

Tipografia generativa: inovações e aplicações criativas no design gráfico

Tipografía generativa: innovaciones y aplicaciones creativas en el diseño gráfico

Generative typography: innovations and creative applications in Graphic design

Crislayne S. Pereira, Josinaldo B. Silva

tipografia generativa, tipografia experimental, design tipográfico

Esta pesquisa explora a tecnologia generativa no design gráfico, investigando seu uso na criação de tipografias digitais por meio da revisão de literatura e do uso de ferramentas como Processing, p5.js, Cavalry e NodeBox. O estudo identifica as variáveis que influenciam o processo criativo e mostra que as tecnologias generativas oferecem várias possibilidades criativas, desde animações simples até sistemas complexos com partículas e objetos 3D, potencializando o processo de criação no design gráfico, permitindo a personalização visual e a comunicação dinâmica.

Generative typography, Experimental typography, Typographic design

This research explores generative technology in graphic design, investigating its use in the creation of digital typefaces through a literature review and tools such as Processing, p5.js, Cavalry, and NodeBox. The study identifies the variables that influence the creative process and shows that generative technologies offer various creative possibilities, from simple animations to complex systems with particles and 3D objects, enhancing the creative process in graphic design for visual customization and dynamic communication.

Tipografía generativa, Tipografía experimental, Diseño tipográfico

Esta investigación explora la tecnología generativa en el diseño gráfico, investigando su uso en la creación de tipografías digitales mediante una revisión bibliográfica y herramientas como Processing, p5.js, Cavalry y NodeBox. El estudio identifica las variables que influyen en el proceso creativo y muestra que las tecnologías generativas ofrecen diversas posibilidades creativas, desde animaciones simples hasta sistemas complejos con partículas y objetos 3D, potenciando el proceso de creación en el diseño gráfico para la personalización visual y la comunicación dinámica.

1 Introdução

A representação do texto escrito passou por diversos processos de produção e impressão ao longo do tempo. Com a evolução dos computadores na década de 80, as tipografias digitais se popularizaram, oferecendo maior liberdade para criar e experimentar tipos, permitindo que qualquer pessoa criasse fontes.

Uma outra forma de gerar fontes é por meio do uso da tecnologia generativa, que tem suas origens em 1979, com a Metafont de Donald Knuth, precursor direto das técnicas de tipografia generativa. No entanto, foi apenas na década de 80 que ela começou a ganhar destaque,

impulsionada pela popularização dos computadores pessoais e pelo trabalho de designers como Erik van Blokland e Just van Rossum, criadores da Beowolf, o primeiro tipo programático da LettError para comercialização.

Para Silva e Silveira (2017), os tipos generativos são fontes com comportamentos variáveis normatizados por algoritmos, funcionando como "organismos" autônomos, mediados pelo usuário por meio de interfaces digitais. Vladimir Kuchinov define a tipografia generativa como qualquer texto interpretado por um algoritmo, projetado pelo autor, em que um sistema ou *script* cria variações a partir da estrutura e dos parâmetros definidos. O pesquisador e artista Phillip Galanter (2003) define a arte generativa como uma prática em que o artista usa um sistema (como um conjunto de regras de linguagem natural, um programa de computador, uma máquina ou outra invenção procedural) que é colocado em movimento com algum grau de autonomia.

Segundo Pereira (2019), a integração dessas tecnologias no design tem aberto novas possibilidades antes inimagináveis. Ele observou como os designers passaram a explorar processos generativos em projetos como identidades visuais dinâmicas e tipografia, superando as limitações das abordagens tradicionais e gerando soluções experimentais e distintas.

2 Objetivos

Este estudo tem como objetivo geral analisar o uso de ferramentas generativas na criação de tipografias experimentais, investigando como esses recursos contribuem para a exploração de novas estéticas, processos e linguagens no design tipográfico. Para isso, os objetivos específicos são: compreender os princípios e variáveis envolvidas na aplicação de métodos generativos à tipografia experimental; avaliar as ferramentas, técnicas e tecnologias utilizadas na produção de tipografias com abordagem generativa; realizar experimentações práticas com ferramentas generativas, observando suas funcionalidades, possibilidades estéticas e técnicas, de modo a identificar os desafios e as potencialidades dessa prática no contexto do design contemporâneo; e, por fim, explorar como a tipografia generativa pode expandir os limites da criação tipográfica tradicional.

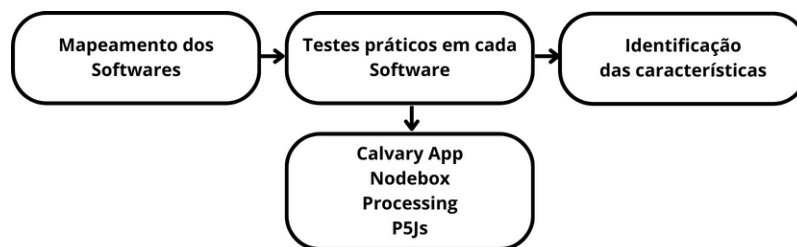
3 Metodologia

A pesquisa começou com uma revisão da literatura sobre tipografia experimental e generativa, com o objetivo de estabelecer uma base teórica sólida. Em relação à tipografia experimental, comparamos diversas definições de autores que, muitas vezes, se contradizem. Barbosa (2019) define-a como uma abordagem artística e menos convencional do espaço visual tipográfico, rompendo limites e permitindo uma exploração funcional e estética. Rocha (2002) aponta que o experimentalismo pode seguir direções como mistura de estilos, distorções óticas e caligrafias inusitadas. Silva (Citado por Vilela, 2011), Espinoza (2018) e Guerrero (2014) veem a tipografia experimental como um processo técnico e expressivo, expandindo os limites

do desenho tipográfico além do lápis e papel. Por outro caminho, David Carson, a descreve como algo ainda não visto ou ouvido, algo inédito (Carson citado por Bilak, 2005), Rocha (2002) e Triggs (2003) consideram-na apenas uma forma de teste ou algo amador.

Posteriormente, realizamos um mapeamento sobre a produção de tipografias generativas, que pode ser feita por meio de códigos em ambientes de programação como Processing e p5.js, ou ainda por meio de sistemas específicos, como o Cavalry, ou diretamente em sites. O passo seguinte foi executar testes práticos nos softwares para compreender seu funcionamento, potencial e identificar padrões entre as ferramentas. O diagrama na figura 1 mostra o desenvolvimento da primeira etapa da pesquisa.

Figura 1: Diagrama do desenvolvimento da primeira etapa da pesquisa (autor)



4 Resultados e discussão

Foram testadas quatro ferramentas tecnológicas que possibilitam a manipulação tipográfica de maneira generativa. O Processing e o p5.js utilizam linguagens de programação como Java e JavaScript para a criação de fontes no modelo generativo. O Cavalry, desenvolvido para animações 2D, também oferece recursos para a criação de artes generativas, permitindo o uso de JavaScript ou programação procedural por meio de conexões e deformadores do sistema. O quarto software testado foi o NodeBox, que possibilita a programação visual por meio de um sistema de "nós".

O Processing e o p5.js são ferramentas essenciais para a programação criativa, com origens interligadas. Criado em 2001, o Processing foi desenvolvido para simplificar a integração entre código e visualização gráfica. Utilizando Java como base, ele reduz a complexidade do código, permitindo criar gráficos com menos linhas em comparação ao Java (Mello, 2015). O p5.js, surgiu posteriormente, sendo uma adaptação do Processing para JavaScript (Santos Costa & Sigrist, 2023, p. 44). Isso o torna uma escolha natural para quem deseja criar gráficos interativos que se integrem facilmente com HTML e outras tecnologias da web.

As ferramentas permitem criar desde animações simples até sistemas complexos de partículas e objetos 3D. O p5.js permite exportar gráficos em formatos como .png, .jpg e .svg, com limitações de resolução. Segundo Voicu (2017), apesar das restrições para impressão de alta qualidade, ambos são poderosas ferramentas no campo da programação generativa, com uma curva de aprendizado suave, ampla documentação e uma comunidade ativa. Isso as torna

ferramentas ideais para artistas e designers que desejam explorar a programação criativa, mesmo sem grande familiaridade com códigos.

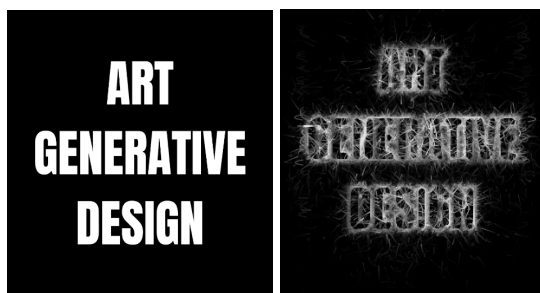
O Cavalry é um aplicativo de animação 2D que combina a flexibilidade do 3D com a facilidade do 2D, sendo totalmente procedural. Ele permite definir regras e parâmetros que controlam automaticamente o comportamento dos elementos animados, criando animações adaptáveis a mudanças de dados ou condições externas. O software também integra dados em tempo real, gerando visões artísticas ou padrões baseados em variáveis. Também permite o uso de *scripts* para automatizar processos e criar efeitos personalizados. Sua compatibilidade com diversos formatos de exportação, como vídeos, GIFs, SVG e gráficos vetoriais, torna-o versátil para diferentes projetos.

O NodeBox é um ambiente de programação visual para criação de gráficos e animações baseadas em dados. Ele gera imagens vetoriais e animações usando uma interface de *nodos*, onde cada *nodo* representa uma função ou processo, e a conexão entre eles define a lógica e o fluxo de dados.

Assim como o Cavalry, o NodeBox utiliza programação procedural, onde algoritmos definem o comportamento e a forma dos elementos gráficos. Na tipografia generativa, isso permite a criação de fontes e estilos adaptáveis a parâmetros dinâmicos ou dados externos. Diferente dos outros softwares, que utilizam Java ou JavaScript, o NodeBox é baseado em Python, com recursos adicionais (Sant'Anna, 2022).

Embora o Cavalry e o NodeBox sejam voltados para usuários sem experiência em programação, ambos têm uma oferta limitada de materiais de estudo, o que dificulta o acesso ao suporte necessário. Além disso, encontramos o Chromata, uma ferramenta de arte generativa criada por Michael Bromley, que usa JavaScript e canvas HTML. Ao contrário dos outros programas, o Chromata apenas requer uma imagem para transformá-la em arte generativa.

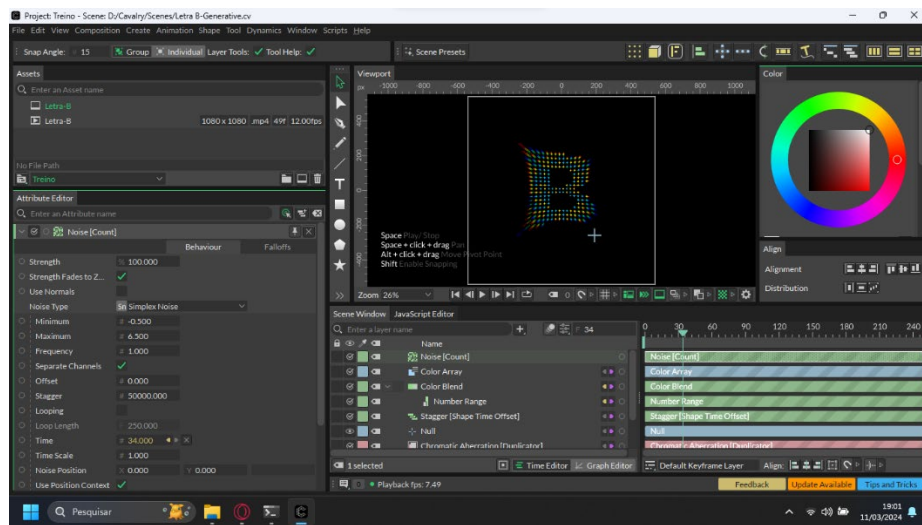
Figura 2: A esquerda uma imagem com texto e a direita a imagem transformada em arte generativa no site chromata – (autor, 2024)



Observamos que todos os softwares, independentemente do tipo de programação, exigem uma tipografia base para manipulação das formas das letras. No entanto, para criar um desenho original sem uma fonte base, é necessário desenvolver os caracteres apenas por meio de código, o que torna o processo trabalhoso e demorado.

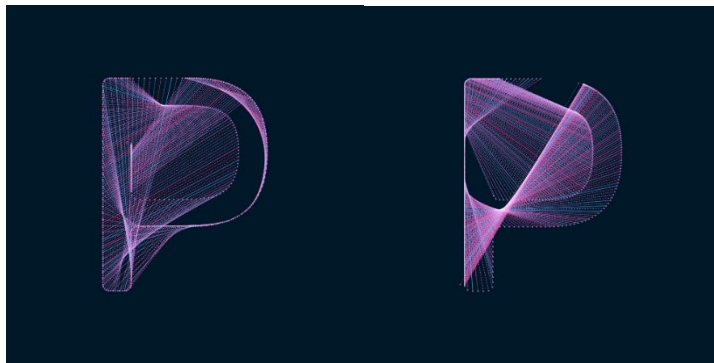
Podemos ver nas figuras abaixo alguns exemplos de tipografia generativa testadas no Cavalry, utilizando os parâmetros e modificadores que o software disponibiliza.

Figura 3: Tipografia Generativa no Cavalry (autor, 2024)



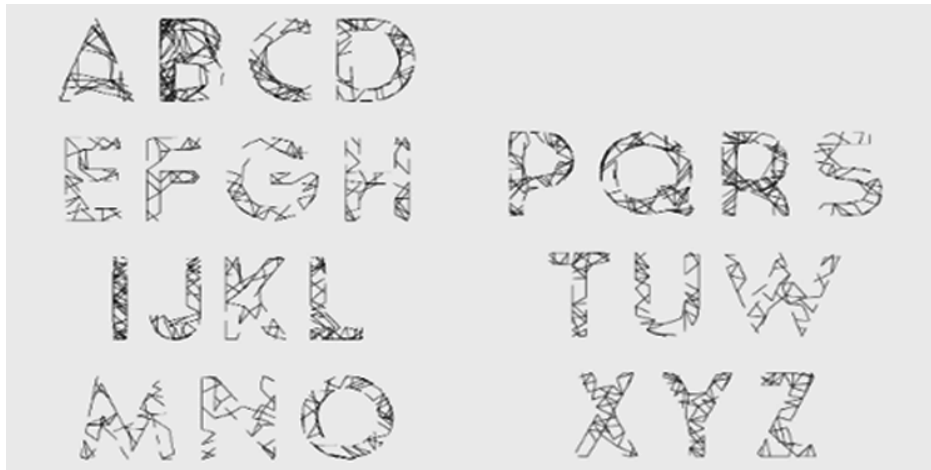
Na figura 4, é apresentada uma possibilidade de gerar tipografias a partir do tempo de captura de uma animação, exemplificada pela letra "P". Cada frame capturado pode dar origem a uma fonte, oferecendo assim inúmeras possibilidades criativas.

Figura 4: Tipografia Generativa no Cavalry e modificadores utilizados (Mario de Mayer, 2024)



A fonte Kocoon (figura 5) é uma fonte experimental criada com o framework de código aberto Nodebox. O conceito principal é o valor "aleatório". Que permite a cada vez exportar um conjunto de tipos ligeiramente diferentes.

Figura 05: Tipografia Kocoon criada no nodebox (Giacomo Vergano, 2012)



5 CONSIDERAÇÕES FINAIS

Segundo Pereira e Silva (2024), a exploração de novas formas de desenvolvimento tipográfico, incluindo o uso de computadores, oferece aos designers a oportunidade de visualizar e experimentar possibilidades tipográficas mais amplas, indo além das questões tradicionais de legibilidade e escolha de fontes.

Dessa forma, podemos concluir que a tipografia generativa é uma ferramenta valiosa para a criação de fontes personalizadas e únicas, aplicáveis a diversos contextos, permitindo transcender os limites tipográficos convencionais. Ela tem o potencial de redefinir não apenas a forma como entendemos a tipografia, mas também como nos conectamos com ela. Em resumo, a arte generativa aplicada às fontes impulsiona uma evolução contínua na maneira como nos comunicamos, interagimos e experienciamos o mundo visual ao nosso redor.

Referências

- Bilak, P. (n.d.). Experimental typography. Whatever that means. Recuperado de <https://www.typotheque.com/articles/experimental-typography-whatever-that-means>. Acesso em 4 de fevereiro de 2023.
- Espinoza, J. L. (2018). Estudio iconográfico de las manifestaciones textiles de la Parroquia Cacha para la creación de una tipografía experimental. Recuperado de <http://dspace.unach.edu.ec/handle/51000/4767>
- Galanter, P. (2003). What is Generative Art? Complexity theory as a context for art theory. In GA2003-6th Generative Art Conference. Recuperado de http://www.philipgalanter.com/downloads/ga2003_paper.pdf. Acesso em 3 de junho de 2024.
- Guerrero, M. (2014). Diseño tipográfico experimental (v. 6). Recuperado de https://www.academia.edu/download/48800462/Diseno_tipografico.pdf
- Mello, P. O. B. D. (2015). *Arte e programação na linguagem Processing* (Dissertação de

Mestrado, Programa de Estudos Pós-Graduados em Tecnologias da Inteligência e Design Digital, Pontifícia Universidade Católica de São Paulo). São Paulo.

Pereira, C. D. S., Anjos, K. F. S. D., & Silva, J. B. D. (2024). Além das palavras: A expressão estética da comunicação através da tipografia experimental com fungos. Blucher Proceedings, São Paulo.

Pereira, F. A. D. S. (2019). O Esqueleto à Forma: Explorações Algorítmicas na Criação de Glifos (Dissertação de Mestrado, Universidade de Coimbra, DEI-FCTUC).

Rocha, C. (2002). Projeto Tipográfico: Análise e produção de fontes digitais (2ª ed.). São Paulo: Edições Rosari.

Santos Costa, D. D., & Sigrist, V. C. (2023). WEB ARTE: A biblioteca p5.js e o estudo de caso do projeto cápsulas sonoras. Revista Extensão, 23(1), 41-50. Recuperado de <https://periodicos.ufrb.edu.br/index.php/revistaextensao/article/view/3031>. Acesso em 20 de abril de 2024.

Sant'Anna, H. C. (2022). Entre foguetes, estrelas e canetas: Relato de experiência de ensino de princípios da Computação para estudantes de graduação em Design. *Projetica*, 13(3), 166–183. <https://doi.org/10.5433/2236-2207.2022v13n3p166>. Acesso em 29 de julho de 2024.

Silva, C. P., & Silveira, L. C. (2017). Sound system: Sistema tipográfico generativo baseado em algoritmos musicais (Trabalho de Conclusão de Curso, Universidade de Brasília).

Vilela, D. N., & Silva, J. B. D. (2011). Um olhar sobre os processos experimentais para criação de fontes tipográficas. p. 4-7.

Voicu, D. (2017). Which to Choose: Processing or p5.js. Recuperado de <https://medium.com/tab-space/which-to-choose-processing-or-p5js-7cbba6fb6e11>. Acesso em 9 de março de 2024.

Sobre o(a/s) autor(a/es)

Crislayne S. Pereira, IFPE, Brasil <csp4@discente.ifpe.edu.br>

Josinaldo B. Silva, Dr, IFPE, Brasil <josinaldobarbosa@recife.ifpe.edu.br>