

# *A poesia expandida pelo trânsito digital*

Carlos Murilo da Silva Valadares  
Serpro – Serviço Federal de Processamento de Dados

*Resumo: Este ensaio propõe uma reflexão acerca da poesia digital, no contexto das interações entre o texto e os algoritmos computacionais para a construção de sentido poético. A principal vertente explorada no artigo será a poesia digital performática, entendida como expressão escrita veiculada por meio da execução de algoritmos elaborados em alguma linguagem de programação de computadores, cuja existência é um acontecimento temporário, que somente pode ser percebida durante a execução dos programas que a definem. Analisaremos algumas possibilidades de fusão/interação entre palavras, imagens e efeitos digitais, a partir de um levantamento sobre como palavras e algoritmos têm sido utilizados como forma de expressão poética. O ensaio aponta então para a poesia expandida, que além de permitir a interferência do leitor, sugere novos olhares sobre a poesia contemporânea. Finalizaremos o artigo pela análise da implementação de um pequeno ensaio poético-digital, desenvolvido por meio do ambiente Processing, um dialeto derivado da linguagem de programação Java.*

*Palavras-chave: Poesia digital, poesia expandida, arte eletrônica.*

## Poesia, poética e computadores

O ensaio debaterá os processos interativos entre a palavra escrita de intenção poética e as tecnologias digitais, como um fenômeno contemporâneo de construção de acontecimentos de sentido estético. Domínio em que prevalece a interdisciplinaridade, onde se resvalam desenho, computação, matemática, física, filosofia, semiótica e poesia. Os objetos específicos de nosso interesse são aqueles que se encaixam na categoria **poesia digital performática**, entendida como **[a] toda manifestação de poesia escrita gerada e exibida necessária e inequivocamente a partir da execução de um algoritmo, escrito em alguma linguagem de programação, e executada exclusivamente em um dispositivo microprocessado (dimensão digital); [b] que solicita a participação do espectador, por meio da ativação dos sentidos (dimensão performática)**. Consideramos as experiências em arte e poesia digital como durações, que encerram permanências de ordem visual e narrativa, matemática e probabilística. Como entidade imaterial, mas de presença perceptível, um programa de computador só existe enquanto está em execução, ocupando um espaço de representação, revelando, portanto, uma dimensão performática. Este momento de existência, combinado à interação com o espectador, acarretará o que Cohen (1998) denomina “ritualizador do instante presente”. Esse momento estruturado de atuação sobre o objeto permitirá que o espectador construa experiências estéticas e significados diferentes e mais amplos do que aqueles obtidos a partir da simples leitura do poema impresso.

A dimensão “performática”, presente em nossa definição do objeto de interesse, possui aqui um significado específico, que atua como um invólucro que delinea as fronteiras entre a obra de arte e a interferência do espectador. Roselee Goldberg<sup>1</sup> reclama para a performance a possibilidade da “experiência do tempo, do espaço e do material”, elementos que antes estiveram relacionados à arte conceitual. Cohen oferece um ponto de vista semelhante, percebendo a performance como uma função do espaço-tempo:  $P = f(s, t)$ , mas que também “acaba tocando os limites tênues que separam a vida e a arte”.<sup>2</sup> Para estes autores, a performance será então uma arte de fronteira. Estas concepções são perfeitamente condizentes com nossa percepção: a arte digital, assim como a performance, caracteriza-se como uma existência em processo. A execução de um programa de

1. GOLDBERG. *A arte da performance: do futurismo ao presente*.
2. COHEN. *Performance como linguagem*.

computador que foi criado para corporificar intenções estéticas, pode ser então reconhecida como uma performance, um acontecimento que existe em um determinado momento e situação do espaço/tempo, e que não pode ser presenciado novamente, a não ser que o programa seja novamente posto a funcionar. A arte digital performática também carrega consigo uma característica que a distingue da performance humana: o deslocamento mais intenso do papel do sujeito/espectador, ao solicitar dele uma atitude participativa, em lugar da mera contemplação. A tecnologia oferece ao indivíduo a oportunidade de interferir na execução dos algoritmos, de muitas formas e em diferentes intensidades. Isto não deve causar estranhamento, nem nos deve levar a pensar que a participação do público destitui a obra de sua originalidade estética. Novamente nos encontramos em terreno movediço, pois a própria performance não possui seus limites totalmente definidos, por constituir uma forma naturalmente híbrida de expressão.

A questão da atualidade da arte digital também surge como uma discussão premente. Serão as obras deste domínio mais ou menos modernas do que poemas impressos? Há um conjunto de fatores que permeiam esta pergunta, pois os artefatos poéticos de gênese virtual constituem híbridos complexos, de lugares ainda não plenamente estabelecidos. Na realidade, nada garante que a tecnologia possa configurar aos objetos que alcança uma atmosfera de modernidade.<sup>3</sup> Do ponto de vista da criação artística, a concepção de manifestações estéticas de natureza tecnológica, tais como vídeo ou filmes de cinema, não representa nenhuma novidade. O que existe de novo encontra-se justamente nas novas possibilidades visuais, viabilizadas pelo considerável avanço das tecnologias eletrônico-digitais no século XXI.

A tecnologia, este domínio de conhecimento muito recente, imprimiu suas feições em praticamente todos os aspectos da sociedade e da produção cultural. Seus princípios são aparentemente claros e perfeitamente reconhecíveis, e, aos olhos do espectador desavisado,<sup>4</sup> não parecem carecer de análise e justificativa. Conduzem uma certa aura, algo que parece nos atirar para um momento futuro, acima de nossa compreensão imediata. Haroldo de Campos chama a isso de a “provisoriedade do estético”, percebendo as novas formas de arte como construções fugazes, características da contemporaneidade, dotadas de velocidade,

3. GALIMBERT. *Psiche e Techne*: o homem na Idade da Técnica.

4. Este caso pode incluir até mesmo especialistas, como por exemplo, analistas de sistemas ou engenheiros de computação.

de um acontecer que logo se extinguirá.<sup>5</sup> Isso tudo delinea algumas feições da arte eletrônica, mas outras facetas são reveladas a cada nova obra. O acaso, a probabilidade, a incerteza, são outros fenômenos que se encontram na gênese da poesia digital e de toda a arte eletrônica.

### Computadores, suas linguagens e a poesia

Procuremos então conhecer um pouco melhor a natureza e os modos operacionais dos computadores, no sentido de entender como suas linguagens podem moldar textos para convertê-los em poesia, e assim visualizarmos que arte disto surgirá. Em termos conceituais, os computadores e suas linguagens são artefatos lógico/matemáticos, operadores de exatidões, enquanto a poesia é criação, subjetividade, abstração, incerteza. Dito assim, parecem fenômenos opostos, e que de uma linguagem computacional será impossível extrair poesia. Mas a capacidade de realização do belo pode ser considerado um atributo existente em qualquer linguagem. Como se pode afirmar, por exemplo, que não encontraremos nenhum sentido de beleza na relação do Princípio da Incerteza de Heisenberg, expressado por

$$\Delta x \Delta p \approx \hbar$$

que representa a impossibilidade de se conhecer simultaneamente, com absoluta precisão, a posição e quantidade de movimento de qualquer partícula fundamental da matéria.<sup>6</sup> Não há, portanto, nenhuma razão para que a poesia não possa acontecer a partir das interações entre regras matemático/computacionais e palavras. Devemos aqui considerar o fato de que as linguagens de programação são, antes de tudo, linguagens. Possuem estrutura gramatical e semântica, permitindo a construção de entidades linguísticas, dotadas de sentido próprio. Articulam todos os atributos de qualquer linguagem de comunicação humana, e por isso, possuem os elementos essenciais à produção de sentido poético. Alguns autores entendem os computadores como máquinas processadoras de símbolos. Esta ideia deriva do pensamento de Allen Newell,<sup>7</sup> que primeiro elaborou o conceito de “sistema

5. CAMPOS. *A arte no horizonte do provável*.

6. ALONSO; FINN. *Física*.

7. NÖTH. *Máquinas semióticas*.

simbólico físico”, para se referir a máquinas capazes de processar números e símbolos. Segundo Nöth, Newell pretendia sintetizar a “ponte teórica entre a ciência dos seres vivos inteligentes, isto é, as ciências cognitivas, e a ciência das máquinas inteligentes, isto é, a ciência da computação”.<sup>8</sup> É longa a discussão sobre a possibilidade de o computador alcançar semiose, pois ele próprio não pode realizar leituras sobre o que ele processa. O entendimento é todo do sujeito humano que o programou. Mas é ainda Nöth que argumenta serem “os processos nos quais as máquinas servem como mediadoras na semiose humana são certamente processos de semiose genuína”. Evidentemente, os símbolos a que estes autores se referem são as linguagens de programação e seu conjunto de instruções, que efetuam operações lógicas e matemáticas bastante específicas. Existem hoje dezenas de linguagens de programação, que atendem a um amplo conjunto de finalidades.

Que máquinas são os computadores em termos operacionais? Na época de sua invenção, em meados do século XX, as tecnologias da informação foram uma das mais importantes conquistas da inventividade humana. Não havia nenhum dispositivo que se comportasse como um computador, a ponto de o cinema chegar a criar interpretações fantasiosas para o funcionamento dos computadores, contribuindo para a formação de uma mitologia das “máquinas pensantes e dominadoras”. No entanto, nenhuma máquina, nem mesmo computadores, podem pensar, na forma como fazem os seres humanos. Constituem, em termos técnicos, dispositivos eletrônicos de processamento de dados, que executam somente tarefas que lhes foram ordenadas e bem definidas, descritas por meio de linguagens de programação. Podem, no máximo, potencializar competências intelectuais e físicas humanas, para a execução de tarefas complexas ou repetitivas, difíceis ou impossíveis de serem desenvolvidas por pessoas.

Constata-se, então, que os computadores são artefatos que comportam paradoxos. Limitados em seu domínio de competências, mas grandemente versáteis naquilo que podem fazer. Desde a sua criação, muitas atividades do complexo cultural da humanidade lhes foram delegadas. Hoje, a variedade e o alcance das atividades executadas por essas máquinas é de tal monta que, mesmo aqueles que trabalham diretamente com tecnologias, desconhecem a verdadeira extensão da interferência destes artefatos em suas vidas. O progresso científico e cultural torna-se mais e mais dependente das tecnologias da informação. Poucas são as

8. NÖTH. Máquinas semióticas.

áreas de ciência que não utilizam diretamente os recursos da computação, que se torna lenta e inexoravelmente, um dos substratos do conhecimento científico.

### Poesia expandida: probabilidades, permutações, acasos e incertezas

Vamos então pensar a poesia expandida pelo viés das tecnologias da eletrônica, embebida de probabilidades, movimento, matemática e velocidade. Uma breve cronologia de realizações nos permitirá compreender como estes artefatos e seus criadores trilharam o caminho da exploração sensorial da palavra.

A partir do início do século XX, os escritos literários passaram a frequentar suportes que, lhes eram, anteriormente, absolutamente estranhos. Das superfícies físicas e à primeira vista inanimadas, papel, madeira, pedra, metal, os textos aportaram em lugares de movimento, cinema, vídeo e, por fim, as telas dos computadores. A torrente de invenções midiáticas da era moderna obrigou o leitor a mudar seu comportamento diante da palavra, não mais se limitando a somente ler o texto, passando a se engajar em operações de caráter participativo, como tocar na palavra ou assistir à sua movimentação em uma tela. Como observa Melo e Castro (2008), desde aquele momento, “as fontes de percepção humana vêm sofrendo uma aceleração”. O número de possibilidades de manipulação das palavras, consideradas aqui como unidades expressivas, cresceu à medida que as tecnologias midiáticas evoluíram. A poesia então passou da impressão plana para objeto perceptivo multidimensional, que lança um evento sensorial em direção ao espectador e o incita a responder como melhor lhe parecer. O encontro entre texto poético e programas de computador só aconteceu depois da metade do século XX, de modo extremamente súbito e contundente. A natureza das linguagens de programação causou, a princípio, certa desconfiança em relação à sua adequação como símbolos realizadores de significado, o que não poderia ser de outra forma. As diferenças aparentes entre os idiomas falados, pensemos na língua portuguesa, e as linguagens simbólicas computacionais são grandes, além de serem conhecidas e praticadas por restrito círculo de desenvolvedores de *software*. Portanto, os não especialistas podem entender que uma linguagem de programação não seria capaz de “dizer” ou “significar” tanto quanto um idioma. De fato, as aplicações de uma linguagem de programação no cotidiano são mais limitadas, por serem muito decodificadas e especializadas, o que inviabiliza seu uso como meio primário de

comunicação. Isto nos leva a um aspecto importante acerca da poesia digital e da arte eletrônica: as primeiras obras desse domínio não foram realizadas por artistas, mas por técnicos, frequentemente engenheiros de computação e matemáticos. À época de seu surgimento, os computadores não eram máquinas de operação facilmente compreensíveis, mas artefatos complexos que exigiam profundos conhecimentos acerca de sua arquitetura interna. Apesar disso, as primeiras manifestações de arte digital não foram desprovidas de sentido estético, e evidenciaram de forma apreciável a capacidade criadora e a curiosidade de quem as produziu.

As primeiras tentativas de produção de poesia eletrônica surgiram no final dos anos de 1950, pelas iniciativas de técnicos como Theo Lutz (1959). Seu trabalho consistiu em uma “poesia artificial cibernética”,<sup>9</sup> intitulado “Texto estocástico”, no qual a ordem de um conjunto pré-determinado de palavras era sorteada, por meio de um programa que executava cálculos probabilísticos, gerando sentenças aleatórias e sempre diferentes, que apresentavam a seguinte configuração:

#### QUADRO 1

Trecho do poema “Texto Estocástico”, de Theo Lutz.

NICHT JEDER BLICK IST NAH. KEIN DORF IST SPAET.  
EIN SCHLOSS IM FREI UND JEDER BAUER IST FERN.  
JEDER FREMDE IST FERN. EIN TAG IST SPAET.  
JEDES HAUS IST DUNKEL. EIN AUGE IST TIEF.

Nenhum outro efeito visual foi programado além da impressão da palavra em papel, pois os sistemas digitais da época não permitiam a elaboração de construções visuais complexas. Mas o algoritmo executava uma real intervenção matemática sobre construção do sentido fraseológico e marcou o início da poesia digital. Max Bense chamou-a de poesia artificial, entendendo ser ela “aquela espécie de poesia na qual – na medida, inclusive, em que ela seja produzida por meios mecânicos – não há nenhuma consciência poética pessoal, com suas experiências, vivências, sentimentos, lembranças, pensamentos, representações de uma faculdade imaginativa”.<sup>10</sup> A ausência de consciência refere-se, evidentemente, ao processamento lógico-matemático do computador, que não pensa, apenas obedece ordens. A discussão estava inaugurada, mas, como perceber os símbolos e os

9. Conceito criado por Max Bense, segundo Jorge Luiz Antonio (2008).

10. BENSE. *Pequena estética*.

resultados do trabalho da linguagem de programação? Uma semiótica da computação, voltada para a apreciação da arte eletrônica, não estava criada e ainda é motivo de debates. Consideremos a definição de Vera Casa Nova, acerca do objeto da semiótica: “O objeto da semiótica será tudo aquilo que se colocar para um sujeito-leitor como passível de leitura, sendo este o momento da produção de sentido.”<sup>11</sup> Podemos então apontar os componentes simbólicos das linguagens de programação como genuínos objetos semióticos, pois são certamente passíveis de leitura. As linguagens são efetivamente formadas por signos, que possuem funções específicas e estabelecem relações de significante e significado. No entanto, existe um componente que torna essa leitura um tanto complexa. Os signos destas linguagens não constituem uma mensagem, e sim os veículos produtores da mensagem. Encontramos então dois níveis de leitura: a leitura dos símbolos da linguagem e a leitura de seus resultados, e ambos não serão necessariamente coincidentes. Os códigos que animam um poema pouco ou nada se parecem com o resultado final. Então, ler os signos que formam um programa não conduzirá o leitor a compreender ou visualizar a obra em execução.

A maioria dos experimentos iniciais em poesia eletrônica se concentraram na utilização de comportamentos combinatórios ou probabilísticos. Deste grupo fez parte o programador Ian Sommerville, que em 1960 escreveu um programa que para gerar o poema “I am that I am”, de Brion Gysin, cuja ordem das palavras eram escolhidas aleatoriamente. As experiências brasileiras desse período muito se assemelharam às obras de artistas de outros países, e colocaram o país no contexto da poesia eletrônica de vanguarda. O primeiro poema elaborado com a utilização de recursos tecnológicos no Brasil, segundo Antonio (2003), recebeu o título de “Poema elétrico”, e foi realizado em 1960 pelo artista neoconcretista Albertus Marques. A obra consistia em um artefato eletro-mecânico, composto por um aparato contendo um botão a ser apertado, que por sua vez era conectado a uma tela. Ao pressionar o botão, aparecia a palavra “FIM”, que desaparecia assim que o botão era solto, encerrando então o evento poético. Antonio comenta que o sentido poético deste trabalho encontra-se justamente na “possibilidade e poder de uma ação”, o simples resultado de uma ação completa. Os primeiros experimentos totalmente digitais foram realizados por Waldemar Cordeiro, artista que se engajou no movimento de arte concreta, juntamente com Décio Pignatari, Haroldo de Campos, e Augusto de Campos. Em um trabalho de

11. CASA NOVA. Introdução à semiótica, p. 11.



1968, denominado “Programa Beabá”,<sup>12</sup> Waldemar elaborou um evento de combinações aleatórias de seis letras, três vogais e três consoantes, em mais um exemplo da colaboração entre o poeta e o técnico. A sequência de procedimentos foi elaborada por Waldemar Cordeiro, e Giorgio Moscati escreveu o algoritmo, utilizando a linguagem FORTRAN IV<sup>4</sup>, em um computador IBM 360/44, na Universidade de São Paulo. A execução do poema obedecia a um conjunto de regras: **[a]** as palavras geradas pelo programa teriam obrigatoriamente 6 letras; **[b]** as palavras seriam formadas com alternância de vogais e consoantes, e **[c]** a probabilidade de ocorrência de conjuntos do tipo cb e bc deveria obedecer à probabilidade de aparecimento destes conjuntos na língua portuguesa.

Em muitos sentidos, a poesia concreta representou um processo de expansão da poesia, e é possível detectar estreitas relações com a poesia digital. Augusto de Campos, em seu texto “Poesia concreta”,<sup>13</sup> propõe que a poesia é “concreta no sentido em que, postas de lado as pretensões figurativas da expressão (o que não quer dizer: posto à margem do significado), as palavras nessa poesia atuam como objeto autônomos”. De fato, muitos artefatos poéticos digitais compartilham estas premissas, e exibem palavras que adquirem comportamento individualizado, derivado de propriedades e operações matemáticas, elaborando significados próprios e momentâneos. Campos dirá ainda que “os poemas concretos caracterizar-se-iam por uma estruturação ótico-sonora irreversível e funcional e, por assim dizer, geradora da ideia, criando uma entidade todo-dinâmica, ‘verbivocovisual’”.<sup>14</sup>

Até o fim dos anos de 1960 e durante praticamente toda a década seguinte, os artefatos poéticos desenvolvidos em computadores não avançaram de modo significativo, em termos técnicos e visuais. Durante esses anos, as interfaces de exibição dos computadores e as linguagens de programação não sofreram melhorias suficientes para a implementação de objetos visuais mais elaborados. Os computadores não sofreram implementações importantes em seus modelos de interatividade. Um livro publicado em 1977, *The Best of Creative*

12. MEDEIROS. Dialética concretista: o percurso artístico de Waldemar Cordeiro.

13. CAMPOS *et al.* *Teoria da poesia concreta: textos críticos e manifestos 1950-1960.*

14. CAMPOS *et al.* *Teoria da poesia concreta: textos críticos e manifestos 1950-1960.*

*Computing*, v. 2, editado por David Ahl, atualizava o público no que havia de mais avançado em arte computacional até aquele momento. Em alguns de seus capítulos, trazia impressões de desenhos elaborados por algoritmos e imagens compostas por arranjos matriciais de letras e números. Em um exemplo emblemático, Paul Emmerich apresenta o algoritmo “Haiku Generator”, cuja finalidade era gerar exemplares de poemas no modelo japonês de mesmo nome. O programa foi escrito na linguagem *Basic*, e parte dele tinha aparência vista no Quadro 2.

#### QUADRO 2

Trecho do código do programa escrito por Paul Emmerich

```
1100 REM *****VERB ROUTINE
1110 IF A>9 GOTO 1130
1120 ON A GOTO 1150, 1190, 1230, 1350, 1390, 1430, 1470
1130 LET A=A-9
1140 ON A GOTO 1170, 1210, 1250, 1290, 1330, 1370, 1410, 1450, 1490
1150 LET A$="RUN"
1160 RETURN
1170 LET A$="SHINE"
```

Frank Popper,<sup>15</sup> pesquisador de história da arte e estética, considera que são poucos os exemplos de arte eletrônica produzida antes dos anos 80 que merecem ser citados. A observação parece um pouco radical, mas os anos de 1980 trouxeram novas perspectivas de desenvolvimento de arte computadorizada, com o advento do *Personal Computer*. Não cabe aqui uma discussão acerca de controvérsias relativas à primazia da invenção dos computadores pessoais, tema de acaloradas discussões. O que importa é que, de repente, aconteceu uma súbita ampliação das possibilidades técnicas e criativas, pelas consideráveis melhorias tecnológicas e pelo barateamento dos computadores. Boa parte dos experimentos desse período exibiam abstrações, como reflexo das formulações matemáticas que geravam as imagens.<sup>16</sup> Mas o caminho para a experimentação digital e interatividade com o espectador estava finalmente aberto: Peter Weibel, diretor do Institute for New Media, declara que “pela primeira vez na história, a imagem é um sistema dinâmico”.

15. Professor Emérito de Estética e Ciência da Arte da Universidade Paris VIII, autor de *Art of the Electronic Age. Works on Art and Technology*.

16. RUSH. *Novas mídias na arte contemporânea*.

## A poesia estendida hoje

É então a partir do início dos anos de 1980 que podemos falar em processos computacionais interativos, por intermédio da mobilização dos sentidos do espectador. Segundo Grau (2009), as tecnologias digitais, linguagens de programação e os computadores, têm a capacidade de influenciar nossas formas de percepção do espaço e do tempo. Os experimentos geralmente procuram atingir estas dimensões, em diversas formas de manipulação. As muitas tentativas de classificações de tipos de poesia informacional contemporânea refletem a variedade e a dificuldade de ordenar conceitualmente a produção contemporânea, no que Vouillamoz afirma ser a “tradução do discurso literário a um formato digital”.<sup>17</sup> Antonio (2003) aponta que a poesia eletrônica de hoje constitui um gênero de poesia que elabora negociações com os processos digitais. Esta negociação significa, essencialmente, articular interações de natureza computacional entre o texto poético e os elementos básicos da comunicação visual: o ponto, a linha, a forma, a direção, o tom, a cor, a textura, a dimensão, a escala e o movimento.<sup>18</sup> A expansão da poesia acontecerá, então, no exato momento em que o poema, modificado materialmente para se fazer existir neste território, alcança nossos sentidos e, por isso, produz semiose.

Dondis (2002) reconhece no movimento uma das forças visuais mais dominantes da experiência sensorial humana. A poesia integrada à dimensão do movimento consiste na indução de deslocamentos espaciais da palavra na tela do computador, em conformidade com as instruções de um programa. É um fenômeno ilusório, que depende inteiramente de processos fisiológicos relacionados à “persistência da visão”,<sup>19</sup> mas muito apropriado para a exploração com finalidade estética. Ao movimento podem-se agregar outros elementos visuais, como direção, textura, dimensão e escala, em uma cadeia de provocações sensoriais.

Como exemplo de poesia em movimento, consideremos o trabalho de Thomasinom, que desenvolveu composições a partir de poemas de Apolinaire.<sup>20</sup>

17. VOUILLAMOZ. *Literatura e hipermedia: la irrupción de la literatura interactiva: precedentes y crítica.*

18. DONDIS. *Sintaxe da linguagem visual.*

19. DONDIS. *Sintaxe da linguagem visual.*

20. ANTONIO. Poesia visual e eletrônica no Brasil: alguns exemplos.

O trabalho consiste em uma corrente descendente de palavras, que caem em quatro fileiras em direção ao canto inferior direito da tela, mas não mantem a direção contínua, fazendo pensar no caminhar titubeante de um inseto. Esta obra permite relacionar a poesia digital ao cinema de animação, no sentido de que os movimentos precisam ser ensaiados e estabelecidos de antemão, para posteriormente serem implementados em um programa ou linguagem específica. Em termos computacionais, a elaboração de animações é hoje uma tarefa razoavelmente simples, dada a oferta de programas-linguagens que viabilizam sua implementação.

### O ambiente *Processing*

Nesta seção, apresentaremos o ambiente *Processing*, um artefato de programação imaginado no intuito de oferecer uma opção de desenvolvimento de objetos visuais para artistas e designers, de um modo que pudesse liberá-los da necessidade de solicitarem a ajuda de cientistas da computação. O objetivo desta apresentação é ilustrar como, em termos práticos, é possível construir poesia textual por meios computacionais. O *Processing* é interpretador da linguagem Java, gratuito, orientado a objetos e distribuído na filosofia do *software* livre, que oferece recursos e facilitadores para o contexto de artes visuais. Foi desenvolvido no MIT Media Laboratory, por um grupo de cientistas representados por Ben Fry, Casey Reas. O *Processing* oferece funções para manipulação de textos, imagens vetoriais e *bitmaps*, animação, transformações escalares e espaciais, configuração de cores, processamento de matrizes, primitivas geométricas e inúmeras funções pré-definidas que facilitam o desenvolvimento de objetos visuais. Também é possível gerar arquivos prontos para serem inseridos no ambiente *web*, no formato de *applets*. Descreveremos o ambiente de programação, algumas de suas funcionalidades mais importantes e como tem sido utilizado para a produção de poesia digital.

Depois de instalado, basta ativar o programa e uma tela de programação se abre, como a que aparece na Figura 1. Os programas em *Processing* recebem a denominação de *sketch*. Um *sketch* possui uma estrutura básica, que pode ser modificada, mas que é utilizada na maioria dos projetos de artes visuais. O Quadro 3 exibe esta estrutura.

### QUADRO 3

Estrutura básica de um programa em *Processing*.

```
void setup() {  
  }  
void draw() {  
  }
```

A instrução *setup()* tem a função de efetuar configurações iniciais, que serão efetuadas sempre que o programa for executado. A instrução *draw()* executa continuamente o código escrito abaixo dela, condição necessária para a programação de animações e eventos interativos.<sup>21</sup> O código no Quadro 4 a seguir implementa este procedimento.

### QUADRO 4

Código *Processing* para repetição de um evento

```
1 void setup() {  
2   size(200, 200);  
3 }  
4 void draw() {  
5   stroke(20);  
6   if(mousePressed) {  
7     line(mouseX, mouseY, pmouseX, pmouseY);  
8   }  
9 }
```

A instrução *size(200,200)*, inserida dentro da função *setup()*, configura o quadro de exibição da animação em 200 *pixels* verticais e 200 horizontais. As instruções dentro de *draw()* serão repetidas indefinidamente, até que a execução do programa seja encerrada. Neste código, a função *stroke(20)* determina a cor cinza para o fundo da tela de exibição. A instrução *if(mousePressed)* detecta o *click* do *mouse* sempre que é pressionado. A instrução *line(mouseX, mouseY, pmouseX, pmouseY)* imprime uma linha na tela de exibição. Como ela está dentro da instrução *if*, uma linha contínua será desenhada, enquanto o *mouse* estiver sendo pressionado, como visto na FIGURA 2.

21. REAS; FRY. *Processing: a Programming Handbook for Visual Designers and Artists*.



FIGURA 1 - Resultado da execução do código descrito no Quadro 4.

O *Processing* oferece um conjunto de funções para a manipulação de textos, com diferentes soluções de exibição e até mesmo de animação. A possibilidade da criação de formatações textuais mais complexas representa um importante avanço na implementação de artefatos de arte digital. As funções disponíveis permitem imprimir textos e, conjuntamente, efetuar operações e efeitos sobre as palavras. Para a exemplificação de algumas possibilidades do uso das funcionalidades do ambiente *Processing*, foi criado um pequeno evento poético visual, que utilizou um considerável conjunto de funções e alguns efeitos visuais. O trabalho foi chamado de “Tente ler o poema”, e o conceito que orientou sua produção foi causar ruídos e adicioná-los em um poema textual, para assim dificultar sua leitura. O poema adotado é de autoria de Vera Casa Nova (2006), intitulado “Numeração falida”. O poema foi convertido para o formato digital, e um código Java foi escrito no *Processing*, que efetuou diversas operações no texto. O questionamento de fundo desta obra é: o leitor estará disposto a superar dificuldades de leitura e visualização para ler e entender um poema, ou ele espera que um texto informatizado seja sempre um objeto imaginado como um facilitador, e não um construtor de obstáculos visuais. Estas considerações nos aproxima das obras de arte conceitual, e até mesmo da arte abstrata, onde as referências reais do espectador são subvertidas, obrigando-o a perceber a obra a partir de outros pontos de vista, que ele pode, inclusive, ignorar. Vejamos o código a seguir e, logo depois, sua análise e resultado.

QUADRO 5

Código para o poema digital “Tente ler o poema”.

```

1 String[] palavras;
2 Void setup() {
3 size(720,536);
4 PImage img = loadImage("fundo4.jpg");
5 Background(img);
6 palavras = new String[10];
7 palavras[0] = "NUMERAÇÃO FALIDA";
8 palavras[1] = "2 ajoelhado o número se percebe infinito";
9 palavras[2] = "3 cortado ao meio se desintegra no fio";
10 palavras[3] = "4 linha cruzada do telefone ocupado";
11 palavras[4] = "5 agarrado ao céu é como pássaro voando";
12 palavras[5] = "6 redondo na espera da volta";
13 palavras[6] = "7 sublime corte no ar";
14 palavras[7] = "8 infinito nascido para além";
15 palavras[8] = "9 redondo na volta do parafuso";
16 palavras[9] = "10 de um e de zero na cota da vida";
17 }
18 void draw() {
19 stroke(20);
20 if(mousePressed) {
21 PImage img = loadImage("fundo4.jpg");
22 background(img);
23 line(mouseX, mouseY, pmouseX, pmouseY);
24 for (int i = 1; i < 10; i = i+1) {
25 PFont myFont;
26 myFont = createFont("FFScala", random(15,20));
27 textFont(myFont);
28 fill(random(255), random(255),random(255));
29 float rf = random(10);
30 int ri = int(rf);
31 rotate(0.01);
32 float mx = random(50);
33 float my = random(200);
34 text(palavras[ri], mouseX+mx, mouseY+my);
35 stroke(100);
36 float radius = random(1);
37 float posX = mouseX+mx;
38 float posY = mouseY+my;
39 float pos2x = posX;
40 float pos2y = posY;
41 for(int deg =0;deg<360*5; deg+=12){
42 float angle = radians(deg);
43 float x = posX+(cos(angle)*radius);
44 float y = posY+(sin(angle)*radius);
45 line(pos2x,pos2y,x,y);
46 radius = radius*1.05;
47 pos2x=x;
48 pos2y=y;
49 }
50 }
51 }
52 }
53

```

A análise dos pontos mais importantes do código demonstrará como o *Processing* manipula objetos visuais. O programa se inicia pela declaração de um vetor de caracteres, *String[] palavras*, que permitirá o armazenamento de todo o poema, linha a linha. Nas linhas 2 a 5, declaramos o tamanho da janela de exibição que o *Processing* irá gerar, 720 linhas horizontais por 536 verticais. Na linha 4, efetuamos a chamada de uma imagem, “fundo4.jpg”, que será utilizada como imagem de fundo para a tela. A instrução *background()* posiciona a imagem na tela. Entre as linhas 7 e 16 inserimos o poema de Vera Casa Nova, sem nenhuma alteração em relação ao original. O texto é composto de 10 linhas, e cada linha é alocada em um índice do vetor *palavras[]*, procedimento que permitirá a manipulação individual de cada linha. O trecho de execução do programa começa na linha 18, na instrução *draw()*. A instrução *if(mousePressed)* estabelece o que, a partir deste ponto, o programa ativará suas funções a cada vez que o mouse for pressionado. Esta operação é que estabelece o ponto de interação entre o observador e o programa, aspecto fundamental no contexto da arte digital. Este procedimento também aponta para a dimensão performática da obra, considerando que a existência do evento, a partir deste momento, dependerá inteiramente da ação do usuário. Nas linhas 23 a 34 encontram-se as instruções que manipulam os versos do poema. A linha 26, *myFont = createFont(“FFScala”, random(15,20))*, seleciona o tipo de fonte que será exibida, *FFScala*, e configura o tamanho da fonte, por meio da função *random*. Desse modo, as linhas do poema serão exibidas em diferentes tamanhos, criando assim uma dinâmica no aspecto visual do poema. A linha 31, *rotate(0.01)*, executa uma inclinação na linha do texto, e as linhas 32 e 33, *float mx = random(50)* e *float my = random(200)* selecionam aleatoriamente a altura em que uma linha de texto aparecerá em relação à outra. Nas linhas 35 a 48 encontra-se o código que executa a geração de linhas em espiral, incluindo assim um ruído gráfico no poema.<sup>22</sup>

O resultado da execução do código pode ser visto na FIGURA 2.

22. O código localizado entre as linhas 35 e 48 foi baseado no exemplo descrito por Reas e Fry, em seu livro *Processing: A Programming Handbook for Visual Designers and Artists*, p. 125.



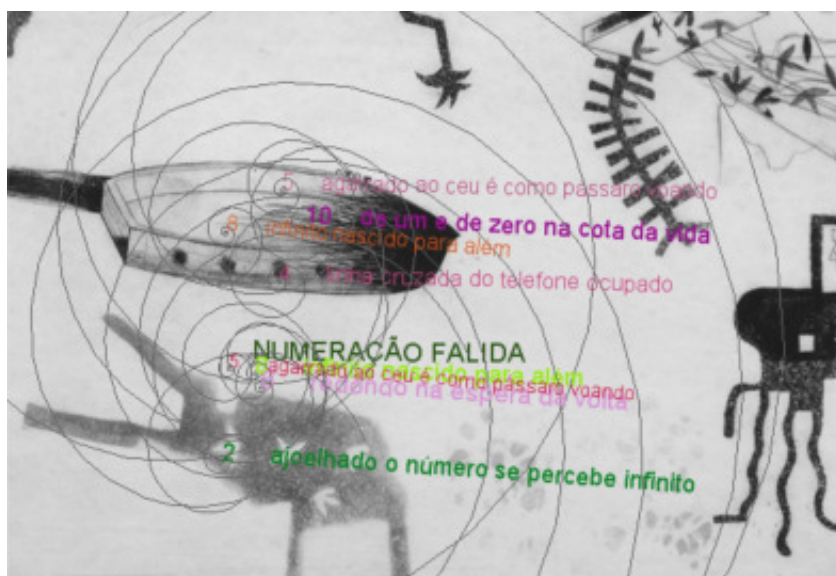


FIGURA 2 - Uma das imagens da obra “Tente ler o poema”, implementado no ambiente *Processing*.

A figura apresenta um dos momentos da execução do programa “Tente ler o poema”. À medida que o usuário pressiona e move o mouse, o poema é reescrito na tela, mas em configuração diferente. Deste modo, o leitor é provocado a tentar ler o poema, mas a ordem e as interferências visuais ao redor do texto dificultam a sua compreensão. O observador deverá tentar várias interações, até conseguir ler todas as partes do texto. A única indicação da ordem em que o poema deve ser lido é a numeração das linhas, que aparecem às vezes sobrepostas ou ilegíveis e sofrem interferências adicionais da imagem de fundo e das linhas espirais que surgem conectadas a cada linha. A proposta desta interação é justamente solicitar ao leitor que tente fazer um esforço de leitura para que ocorra entendimento do poema. Esta é uma exigência que, provavelmente, só tem lugar na arte, pois que a tônica da contemporaneidade parece ser justamente o constante e saturador provimento de facilidades e confortos aos indivíduos, sensações que são, no entanto, inteiramente ilusórias.

Para onde caminha a poesia digital? Vimos que, ainda hoje, os encontros entre a arte e as tecnologias digitais são permeados por dificuldades e paradoxos. Nos ambientes virtuais, as possibilidades visuais e participativas das interfaces são limitadas unicamente pelos códigos computacionais que as suportam,

determinando, por exemplo, o grau de incerteza no comportamento e os parâmetros funcionais e tateis oferecidos. Os autômatos, nossos conhecidos robôs, também se integram a essa discussão, por sua condição de máquinas comportamentais, nas quais podemos injetar programas, e cujos sensores podem capturar estímulos ambientais e, a partir de inferências matemáticas, criar acontecimentos poéticos permeados de movimento. Certamente, a arte eletrônica continuará a integrar em seu repertório as descobertas derivadas de outras ciências, como os algoritmos evolucionários e as biotecnologias.

### Poetry Expanded by Digital Interaction

*Abstract: This essay proposes a reflection about digital poetry, in the interaction context between the text and computer algorithms to construct a poetic sense. The main course explored in this article will be the digital poetry performance, known as a write expression exposed by the execution of algorithms created in any computer programming language, its existence is a temporary event, and only noticed during the program accomplishment which defines it. The essay point towards expanded poetry, in addition to allow the readers interference, suggests a new glance over contemporary poetry. At the end of this article we present the implementation analysis of a small digital-poetic essay developed by Processing environment, a dialect derived from Java language programming.*  
*Keywords: Digital Poetry, expanded poetry, Eletronic art.*

### Referências

- ALONSO, M.; FINN, E. J. *Física*. Harlow: Addison Wesley, 1999.
- ANTONIO, Jorge Luiz. Poesia visual e eletrônica no Brasil: alguns exemplos. II Colóquio Internacional "A crise da Poesia no Brasil, na França, na Europa e em outras latitudes" – Universidade Estadual de Feira de Santana (Brasil)/Université d'Artois (França) – Feira de Santana, BA, 3 a 5 set. 2003.
- BENJAMIN, Walter. A obra de arte na era de sua reprodutibilidade técnica. In: BENJAMIN, W. *Obras escolhidas - Magia e técnica, arte e política*. 7. ed. São Paulo: Brasiliense, 1994. p. 165-196.
- BENSE, Max. *Pequena estética*. Tradução de J. Guinsburg e Ingrid Dormien. Haroldo de Campos (Org.). São Paulo: Perspectiva, 1975.

- CAMPOS, Augusto; CAMPOS, Haroldo; PIGNATARI, Décio. *Teoria da poesia concreta: textos críticos e manifestos 1950-1960*. São Paulo: Brasiliense, 2006.
- CAMPOS, Haroldo de. *A arte no horizonte do provável*. São Paulo: Perspectiva, 1977.
- CASA NOVA, Vera. Introdução à semiótica. In: \_\_\_\_\_; PINTO, Julio. *Algumas semióticas*. Belo Horizonte: Autêntica, 2009.
- CASA NOVA, Vera. *Rastros*. Rio de Janeiro: 7Letras, 2006.
- COHEN, Renato. *Performance como linguagem*. São Paulo: Edusp, 1998.
- DONDIS, Donis A. *Sintaxe da linguagem visual*. São Paulo: Martins Fontes, 2002.
- GALIMBERT, Umberto. *Psiche e Techne: o homem na Idade da Técnica*. São Paulo: Paulus, 2006.
- GOLDBERG, Roselee. *A arte da performance: do futurismo ao presente*. São Paulo: Martins Fontes, 2006.
- GRAU, Oliver. Lembram a fantasmagoria! Política da ilusão do século XVIII e sua vida após a morte multimídia. In: DOMINGUES, Diana (Org.). *Arte, ciência e tecnologia: passado, presente e desafios*. Tradução de Flavia Gisele Saretta et al. São Paulo: UNESP, 2009.
- LUTZ, Theo. *Nach Franz Kafka*. Stuttgart, Alemanha, 1959. Disponível em: <<http://www.reinhard-doehl.de/poetscorner/lutz2.htm>>. Acesso em: 7 fev. 2011.
- MEDEIROS, Givaldo. Dialética concretista: o percurso artístico de Waldemar Cordeiro. *Revista do Instituto de Estudos Brasileiros*, São Paulo, n. 45, 2007.
- MELO E CASTRO, E. M. Para uma outra literacia. In: ANTONIO, Jorge Luiz. *Poesia eletrônica: negociações com os processos digitais*. Belo Horizonte: Veredas & Cenários, 2008.
- NÖTH, Winfried. Máquinas semióticas. *Galáxia: Revista do Programa de Pós-Graduação em Comunicação e Semiótica*, São Paulo, v. 1, n. 1, p. 51-73, 2001.
- POPPER, Frank. *Art of the Electronic Age*. New York: Thames and Hudson, 1993.
- REAS, Casey; FRY, Ben. *Processing: a Programming Handbook for Visual Designers and Artists*. MA: MIT Press, 2007.
- RUSH, Michael. *Novas mídias na arte contemporânea*. Tradução de Cassia Maria Nasser. São Paulo: Martins Fontes, 2006.
- VOUILLAMOZ, Núria. *Literatura e hipermedia: la irrupción de la literatura interactiva: precedentes y crítica*. Barcelona; Buenos Aires; México: Paidós, 2000.