
        ATUALIZAR posição da partícula
    FIM PARA
FIM
```

Fonte: elaborada pelo autor.

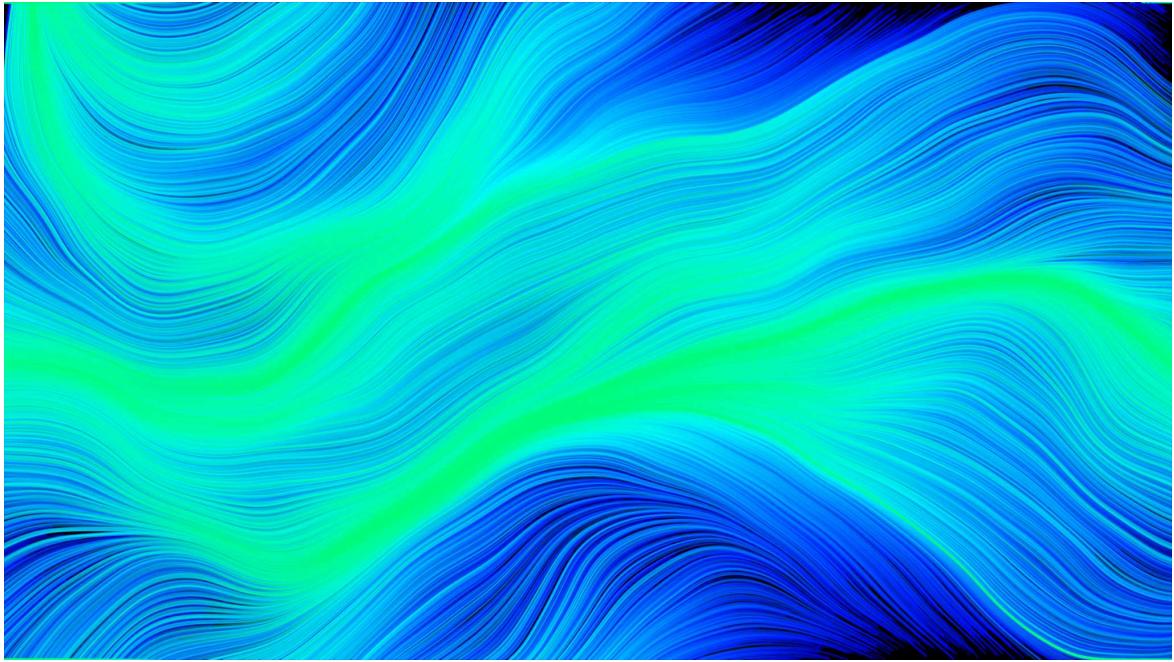
Graficamente, optou-se pela redução da opacidade das cores com o intuito de permitir o surgimento de novas cores a partir da sobreposição das partículas (Figura 15). Essa característica, associada a um fator que diminui progressivamente a opacidade destas, permitiu que resultados inesperados fossem obtidos. Além disso, a cor inicial escolhida determina não somente a si mesma, mas também a cor secundária, isto é: outro fator descrito no código reduz a matiz base, utilizada para dar cor às partículas, com o passar dos quadros do algoritmo. Esse recurso permite que cada cor utilizada produza um efeito único no resultado final, com algumas variações possuindo um aspecto bem vibrante (Figura 16), qualidade imprevista inicialmente.

Figura 15: Resultado I do algoritmo baseado no Design Generativo.



Fonte: elaborada pelo autor.

Figura 16: Resultado II do algoritmo baseado no Design Generativo.



Fonte: elaborada pelo autor.

Sinteticamente, no pseudocódigo, temos a inicialização de algumas variáveis, que determinam os elementos associados ao campo de fluxo. No *setup*, a tela é posta na sua máxima resolução, o modo de cor é posto no seu modo *HSB*¹¹, a cor preta (#282828) é associada ao fundo da tela e o campo de vetores é gerado. Na função *draw*, a partir do sistema de partículas descrito anteriormente, exibe-se cada uma e em seguida sua posição é atualizada. Isso é feito para cada uma delas, por meio de uma estrutura de repetição. Com isso, na próxima chamada desta função, suas posições vão ser distintas, permitindo a sua movimentação.

¹¹ Termo em inglês que significa *hue, saturation e brightness*, que corresponde a matiz, saturação e luminosidade, em português.

7 RESULTADOS E DISCUSSÕES

Conforme descrito na metodologia, a análise passou por quatro etapas: redução, categorização, interpretação e redação. Na redução, todos os objetos produzidos foram simplificados, de acordo com suas características constitutivas. A categorização classificou esses objetos. A interpretação discorre a respeito destes dados agrupados, buscando associá-los aos objetivos específicos do trabalho. Finalmente, tudo isso foi descrito em um relatório, fechando a última etapa da investigação.

Podemos categorizar os algoritmos conforme o seu paradigma do Design Computacional e quanto a sua natureza determinística ou randômica, conforme discutido na sessão anterior. Nesse sentido, temos dois algoritmos determinísticos e um randômico. Eles seguem os paradigmas originários do Design Computacional, sendo um paramétrico, um algorítmico e um generativo. A interpretação deles vai se pautar nos objetivos específicos definidos anteriormente.

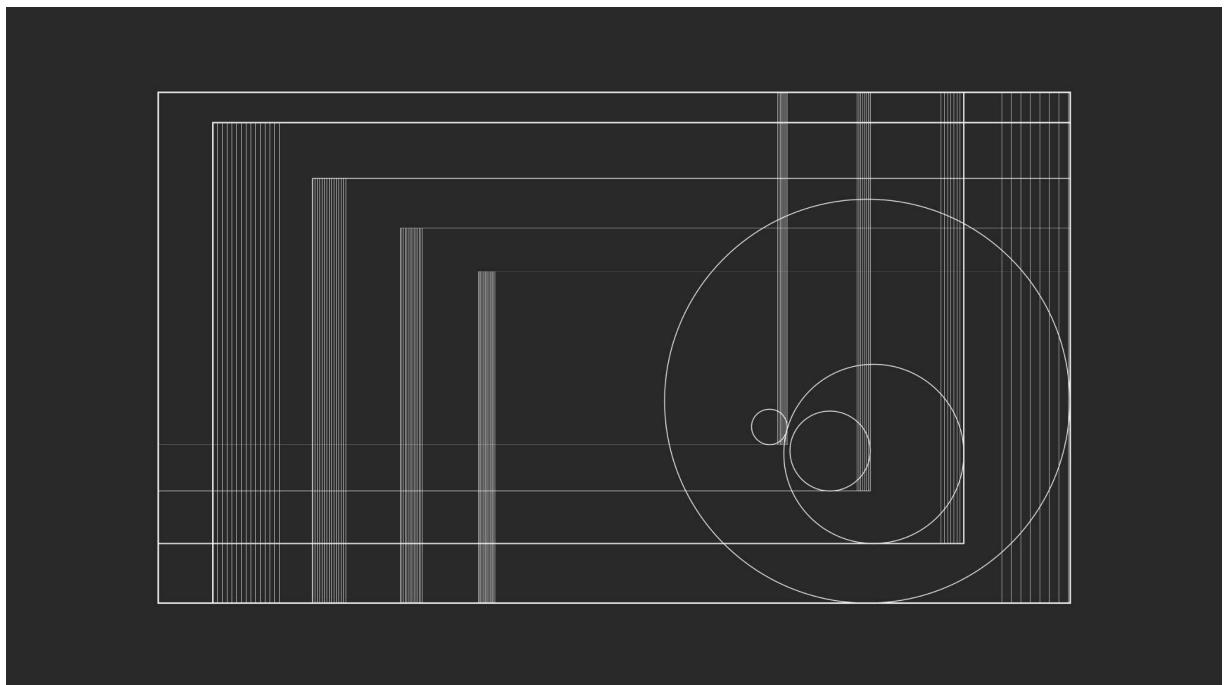
Iniciando a partir do Algoritmo baseado no Design Paramétrico, o maior ganho criativo deste código foi a possibilidade de aplicar regras de distribuição modificáveis pelos parâmetros descritos anteriormente. Ao serem aplicadas e iteradas repetidas vezes ao longo da execução do código, estas regras originaram diferentes resultados gráficos. Nesse sentido, houve um grande ganho de experimentação, uma vez que a modificação de um simples parâmetro e a execução do código correspondente gerou rapidamente um novo resultado observável, sem necessidade de grandes alterações.

Além disso, foi possível se utilizar da natureza numérica da estrutura de repetição para aplicar certas distribuições de elementos somente em iterações ímpares, por exemplo. Nesse sentido, o processo computacional se demonstrou altamente efetivo na execução destas tarefas, por serem calculáveis e cíclicas. O paradigma do Design Paramétrico, por sua vez, serviu como base para pensar em como desenvolver um algoritmo que não definia a forma final, mas que somente declarava elementos e regras, que eram reguladas pelos *inputs* estabelecidos.

Os parâmetros deste algoritmo determinam, respectivamente, o tamanho do frame, o número máximo de ciclos, a quantidade de linhas desenhadas e o expoente de cantor. Nesse sentido, os parâmetros do produto final I (figura 17) foram: 1, 7, 75 e 0.94. Os do produto

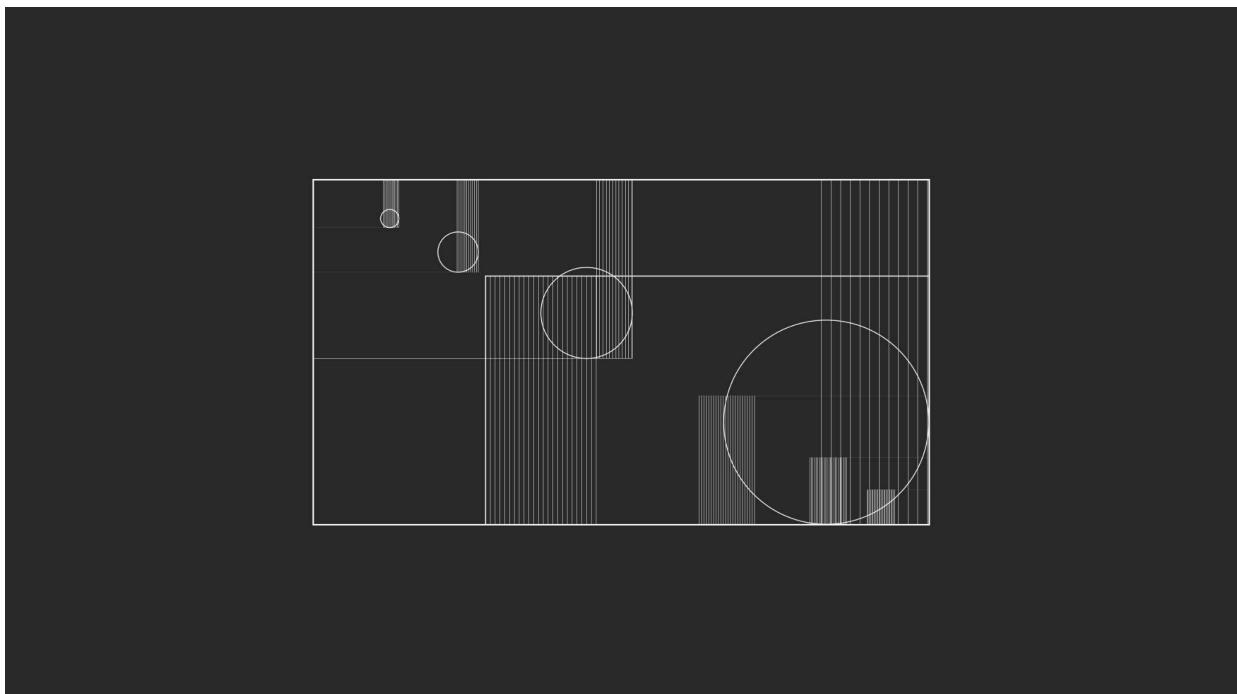
final II (figura 18): 2, 7, 120 e 0.72. Finalmente, os do produto final III (figura 19): 3, 6, 100 e 1.09.

Figura 17: Resultado final I do algoritmo baseado no Design Paramétrico.



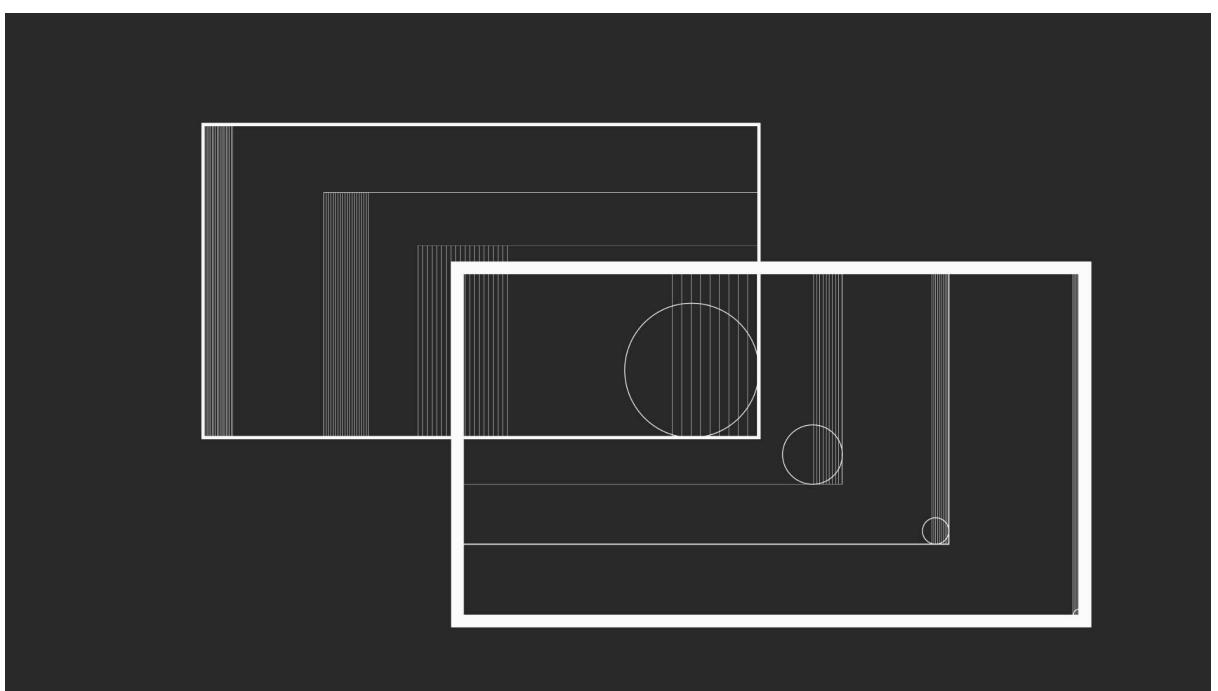
Fonte: elaborada pelo autor.

Figura 18: Resultado final II do algoritmo baseado no Design Paramétrico.



Fonte: elaborada pelo autor.

Figura 19: Resultado final III do algoritmo baseado no Design Paramétrico.

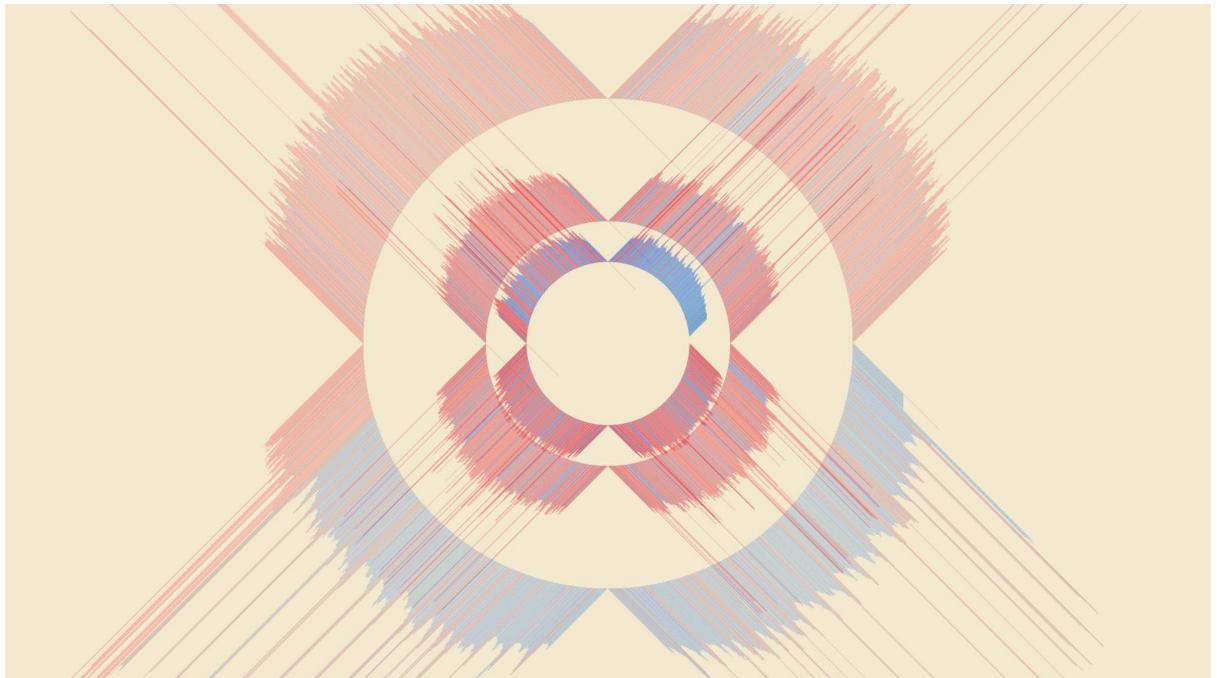


Fonte: elaborada pelo autor.

O Algoritmo baseado no Design Algorítmico, inspirado em Kandinsky, por sua vez, teve um grande ganho criativo ao permitir o uso de elementos físicos, como a amplitude e a forma das ondas sonoras de determinada música, como *inputs* para o resultado produzido. Essa possibilidade não se restringe ao mundo computacional, mas quando comparado a versão analógica, o computador demonstra uma de suas maiores qualidades: a capacidade de lidar com grandes volumes de dados. Isso permite não somente o processamento destes dados em tempo real, mas também incorporar diversos elementos para computação simultaneamente. Essa oportunidade permitiu analisar a amplitude e a forma das ondas de modo concomitante, gerando um resultado mais sensível aos sons em questão usados como base. Isso garantiu que ao usar diferentes músicas como *input*, produtos distintos fossem obtidos, evidenciando uma maior riqueza de aplicações do algoritmo em questão.

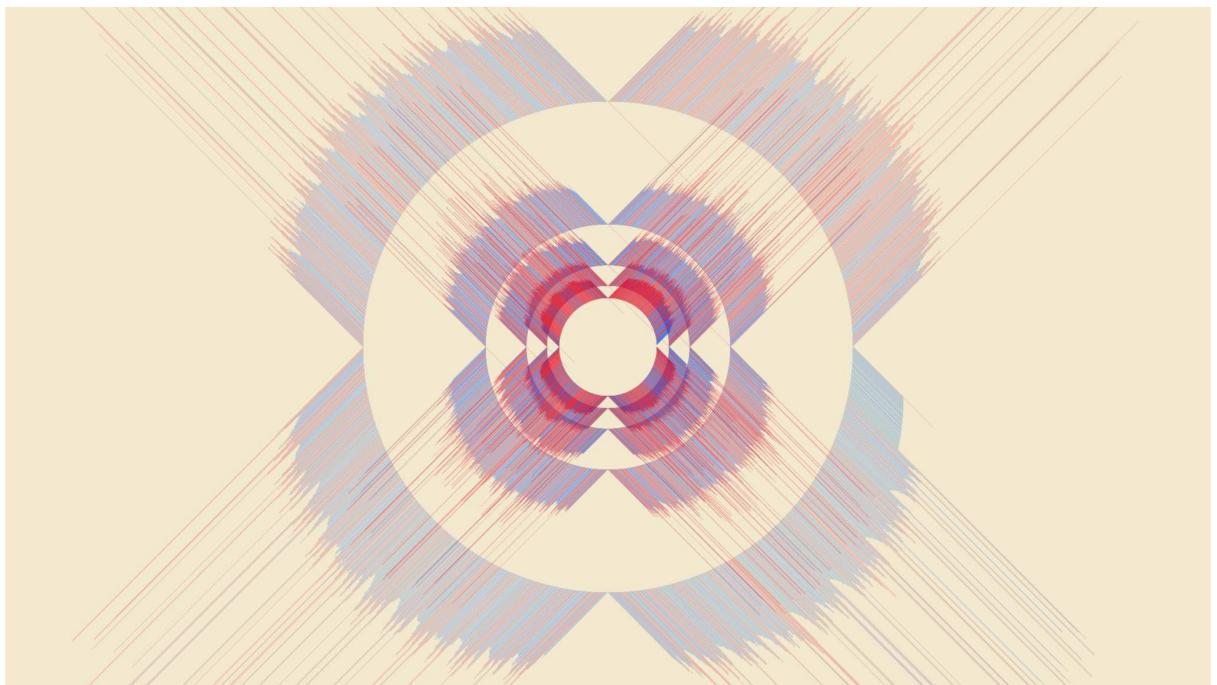
Foram utilizadas três músicas para a geração dos resultados: “Luv(sic.) pt3”, do artista Nujabes (Figura 20), “Disorder”, da banda Joy Division (Figura 21) e “Computadores Fazem Arte”, da banda Chico Science e Nação Zumbi (Figura 22).

Figura 20: Resultado final I do algoritmo baseado no Design Algorítmico.



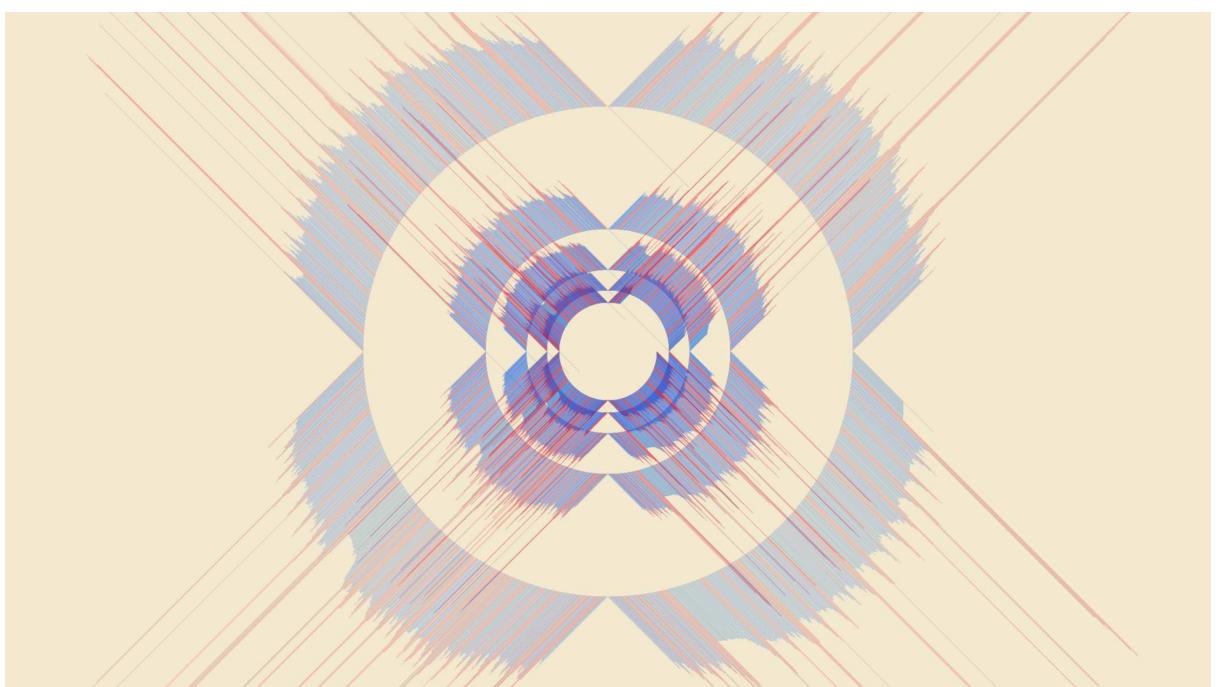
Fonte: elaborada pelo autor.

Figura 21: Resultado final II do algoritmo baseado no Design Algorítmico.



Fonte: elaborada pelo autor.

Figura 22: Resultado final III do algoritmo baseado no Design Algorítmico.



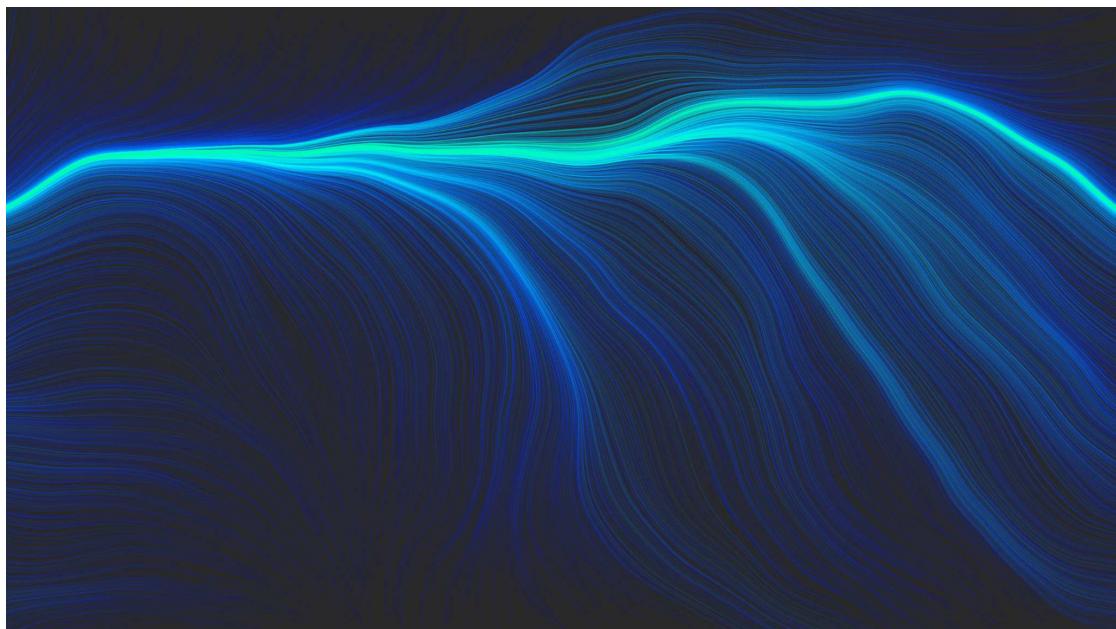
Fonte: elaborada pelo autor.

Por fim, temos o Algoritmo baseado no Design Generativo, inspirado em Pollock. Ele se destacou muito por sua natureza imprevisível, influenciada pelos diversos fatores aleatórios, como a posição inicial dos elementos e a direção dos vetores, usados como base para o funcionamento do código. Ao serem vinculados, esses elementos geram combinações imprevistas a cada execução do algoritmo (Figura 23). Conforme descrito anteriormente, essa característica também não se restringe ao mundo computacional, mas a capacidade de utilizar esses fatores em escala exponencialmente maior se destaca como a principal característica de algoritmos que usam o Design Generativo como base.

Para sintetizar tal afirmação, vale lembrar que a direção de todos os vetores e a posição inicial de todas as partículas utilizadas no algoritmo são determinadas deste modo, o que permite possibilidades na casa dos milhões. Essa vastidão permite determinar somente como o algoritmo vai funcionar, mas os resultados surgem de uma forma curiosamente singular, característica que merece grande destaque.

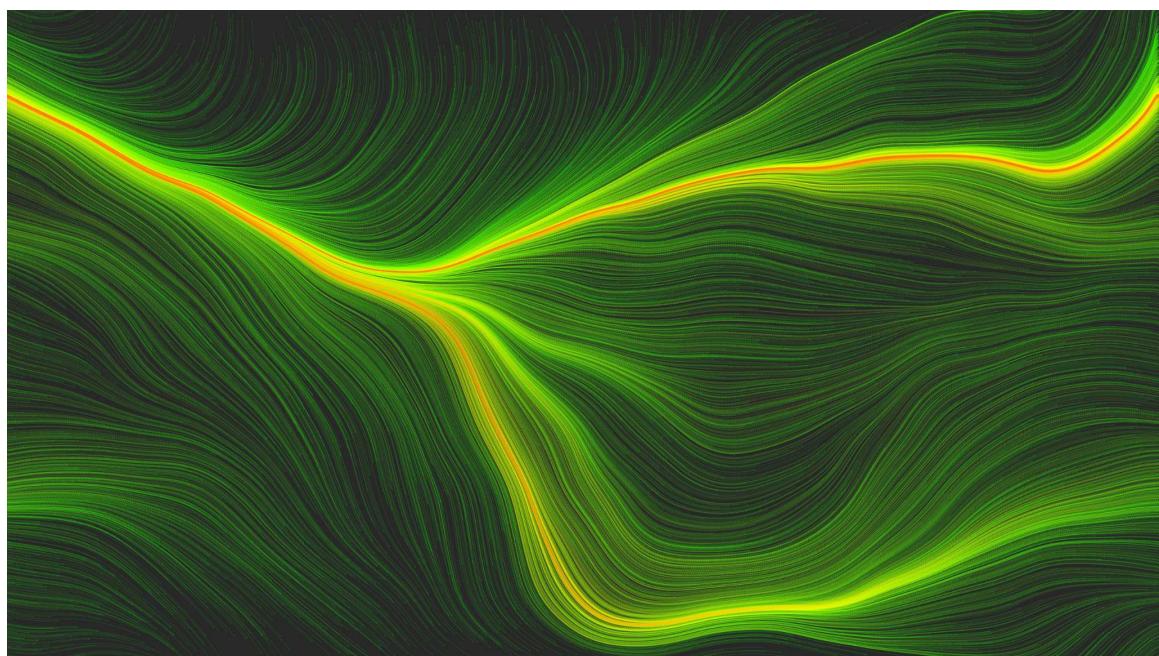
Outra qualidade que vale ser realçada é o resultado das cores, obtido a partir da redução da opacidade das partículas, permitindo que elas se misturem e criem combinações (Figura 24), reforçando ainda mais o aspecto inesperado dos resultados. Por conta disso, a capacidade de experimentação ganha bastante ênfase, uma vez que a mudança dos valores associados às cores utilizadas gera uma nova gama de resultados (Figura 25).

Figura 23: Resultado final I do algoritmo baseado no Design Generativo.



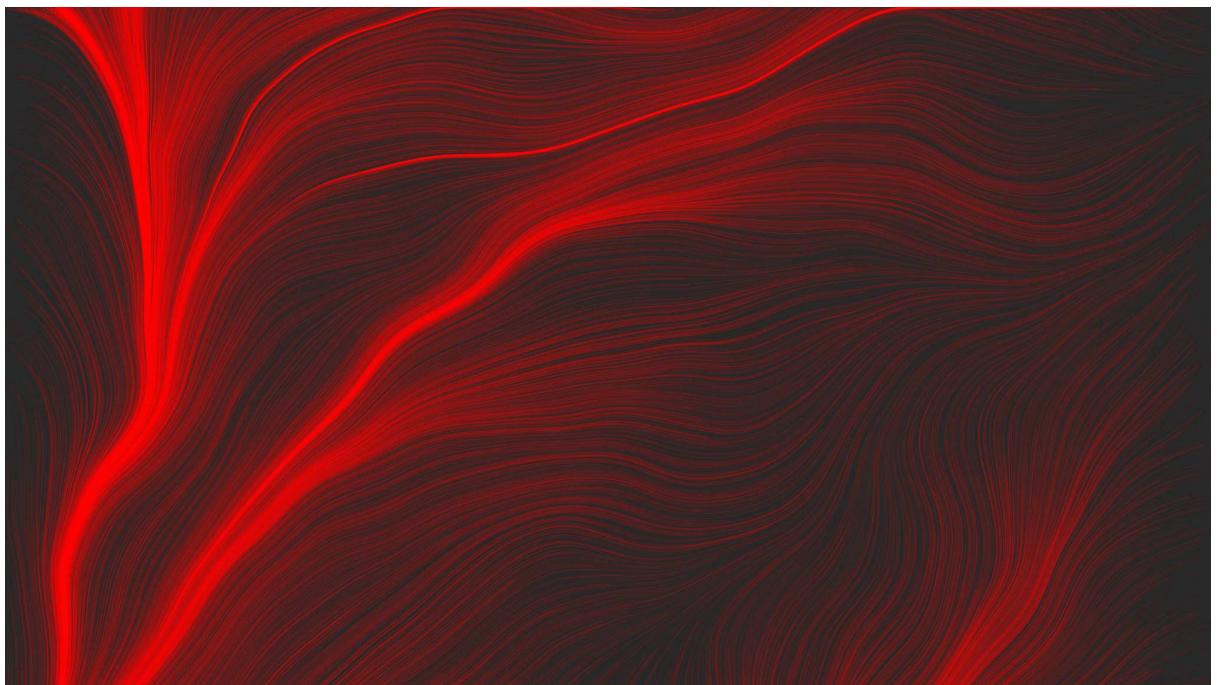
Fonte: elaborada pelo autor.

Figura 24: Resultado final II do algoritmo baseado no Design Generativo.



Fonte: elaborada pelo autor.

Figura 25: Resultado final III do algoritmo baseado no Design Generativo.



Fonte: elaborada pelo autor.

8 CONSIDERAÇÕES FINAIS

A partir da investigação realizada, utilizando os paradigmas do Design Computacional enquanto processos e técnicas, foi possível identificar diversas possibilidades de aplicação da área para finalidades relacionadas à expressão criativa, conforme evidenciado pelos resultados obtidos. Todavia, vale destacar que foi realizado somente um estudo inicial sobre essa forma de aplicação, delimitado pelo escopo do presente trabalho. Nesse sentido, os paradigmas conjugados do Design Computacional, como o Design Paramétrico e Generativo ou Algorítmico, se evidenciam como possibilidades não exploradas. Um possível uso baseado neles pode ser manifesto em algoritmos que incorporam parâmetros e um fator randômico que dialoga com estes inputs para interagir com o público, condição que cria novas dinâmicas de interação entre obra e usuário, e que merece ser explorada mais a fundo.

Quando comparada à pintura, o uso desse segmento do design se destaca por incorporar características computacionais, como a capacidade de processar dados e de realizar milhares de cálculos em alta velocidade. Isso permitiu não somente a utilização destes resultados numéricos para geração das obras - como evidenciado pelos produtos do algoritmo

baseado no Design Paramétrico -, mas também a criação de obras de modo contínuo, que se mantêm em execução enquanto o código for reproduzido - exemplificado pelo algoritmo baseado no Design Algorítmico. Essas características oferecem novas condições de expressão criativa, que dificilmente poderiam ser alcançadas sem o auxílio da computação.

Além disso, a incorporação do computador no processo criativo proporcionou um grande ganho para a experimentação. Isso se originou pela extrema flexibilidade e facilidade que este tipo de aplicação permitiu para a reformulação da obra, uma vez que é possível modificar variáveis ou parâmetros com grande facilidade, e, a depender do resultado, tornar essa mudança definitiva ou reverter para a versão anterior. Nesse sentido, o processo de testar hipóteses foi facilitado, uma vez que se criou um cenário seguro para análises e modificações, característica extremamente positiva em um contexto de expressão.

Espera-se que este estudo represente um avanço no debate sobre a utilização do Design Computacional para finalidades criativas, abrindo espaço para demais investigações, que possam explorar ainda mais as interseções entre computação, criatividade e design.

REFERÊNCIAS

- ADAMATZKY, Andrew (Ed.). **Game of Life Cellular Automata**. London: Springer, 2010. ISBN 9781849962179.
- ARGAN, G Carlo. **Arte Moderna**. São Paulo: Companhia das Letras, 1992.
- BURRY, J. R.; BURRY, M. C. **Gaudí and CAD**. *Journal of Information Technology in Construction*, v. 11, p. 437-446, 2006.
- CAETANO, Inês; SANTOS, Luís; LEITÃO, António. **Computational design in architecture: Defining parametric, generative, and algorithmic design**. *Frontiers of Architectural Research*, 2020.
- CALDAS, L. **GENE_ARCH: An Evolution-Based Generative Design System for Sustainable Architecture**. In: SMITH, I. F. C. (Ed.). *Intelligent Computing in Engineering and Architecture*. EG-ICE 2006. Lecture Notes in Computer Science, v. 4200. Berlin, Heidelberg: Springer, 2006. Disponível em: https://doi.org/10.1007/11888598_12. Acesso em: 5 jun. 2024.
- CAMBRIDGE UNIVERSITY PRESS. **Cambridge Dictionary**. 2024. Disponível em: <https://dictionary.cambridge.org>. Acesso em: 12 out. 2024.
- CARVALHO, Paulo Roberto de; MANSANO, Sonia Regina Vargas. **Da arte figurativa à arte abstrata: uma análise psicológica**. Rev. Polis Psique, Porto Alegre, v. 10, n.1, p. 30-46, abr. 2020. Disponível em: [Da arte figurativa à arte abstrata: uma análise psicológica](#). Acesso em: 21 nov. 2024.
- CSTA. ACM. **CSTA K –12 Computer Science Standards**, (2016). Disponível em: <http://k12cs.org/wp-content/uploads/2016/09/K%20Computer-Science-Framework.pdf>. Acesso em 14 jan. 2025
- GIL, Antonio C. **Como Elaborar Projetos de Pesquisas**. 4. ed. 7. tir. São Paulo: Atlas, 2002.
- GOMBRICH, E. H. **História da arte**. São Paulo: Círculo do Livro, 1972
- JOY DIVISION. **Disorder**. *Unknown Pleasures*. Reino Unido: Factory Records, 1979. Disponível em: [Disorder \(2007 Remaster\)](#). Acesso em: 21 fev. 2025.
- KANDINSKY, Wassily. **Concerning the spiritual in art**. Revised edition. New York: Dover Publications, 1977.
- KISS, E. "Criatividade, design e inovação". **Portal Design Brasil**, 17 dez. 2005. Disponível em: <https://acervo-digital.espm.br/Artigos/ART/119428.pdf>. Acesso em: 09 set. 2024.
- KOLAREVIC, B. (2000). **Digital morphogenesis and computational architectures**. Em Congress of the Iberoamerican Society of Digital Graphics, 10. (2000), Rio de Janeiro.
- LIESER, Wolf. **Arte digital**. Königswinter: h.f.ullmann, 2009.

MARTINHON, Carlos A. **Algoritmos Randônicos em otimização combinatória**. In: XXXIV Simpósio Brasileiro de Pesquisa Operacional, 2002. Rio de Janeiro, RJ.

MINEIRO, Érico; JOSÉ DA SILVA, Fernando. **Design computacional: escopo e alcance de uma área em formação**. Pensamentos em Design, [S. l.], v. 3, n. 2, p. 123–133, 2024. DOI: 10.36704/pendes.v3i2.8585. Disponível em: <https://revista.uemg.br/index.php/pensemdes/article/view/8585>. Acesso em: 10 mar. 2025.

NAÇÃO ZUMBI. **Computadores Fazem Arte. Da Lama ao Caos**. Brasil: Sony Music Entertainment Brasil ICL, 1994. Disponível em: <https://www.youtube.com/watch?v=JyfJyfJyfJy>. Acesso em: 21 fev. 2025.

NUJABES. **Luv(sic.) pt3 (feat. Shing02)**. *Modal Soul*. Japão: Hydeout Productions, 2005. Disponível em: <https://www.youtube.com/watch?v=JyfJyfJyfJy>. Acesso em: 21 fev. 2025.

PARENTE, A. (Org.) (1993). **Imagen máquina: A era das tecnologias do virtual**. Rio de Janeiro: Editora 34.

QUEIROZ, N.; VAZ, C. **Designing a building envelope using parametric and algorithmic processes**. In: CONFERENCE OF THE IBEROAMERICAN SOCIETY OF DIGITAL GRAPHICS, 19., 2015. *Anais...* [S.l.: s.n.], 2015.

REAS, C.; FRY, B. **Processing: programming for the media arts**. *AI & Society*, v. 20, p. 526–538, 2006. Disponível em: <https://doi.org/10.1007/s00146-006-0050-9>. Acesso em: 5 jun. 2024.

ROHENKOL, R. A. S. **Criatividade e Design: uma análise da habilidade criativa no processo projetual**. *Unoesc & Ciência - ACSA*, v. 3, n. 1, p. 45–54, 2012. Disponível em: <https://periodicos.unoesc.edu.br/acsa/article/view/1484>. Acesso em: 9 set. 2024.

SCHAPIRO, Meyer. **A Dimensão Humana na Pintura Abstrata**. São Paulo: Cosac Naify, 2001.

SETZER, Valdemar W. **O computador como instrumento de anti-arte**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 8., 1997, São José dos Campos. *Anais [...]*. São José dos Campos: Sociedade Brasileira de Computação, 1997. p. 509-530. Disponível em: <https://www.ime.usp.br/vwsetzer/comp-art-port.html#:~:text=Um%20computador%20pode%20produzir%20desenhos,algo%20que%20aparentemente%20é%20semelhante>. Acesso em: 4 ago. 2024.

TERZIDIS, Kostas. 2004. **Algorithmic Design: A Paradigm Shift in Architecture?**

VERBEECK, Kenny. **Randomness as a Generative Principle in Art and Architecture**. Instituto de Tecnologia de Massachusetts. 2006.

APÊNDICE A – LINK DE ACESSO PARA OS CÓDIGOS NO GITHUB

<https://github.com/gusholz/tcc>

APÊNDICE B - LINK DE ACESSO AOS ALGORITMOS EM FUNCIONAMENTO

https://www.youtube.com/watch?v=p4JPj_E5ujQ&list=PLJmMJ179knestu1Rn406J-C0P_NP6_qSJ&index=1