

CAPÍTULO 04

Introdução ao Processing

1. O que é Processing?

Processing é uma linguagem de programação desenvolvida para ambiente compartilhado e participativo on-line. Desde 2001, ele vem permitindo desenvolver programas para as artes visuais. Inicialmente foi criado para permitir desenvolver esboço de software e para ensinar os fundamentos básicos de programação num contexto visual. O processamento evoluiu para uma ferramenta de desenvolvimento para profissionais. Hoje, existem muitos estudantes, artistas, designers, pesquisadores e amadores que utilizam o Processing para aprendizagem, realização de protótipos, e produção. Ele é um software livre que pode ser obtido na Internet: é um *open source*. Permite desenvolver programas interativos. Ele foi desenvolvido para ser executado em ambiente Linux, Mac OS X e Windows. Possui mais de 100 bibliotecas para atender ao software central.

O **Processing** relaciona conceitos de programação para princípios de forma visual, movimento e interação. Ele integra uma linguagem de programação, com um ambiente de desenvolvimento e metodologia de ensino em um sistema unificado. O **Processing** foi criado para ensinar fundamentos da programação de computadores dentro de um contexto visual, para servir como um software de desenho, e para ser usado como uma ferramenta de produção para contextos específicos. Os estudantes, artistas, profissionais de design e pesquisadores usam para a aprendizagem, prototipagem e produção.

O **Processing** é uma linguagem de programação que permite, entre outras coisas, gerar e modificar imagens. O **Processing** permite um equilíbrio entre processamento simples e recursos avançados. Iniciantes podem escrever seus próprios programas depois de apenas alguns minutos de instrução, mas os usuários mais avançados podem escrever a partir de bibliotecas, com funções adicionais. Ele permite trabalhar com computação gráfica, técnicas de interação com desenho vetorial, e bitmap (raster), processar imagens, modelos de cores,

utilizar mouse e teclado, eventos, comunicação de rede e programação orientada a objetos. Com as bibliotecas podemos ampliar a capacidade de processamento para gerar som, enviar e receber dados em diversos formatos e, por fim, importar e exportar arquivos 2D e arquivo 3D.

2. A Arte Algorítmica e o Processing

Acompanhando as mudanças das tecnologias contemporâneas, a arte, parte inerente da cultura, constantemente se apresenta em novos formatos e suportes. Isso não poderia ser diferente com o surgimento das tecnologias da computação. Desde o início da computação nos anos 1950 e 1960, diversos artistas vêm realizando experimentos com os computadores. Utilizando algoritmos e transferindo parte da autoria para o computador, os artistas computacionais experimentam novas possibilidades a cada inovação tecnológica. No Brasil, Waldemar Cordeiro e Abraham Palatinik são grandes nomes pioneiros da arte computacional.

No princípio, computadores eram máquinas inacessíveis às pessoas que não eram engenheiros e a programação era uma linguagem extremamente difícil. No entanto, desde a década de 1960 diversos pesquisadores têm procurado criar interfaces, ferramentas e linguagens acessíveis além dos limites da engenharia. Alguns exemplos importantes são *Logo*, *Dynabook*, *Max*, *Arduino* e *Scratch*.

A linguagem *Logo*, foi criada em 1967 e tinha o objetivo de ensinar matemática para as crianças. Alan Kay, na década de 1970, idealizou o *Dynabook*: um computador pessoal com uma interface gráfica que poderia ser facilmente utilizado por pessoas de todas as idades para a produção multimídia. Na década de 80, surgiu a linguagem *Max*, muito usada para composição musical e, principalmente, para performances ao vivo. O *Arduino*, uma plataforma de prototipagem que foi projetada para a interação física entre o ambiente e o computador, foi lançado em 2005. Mais recentemente, em 2007, surgiu o *Scratch*, uma linguagem de programação direcionada para o público infantil. Todas essas iniciativas compartilham o mesmo objetivo: tornar a programação acessível para pessoas de idades e formações diversas. Nesse contexto de acessibilidade da computação está inserido o *Processing*. O projeto surgiu em 2001 no *MIT Media*

Lab (Massachusetts Institute of Technology). Dois alunos, Casey Reas e Ben Fry, em parceria com o Professor John Maeda, do grupo Estética Computação (*Aesthetics and Computation Group*), conceberam o *Processing*, uma interface de desenvolvimento e uma linguagem de programação destinadas às artes visuais. O objetivo deles era elaborar uma linguagem de programação acessível e que inspirasse o processo criativo, para ser utilizada por artistas e designers – pessoas com um pensamento visual e que criam rascunhos e experimentam durante o processo criativo.

Na linguagem de programação *Processing*, computação e arte se encontram de uma maneira diferente do que havia acontecido nas décadas anteriores, tanto em relação à acessibilidade da programação para artistas quanto nos aspectos de produção artística. O *Processing* não é uma linguagem de programação qualquer. É uma linguagem concebida especificamente para as artes visuais. Assim como diferentes materiais possibilitam diferentes resultados artísticos na escultura, na pintura ou na música, também na arte computacional diferentes linguagens de programação possibilitam diferentes resultados estéticos (REAS et al, 2010). O *Processing* apresenta funções nativas e uma estrutura de processamento interno otimizadas para a produção de imagens.

Assim, ela tornou-se uma linguagem amplamente adotada em escolas de arte de todo o mundo para o ensino de programação para artistas e designers. E mais do que isso, se tornou ainda uma linguagem de grande popularidade entre artistas para a produção final das obras. Desde o seu lançamento, o seu download já foi realizado na internet quase 2 milhões de vezes, segundo informação no site do projeto. Procurando compreender o atual processo de criação de arte computacional a partir do código, esta pesquisa propõe uma investigação sobre as relações entre arte e tecnologia na linguagem de programação *Processing*, respondendo à seguinte questão: quais são as relações entre desenvolvimento tecnológico e arte computacional na linguagem de programação *Processing*? A arte computacional não é um tema recente. Quando o computador surgiu, nas décadas de 1940-1950, alguns artistas começaram a experimentar com o novo meio, se interessando pelo uso do computador como ferramenta de criação e expressão (BARRIÈRE, 2010). Paula Perissinoto (2010), no artigo Arte e Tecnologia: uma história por vir, publicado no livro comemorativo de dez anos do popular Festival

Internacional de Linguagem Eletrônica, explora as relações entre arte e tecnologia e remete esse vínculo ao trabalho de alguns artistas da vanguarda modernista, que queriam dar vida às artes plásticas. Ela ressalta o Manifesto do Maquinismo, de 1955, em que Bruno Munari anuncia que os artistas deveriam adentrar o mundo das máquinas, e trazendo esse conceito para o mundo presente, propõe que os artistas atuais também devem adentrar o mundo do software. “Se o mundo em que vivemos é o mundo dos códigos, devemos, como artistas, produtores culturais e pensadores contemporâneos, adentrar esse universo de zero e uns e dominá-los” (PERISSIOTO, p.374, 2010).

A arte computacional não é um tema recente. Quando o computador surgiu, nas décadas de 1940 e 1950, alguns artistas começaram a experimentar com o novo meio, se interessando pelo uso do computador como ferramenta de criação e expressão (BARRIÈRE, 2010). Paula Perissinoto (2010), no artigo Arte e Tecnologia: uma história por vir, publicado no livro comemorativo de dez anos do popular Festival Internacional de Linguagem Eletrônica, explora as relações entre arte e tecnologia e remete esse vínculo ao trabalho de alguns artistas da vanguarda modernista, que queriam dar vida às artes plásticas. Ela ressalta o Manifesto do Maquinismo, de 1955, em que Bruno Munari anuncia que os artistas deveriam adentrar o mundo das máquinas, e trazendo esse conceito para o mundo presente, propõe que os artistas atuais também devem adentrar o mundo do software. “Se o mundo em que vivemos é o mundo dos códigos, devemos, como artistas, produtores culturais e pensadores contemporâneos, adentrar esse universo de zero e uns e dominá-los” (PERISSIOTO, p.374, 2010).

Nas primeiras décadas da arte computacional, muitas pessoas se perguntavam qual era o papel do artista e qual era o papel da máquina na criação. Recentemente, em abril de 2014, em um editorial da revista Leonardo, Frieder Nake, um dos pioneiros da arte computacional, escreveu sobre essa questão. O editorial, em resposta à pergunta sobre qual é o papel do artista na arte computacional, levantada na edição anterior da revista, explora a importância de se pensar o algoritmo como criação humana para então se pensar arte computacional. O autor, que prefere o termo arte algorítmica à arte computacional – apesar de arte algorítmica poder ser feita sem computadores –,

traz a questão do *software* à tona. O computador não funciona sem *software* e, portanto, para se pensar arte feita no computador é necessário se pensar também o *software*.

Arte algorítmica começa com a criação de um algoritmo. Isso é trabalho humano. O processo generativo termina em um objeto material, como tinta no papel. Isso é trabalho da máquina. O que a máquina realiza é uma possibilidade de potencialmente infinitos conjuntos de partes. O artista, entretanto, descreveu um conceito: toda a classe dessas partes. Dependendo do poder expressivo dos parâmetros contidos na descrição algorítmica, as diferenças na aparência visual das partes podem ser imensas. Não há limite para a nossa capacidade descritiva (NAKE, p.108, 2014).¹

No artigo *Arte y algoritmos* (Arte e algoritmos), Barrière (2010) também defende o argumento de que o algoritmo é parte da obra de arte e não somente um método para se chegar ao resultado final e, da mesma forma, explica que um único código pode produzir diversos resultados. A questão do algoritmo computacional, e portanto do *software*, na produção de arte computacional é central para a presente pesquisa. Como foi descrito anteriormente, a linguagem *Processing* foi criada para a produção e ensino das artes visuais. Para Reas e Fry (2004 e 2006), criadores do *Processing*, cada linguagem de programação, assim como cada tecnologia, é como um material diferente que o artista pode usar. Assim, eles elaboraram o *Processing* especificamente para gerar e modificar imagens.

A História mostra que tecnologias como tinta a óleo, câmeras e filme transformaram a prática e o discurso artísticos, e enquanto nós não estamos defendendo que novas tecnologias *melhoram* a arte, nós acreditamos que elas possibilitam diferentes formas de comunicação e expressão. Software é único dentre os materiais artísticos em sua habilidade de produzir formas dinâmicas, processar gestos, produzir comportamentos, simular sistemas naturais e integrar vários outros meios, como som, imagem e texto (REAS; FRY, p.528, 2006, grifo no original).²

¹ Traduzido pelo autor a partir de: Algorithmic art starts with the development of an algorithm. That's human work. The generative process ends in some material object, say ink on paper. That's the machine's work. What the machine realizes is one instance of a potentially infinite set of pieces. The artist, however, has described a concept: the entire class of those pieces. Depending on the expressive power of the parameters contained in the algorithmic description, the differences in visual appearance of the pieces may be immense. There is no limit to our descriptive capacities.

² Traduzido pelo autor a partir de: History shows technologies such as oil paint, cameras, and film have changed artistic practice and discourse, and while we do not claim new technologies *improve* art, we do feel they enable different forms of communication and expression. Software is unique among artistic mediums in its ability to produce dynamic form, process gestures, to produce behavior, simulate natural systems, and integrate various media including sound, image, text.

Em 2013, Lev Manovich publicou o livro *Software Takes Command* (*Software Assume o Comando*), no qual explora os aspectos culturais do *software* na produção multimídia. No livro, o autor se propõe a responder à seguinte questão: o que é mídia depois do *software*? Ele explora os conceitos dos principais pensadores do início da computação e os relaciona com ferramentas de produção multimídia atuais. O livro, apesar de não tratar especificamente do *Processing*, é de extrema relevância para a presente pesquisa, pois investiga o papel do *software* na produção cultural multimídia. O autor, um dos fundadores da área de pesquisa denominada Estudos do *Software*, explora em profundidade o projeto *Dynabook*. Alan Kay e sua equipe na XEROX PARC, idealizadores do projeto, estavam, na década de 1970, trabalhando para criar um computador que pudesse ser uma ferramenta de criatividade: utilizado na produção de música, animação, gráficos, desenhos e poesia por qualquer pessoa. Segundo Manovich (2013), Kay e sua equipe na XEROX PARC reinventaram o computador, transformando-o em um sistema interativo para o pensamento criativo. O *Dynabook* trazia uma nova proposta de computador pessoal: um computador acessível que permitia que o usuário fosse também o programador.

Essa questão da acessibilidade é um aspecto central do *Processing*, pois ele foi projetado para ser uma linguagem de programação de fácil acesso que pudesse ser também utilizada no ensino da programação para artistas e designers. Em um artigo publicado em 2010, nos Anais do 9º Encontro Internacional de Arte e Tecnologia (#9ART), Suzete Venturelli, pesquisadora, artista e organizadora do encontro, defende que é fundamental para o novo artista computacional aprender a programar:

Aprendemos a programar pela necessidade imposta pelo próprio meio. Não se faz arte computacional sem saber programar. Um desafio para o artista atual? Sim, um desafio, pois as universidades ainda não estão preparadas para o ensino da arte computacional, área transdisciplinar, em gestação (VENTURELLI, p. 423, 2010).

Em relação ao desafio do ensino de programação para alunos das Artes, Madalena Ribeiro da Universidade Castelo Branco em Portugal, explora, no artigo *As Linguagens de Programação para Artes - Metodologias de Ensino-Aprendizagem Adaptadas*, os aspectos da adoção do *Processing* como ferramenta de ensino de programação. Ela destaca as diferenças entre alunos das áreas de engenharias e afins e alunos das áreas de artes e explica as dificuldades de se ensinar programação para artistas:

A aprendizagem de linguagens de programação, essencialmente ao nível inicial, reveste-se de um conjunto de dificuldades, especialmente pela natureza da própria temática em si e pelo facto de implicar no aluno a habilidade de conciliar, simultaneamente, a sua capacidade de abstração, o domínio da sintaxe e o talento para criar um algoritmo (RIBEIRO, 2013).

A partir dessas dificuldades, a autora percorre metodologias de ensino utilizando *Processing*, as quais podem resultar em “um processo criativo e desafiante, promovendo a motivação e a satisfação dos alunos e o conseqüente sucesso da aprendizagem” (RIBEIRO, 2013). A autora explica, ainda, que a possibilidade de criar efeitos visuais interessantes de forma quase imediata e simples é o principal atrativo da linguagem *Processing*, pois desafia os sentidos dos alunos e os estimula a querer continuar aprendendo.

No livro *Getting started with Arduino* (Começando com o *Arduino*), Massimo Banzi, criador da plataforma de prototipagem para interação física com o computador – e que teve como inspiração direta o *Processing* –, também aborda a questão do ensino da programação para artistas e designers, propondo uma metodologia baseada na experimentação. “Nós acreditamos que é essencial brincar com a tecnologia, explorando diferentes possibilidades diretamente no hardware e no software – às vezes sem um objetivo definido” (BANZI, p.7, 2009)³.

Esses novos artistas, que aprendem e utilizam a programação eles próprios, são chamados de artistas-programadores nas pesquisas de mestrado de Antonio Francisco Moreira Neto e Alexandra Cristina Moreira Caetano, ambos da Universidade de Brasília. Neto (2010) define o artista-programador como aquele que trabalha diretamente com o *software*, investigando os processos de criação com foco na poética artística da construção e reconstrução de algoritmos. Esse artista-programador insere a sua poética no código-fonte do programa que gera a obra de arte. Neto define a arte produzida a partir do código também como *software arte*.

O artista programador é aquele que ao explorar a software arte investiga os processos de criação com foco principalmente na poética artística da reconstrução dos algoritmos. Permite-se inserir no código-fonte a sua poética, ou sobre a execução do código e as imagens e processos artísticos automaticamente gerados, ou ainda sobre o uso das linguagens de programação e a estética visual gerada na construção das interfaces

³ Traduzido pelo autor a partir de: We believe that it is essential to play with technology, exploring different possibilities directly on hardware and software—sometimes without a very defined goal

computacionais, mas em qualquer abordagem o código e a construção algorítmica é [sic] parte da poética e do processo criativo da obra (NETO, p.16, 2010).

Sob a perspectiva da interatividade, Caetano (2010) igualmente explora o conceito do artista-programador em sua pesquisa de mestrado na UNB. Segunda a pesquisadora, o “artista-programador tanto manipula códigos já existentes em busca de novas configurações, como constrói seus próprios algoritmos / programas, criando os parâmetros que julga interessante manipular” (CAETANO, p.56, 2010). Também de extrema relevância para esta pesquisa é o artigo *Scratch: Programming for All* (*Scratch: programação para todos*), de 2009. *Scratch* é uma linguagem de programação inspirada no jogo Lego e voltada para o público infantil. Apesar de terem públicos e possibilidades de produção diferentes, a linguagem *Scratch* apresenta características muito semelhantes ao *Processing*, principalmente em relação à comunidade online, que troca experiências e aprende em conjunto:

Para o Scratch ser bem-sucedido, a linguagem precisa estar relacionada a uma comunidade onde as pessoas podem se ajudar, colaborar, criticar e construir em cima do trabalho uma das outras [...] O site é também um terreno fértil para a colaboração. Membros da comunidade estão constantemente pegando emprestado, adaptando e construindo com base nas ideias, imagens e programas de outras pessoas. Mais de 15% dos projetos no site são remixes de outros projetos (RESNICK et al, p.65, 2009).⁵

Segundo Takhteyev (2013), o mundo do *software* é dominado pelo efeito da rede: quanto mais gente usa um *software*, mais valioso ele se torna. Isso acontece porque mais pessoas produzem referências, tutoriais, módulos de código e projetos na linguagem, ampliando a documentação e a possibilidade de aprendizado. Além disso, no caso do *software* livre (como são o *Processing*, o *Scratch* e o *Arduino*), mais desenvolvedores se interessam por participar da comunidade que mantém e estende o *software*, consertando *bugs* e desenvolvendo novas bibliotecas. Assim, se torna essencial comparar o *Processing* a outras linguagens open source que igualmente apresentam fortes comunidades online.

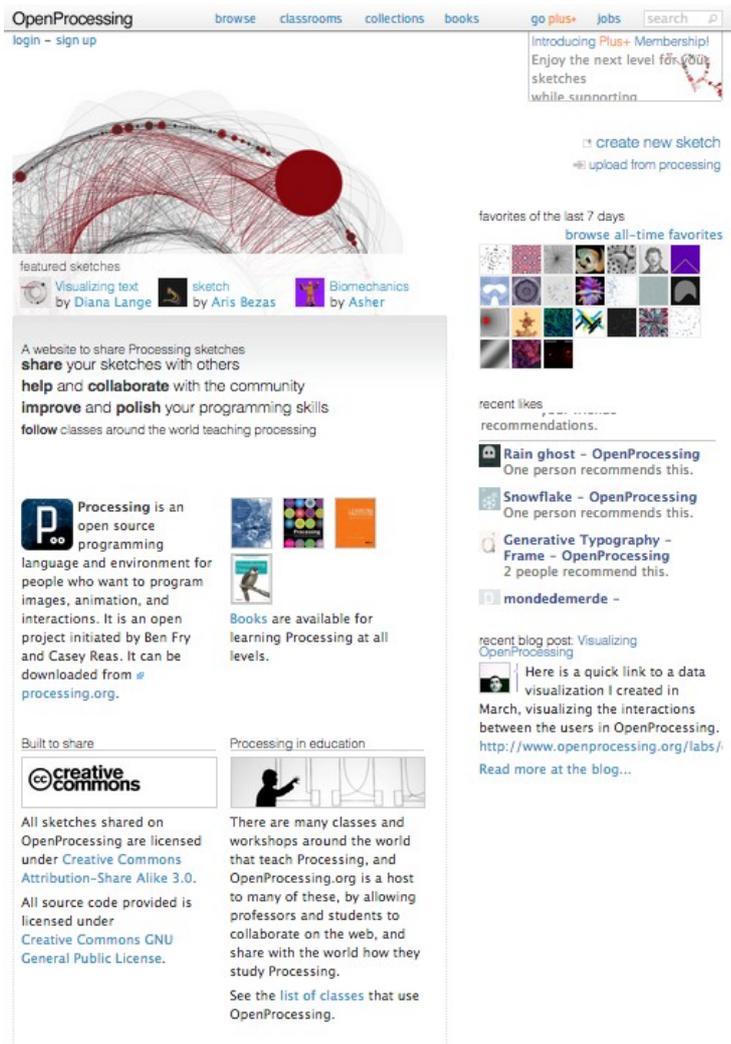
3. Ferramentas e Interfaces do Processing

O site *OpenProcessing* foi criado em 2008 por Sinan Ascioğlu, na época estudante do curso *Interactive Telecommunications Program* da *Tisch School of the Arts*, na Universidade de Nova York. Ascioğlu conta em entrevista concedida para esta investigação que, na época, estava fazendo sua pesquisa de mestrado: um projeto experimental para a criação de uma comunidade online de compartilhamento de visualizações de dados criadas em *Processing*. Uma das dificuldades técnicas para a criação do website da comunidade era integrar os *sketches* em *Processing* numa página web. Naquela época ainda não havia a biblioteca de *Processing* em JavaScript (*processing.js*) que existe hoje e que tornou esse tipo de publicação na web extremamente simples. Dessa maneira, ao conseguir resolver esse problema, Ascioğlu conta que o Professor Daniel Shiffman, seu orientador e uma das lideranças do *Processing*, sugeriu ampliar a comunidade para qualquer sketch, criando, dessa forma, algo como um *Flickr* do *Processing*. Ascioğlu seguiu a sugestão do professor e, assim, surgiu o *OpenProcessing*.

Ascioğlu explica que, na época (2008), as pessoas compartilhavam seus *sketches* em seus blogs pessoais e faziam perguntas no fórum oficial do *Processing* (forum.processing.org). A primeira transformação que ele notou com o surgimento da comunidade é que as pessoas passaram a publicar os *sketches* no *OpenProcessing* e apontar para lá quando faziam perguntas no fórum.

Quando criou o site, Ascioğlu queria manter a interação entre os usuários simples e minimizar qualquer problema com privacidade e segurança de dados. Não sendo um programador de *backend* e por estar trabalhando sozinho, Ascioğlu quis simplificar ao máximo. Dessa maneira, optou por manter tudo público. As únicas informações privadas no site inicialmente eram e-mail e senha. Fora isso tudo era aberto. Ele conta que essa abertura atraiu mais gente e, assim, a comunidade se formou.

Figura XX- OpenProcessing



Fonte: www.openprocessing.org

Essa abertura dos dados, portanto, foi um fator crucial para a formação da comunidade. O *Processing* é um *software* livre que já contava com uma comunidade de dados abertos e colaboração livre naquela época. Dessa maneira, o *OpenProcessing* foi criado em alinhamento com a cultura do *Processing*, mas, além disso, também foi criado em sintonia com a cultura inicial da internet, da tecnomeritocracia e da cultura *hacker*:

A cultura hacker é, em essência, uma cultura de convergência entre seres humanos e suas máquinas num processo de interação liberta. É uma cultura de criatividade intelectual fundada na liberdade, na cooperação, na reciprocidade e na informalidade (CASTELLS, p.45, 2003).

Castells (2003) faz uma distinção importante entre os termos *hacker* e *cracker*. Apesar de atualmente se usar "*hacker*" de uma maneira negativa, ele explica que a cultura *hacker* é uma cultura de colaboração online baseada na liberdade, enquanto que "*crackers*" seriam aqueles que utilizam suas habilidades técnicas para cometer crimes, como roubo de números de cartão de crédito, acesso indevido a sites de terceiros etc. Dessa maneira, incentivando a colaboração e a abertura do código-fonte, o *OpenProcessing* surgiu e se tornou um importante destino para o compartilhamento e para o aprendizado da linguagem *Processing*. O surgimento de uma comunidade como o *OpenProcessing* não é uma surpresa na história do *Processing*, no entanto. Atentos para o potencial das comunidades online na internet, os criadores do *Processing* mantiveram a formação de uma comunidade online como meta desde a concepção do projeto:

Processing foi projetado para utilizar-se da força das comunidades online, o que permitiu que o projeto crescesse de maneiras inesperadas. Milhares de alunos, educadores e praticantes nos cinco continentes estão usando o software (REAS; FRY, p.1, 2004).⁴

Wenger (2000) explica que em comunidades de prática – como o *OpenProcessing* –, o aprendizado é definido em uma dinâmica entre a competência do grupo e a experiência pessoal. Parte do conhecimento que uma pessoa “tem” é, na verdade, conhecimento gerado pela comunidade no decorrer dos anos. Dessa maneira, o aprendizado combina transformação pessoal com a evolução de estruturas sociais. As comunidades de prática apresentam três características (WENGER, 2000):

- Membros unem-se por meio de um entendimento coletivo sobre o que é a comunidade;
- A comunidade é construída por engajamento mútuo; membros interagem uns com os outros estabelecendo normas e relacionamentos;
- E a comunidade produz um repertório compartilhado de recursos.

⁴ Traduzido pelo autor a partir de: Processing is built to take advantage of the strengths of web-based communities, which has allowed the project to grow in unexpected ways. Thousands of students, educators, and practitioners across five continents are involved in using the software.

Para que um membro tenha competência em uma comunidade, ele precisa conhecer a comunidade e saber navega-la, interagir com outros membros se tornando um contribuidor confiável e, por último, ter acesso ao repertório compartilhado e saber usá-lo. O *OpenProcessing* é uma comunidade de nicho, na qual as pessoas colaboram e interagem criando aplicações em *Processing*. A comunidade é formada por artistas-programadores em *Processing* (iniciantes e experientes) que querem compartilhar suas aplicações e o código fonte, contribuindo para o desenvolvimento de novos programadores e para a criação de novos *sketches*. A abertura da informação, com a postagem pública dos *sketches* e dos respectivos códigos-fonte de *Processing* é um dos principais pilares culturais da comunidade, formando a base do entendimento do que ela é e também o repertório compartilhado.

A distribuição aberta dos códigos-fonte permite a qualquer pessoa modificar o código e desenvolver novos programas e aplicações, numa espiral ascendente de inovação tecnológica, baseada na cooperação e na livre circulação de conhecimento técnico (CASTELLS, p.35, 2003).

O *OpenProcessing* tem atualmente 38.550 usuários cadastrados, dentre eles 4,795 alunos, e 89.556 *sketches* criados. O número de visualizações de páginas foi 3.292.955, com uma média de sessão, isto é, de cada visita de cada usuário, de 5 minutos e 32 segundos. Se comparado com o mesmo período ano passado, 2013, há um aumento de 11,49% no número de páginas visitadas por sessão e um aumento de 17,22% no tempo médio de cada sessão. Ainda de acordo com o *Google Analytics*, a maior parte dos usuários do *OpenProcessing* estão nos Estados Unidos. O Brasil está em nono lugar:

Tabela - Países mais populares no *OpenProcessing*

1º	Estados Unidos	6º	Japão
2º	Inglaterra	7º	Espanha
3º	Alemanha	8º	Itáli
4º	França	9º	Bras
5º	Canadá	10º	Austrália

Fonte: *Google Analytics*

Contudo, dentre todos os dez primeiros países, o Brasil foi o que apresentou a maior taxa de crescimento no número de sessões em relação ao mesmo período ano passado (01 de janeiro a 25 de maio): 29,47% de crescimento. Depois do Brasil, vem a Espanha, com um crescimento de 10,20% no número de sessões. Esses números indicam um crescimento no uso do *OpenProcessing* e, portanto, também do *Processing* no Brasil.

A comunidade apresenta uma estrutura simples, com funcionalidades comuns em outras redes sociais e comunidades online. Cada usuário tem uma página de perfil, chamada de portfólio. As interações entre os membros da comunidade acontecem também de maneira simples. Essa cultura de compartilhamento dos *sketches* inspira a colaboração e o aprendizado em conjunto. Quando alguém em uma comunidade compartilha o código fonte de suas aplicações, várias pessoas podem aprender. Isso pode acontecer pela observação e pelo entendimento de como aquela aplicação foi construída ou por meio da extensão daquele programa. Dessa maneira, a comunidade toda cresce (REAS; FRY, 2006).

Ainda na página do *sketch*, são apresentadas as licenças de publicação dos *sketches*. No canto inferior direito é possível observar duas licenças: *Creative Commons*³⁴ e GNU8GPL. A licença *Creative Commons* usada pelo *OpenProcessing* (CC BY8 SA 3.0) permite a ampla distribuição e reutilização do *sketch* por meio de *remix*, transformação ou extensão para qualquer fim, inclusive comercial. As exigências são a atribuição de autoria e a continuação da licença, não sendo possível, portanto, estender uma aplicação do *OpenProcessing* e publicá-la com uma licença de direitos autorais totalmente fechada, por exemplo. Já a licença GNU8 GPL cobre o *software* em si, o código. A licença protege o direito ao acesso livre ao código e a sua reutilização.

Essa licença utilizada no *OpenProcessing*, a CC8GNU GPL, surgiu no Brasil em 2003. Mendes (2006) explica que devido ao apoio do governo brasileiro ao *software* livre, foi feito um acordo entre a *Free Software Foundation*, o Instituto Nacional de Tecnologia da Informação, por intermédio da Escola de Direito da Fundação Getúlio Vargas, no Rio de Janeiro, e da organização americana *Creative Commons*.