

PADRÕES GENERATIVOS COMPLEXOS APLICADOS NO DESIGN DE SUPERFÍCIES

COMPLEX GENERATIVE PATTERNS APPLIED ON SURFACE DESIGN

Mariana Araújo Laranjeira¹

Marizilda dos Santos Menezes²

Resumo

A complexidade é uma característica, estudada dentro das ciências, cada vez mais presente em projetos interdisciplinares. Ao trazer inovações na construção do pensamento, ela pode contribuir para a área do design ao influenciar nos métodos de criação e de composição visual. Neste contexto, o design de superfícies é uma especialidade que pode utilizar os conceitos pertinentes à complexidade e aos sistemas complexos, principalmente no uso de novas tecnologias digitais com algoritmos de programação e em novos processos produtivos computadorizados. Este artigo resume o estudo dessa temática, explorado com mais detalhes na dissertação de mestrado “*Design e Complexidade: Investigação do design de padrões generativos e o uso algoritmos computacionais*”, apresentado no programa de pós-graduação em Design na Universidade Estadual Paulista ‘Julio de Mesquita Filho’. Por meio de um estudo de caso, foram pesquisados os projetos que envolvessem os princípios do desenvolvimento de padrões generativos aplicados à área têxtil, estabelecendo a conexão de superfícies bidimensionais e tridimensionais com os sistemas complexos e a utilização de softwares de programação algorítmica. A pesquisa culminou em experimentações para criação de algoritmos de padronagem complexa cujos resultados são apresentados resumidamente no presente artigo.

Palavras-chave: design de superfície; complexidade; design têxtil; algoritmos; padronagem.

Abstract

Complexity is a characteristic, studied within the sciences, increasingly present in interdisciplinary projects. By bringing innovations in the construction of thought, it can contribute to design field by orienting in the methods of creation and visual composition. In this context, surface design is a specialty that can use the concepts relevant to complexity and complex systems, mainly in new digital technologies through programming algorithms and in new computerized production processes. This article reports the study of this theme, explored in more detail in the master's thesis “*Design and Complexity: Investigating the design of generative patterns and the use of computational algorithms*”, presented in the Design graduate program at Universidade Estadual Paulista 'Julio de Mesquita Filho', at a master's level. Through a case study, projects that involve the principles of the development of

¹ Mestre, UNESP - FAAC - Departamento de Design, Bauru, SP, Brasil, mariana.laranjeira@unesp.br; ORCID: 0000-0002-2436-2220.

² Professora Doutora, UNESP – FAAC - Departamento de Artes e Representação Gráfica, Bauru, SP, Brasil. marizil@faac.unesp.br; ORCID:0000-0003-4242-0698.

generative patterns related to the textile area were researched, establishing a connection of two-dimensional and three-dimensional surfaces with complex systems and the use of algorithmic programming software. The research culminated in experiments to create complex patterning algorithms. These results are summarized in the present article.

Keywords: surface design; complexity; textiles; algorithms; pattern.

1. Introdução

A complexidade é um princípio de fundamental importância nas ciências, e tem se apresentado como um conhecimento holístico em pesquisas científicas recentes. Ela é uma característica presente nos sistemas e representa uma condição de organização muito significativa para o entendimento, tanto do meio ambiente e seus elementos biológicos, como das sociedades e as suas interações humanas.

Definir que um objeto ou um sistema apresentam características de complexidade não significa que estes sejam necessariamente complicados, ou de difícil compreensão. A complexidade, na verdade, entra na questão das regras de organização desse sistema, independentemente de haver ordem ou desordem dos elementos. Assim, os sistemas complexos podem ser definidos como sistemas que apresentam como característica principal um grande número de elementos simples que interagem e se organizam de maneira complexa, sem que tenham necessariamente um controlador central, favorecendo a construção de padrões visuais e a manipulação de informações que podem se desenvolver e evoluir (MITCHELL, 2009).

Na ciência dos sistemas complexos, pode-se considerar o design como uma metodologia essencial (ALEXIOU; JOHNSON; ZAMENOPOULOS, 2010), principalmente pela sua capacidade de configuração de projetos e o uso de ferramentas representativas. Por ser um campo de estudo que abrange diversas disciplinas, o design pode utilizar o pensamento sistêmico e a complexidade como interessantes métodos na solução de problemas e na criação de produtos diferenciados, abrindo portas para novas perspectivas de aplicação.

Com o exercício de projetos de design cada vez mais complexos, evoluem também as metodologias de design e as ferramentas criativas. O uso indiscriminado do computador abre caminhos para tecnologias digitais de programação e desenvolvimento de algoritmos, assim como novos processos de fabricação como a manufatura aditiva. Com isso, este artigo relata a pesquisa realizada para o desenvolvimento da dissertação de mestrado intitulada “Design e Complexidade: Investigação do design generativo com o uso de algoritmos computacionais” no Programa de Pós-graduação em Design da UNESP. O projeto teve o intuito de entender o contexto da complexidade aplicada ao design, considerando as especialidades que lidam com o design paramétrico, generativo, e o uso do pensamento sistêmico para geração específica de formas e padrões para superfícies têxteis.

Neste cenário, a pretensão da investigação do design de superfícies esteve em aceitar a viabilidade dessa especialidade do design em poder se beneficiar com esse novo paradigma que é a complexidade, entendendo as implicações dessa temática nos processos criativos e produtivos do designer. Por meio de estudo de casos com o projeto de formas e estruturas com características particulares de sistemas complexos, a pesquisa culminou em superfícies desenvolvidas com algoritmos computacionais. Desta forma, este trabalho tem valor pela sua atualidade e aderência às novas tecnologias e às tendências criativas e projetuais, onde o

designer passa a ter um papel ativo no desenvolvimento de códigos e regras para a construção das formas, inserindo cada vez mais a condição da complexidade na atividade projetual.

2. Design e Complexidade

Cham (2010) situa que as teorias sobre a complexidade apenas começaram a se estabelecer em diversas disciplinas de conhecimento e que não há uma teoria unificada entre elas. Entretanto, mesmo que uma definição de teoria da complexidade esteja em processo de consolidação, é possível determinar algumas características principais apresentadas pelos sistemas complexos. A Complexity Labs (2015) descreve tais propriedades, como pode ser visto no Tabela1.

Tabela 1: Propriedades do sistema complexo

PROPRIEDADE	DESCRIÇÃO
AUTO-ORGANIZAÇÃO E EMERGÊNCIA	A organização emerge por meio de auto-organização dos elementos do sistema. Sem um controle central.
NÃO-LINEARIDADE	Imprevisíveis. A soma dos elementos não é uma simples adição e o resultado pode ser completamente diferente do que se imagina. Representação por meio de equações não lineares.
CONNECTIVIDADE	Altos níveis de interconectividade e interação entre os elementos, estrutura em rede.
ADAPTAÇÃO	Autonomia do elemento em se adaptar ao ambiente e aos demais elementos. Capacidade de evolução. Comportamento dinâmico.

Fonte: Adaptado de Complexity Labs (2015)

O conceito de complexidade, segundo Morin (2008) está diretamente associado com a organização do pensamento e, portanto, mediado por uma abordagem transdisciplinar e holística. Para o autor, nenhuma disciplina oferece uma maneira de integrar completamente todas as informações, e nesse sentido, o paradigma da complexidade deve ser visto como um método de conduzir a organização sistêmica do nosso pensamento.

Nesse contexto, talvez o design, enquanto uma área capaz do projeto interdisciplinar, possa se tornar uma ferramenta significativa na compreensão e acepção da complexidade. "O design e a complexidade compartilham muitos pressupostos epistemológicos e metodológicos, tentando adotar uma perspectiva construtiva e holística do mundo ao nosso redor." (ALEXIOU; JOHNSON; ZAMENOPOULOS, 2010, tradução nossa)

A complexidade pode ser enxergada no setor criativo do design em diversas instâncias, desde o processo até o artefato, influenciando no desenvolvimento de soluções e projetos inovadores. Considerando a importância dos processos artísticos e representativos do design, Cham (2010) afirma que "é nas artes que se encontram conhecimentos e práticas especializadas para a inovação na representação; documentação, visualização, simulação e incorporação são todos métodos artísticos que podem representar sistemas complexos" (CHAM, 2010, p.123, tradução nossa).

Aceitando a importância da complexidade associada ao design, a especialidade do design generativo propõe o encontro da atividade projetual com os princípios dos sistemas

complexos da natureza, combinando os códigos - que podem ser ou não programados no computador - com os processos naturais para a criação de sistemas de linguagem visual com complexidade. Assim, o design generativo é, sem dúvidas, um dos principais cenários de aplicação da complexidade no design.

O computador, atualmente, é uma das ferramentas mais usadas no desenvolvimento de projetos generativos. Por meio de uma prática focada na organização do sistema, onde um conjunto de regras de linguagem definem a composição visual, o uso de algoritmos computacionais pode trazer novas características para a formação de objetos. (KHABAZI, 2012)

As ferramentas de computação e programação de dados possibilitam meios de canalizar o conhecimento do design baseado em parâmetros e variáveis onde elementos geométricos simples podem ser combinados por regras estabelecidas pela programação, gerando sistemas visuais complexos onde a organização e interação dos elementos surgem de maneira fluida e inteligente. A competência em gerar complexidade está associada com as propriedades do sistema em se construir, se manter e se organizar por conta própria a partir das funções estabelecidas (MCCORMACK; DORIN; INNOCENT, 2004). Não obstante, o uso de programação irá aportar ao sistema e seus elementos uma abordagem paramétrica capaz de carregar informações formais, estruturais, funcionais, estéticas e culturais aos objetos (YU; GERO; GU, 2013).

Nesse cenário, um algoritmo computacional se destaca pela sua capacidade de gerar uma infinidade de soluções para um único problema. Em algoritmos para construção de superfícies, essas soluções tendem a ser de ordem geométrica e estética, ou de ordem estrutural e física de acordo com a função da superfície desenvolvida. Esse dinamismo do design generativo permite que a função do computador deixe de ser a de apenas um auxiliar do designer e passe a atribuir à máquina o papel de gerador de conteúdo, colaborando no processo de projeto e permitindo a geração de ideias e soluções.

Desta forma, a função do designer estará em instaurar uma metodologia para a definição do problema, a aplicação dos conceitos propostos, a criação de variáveis e a determinação das etapas a serem executadas pelo algoritmo, independente das soluções a serem oferecidas no final pelo programa. Assim, os parâmetros inseridos irão seguir as regras estabelecidas e ocasionar em diferentes resultados obtidos, dos quais o designer terá pouca ou nenhuma influência sobre eles.

Dentre as ferramentas disponíveis para se tratar a complexidade no contexto do design, foram selecionadas duas principais para menção nesta pesquisa. A primeira ferramenta escolhida é o Grasshopper, plug-in que apresenta funções específicas para o uso de algoritmos dentro do software gráfico Rhinoceros, voltado à modelagem tridimensional. Por meio de uma estrutura de fluxograma, os algoritmos criados neste ambiente facilitam a formação de estruturas complexas. O Grasshopper é uma ferramenta de modelagem algorítmica das mais populares e avançadas, sendo disponibilizada gratuitamente para rodar no Rhinoceros licenciado (TEDESCHI, 2014). Essas ferramentas adjacentes permitem que o designer aborde a complexidade dentro da modelagem 3D com mais facilidade (BUENO, 2016).

O Processing foi a segunda ferramenta selecionada por apresentar uma linguagem de programação open source, ou seja, de acesso gratuito, com o intuito de desenvolver algoritmos voltados para a representação visual estática ou dinâmica. Foi criado em 2001 no Massachusetts Institute of Technology (MIT) por Casey Reas e Ben Fry (REAS; FRY, 2014), e possibilita novas formas de expressão por meio de códigos escritos, incentivando a colaboração artística por ser um software livre (MELLO, 2015). Pode ser definido como “um

ambiente que permite a criação de narrativas interativas, jogos, simulações e produções computacionais que são desenvolvidas em linguagem de programação e que são compartilhadas em uma comunidade online” (BOZOLAN, 2016, p.43).

O uso da programação como ferramenta artística é algo que se baseia na experimentação. Por meio de códigos programados é possível criar, manipular e controlar a complexidade de objetos digitais, assim como prever estratégias de fabricação desde as primeiras etapas de criação.

No entanto, a programação de dados envolve muito mais do que o conhecimento de determinada linguagem e a sua respectiva sintaxe envolvida. Programar significa estabelecer um processo de cognição ampliada por meio de princípios de lógica. É todo um universo à parte que abre espaço para uma infinidade de aplicações matemáticas e de composição visual, que são direcionadas facilmente ao campo do design, seja por meio de projeto ou de pesquisa.

Para Vieira (2014, p.101), “a ferramenta por si só não é criativa. A exploração da linguagem acontece ao integrar a capacidade das ferramentas utilizadas com o interesse, a intenção, a prática e os conhecimentos do designer”.

3. Design de Superfícies

As superfícies são importantes na delimitação de um objeto e desencadeiam estímulos sensoriais através de suas formas, de seu volume, e suas texturas visuais e táteis. Considerando que qualquer volume é dotado de uma superfície que o reveste, é possível aceitar que a atividade projetual, por meio de metodologias do design, para a criação e desenvolvimento de uma superfície, tem uma abrangência significativa na configuração de artefatos.

O design voltado para superfícies pode se responsabilizar pela resolução de problemas estéticos, formais e estruturais dos objetos, com a exploração de abordagens sociais e culturais. As definições do design de superfícies apresentam um crescimento emergente que traz à tona investigações para a construção de superfícies variadas por meio de metodologias do design, além de novas tecnologias de materiais e processos produtivos. Assim, em uma definição abrangente sobre a atividade, podemos concordar com o que estabelece Ruthschilling (2008) de que:

Design de superfície é uma atividade criativa e técnica que se ocupa com a criação e desenvolvimento de qualidades estéticas, funcionais e estruturais, projetadas especificamente para constituição e/ou tratamentos de superfícies, adequadas ao contexto sociocultural e às diferentes necessidades e processos produtivos. (RUTHSCHILLING, 2008)

Como um campo em processo de consolidação e aceitação, Rinaldi (2013, p.1) afirma que “a especialidade ainda é vista como um desdobramento do design têxtil, do design de produto ou como modo de expressão do design gráfico”, complementando que na verdade, o design de superfícies apresenta um hibridismo de todas essas áreas. Desta maneira, pode-se dizer que o design de superfícies é um campo interdisciplinar e transita entre outras áreas de conhecimento conforme a sua aplicação.

É importante destacar que todas as superfícies oferecem uma experiência sensorial, e que a escolha do processo criativo será muito significativa para determinar a sua construção. Elas são veículos de informação da forma, e seu estudo pode trazer contribuições para o diálogo entre materiais e processos, buscando a resolução de problemas, que é o propósito do design.

Nesse sentido, as metodologias de design surgem como um conjunto de orientações a serem seguidas com o intuito de organizar o pensamento e guiar o designer no desenvolvimento do projeto, permitindo a noção de uma sequência de etapas a serem realizadas. No caso do design de superfícies, Rinaldi (2013, p.17) reconhece que não existe uma metodologia correta ou “um método que seja extremamente infalível ou eficaz. Cada projeto dependerá de um processo único, pautado em fases comuns, mas que guardam em si características peculiares dependendo do produto a ser desenvolvido”.

Sob a ótica da atividade projetual, Schwartz (2008) descreve que a superfície está relacionada com aspectos construtivos bidimensionais e que por ser um elemento de circunscrição de corpos, é percebida no espaço tridimensional, com sua conformação e manipulação em um ambiente físico e real. Por outro lado, a autora considera que ao se entender a superfície como um elemento autônomo de objetos, que possui um caráter de interação entre informação, matéria e usuário, ela pode ser percebida também em um ambiente digital e virtual.

Nesse contexto, a busca por se estabelecer uma teoria de classificação da atividade projetual, e conseqüentemente, por um processo de design, deu origem a alguns princípios referentes ao caráter e ao tratamento da superfície. Tanto Rinaldi (2013) quanto Schwartz (2008) definem a superfície por três tipos de abordagens projetuais, considerando-a como um suporte de **representação** gráfica, com uma **constituição/estrutura** material variada e com um propósito de estabelecer **relações** entre o objeto e seus usuários.

Tratando de estabelecer uma metodologia do design de superfícies abrangente, acredita-se que possa ser formalizada uma estrutura de etapas de projeto que considerem como ponto de partida a identificação ou definição de um problema, e a execução sequencial de etapas de criação, produção e aplicação, finalizadas com a obtenção de uma ou mais soluções de design.

Schwartz (2008) considera a superfície por sua condição enquanto **envoltório** de um volume, e por **objeto** constituído. Essa classificação abre caminhos para se investigar, dentro das etapas de criação e representação, quais seriam os conhecimentos a serem levados em consideração, não apenas para a construção bidimensional e tridimensional de superfícies, como da aplicação dos conceitos pertinentes à complexidade, proposta desta pesquisa.

Padrões visuais são importantes representações organizacionais que podem se aplicar em todas as coisas. Considerando que o ponto, a linha e o plano compõem os alicerces das formas no design, Lupton e Philips (2008, p.13), afirmam que “texturas e padrões são construídos a partir de grandes grupos de pontos e linhas que se repetem, revezando-se ou interagindo na formação de superfícies singulares e atraentes”. À vista disso, os autores ressaltam que “entendendo como se produzem padrões, os designers aprendem a tecer complexidade a partir de estruturas elementares, participando de uma das práticas artísticas mais antigas e recorrentes do mundo” (LUPTON; PHILIPS, 2008, p.185).

Para Wiener (apud TOUSSAINT; TOUSSAINT, 2014), o mundo é todo constituído de padrões. Eles estão presentes ao nosso redor, não apenas na natureza que nos rodeia e nos sistemas biológicos, como em nosso próprio ser. (MACNAB, 2012).

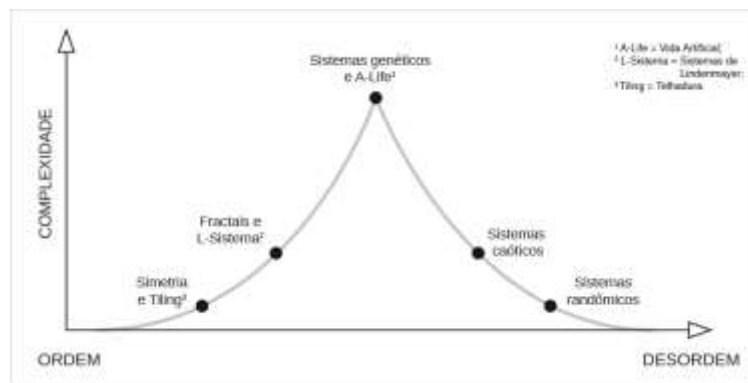
Dentro de um sistema gráfico visual com repetição, os módulos se repetem de modo a criarem uma textura que pode ser visual ou tátil, e por isso os padrões podem fornecer não apenas qualidades estéticas, como também relevos com propriedades físicas, químicas e funcionais. Além disso, eles podem servir como meios de comunicação e sentido, transmitindo mensagens culturais e sociais.

Assim, pode-se afirmar que padrões são o resultado da repetição de um elemento (módulo) de maneira contínua dentro de um sistema, e é possível classificá-los entre dois tipos: regulares e irregulares; sendo os regulares baseados em repetição de unidades com medida, e os irregulares baseados em aleatoriedade e repetição não-mensurável (PROCTOR, 1990).

O método de criação de padrões de repetição regulares, pelo uso de simetria por exemplo, pode servir como fundamentação para o desenvolvimento de padrões de repetição irregulares, ou seja, onde a repetição não pode ser prevista ou controlada. Desta forma, entender as bases conceituais da composição de padrões simples é o primeiro passo para se atingir superfícies visuais com padrões complexos.

Em uma análise mais profunda, pode-se comparar o nível de regularidade de um padrão com o nível de ordem de um sistema e, a partir disso, é válido apreciar a classificação dos sistemas generativos visuais (Figura 1) determinada por Galanter (2010) em sua publicação “O que é a arte generativa?”.

Figura 1: Sistemas visuais generativos



Fonte: Adaptado de Galanter (2010)

Seguindo o gráfico adaptado de Galanter, os sistemas visuais generativos têm seu nível de complexidade influenciado pelo equilíbrio entre o parâmetro de ordem e desordem. Ao se considerar os padrões regulares como sistemas complexos com ordem e, por conseguinte, os padrões irregulares como sistemas em desordem, pode-se afirmar então que o nível de complexidade de um sistema será diretamente influenciado pelo algoritmo e suas regras de estruturação de formas.

Assim, todos os tipos de organizações descritos no gráfico dão origem a sistemas complexos, e a diferença está no nível de estruturação do conteúdo visual e na gramática da forma utilizada. Além disso, o autor deixa claro que o grau de complexidade de um sistema pode variar dentro de uma mesma categoria (GALANTER, 2010).

É relevante esclarecer também, que não existe uma condição específica para se medir o nível de complexidade de um sistema, mas que algumas características podem influenciar. Mitchell (2009) afirma que essas características podem ter a ver com tamanho, entropia, informação algorítmica, profundidade da lógica, e dimensão.

Não obstante, independente do padrão ser regular ou irregular, Reas e McWilliams (2010), afirmam que a repetição é uma característica constante em ambos e tem a função de fornecer ritmo às formas, de modo que pelo uso de regras, todos os padrões visuais são, em sua essência, composições algorítmicas que podem ser trabalhadas por meio de linguagens de programação.

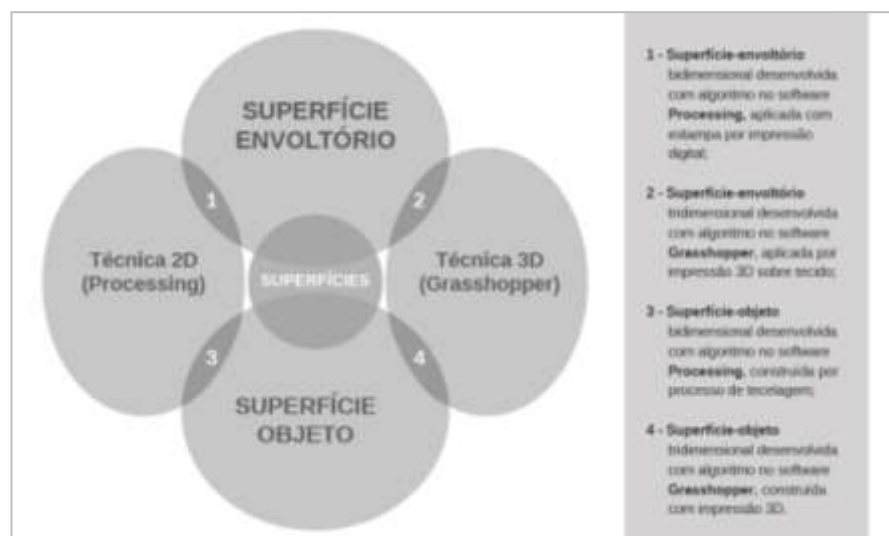
4. Materiais e métodos

Os procedimentos metodológicos da pesquisa incluíram a etapa teórica, onde ocorreu uma revisão bibliográfica para fundamentação, e a etapa prática composta por uma investigação de campo por meio de estudo de casos e da prática experimental com a reprodução de algoritmos construtivos.

Para os estudos de casos determinaram-se os seguintes procedimentos: fundamentação dos temas tratados (1); definição das unidades-caso (2); determinação do número de casos (3); elaboração do protocolo de estudo (4); coleta de dados (5); análise e interpretação dos dados (6).

O critério de escolha dos casos baseou-se nas duas vertentes construtivas da superfície, mencionadas por Schwartz (2008). Assim, pode-se afirmar que a superfície têxtil nesta pesquisa foi tratada basicamente pela sua condição enquanto suporte para padronagens, o que no caso seria superfície-envoltório; e pela sua capacidade enquanto estrutura tramada têxtil, ou superfície-objeto. Mais do que isso, é importante lembrar que o conceito unificador para esse paralelismo observado foi o design generativo, ou seja, a complexidade em sistemas visuais e o uso de algoritmos computacionais. Com isso, a Figura 2 formaliza os critérios de escolha dos projetos a serem abordados:

Figura 2: Critérios de seleção dos estudos de caso



Fonte: Elaborado pelas autoras

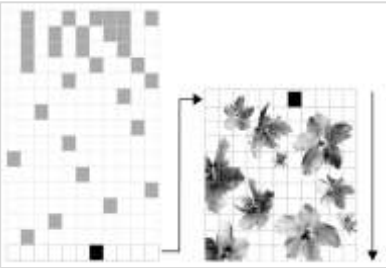

Após o estudo de casos, foram desenvolvidas experimentações com os softwares mencionados. Com isso, exploraram-se alguns dos recursos algorítmicos disponibilizados pelo Processing para superfícies 2D com o uso de programação de dados.

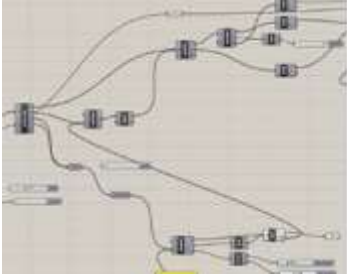
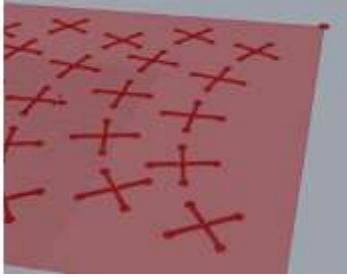


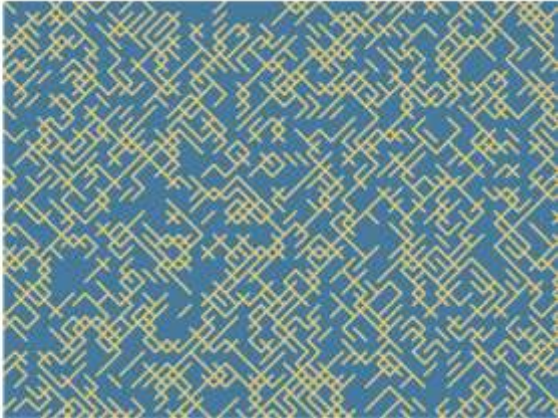
5. Estudos de Caso

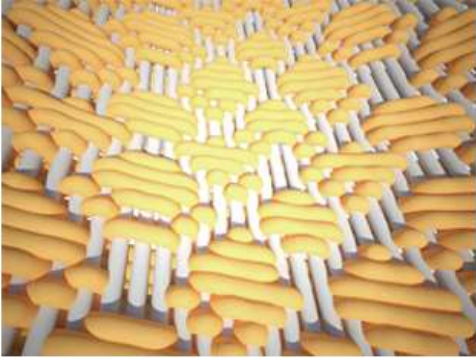
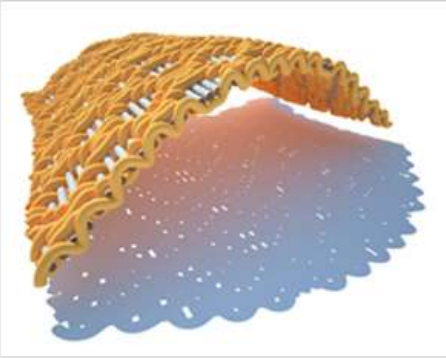
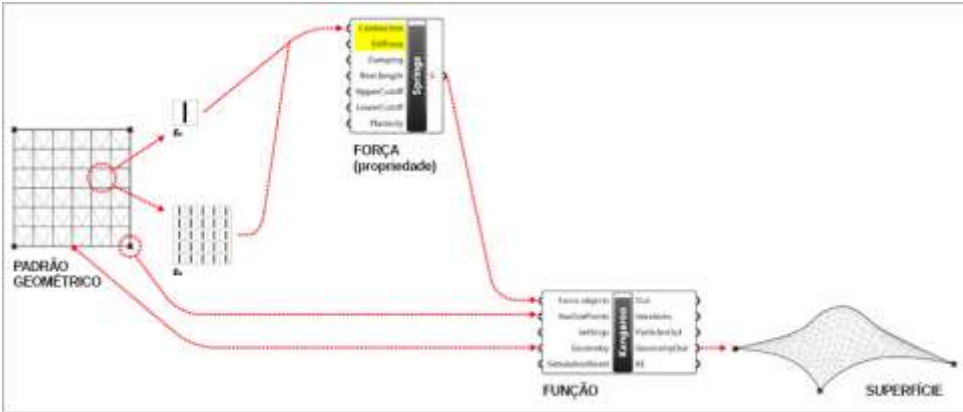
Em busca de um estado da arte do design de superfícies complexas, os casos selecionados para o estudo foram projetos aplicados ou acadêmicos que abordassem a complexidade e a atividade da programação de dados aplicadas ao design. Foram 6 (seis) casos escolhidos, classificados pelas suas ferramentas de criação 2D ou 3D utilizadas em aplicações de superfícies têxteis. Eles foram analisados, principalmente, com o propósito de investigar as principais associações com os temas sugeridos nesta pesquisa e as consequentes inovações existentes nos projetos.



Os estudos de casos estão apresentados na Tabela 2 a seguir, com o tipo de superfície trabalhada, os autores responsáveis e uma breve descrição de cada atividade.

Tabela 2: Estudos de casos escolhidos e analisados

Tipo de superfícies	Autor (es)	Descrição
Superfície envoltório 2D	Alex Russell (2014)	Impressão de tecidos com tecnologia de estampa digital em tempo real. A padronagem foi desenvolvida por meio de programação de dados utilizando autómatos celulares, de modo que os elementos gráficos fazem parte de um sistema de organização complexo com características de imprevisibilidade. Deste modo, os motivos da estampa se auto-organizam aleatoriamente, alterando o padrão conforme a mesma está sendo impressa.
Autómatos celulares		Padronagem aleatória
		
Fonte: Adaptado de Russell (2014)		
Superfícies Envoltório 3D	KISD (2016)	Projeto de impressão 3D em tecidos. O padrão é desenvolvido por algoritmos de modelagem 3D no Grasshopper. A combinação dos filamentos em estruturas com forma de pirâmide ocasiona em um sistema complexo que fornece ao tecido propriedades de isolamento acústico e a capacidade de adaptabilidade em diferentes ambientes.

Tipo de superfícies	Autor (es)	Descrição
<p>Algoritmo no Grasshopper</p> 	<p>Modelagem 3D</p> 	<p>Tecido com impressão 3D</p>  <p>Fonte: Adaptado de KISD (2016)</p>
<p>Superfícies objeto 2D</p>	<p>WOVNS (2019)</p>	<p>Tecidos tramados por tecelagem utilizando padrões gráficos criados por programação de dados com o Processing. A empresa WOVNS disponibiliza os algoritmos das padronagens e permite que os clientes alterem os parâmetros do código e personalizem suas estampas. A aleatoriedade e imprevisibilidade dos resultados são as principais características dos sistemas visuais complexos obtidos.</p>
<p>Biblioteca WOVNS e algoritmo no Processing</p> 	<p>Padronagem aleatória</p> 	<p>Fonte: Adaptado de WOVNS(2019)</p>
<p>Superfícies objeto 3D</p>	<p>KHABAZI (2010) tecelagem</p>	<p>Desenvolvimento de algoritmos generativos para modelagem e impressão 3D utilizando o <i>software</i> Rhinoceros com o Grasshopper para a construção de superfícies. Fazendo uma análise do processo construtivo da técnica de tecelagem, o autor propõe um olhar mais atento para processos de fabricação tradicionais como fundamentação para processos produtivos contemporâneos. Os padrões complexos gerados pelos algoritmos são utilizados para aplicações arquitetônicas.</p>

Tipo de superfícies	Autor (es)	Descrição
		<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Modelagem 3D</p>  </div> <div style="text-align: center;"> <p>Simulação de aplicação 3D</p>  </div> </div> <p style="text-align: center;">Fonte: Adaptado de Khabazi (2010)</p>
	<p>McKnelly (2010)</p> <p>tricô</p>	<p>Pesquisa que investigou o uso de padrões da trama do tricô aplicados em algoritmos programados e modelados no Grasshopper. Utilizando conhecimentos referentes ao processo de tricô parcialmente mecanizado, o estudo buscou identificar as propriedades de rigidez e força apresentadas nos modelos físicos desenvolvidos em tricô e aplicá-las no modelo digital. Entendendo os tipos de pontos desta técnica, os padrões formados pelo algoritmo apresentam características de complexidade e determinam as condições de flexibilidade, densidade e direção da trama estruturada.</p>
		<p style="text-align: center;">Processo de modelagem algorítmica 3D</p>  <p style="text-align: center;">Fonte: Adaptado de McKnelly (2010)</p>
<p>Worden (2011)</p> <p>crochê</p>	<p>Usando como premissa a técnica do crochê para geração da forma e o uso de algoritmos criados no Grasshopper e no Rhinoceros, o projeto explora uma série de superfícies construídas manualmente com a técnica do crochê utilizando materiais diferenciados. O entendimento do processo analógico permitiu que o conhecimento pudesse ser digitalizado por meio de algoritmos. Os algoritmos desenvolvidos com modelagem tridimensional e o Grasshopper são exemplos da aplicação de técnicas generativas para padrões tramados.</p>	

Tipo de superfícies	Autor (es)	Descrição
	 <p>a - modelagem no crochê</p>	 <p>b - algoritmo no Grasshopper</p> <p>c - Superfície modelada em 3D</p> <p>Fonte: Adaptado de Worden (2010)</p>

Fonte: Elaborado pelas autoras

Os casos estudados dão oportunidade para novos métodos de criação que utilizam a programação de dados e, como fonte de informação, o conhecimento milenar das técnicas de trama, como a tecelagem, o tricô e o crochê. A importância dos têxteis enquanto superfícies construídas se apresentam em suas tramas dotadas de propriedades matemáticas, e consequentemente topológicas, sendo responsáveis por estruturas que exploram e delimitam os espaços que nos rodeiam.

Por meio de um processo híbrido, que combina códigos programados com o processo tradicional de produção de tecido, é possível novos métodos para a estruturação de superfícies complexas, sejam elas em âmbito virtual ou físico. Esses novos métodos de criação dão oportunidade de que as estampas deixem de ser estáticas e passem a ser também imagens dinâmicas, se alterando e evoluindo progressivamente. Um dos benefícios disso é a possibilidade que o código fornece de que um módulo não seja igual ao anterior, evoluindo e emergindo conforme as variáveis e funções determinadas no sistema.

Com o uso de algoritmos, se torna possível pensar na tridimensionalidade da superfície por meio de regras de organização e combinação de linhas e na formação da malha. Isso auxilia na visualização de estruturas com texturas diferenciadas e atribui plasticidade nas formas, de modo que as características da superfície variam não apenas no quesito dos materiais utilizados, como também nos métodos construtivos e processos de fabricação.

6. Experimentação

Levando em consideração os projetos analisados nos estudos de caso, foram desenvolvidas na pesquisa algumas experimentações, com o propósito de alterar, reproduzir e criar algoritmos por meio de linguagem de programação. Este artigo relata a experimentação que teve como objetivo criar um algoritmo de padronagem com base nos princípios dos sistemas complexos vistos anteriormente.

Dentre os softwares mencionados na pesquisa, o aplicativo escolhido para a execução do código programado foi o Processing. A escolha ocorreu devido à facilidade de se encontrar

tutoriais e cursos online, de modo gratuito, que disponibilizassem as informações necessárias para sua utilização, o que incluiu a orientação para se desenvolver algoritmos generativos e com condições condizentes aos sistemas de complexidade. Além disso, sua classificação de software livre tornou possível e acessível o download e a instalação do programa no computador.

É interessante destacar que o Processing permite a realização de sistemas visuais que exploram tanto o âmbito bidimensional quanto o tridimensional. Entretanto, para interesse deste trabalho, o foco foi no desenvolvimento de superfícies bidimensionais com padrões de repetição, a fim de estabelecer os parâmetros primordiais para este tipo de projeto digital.

O Processing é uma linguagem baseada em Java, o que significa que o algoritmo é originado de um código escrito em inglês, com regras e expressões que irão determinar a execução de uma tela do programa. A tela de visualização é o que, no caso de um projeto de design de superfícies, pode se referir à uma peça gráfica com elementos de linguagem visual que constituem uma padronagem - que como visto, é composta de padrões de repetição e regras de organização dos módulos por meio de simetrias.

De uma maneira sucinta, pode-se considerar que um código do Processing deve apresentar dois componentes básicos: Variáveis, responsáveis pela inclusão de dados por meio de parâmetros; e Funções, responsáveis pela execução de comandos. Obviamente, existem muitos outros elementos de programação de um algoritmo que devem ser levados em consideração, mas Variáveis e Funções são as ferramentas mais essenciais para se adentrar ao desenvolvimento do projeto de um algoritmo.

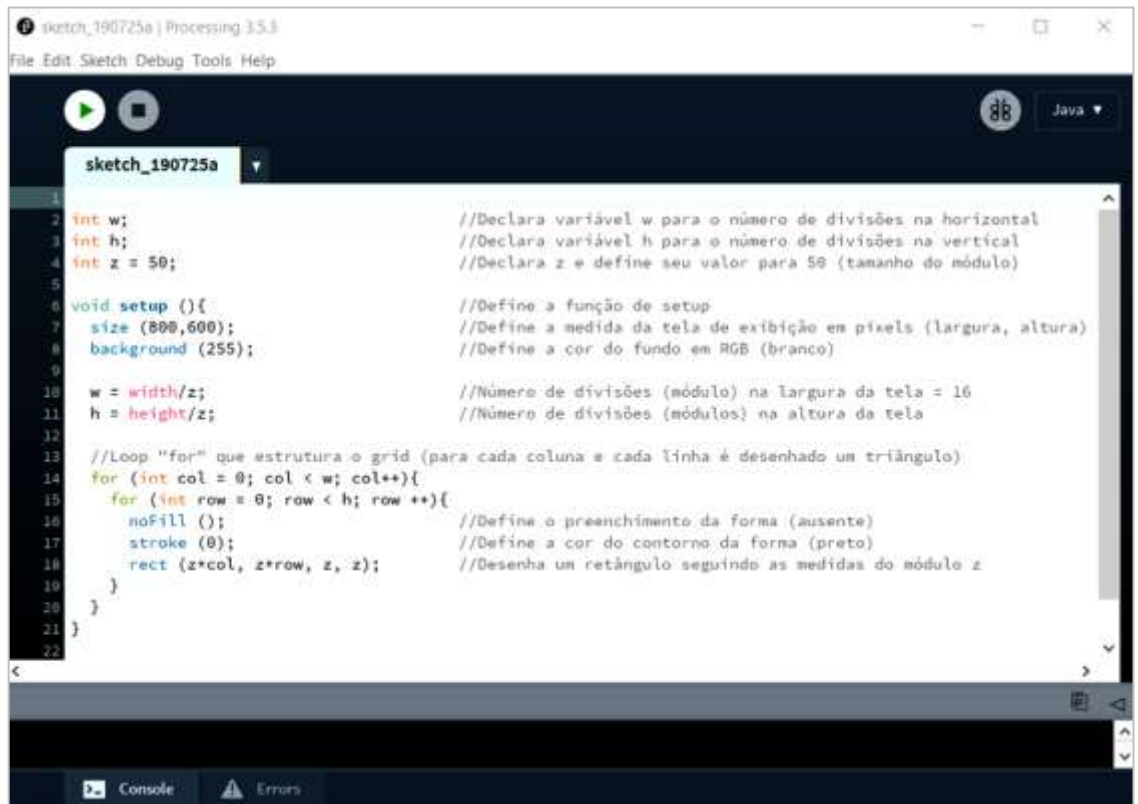
Com ferramentas voltadas para designers e artistas, Reas e Fry (2014, p.11) afirmam que “o Processing foi projetado para facilitar a criação de elementos gráficos, como linhas, elipses e curvas na tela de exibição”. Para a criação de tais formas é preciso utilizar Funções previamente determinadas pela sintaxe do software, assim como a inserção de parâmetros que especificam as coordenadas e medidas da forma.

A imprevisibilidade, não-linearidade e capacidade de adaptação das formas por meio da alteração do código de programação consolidam a aplicação de princípios dos sistemas complexos no design de superfícies.

O primeiro passo para o desenvolvimento de uma padronagem foi esmiuçar os elementos básicos de sua composição. Para isso, deve-se considerar que a construção de uma grade quadriculada é a maneira mais simples de se estruturar um padrão, regular ou irregular, onde o módulo irá se repetir. A partir disso, a primeira etapa da experimentação no Processing foi reproduzir um algoritmo responsável pela execução de um grid (Figura 3).

Como pode ser observado na Figura 1, os comentários que antecedem o uso de “//” esclarecem linha a linha da sintaxe inserida. A estrutura do código consiste em definir e atribuir valor às variáveis (w, h, z) que representam a quantidade e o tamanho dos módulos que serão inseridos e repetidos no grid. Em seguida, é definida a condição lógica (*for*) para garantir um loop onde para cada coluna (*col*) e linha (*row*) seja inserido um quadrado (*rect*) com os parâmetros previamente definidos nas variáveis.

Figura 3: Algoritmo do Processing



```
1 sketch_190725a | Processing 3.5.3
2 File Edit Sketch Debug Tools Help
3
4 sketch_190725a
5
6 1
7 2 int w; //Declara variável w para o número de divisões na horizontal
8 3 int h; //Declara variável h para o número de divisões na vertical
9 4 int z = 50; //Declara z e define seu valor para 50 (tamanho do módulo)
10 5
11 6 void setup () { //Define a função de setup
12 7 size (800,600); //Define a medida da tela de exibição em pixels (largura, altura)
13 8 background (255); //Define a cor do fundo em RGB (branco)
14 9
15 10 w = width/z; //Número de divisões (módulo) na largura da tela = 16
16 11 h = height/z; //Número de divisões (módulos) na altura da tela
17 12
18 13 //Loop "for" que estrutura o grid (para cada coluna e cada linha é desenhado um triângulo)
19 14 for (int col = 0; col < w; col++){
20 15 for (int row = 0; row < h; row++){
21 16 noFill (); //Define o preenchimento da forma (ausente)
22 17 stroke (0); //Define a cor do contorno da forma (preto)
23 18 rect (z*col, z*row, z, z); //Desenha um retângulo seguindo as medidas do módulo z
24 19 }
25 20 }
26 21 }
27 22
```

Fonte: Elaborado pelas autoras

Desta maneira, o grid será formado com fundo branco (*background(255)*) e linhas em preto (*stroke(0)*), podendo variar o tamanho de cada módulo de acordo com o valor estipulado pelo designer na variável z.

O algoritmo foi reproduzido completamente, tendo como base uma estrutura lógica existente, mas sem necessariamente utilizar as mesmas variáveis ou definições de construção de formas.

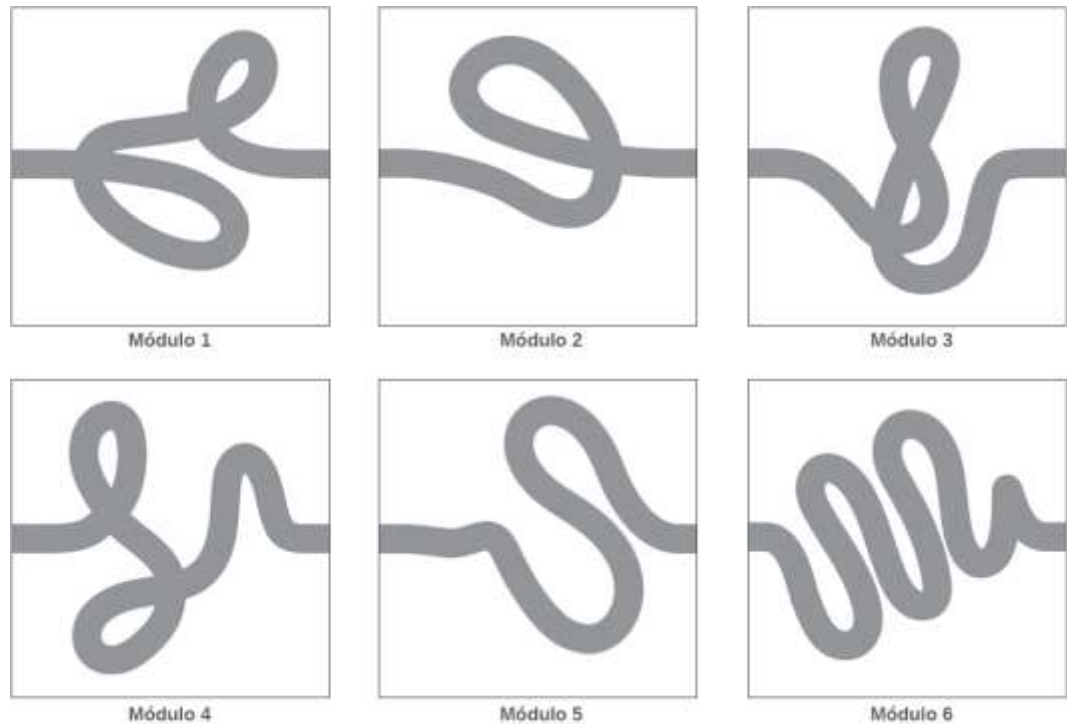
Devido à complexidade que um algoritmo pode chegar a ter, a prática da reprodução é algo recorrente, que permite ao designer-programador a possibilidade de facilitar o seu trabalho e produzir projetos mais elaborados e diferenciados em menos tempo. Por esse motivo, além de existir uma grande quantidade de códigos abertos e disponibilizados online gratuitamente, existe também uma ampla discussão sobre direitos autorais e sobre até que ponto um designer-programador é detentor de determinado código. Apesar desse assunto não ser explorado neste trabalho, vale a pena deixar claro que esses questionamentos existem dentro da área do design com linguagem de programação.

Tentando ir além da simples substituição de parâmetros e formas geométricas, surgiu a proposta da criação de um algoritmo de padronagem que, ao invés de funções de formas geométricas, fosse além e utilizasse como parâmetro o uso de imagens, previamente desenvolvidas, como elemento modular.

Assim, considerando questões relativas à constituição de rapport, onde para a realização do padrão o módulo deve apresentar característica de encaixe em suas laterais, foram desenvolvidos 6 tipos de módulos diferentes, todos permitindo um padrão contínuo

independente da combinação. Fazendo referência às linhas contorcidas apresentadas nos têxteis, os módulos foram criados em vetor no software Adobe Illustrator (Figura 4).

Figura 4: 7 tipos de módulos de repetição



Fonte: Elaborado pelas autoras

Inicialmente as imagens foram desenvolvidas em tons de cinza com fundo branco, e todos os módulos se encaixam entre si no sentido horizontal (Figura 5).

Figura 5: Módulos se encaixam na lateral horizontal



Fonte: Elaborado pelas autoras

Seguindo então o código do grid criado e realizando as adaptações necessárias para a inserção de imagens no algoritmo, foi desenvolvida uma função responsável por inserir os 6 módulos diferentes de maneira aleatória. A regra de repetição utilizada foi o *Random*, função que dentro de uma margem numérica escolhida randomiza qual valor ou objeto será apresentado na tela. A aleatoriedade aqui se apresenta como propriedade muito significativa para a construção de padrões variados e complexos.

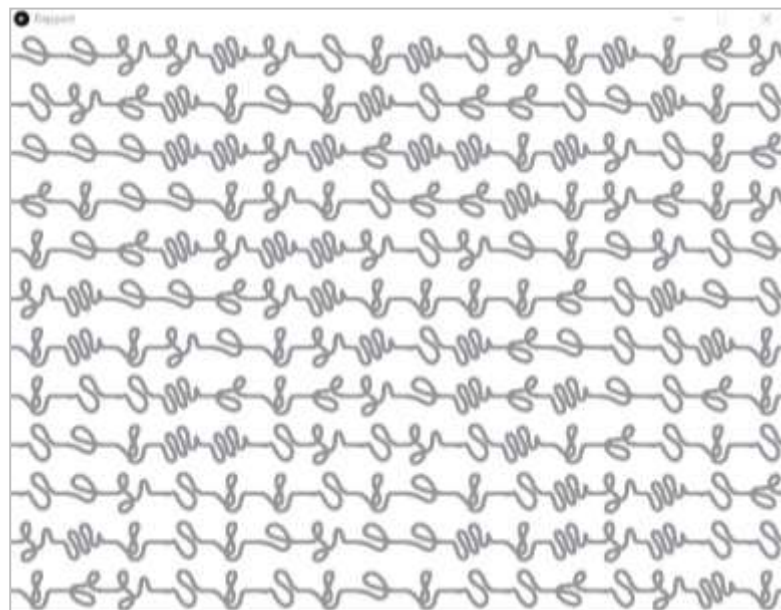
Figura 6: Algoritmos de padronagem aleatória no Processing

```
1 //Codigo //DESCRIÇÃO DO CÓDIGO
2
3 int maxImages = 6; //Declara variável 'maxImages' para o número máximo de imagens inseridas
4 int w; //Declara variável w para o número de divisões na horizontal
5 int h; //Declara variável h para o número de divisões na vertical
6 int z = 58; //Declara z e define seu valor para 58 (tamanho do módulo)
7
8 Image [] modulo = new PImage [maxImages]; //Declara a variável array 'modulo' para inserção de imagens
9
10 void setup () { //Define a função de setup
11   size (800,600); //Define a medida da tela de exibição em pixels (largura, altura)
12   background (255); //Define a cor de fundo em RGB (branco)
13 }
14
15 for (int i = 0; i < modulo.length; i++){ //Loop 'for' que adiciona todas as imagens na variável 'modulo'
16   modulo[i] = loadImage ("modulo"+i+".png");
17 }
18
19 w = width/z; //Número de divisões (módulos) na largura da tela
20 h = height/z; //Número de divisões (módulos) na altura da tela
21
22
23
24 for (int col = 0; col < w; col++){ //Loop 'for' que estrutura o grid (para cada coluna e cada linha é inserido um módulo)
25   for (int row = 0; row < h; row++){
26     int index = int (random (5)); //Define que o módulo escolhido seja randômico
27     noFill (); //Define o preenchimento da forma (ausente)
28     stroke (0); //Define a cor do contorno da forma (preto)
29     image (modulo[index], z*col, z*row, z, z); //Exibe uma a uma as imagens na tela na posição do grid
30   }
31 }
32 }
```

Fonte: Elaborado pelas autoras

Cada vez que o algoritmo reinicia, um novo padrão é gerado (Figura 7), nunca se repetindo uma padronagem anterior. A sequência dos módulos é completamente aleatória e imprevisível, constituindo em uma superfície bidimensional com propriedades de complexidade.

Figura 7: Padronagem gerada pelo Processing



Fonte: Elaborado pelas autoras

Ao realizar essa experimentação, observou-se que entre um padrão e outro dificilmente é possível verificar que houve muita alteração na composição dos módulos. Isso ocorre por alguns motivos, como por exemplo o fato de os módulos serem muito parecidos, estando todos na mesma cor e com desenhos similares. Para resolver essa questão, podem ser realizadas alterações na cor, na quantidade de módulos, desenhos de linhas mais diferenciados entre si, entre outras soluções.

É significativo mencionar que a randomização, principal característica de complexidade inserida no algoritmo desenvolvido, se apresentou principalmente na inserção e no posicionamento dos módulos. Entretanto, ela poderia ser uma propriedade facilmente aplicada em outros parâmetros do algoritmo como na cor e no tamanho, por exemplo. Trazendo assim, uma ampla gama de possibilidades visuais para os padrões gerados.

Fazendo uma análise final sobre o processo de experimentação, é significativo considerar as duas fases que envolvem um projeto de design de superfícies com programação. Em primeiro lugar, deve-se considerar o processo criativo, no qual o designer precisa entender sobre os métodos de construção de padrões e estruturas de repetição, relativos aos conhecimentos de superfícies, sejam estas do tipo envoltório ou objeto. Nesse sentido, foi valorizado o desenvolvimento dos módulos e os respectivos encaixes, assim como a preocupação com a continuidade da repetição, e com questões estéticas e de composição visual.

Em um segundo momento, deve-se dar atenção ao processo de construção do algoritmo. A simplicidade do objetivo esperado não necessariamente irá condizer com a simplicidade do código. Para isso, é preciso considerar a sintaxe e lógica de programação envolvidas.

Essas duas questões são essenciais para que qualquer programa desenvolvido com linguagem de programação possa ser executado. A sintaxe faz referência ao uso correto de expressões e funções determinadas pelo software utilizado, o que neste caso foi o Processing. Se alguma palavra ou mesmo algum caractere estiver mal colocado, o algoritmo apresentará uma mensagem de erro e não funcionará, ou executará com deficiência. Deste modo, para que não ocorram problemas de sintaxe, é importante que o designer adquira os conhecimentos necessários para escrever o código, o que em outras palavras significa aprender a falar o idioma do software utilizado. Por outro lado, ainda que o código esteja com uma sintaxe correta, se ele não apresentar uma lógica de execução, não haverá resultado nenhum.

Entender a importância dessas questões e saber alinhar com as metodologias de design específicas é o mais importante passo para o desenvolvimento do design de superfícies digitais e complexas.

7. Considerações Finais

Acompanhando a mudança de paradigma que vem ocorrendo em muitos campos da ciência, este trabalho teve como propósito a consideração e o entendimento do conceito de complexidade e sua influência dentro das pesquisas na área do design.

Os estudos sobre a complexidade, e o que se acredita ser coerente com uma possível teoria da complexidade, são relativamente extensos e se enredam por todas as áreas de conhecimento. Com isso, é importante ressaltar que o objetivo não foi buscar todas as visões relativas ao assunto, mesmo porque isso seria inviável, mas que foi sim do interesse deste trabalho encontrar as principais referências e conceitos que se encaixassem especificamente com as pesquisas do campo do design de superfícies.

Observou-se que as teorias da complexidade podem se apresentar como um modo de se definir métodos de criação e produção, se tornando uma ferramenta para a organização de sistemas visuais e formais. Por essa vertente, ela foi tratada nessa pesquisa como um conceito fundamental para a estruturação de sistemas de formas geométricas e a composição de padrões visuais com aplicações em superfícies reais ou virtuais mais complexas.

De maneira mais específica, o objetivo principal foi entender os princípios básicos para se criar um algoritmo com programação de dados que tivesse enfoque no desenvolvimento de padrões de repetição e, assim, visualizar o modo mais simples de se programar o código e estabelecer precedente para algoritmos cada vez mais complexos.

Neste contexto, as técnicas de construção de trama, como os processos de tecelagem, tricô e crochê, são exemplos típicos de como a aplicação da matemática associada com o trabalho manual permite o manuseio e a objetificação de problemas e cálculos. Estas técnicas artesanais identificadas comportam um número vasto de aplicações interessantes, mostrando a conveniência de se revelar a estrutura geométrica subjacente e de se conduzir questões através de uma perspectiva lógica. Ao estabelecer uma conexão entre as superfícies têxteis com a programação de algoritmos e o desenvolvimento de formas por meio de softwares digitais, foi possível estabelecer a importância do uso de novas tecnologias no processo de superfícies com mais complexidade.

A principal justificativa desse tipo de pesquisa, na qual as tramas têxteis são inseridas no ambiente digital e transformadas em padrões complexos por meio de ferramentas de programação, está em entender novas maneiras de se construir superfícies, explorando texturas e formas dentro do atual contexto tecnológico. Novas metodologias híbridas de projeto trazem novas oportunidades, não apenas para que designers-artesãos explorem mais o conhecimento tradicional da realização do trabalho manual, mas também para que designers-programadores possam aplicar as muitas funcionalidades e propriedades em projetos digitais e inovadores. O profissional do futuro pode se revigorar com as possibilidades expressivas que um conhecimento híbrido, entre digital/virtual e físico/analógico tem a oferecer.

Em acordo com esses conhecimentos investigados, o profissional de design que se interesse em explorar as propriedades organizacionais e evolutivas dos sistemas complexos, deve buscar complementar sua formação para realizar uma alfabetização digital com relação às novas linguagens de programação e aplicativos voltados ao design, como é o caso do Processing e do Grasshopper.

É importante considerar que o aspecto colaborativo do Processing, por exemplo, abre uma porta de possibilidades para os designers e artistas-programadores que estão iniciando nesse meio de linguagens de programação, e conseqüentemente, auxilia no condicionamento destes profissionais permitindo, muitas vezes, que o foco da criação seja mais no lado criativo e estético do projeto do que no lado lógico e matemático envolvido.

O foco no uso das novas tecnologias deve estar em encontrar os benefícios que estas apresentam quando comparadas com técnicas tradicionais. Alguns destes benefícios podem ser referentes à melhora do processo de projeto e, conseqüentemente, no trabalho do designer. As possibilidades criativas e diferenciadas que as linguagens de programação proporcionam para o design podem influenciar significativamente nas etapas de criação e execução da composição visual. A otimização e variedade dos resultados, obtidos por meio de algoritmos computacionais, também devem ser levadas em conta. Além disso, o projetista

pode se beneficiar de uma possível diminuição da carga de trabalho e a possibilidade de uma grande quantidade de imagens esteticamente diferenciadas geradas por um único algoritmo.

Levando em consideração os estudos de caso realizados, a pesquisa proporcionou o entendimento de uma evolução funcional e estética em códigos abertos que podem ser alterados e transformados de modo que uma mesma função de criação de padrão pode variar de acordo com os propósitos e as aplicações da superfície. Foi possível identificar que os projetos de superfícies variam entre bidimensionais e tridimensionais, e que apesar dos exemplos escolhidos mostrarem aplicações variadas, a essência do desenvolvimento de todos faz referência direta com o hibridismo entre design (de superfícies), conhecimento analógico têxtil, comportamento de complexidade e uso de tecnologia digital. Confirmando os parâmetros estabelecidos no início deste trabalho.

Sobre as experimentações, foram absorvidos os fundamentos básicos, observados ao longo da pesquisa, para a realização de superfícies complexas simuladas em ambiente digital, como as propriedades que um sistema de complexidade deve apresentar, as metodologias de design envolvidas no processo do projeto de superfícies com padronagens, e o uso das ferramentas de programação e codificação de algoritmos por meio do software digital.

Pesquisas futuras podem aproximar ainda mais as relações entre a superfície física com as tecnologias computacionais, por meio do desenvolvimento de formas e estruturas que sejam influenciadas pela programação de dados e suas características. Nesse sentido, em uma proposta além, novas investigações podem trazer a possibilidade de se associar ao design de superfícies complexas com outras tecnologias como o uso de sensores e plataformas eletrônicas, como o Arduino. Também podem-se considerar evoluções que ocorram nas superfícies digitais e interativas, onde o uso de algoritmos generativos e evolutivos pode se apresentar de modo muito inovador quando se levantam, por exemplo, questões como a interação com o usuário ou a entrada de dados aleatórios, sustentados pelos princípios de emergência da forma e organização adaptativa dos elementos do sistema visual.

Em suma, pode-se afirmar que a pesquisa atendeu as demandas da hipótese proposta no início do projeto, e mostrou que a assimilação dos conceitos referentes à complexidade e o consequente uso de tecnologia digitais de programação de algoritmos, podem contribuir significativamente para o desenvolvimento do design de superfícies cada vez mais eficientes e inovadoras.

Referências

ALEXIOU, K.; JOHNSON, J.; ZAMENOPOULOS, T. Embracing complexity in design: Emerging perspectives and opportunities. In: INNS, T. (ed.). **Designing for the 21st century: Interdisciplinary methods and findings**. Vol. II, New York: Gower Publishing, 2010.

BORATO, Mariana Araújo Laranjeira. **Design de superfície e complexidade: investigação do design de padrões generativos e o uso de algoritmos computacionais**. 2019. 103 f. Dissertação (Mestrado) - Curso de Design, Unesp, Bauru, 2019. Disponível em: <https://repositorio.unesp.br/handle/11449/190906>. Acesso em: 25 ago. 2020.

BOZOLAN, S. M. **O pensamento computacional: ensino e aprendizagem através do software Processing**. 2016. 145f. Dissertação (Mestrado em Tecnologias da Inteligência e Design Digital) – Pontifícia Universidade Católica de São Paulo - PUC, São Paulo, 2016.

- BONIME, W. **How WOVNS is revolutionizing the interior design textile market.** 2017. Disponível em: <https://www.forbes.com/sites/westernbonime/2017/09/09/how-wovns-is-revolutionizing-the-interior-design-textile-market/?fbclid=IwAR3j1xf-p1Hy4qotcLYJ-xlPCrszjVZPNcOPpGPF4kGyLt0Wo0OY6QpLG_0#62776c38247f> Acesso em março de 2019.
- BUENO, E. **Grasshopper.** In: GASPAR, J. (ed.) 101 Conceitos de arquitetura e urbanismo na era digital. São Paulo: Probooks, 2016.
- CHAM, K. The art of complex systems science. In: ALEXIOU, K.; JOHNSON, J.; ZAMENOPOULOS, T. (Ed.). **Embracing complexity in design.** New York: Routledge - Taylor & Francis Group. cap. 7, p. 121-142, 2010.
- COMPLEXITY LABS. **Complexity Theory Course.** 2015. Disponível em: <https://www.youtube.com/watch?v=vp8v2Udd_PM&index=2&list=PLsJWgOB5mIMDRt8-DBLLVfh-XeKs2YAcg> Acesso em: Agosto de 2018.
- GALANTER, P. **The problem with evolutionary art is...**In: Evo Applications 2010, Part II, LNCS 6025. Berlin: Springer-Verlag Heidelberg, p. 321-330, 2010.
- _____. **What is generative art?** Complexity theory as a context for art theory. In: GA2003 - 6th Generative Art conference, Milan, Italy. 2003.
- KHABAZI, Z. M. **Generative algorithms concepts and experiments: Weaving.** E-book. Disponível em: <www.morphogenesisism.com>, USA, 2010.
- _____. **Generative algorithms: using Grasshopper.** E-book. Disponível em: <www.morphogenesisism.com>, USA, 2012.
- KISD, Köln International School of Design. **3D-Printing-Design Award: KISD Project "sonogrid" Among Top 5.** 2016. Disponível em: <<https://kisd.de/en/projects/3d-printing-design-award-kisd-project-sonogrid-among-top-5/>> Acesso em: março de 2019.
- LUPTON, E.; PHILIPS, J. C. **Novos fundamentos do design.** São Paulo: Cosac Naify, 2008.
- MACNAB, M. **Design by Nature – Using universal forms and principles in design.** New Riders: Canada, 2012.
- MCCORMACK, J.; DORIN, A.; INNOCENT, T. **Generative design: a paradigm for design research.** In: Redmond, J. et. al. (eds) Proceedings of Futureground, Design Research Society, Melbourne, 2004.
- MCKNELLY, C. L. **Knitting Behavior: A material-centric design process.** 2015. 57f. Dissertação (Master of Science in Architecture Studies) - MIT Massachusetts Institute of Technology, 2015.
- MELLO, P. O. B. **Arte e programação na linguagem Processing.** 2015. Dissertação (Mestrado em Tecnologias da Inteligência e Design Digital) – Pontifícia Universidade Católica de São Paulo - PUC, São Paulo.
- MITCHELL, M. **Complexity: A guided tour.** New York: Oxford University Press, 2009.
- MORIN, E. **On complexity: Advances in systems theory, complexity, and the human sciences.** New Jersey: Hampton Press Inc., 2008.
- PROCTOR, R. M. **Principles of Pattern Design.** Dover Publications: New York, 1990.
- REAS, C.; FRY, B. **Processing: a programming handbook for visual designers and artists.** The MIT Press, 2nd edition. Cambridge, 2014.

REAS, C.; MCWILLIAMS, C. **Form + code in design, art, and architecture**. Nova York: Princeton Architectural Press, 2010.

RINALDI, R. M. **A intervenção do design nas superfícies projetadas: processos multifacetados e estudos de caso**. 2013. 190 f. Tese (doutorado) - Universidade Estadual Paulista Júlio de Mesquita Filho, Faculdade de Arquitetura, Artes e Comunicação de Bauru, 2013.

RUSSELL, A. **Repeatless: transforming surface pattern with generative design**. In: Shapeshifting Conference: Auckland University of Technology, Nova Zelândia, 2014.

RUTHSCHILLING, E. A. **Design de superfície**. Porto Alegre: Editora da UFRGS, 2008.

SCHWARTZ, A. R. D. **Design de superfície: por uma visão projetual geométrica e tridimensional**. 2008. 200 f. Dissertação (Mestrado em design) - Universidade Estadual Paulista 'Júlio de Mesquita Filho', UNESP, 2008.

TEDESCHI, A. **Algorithms-Aided Design - AAD: Parametric strategies using Grasshopper**. Brienza: Le Penseur, 2014.

TOUSSAINT, E. R; TOUSSAINT, G.T. **What is pattern?** In: Proceedings of Bridges 2014: Mathematics, Music, Art, Architecture, Culture. Seoul, Korea. 2014

VIEIRA, A. K. **Design generativo – estudo exploratório sobre o uso de programação no design**. 2014. 117f. Monografia (Graduação em Design) – USP, Universidade de São Paulo. São Paulo, 2014.

WORDEN, A. G. **Emergent Explorations: Analog and Digital Scripting**, 2011, 161 f. Dissertação (Mestrado em Arquitetura) - Virginia Polytechnic Institute and State University, Blacksburg, 2011.

WOVNS. **A platform for custom woven textiles**. Disponível em: <<https://www.wovns.com/>> Acesso em: março de 2019.

YU, Rongrong; GERO, John; GU, Ning. **Impact of Using Rule Algorithms on Designers' Behavior in a Parametric Design Environment: Preliminary Result from a Pilot Study**. CAAD Futures, CCIS 369, pp. 13–22, 2013.