

14º Congresso Brasileiro de Pesquisa e Desenvolvimento em Design: Ensino de programação criativa aplicado à prática do estudante de design

14th Brazilian Congress on Design Research: The teaching of creative coding applied to the design student's practice

DURÃO, Guilherme Marandino; Bacharelado em Design em andamento, Bolsista de Iniciação Científica; Pontifícia Universidade Católica do Rio de Janeiro

guilhermemdurao@gmail.com

NOVAES, Luiza; Doutora; Pontifícia Universidade Católica do Rio de Janeiro

Inovaes@puc-rio.br

BONELLI, João; Doutor; Pontifícia Universidade Católica do Rio de Janeiro

joao-bonelli@puc-rio.br

Esta pesquisa explora métodos para o ensino de programação criativa para estudantes de design, com foco em aplicações capazes de gerar respostas multimidiáticas para questões e necessidades de projetos de design na contemporaneidade. A programação criativa é uma maneira de abordar a computação que possibilita uma nova forma de expressão em projetos de design no contexto digital, promovendo a criação de peças que dialogam com um mundo tecnológico e interativo, atualizando e potencializando as possibilidades do Design. Experimentos de ensino de programação criativa foram elaborados tendo como referência a metodologia da Prática Reflexiva, proposta por Donald Schön (1983), que pressupõe um processo de conhecer na ação e envolve dinâmicas de reflexão-na-ação e reflexão sobre a ação, em um processo de retroalimentação de ideias. No desenvolvimento da pesquisa foram observadas práticas existentes de ensino de programação criativa em cursos livres e posteriormente foi elaborado e realizado um workshop de ensino em turmas iniciais de cursos de graduação em design.

Palavras-chave: Programação Criativa; Educação; Tecnologia.

This research explores methods for teaching creative programming to design students, focusing on applications capable of generating multimedia responses to questions and needs of contemporary design projects. Creative programming is a way of approaching computing that enables a new form of expression in design projects in the digital context, promoting the creation of pieces that dialogue with a technological and interactive world, updating and enhancing the possibilities of Design. Creative programming teaching experiments were developed based on the Reflective Practice methodology, proposed by Donald Schön (1983), which presupposes a process of knowing in action and involves dynamics of reflection-in-action and reflection on action, in a feedback process of ideas. In the development of the research, existing

practices of teaching creative programming in open courses were observed and later a teaching workshop was prepared and carried out in initial classes of undergraduate design courses.

Keywords: Creative Coding; Education; Technology

1 Introdução

A programação criativa é uma maneira de abordar a computação que possibilita uma nova forma de expressão em projetos de design no contexto digital, promovendo a criação de peças únicas que dialogam com um mundo mais tecnológico e interativo, atualizando e potencializando as possibilidades do Design para as novas tecnologias. O presente estudo tem como referência o trabalho de Iniciação Científica desenvolvido por Augusto dos Santos Pereira (2019-2020) desenvolvido no Laboratório de Interfaces Físicas Experimentais - LIFE¹ da PUC-Rio, como bolsista do Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação - PIBITI, do Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq, que buscou compreender as principais diferenças entre linguagens de programação de baixa complexidade, para elaborar formas de transmitir conteúdos de programação de computadores para alunos de design.

Neste artigo é relatado o desenvolvimento da pesquisa, que se iniciou pela observação e pelo fichamento de diferentes linguagens de programação, seguidos por propostas de ensino de programação criativa para designers, a fim de compreender as similaridades e diferenças entre as propostas, bem como compreender os limites e vantagens de cada linguagem pesquisada.

A pesquisa tem como objetivo elaborar reflexões e diretrizes a respeito do ensino de programação criativa para estudantes de design, com foco em aplicações reais capazes de gerar respostas multimidiáticas para questões e necessidades de projetos de design na contemporaneidade.

2 Objetivos

Refletir a respeito de programação criativa e sua relação com o design;

Propor soluções aplicáveis à realidade do estudante de design;

Reunir experimentos e exercícios a partir de programação criativa para questões e necessidades de projetos contemporâneos de design.

¹ Sobre o LIFE <http://www.life.dad.puc-rio.br/sobre.html> Acesso em 14/04/2022

3 Metodologia

O conceito de prática reflexiva, definido por Donald Schön (1983), que pressupõe um processo de conhecer na ação e envolve dinâmicas de reflexão-na-ação e reflexão sobre a ação, em um processo de retroalimentação de ideias, foi adotado nesse estudo como referência para as propostas de experimentação. Considerando a base teórica e as conclusões registradas em relatórios de pesquisa desenvolvidos pelo bolsista Augusto dos Santos Pereira, primeiramente fizemos uma reflexão sobre as diversas linguagens de programação e suas aplicações no contexto de ensino para alunos de design.

Paralelamente, foram pesquisadas diferentes metodologias de ensino de programação criativa por meio de revisão bibliográfica e observação em disciplinas e cursos de programação criativa. Assim, buscamos identificar em que situações os conceitos de programação criativa têm aplicabilidade real para os estudantes de design e dessa forma foram propostos experiências e exercícios, pensando em diferentes possibilidades de ensino.

4 Desenvolvimento: O Código

Na pesquisa, o código é tratado como um conjunto de regras e comandos utilizados na programação de computadores, matéria-prima necessária para o desenvolvimento de qualquer projeto de programação criativa. No entanto, nem toda forma de código ou programação foi desenvolvida com foco nos estudantes de design, usuários beneficiados com esta pesquisa. Na verdade, ao longo da história da computação e de suas linguagens, uma realidade distinta foi apresentada, já que a maioria dos inventos da área foi desenvolvida como ferramenta para engenheiros, físicos, matemáticos e estudantes ou profissionais de áreas correlatas.

Entretanto, é fato que ao longo da história houve momentos em que intelectuais e estudiosos perceberam e vislumbraram, para além do estabelecido, formas como a computação afetaria outras áreas, incluindo a arte e o design, como foi o caso da matemática Ada Lovelace. A matemática inglesa foi convidada por Charles Babbage, o inventor da Máquina Analítica, considerado o primeiro computador analógico, para traduzir um artigo que um jovem engenheiro italiano chamado Luigi Menabrea² havia escrito em francês. O artigo em questão é a transcrição de uma apresentação da máquina que Charles havia feito na Universidade de Turim. Mesmo com o pedido objetivo, Ada Lovelace não se limitou a traduzir a apresentação para o inglês e decidiu somar à obra algumas longas anotações suas sobre o funcionamento e as possibilidades da máquina analítica. O resultado final ampliou em três vezes o tamanho do

² MENABREA, Luigi. Notions sur la machine analytique de M. Charles Babbage. Bibliothèque universelle de Genève, nouvelle série 41 (1842) 352–76

artigo. As notas de Lovelace foram organizadas em ordem alfabética de "A" a "G" e foram publicadas, juntamente com o artigo traduzido, na edição de setembro de 1843 da série de livros Scientific Memoirs, editados e publicados por Richard Taylor.

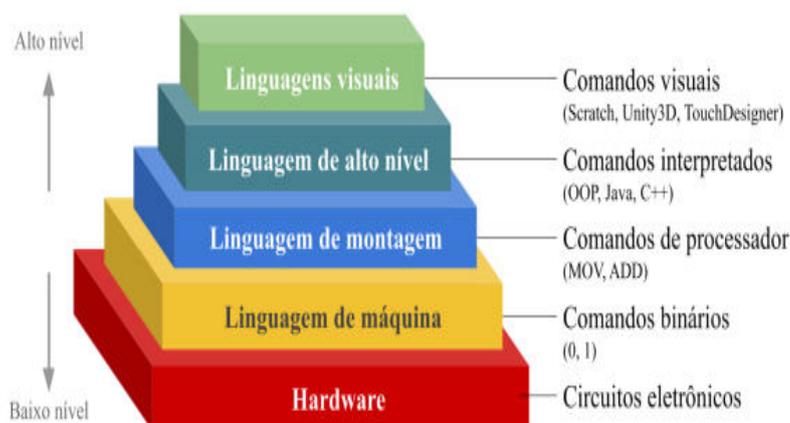
Na "nota G" Ada Lovelace descreve uma expressão, considerada o primeiro algoritmo de computador e logo depois há um trecho específico em que a matemática abre caminhos para a tese de que, no futuro, seria possível utilizar a máquina para desempenhar funções artísticas:

Supondo, por exemplo, que as relações fundamentais dos variados sons na ciência da harmonia e da composição musical sejam suscetíveis de tais expressões e adaptações, a Máquina poderia compor peças elaboradas e científicas de música de qualquer grau de complexidade ou extensão. (LOVELACE, 1843 apud OLIVEIRA, 2021).

Apesar desses pressupostos, foi necessário um longo período de desenvolvimento para que fossem criadas ferramentas adequadas para que pessoas criativas pudessem se expressar por meio de códigos de computador. Esse avanço só foi possível com o desenvolvimento de linguagens visuais e de linguagens de programação de alto nível, adaptadas para uma curva de aprendizado mais suave e com um funcionamento desenvolvido com foco em gerar resultados visuais.

O gráfico a seguir foi desenvolvido por Carlos de Oliveira (2020), também conhecido pelo nome artístico Vamoss, e mostra os diversos níveis das linguagens de programação existentes organizados a partir dos seus graus de dificuldade. Nessa lógica, as linguagens de alto nível são as mais próximas da linguagem escrita (geralmente o inglês) e as linguagens de baixo nível as mais próximas da língua de máquina (comandos binários, hexadecimais ou mecânicos).

Figura 1- Níveis de linguagens de programação.



Fonte: Oliveira, 2020.

Segundo Carlos de Oliveira (2020) a Programação Criativa é um termo utilizado para representar toda “expressão criativa através da manipulação de códigos escritos em computadores”. Esse termo, no entanto, não é estável e único, podendo se apresentar de diversas formas como por exemplo Codificação Criativa, Computação Poética, Arte Eletrônica, Arte Digital e até mesmo Aesthetic Computing, mas que de uma maneira mais ampla simbolizam a mesma coisa, o uso de tecnologia de forma criativa para gerar resultados artísticos e estéticos, sejam estes visuais ou multissensoriais.

Pode-se traçar a origem da programação criativa no campo do design como fruto da fundação do Visible Language Workshop em 1974³ pela pesquisadora e editora chefe do MIT Press, Muriel Cooper, no Massachusetts Institute of Technology (MIT), uma instituição pioneira no ensino de novas tecnologias aplicadas ao design. As aulas de Cooper influenciaram um de seus estudantes, John Maeda, que mais tarde fundou o grupo de pesquisa Aesthetics + Computation Group⁴ no MIT Media Lab e o projeto Design By Numbers - DBN⁵, uma das primeiras linguagens de programação desenvolvidas com foco em ensinar programação para artistas, designers e estudantes de currículos criativos e fornecer uma nova ferramenta para o desenvolvimento de projetos criativos em computação.

Figura 2 - Interface de desenvolvimento do Design by Numbers.



Fonte: Maeda, 1999

A partir de reflexões feitas sobre o DBN, dois doutorandos orientados por John Maeda, Ben Fry e Casey Reas, decidiram desenvolver o ambiente de programação Processing (RANOYA, 2021), um sistema com mais possibilidades, baseado na já existente linguagem de programação Java⁶, que ampliou a viabilidade da programação criativa como ferramenta de criação visual para profissionais criativos.

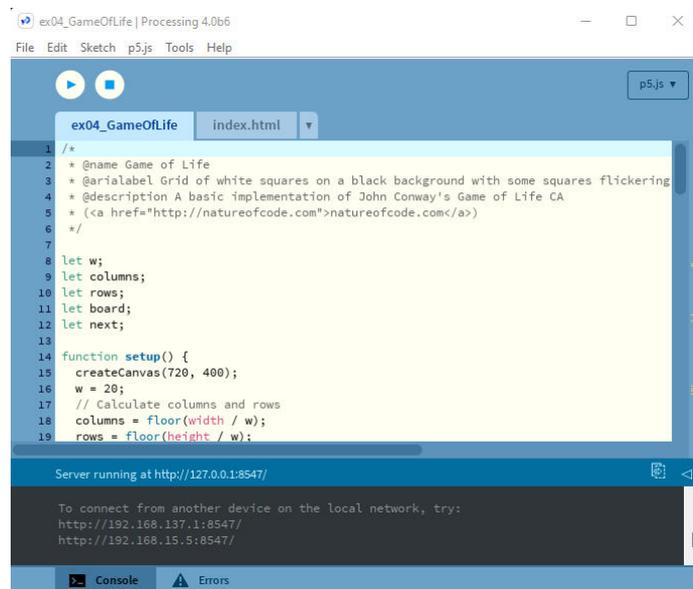
³ Visible Language Workshop: <https://act.mit.edu/special-collections/vlw-archive/> Acesso em 14/04/2022

⁴ MIT Aesthetics + Computation Group: <https://acg.media.mit.edu/> Acesso em 14/04/2022

⁵ Design by Numbers: <https://dbn.media.mit.edu/> Acesso em 14/04/2022

⁶ Java: <https://www.java.com/pt-BR/about/> Acesso em 14/04/2022

Figura 3 - Interface de desenvolvimento do Processing



Fonte: Processing Foundation, 2022

O Processing foi lançado em 2001 e desde então vem crescendo em popularidade entre artistas e designers, que muitas vezes têm seu primeiro contato com o código de computadores usando a plataforma. Sua interface amigável, linguagem simplificada e suas múltiplas possibilidades criativas garantiram para o ambiente a condição de ferramenta de ensino, sendo utilizada em disciplinas de design de diversas faculdades pelo mundo, incluindo cursos do Departamento de Artes e Design da PUC Rio.

6 A Programação Criativa e o ensino de Design no Brasil

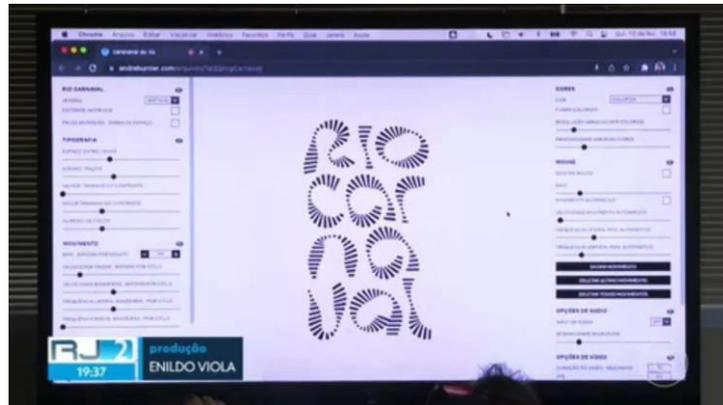
O ensino de Design no Brasil começou em 1963 com a fundação da Escola Superior de Desenho Industrial - ESDI, posteriormente vinculada à Universidade do Estado do Rio de Janeiro - UERJ, na cidade do Rio de Janeiro⁷. Desde então, 58 anos se passaram e o ensino de Design no país continua predominantemente focado nos eixos de Design Gráfico e Design de Produto, com exceções raras que se apresentam normalmente com uma abrangente denominação de Design Digital ou Design de Mídia Interativa (PINHEIRO, 2011). No contexto de ensino de Design Digital, a PUC-Rio ocupa um papel de pioneirismo no cenário nacional, sendo uma das poucas graduações com uma habilitação destinada a Mídia Interativa (BONELLI, 2016). No Departamento de Artes & Design da PUC-Rio o Laboratório de Interfaces Físicas Experimentais - LIFE⁸ é um dos responsáveis pelo ensino da programação criativa na graduação

⁷ Dado disponível em <http://www.esdi.uerj.br/a-esdi/historia> Acesso em 23/04/2021

⁸ Laboratório de Interfaces Físicas Experimentais LIFE PUC-Rio: <http://life.dad.puc-rio.br/> Acesso em 10/04/2022

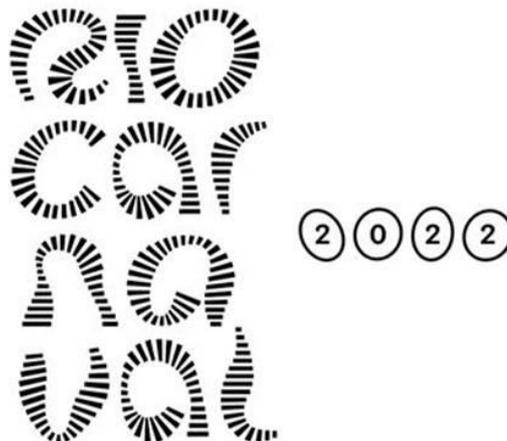
profissionais atuantes no campo da programação criativa, e responsáveis por práticas observadas e estudadas na pesquisa.

Figura 5 - Captura de tela do software desenvolvido para a identidade visual do Carnaval Rio



Fonte: RJTV, 2022

Figura 6 - Identidade visual do Carnaval Rio 2022



Fonte: Tátil Design, 2022

Inspirada no movimento das porta-bandeiras das escolas de samba, a nova marca desenvolvida a pedido da LIESA¹⁶ (Liga Independente das Escolas de Samba do Rio de Janeiro) é viva e se movimenta seguindo os diferentes parâmetros disponíveis em seu software, desenvolvido com a linguagem de programação p5.js¹⁷ por André Burnier. As várias opções de cores também disponíveis permitem que a marca se adapte à identidade visual de cada uma das escolas de samba participantes do desfile. A versatilidade do programa permite que qualquer pessoa acesse o software em um navegador de internet comum, alterne entre os diferentes modos e intensidades de movimento e salve a opção gerada, para então aplicar o resultado no suporte desejado.

Figura 7 - Aplicação da identidade visual do Carnaval Rio 2022



Fonte: Tátil Design, 2022

Durante o processo de projeto, foram também desenvolvidos experimentos visuais por Marlus Araujo usando o ambiente de programação visual TouchDesigner¹⁸ que, através da interpretação do movimento das imagens captadas de sambistas e porta-bandeiras, serviram de inspiração para o resultado final.

Já no setor das telecomunicações, a atual identidade visual e logo da empresa Oi foi desenvolvida em uma parceria entre as agências Wolff Olins e FutureBrand¹⁹ e apresentada em 2016 para o público. Sua proposta inovadora usou o design generativo ou paramétrico (baseado em processos algorítmicos) para gerar um guia com 70 formas similares, porém distintas, que juntamente às três variações de degradê disponíveis resultam em um conjunto de 210 variações de logomarcas que compõem a identidade da empresa²⁰. Ainda de acordo com o manual de marca da Oi, a possibilidade de usar variações reflete a diversidade das diferentes vozes das pessoas que compõem a companhia. Todas as 210 formas selecionadas e

¹⁶ Ligas das Escolas de Samba do Rio apresenta nova logomarca do carnaval:

<https://g1.globo.com/rj/rio-de-janeiro/carnaval/2022/noticia/2022/02/10/ligas-das-escolas-de-samba-do-rio-apresenta-nova-logomarca-do-carnaval.ghtml> Acesso em 14/04/2022

¹⁷ A linguagem de programação p5.js é uma adaptação da linguagem JavaScript para o ambiente de desenvolvimento simplificado do Processing. Referência: <https://p5js.org/> Acesso em 10/4/2022

¹⁸ O processo de estudo do movimento no TouchDesigner foi documentado em: <https://www.instagram.com/p/CbfKSRCIx3m/> Acesso em 14/04/2022

¹⁹ Fonte: <https://www.b9.com.br/64134/oi-apresenta-nova-identidade-visual-e-logos/> Acesso em 14/04/2022

²⁰ Fonte: Manual de identidade Visual Oi <https://www.refinariadesign.com.br/manuais/Oi/brandbook-manual-de-identidade-oi-2016.pdf> Acesso em 14/04/2022

as outras infinitas possíveis têm forma viva e orgânica, formadas a partir de uma ferramenta desenvolvida especialmente para o projeto.

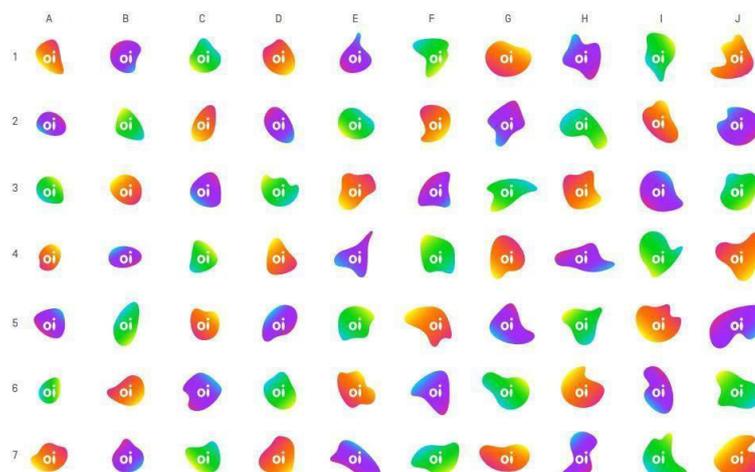
O software em questão foi desenvolvido pelos designers/programadores Henrique Périgo e Dimitre Lima na linguagem de programação C++, utilizando a biblioteca de software Cinder que, de acordo com seu próprio site é “uma biblioteca para programação com intenções estéticas – o tipo de desenvolvimento que geralmente é chamado de programação criativa”²¹. O produto final, criado para a empresa, consiste em uma interface intuitiva que manipula a logomarca, tanto de forma programática seguindo parâmetros numéricos, quanto reativa a áudio, resultando na criação das formas disponibilizadas no manual e de outras infinitas combinações visuais possíveis geradas pelo programa.

Figura 8 - Captura de tela do software desenvolvido para a identidade visual da Oi



Fonte: Dimitre Lima, 2016

Figura 9 - Opções de logomarca indicadas no manual de marca da Oi



²¹ No original em inglês: “Cinder is a C++ library for programming with aesthetic intent - the sort of development often called creative coding”. Tradução dos autores. Referência: <https://libcinder.org/about> Acesso em 14/04/2022

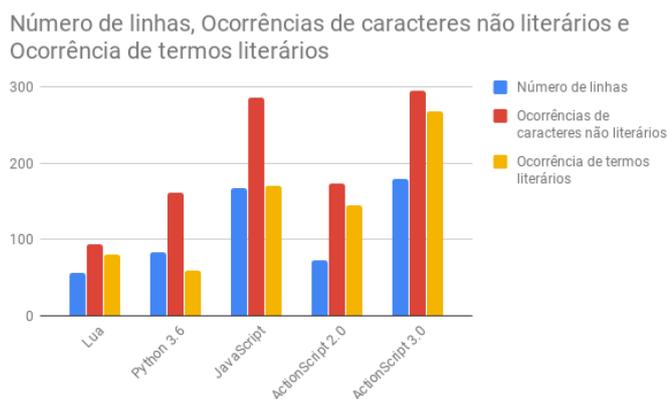
Fonte: Manual de marca Oi, 2016

O mundo contemporâneo demanda por profissionais capazes de propor e implementar novas formas de interação com marcas, conteúdos e recursos, que hoje se concentram em meios digitais. Essas novas formas de interação pouco se assemelham com suas contrapartes analógicas ou são inteiramente novas. Nos casos exemplificados acima, é possível observar como mudanças nos suportes (ex: de impressos para painéis digitais) permitem que marcas adotem movimentos e elementos generativos como características principais em seu desenvolvimento. A programação criativa surge, portanto, como uma das abordagens de ensino capazes de formar profissionais preparados para desenvolver projetos de design na contemporaneidade.

8 Linguagens de programação adotadas na prática do design

Como citado na introdução, a pesquisa realizada pelo bolsista Augusto dos Santos Pereira (2019-2020), em um de seus levantamentos de dados, analisou as linguagens Processing, Lua, Python 3.6, Actionscript 2.0, Actionscript 3.0 e Javascript, para definir a mais adequada para o ensino de programação criativa, fazendo uma comparação de como seria escrever o mesmo código em cada uma das plataformas, de acordo com três parâmetros selecionados: número total de linhas de código; ocorrência de caracteres não utilizados na escrita convencional, como colchetes, chaves, asteriscos e similares; e ocorrência de termos presentes na língua portuguesa ou inglesa em sua forma escrita. Esse levantamento de dados concluiu que, segundo os parâmetros apresentados, as linguagens de programação Lua, Python 3.6 e ActionScript 2.0 apresentavam um menor número de linhas, porém, segundo o autor, a linguagem Python 3.6 apresentava um número de ocorrências de caracteres não literários 63.6% maior do que o número de ocorrências de termos literários (PEREIRA, 2018), característica essa que afasta alunos sem conhecimento prévio de programação por seu formato não convencional.

Figura 10 - Comparação entre linguagens de programação



Fonte: Pereira, 2018.

Por fim, como critério de desempate foi constatado que a linguagem Lua, apesar de ideal, não tinha disponível um ambiente de desenvolvimento integrado (IDE - Integrated Development Environment) próprio, dificultando seu uso e desenvolvimento por pessoas menos experientes. Além disso, verificou-se que a linguagem ActionScript 2.0, apesar de contar com ambiente próprio, foi descontinuada em dezembro de 2020 e seria bloqueada para reprodução em navegadores comuns²² em 2021, se tratando, portanto, de uma tecnologia ultrapassada.

Nesse cenário, foi importante procurar uma outra alternativa que possuísse IDE próprio, e a solução identificada como adequada foi o Processing, um ambiente de programação baseado na linguagem Java, com resultados bastante similares ao Lua, IDE próprio e a capacidade de também ser programada na versão Python e Javascript. Não por acaso, o Processing já era amplamente utilizado no Departamento de Artes & Design da PUC-Rio.

9 Processing, P5.js e Processing.py

Com a definição do Processing como linguagem nos deparamos com mais uma questão: como o título da pesquisa diz, as práticas que serão propostas nesta pesquisa devem servir ao estudante de design na medida em que ofereçam a ele uma possibilidade real de publicar ou compartilhar seus projetos, evitando assim possíveis dificuldades na publicação de suas ideias.

No caso do ambiente de desenvolvimento Processing, existe uma linguagem padrão baseada em Java e dois modos auxiliares baseados nas linguagens Python (Processing.py) e Javascript (p5.js), sendo cada um desses modos uma alternativa oficial à linguagem original com características novas.

A fim de atender a demanda por um ambiente de fácil publicação, foi selecionado o modo p5.js, que por se basear na linguagem JavaScript permite um fácil acesso por qualquer aparelho ou computador usando apenas um navegador (browser) comum de internet, expandindo as possibilidades de publicação e divulgação da peça visual desenvolvida, que pode ser publicada tanto como imagem exportada do programa ou como um software em uma página web.

10 Práticas observadas

Dando continuidade à pesquisa e considerando as limitações de deslocamento impostas pela pandemia de COVID-19, diversas atividades on-line serviram de material de estudo. Nelas foi possível analisar métodos de ensino, temas e exercícios, além de servir para observar as dificuldades dos programadores iniciantes que optaram por realizar esses cursos, permitindo uma análise de quais eram os principais desafios para esses participantes. É importante reforçar que, em todos os cursos e atividades frequentados, o público alvo era

²² Comunicado sobre o fim do suporte ao Flash: <https://www.blog.google/products/chrome/saying-goodbye-flash-chrome/> Acesso em 14/04/2022

majoritariamente formado pelo grupo beneficiado por essa pesquisa, ou seja, estudantes ou profissionais de design e áreas correlatas.

A seguir, uma lista de eventos ou cursos e seus organizadores que foram ou estão sendo observados para a pesquisa:

Festival Multiverso 2021 - Oi Futuro²³;

Processing Community Day 2021 - Processing Foundation²⁴;

LAB Multiverso - André Anastácio e Carlos Oliveira oferecido durante o Festival Multiverso 2021;

Programação Visual: Introdução à programação para designers e artistas - Curso ministrado por André Burnier no Espaço CC²⁵;

Distorcendo Letras - Curso de Processing e tipografia ministrado por André Burnier no Espaço CC;

Design de Objetos Inteligentes - Disciplina ministrada em 2021 pelo professor e orientador desta pesquisa João Bonelli no curso de Design da PUC-Rio.

Interfaces Físicas e Lógicas - Disciplina ministrada em 2022 pelo professor e orientador desta pesquisa João Bonelli no curso de Design da PUC-Rio.

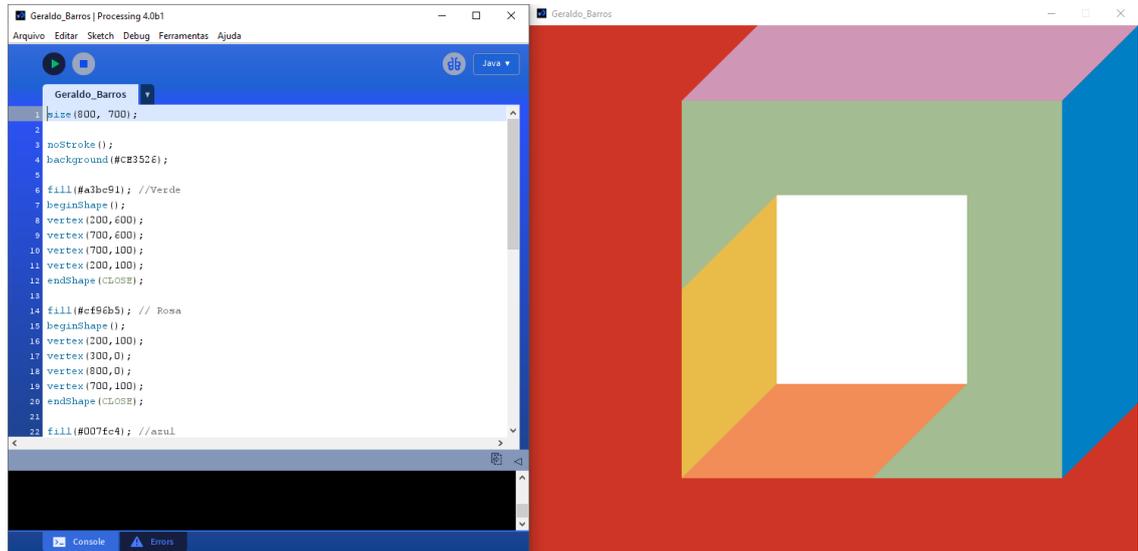
Destes eventos, é importante destacar os cursos oferecidos por André Burnier no Espaço CC, em formato on-line, dos quais um dos autores deste artigo participou como aluno e as disciplinas Design de Objetos Inteligentes e Interfaces Físicas e Lógicas, oferecidas pelo professor João Bonelli no curso de Design da PUC-Rio. O destaque dessas atividades é devido à proposta que elas têm de ensinar alguma modalidade de Processing com foco em iniciantes que estudam ou atuam na área de design ou em áreas correlatas.

Figura 12 - Desenho desenvolvido na Aula 1 do curso oferecido por André Burnier

²³ Festival Multiverso: <https://multiverso.cc/> Acesso em 14/04/2022

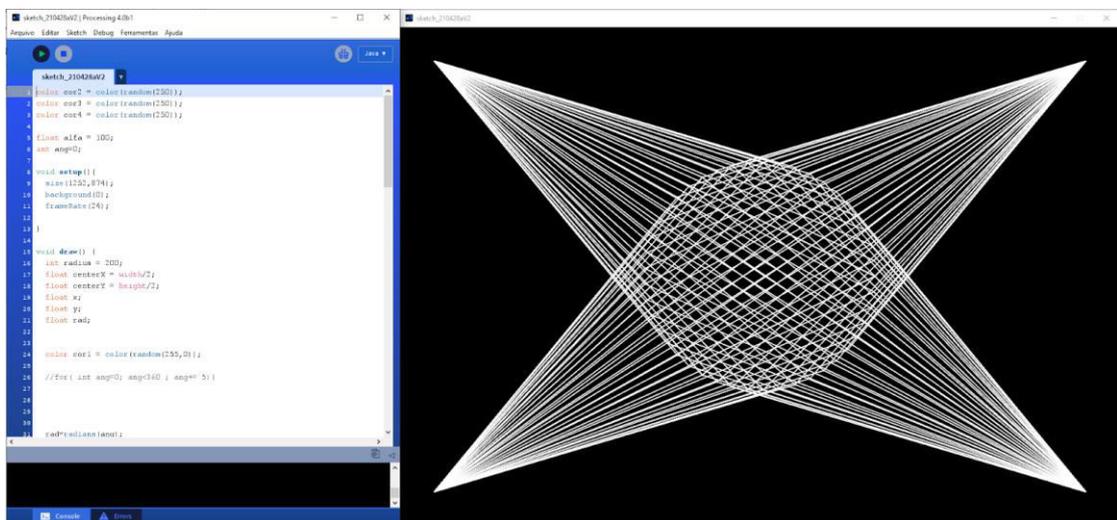
²⁴ Processing Community Day: <https://pcd.encontrosdigitais.com.br/> Acesso em 14/04/2022

²⁵ Espaço CC: <https://espaco.cc/sobre> Acesso em 14/04/2022



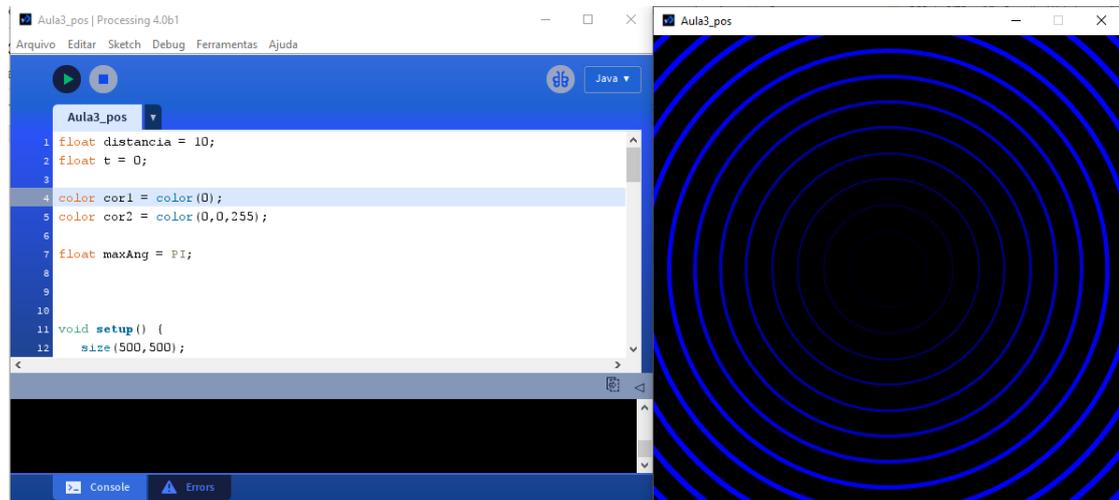
Fonte: Durão, 2021.

Figura 13 - Desenho desenvolvido na Aula 2 do curso oferecido por André Burnier



Fonte: Durão, 2021.

Figura 14 - Desenho desenvolvido na Aula 3 do curso oferecido por André Burnier



Fonte: Durão, 2021.

Apesar de dedicadas a pessoas com um perfil similar, é importante reforçar que todas as práticas observadas têm particularidades que as distinguem, tanto em relação aos assuntos abordados quanto em relação à estrutura, metodologia e tempo de duração, embora com um objetivo em comum: ensinar programação criativa para iniciantes. No caso dos cursos de curta duração, Programação Visual e Distorcendo Letras, o objetivo do primeiro era apresentar as possibilidades da linguagem Processing no modo Java, a partir de um programa que incluía seis aulas que misturavam conceitos básicos de programação com particularidades do Processing e exercícios visuais. O segundo curso, que se trata de uma continuação do primeiro, era focado especificamente em experimentos tipográficos utilizando o Processing no modo Java com exercícios um pouco mais avançados, embora também tenha contado com alunos iniciantes matriculados.

Figura 15 - Desenho desenvolvido no curso Distorcendo Letras

Me
deixa
confuso

Fonte: Durão, 2021.

Figura 16 – Desenho interativo desenvolvido no curso Distorcendo Letras



Te Quero Longe

Fonte: Durão, 2021.

Já as disciplinas Objetos Inteligentes e Interfaces Físicas e Lógicas são oferecidas com duração de um semestre letivo no curso Design do Departamento de Artes e Design da PUC Rio. A disciplina Objetos Inteligentes, que foi acompanhada como objeto de estudo ao longo do segundo semestre de 2021, tem como foco primeiramente apresentar os conceitos básicos do p5.js (Processing no modo JavaScript) e de inteligência computacional para então propor que os alunos estudem por conta própria os tópicos avançados que serão necessários no projeto individual que cada um irá desenvolver. Nesse formato, o aluno é incentivado a aprender fazendo, um dos pilares da metodologia de Prática Reflexiva definida por Donald Schön, que acaba por tornar a aula um grande ateliê de experimentação prática. Já a disciplina Interfaces Físicas e Lógicas, que foi cursada pelo bolsista como aluno ao longo do primeiro semestre de 2022, foca no ensino de Processing no modo Java e na plataforma de microcontroladores embarcados Arduino²⁶, possibilitando que alunos criem projetos com programação criativa que ultrapassam a barreira do digital e se comunicam com objetos físicos do nosso dia a dia.

²⁶ Arduino: <https://www.arduino.cc/>

Figura 16 - Tinta Solta, exemplo de projeto de design de interação física desenvolvido na graduação em Design da PUC-Rio pelo aluno João Whitaker



Fonte: Whitaker, 2014.

Os conceitos básicos de programação ensinados em todos os cursos são:

Formas básicas

Cores e estilos

Variáveis

Condicionais

Funções

Loops

Fontes, Imagens e Visão Computacional

A análise das atividades acima permitiram que uma proposta de minicurso de um dia para alguns calouros de Design da Pontifícia Universidade Católica do Rio de Janeiro fosse esboçada, planejada e realizada em dezembro de 2021, contando com a presença de 5 participantes, um número que serviu para validar a barreira de dificuldade da atividade e para fazer um ensaio para futuras atividades.

11 Minicurso “Programa Cartaz”

O minicurso “Programa Cartaz” nasceu como resposta a um convite do Departamento de Artes e Design da PUC-Rio que desejava desenvolver uma série de atividades para receber no campus, pela primeira vez, alunos que ingressaram no curso de Design durante a pandemia de COVID-19 e ainda não tinham tido a oportunidade de estar presencialmente na universidade.

O pedido feito aos laboratórios de ensino do departamento, incluindo o Laboratório de Interfaces Físicas Experimentais LIFE PUC-Rio, foi de que desenvolvêssemos atividades de curta duração para introduzir aos alunos os assuntos pesquisados no local, apresentando a prática do design com os maquinários que estão disponíveis para uso dos alunos nos laboratórios e incentivando o intercâmbio entre novos frequentadores, veteranos, monitores, estagiários e técnicos nos espaços em questão. Para esse evento, o LIFE decidiu formar dois grupos de trabalho que seriam acompanhados pelos orientadores desta pesquisa, Luiza Novaes e João Bonelli. O primeiro grupo foi formado pelos monitores, que no dia a dia são responsáveis pelo apoio às atividades do laboratório e aos alunos que necessitam de suporte técnico. Essa equipe focou em introduzir o tópico de computação física, eletrônica básica e apresentar o sistema Arduino aos visitantes. O segundo grupo, formado por um dos autores desse artigo, que é bolsista PIBITI do CNPq, focou no tópico da programação criativa e das possibilidades dessa ferramenta em projetos de design, principalmente de natureza gráfica e visual.

Durante as conversas iniciais com os orientadores foi decidido que a melhor forma de apresentar um minicurso introdutório sobre programação criativa seria propondo um roteiro, ou exercício, que unisse alguns dos principais tópicos ensinados nas atividades que observei, ou seja, como programar formas básicas, alterar cores, estilos e fontes, guardar informações em variáveis, utilizar condicionais lógicas e loops, criar funções e manipular imagens e vídeo da webcam. Além disso, era de extrema importância que resultasse em um projeto de design, ou seja, que o resultado final fosse algum objeto ou forma que pudesse ser remixado e adaptado para algum uso prático do aluno. Seguindo o último aspecto, foi decidido que o exercício teria como objetivo o desenho de um cartaz usando somente programação. Para isso, foram desenvolvidas três propostas de exercícios, com complexidades e com resultados de aspectos visuais distintos, a fim de que, em uma futura análise, fosse decidido qual seria o exercício selecionado para a aula.

Os exemplos chamados de Cartaz 01, Cartaz 02 e Cartaz 03 se apresentavam visualmente da seguinte forma:

Figura 17 - Cartaz 01, 02 e 03 respectivamente



Fonte: DURÃO, 2022.

Cartaz 01 - Gerador "Athos Bulcão"

Tópicos utilizados: formas, estilo, condicionais, fontes, variáveis, funções, loops e rotação/translação.

Esse exemplo consiste na criação de um cartaz interativo que sorteia diferentes posições para cada forma criada nos quadrantes. Clicando no teclado o usuário pode escolher quando sortear uma nova forma e quando parar em uma forma desejada.

Cartaz 02 - Cartaz Noise

Tópicos utilizados: formas, estilo, condicionais, fontes, variáveis, funções, loops e noise.

No Cartaz 02 o diferencial é a criação de uma malha de pontos que ocupa toda a extensão do cartaz, sendo que cada ponto recebe um valor que indica seu tamanho. Esse valor varia com o tempo, tendo como base uma função randômica de noise, gerando esse visual que lembra nuvens.

Cartaz 03 - Cartaz concretista

Tópicos utilizados: formas, estilo, condicionais, fontes, variáveis, funções e loops.

Nesse último exemplo, o usuário escreve as linhas de texto conforme o mouse se movimenta na tela, gerando um layout novo a cada experimento. Além disso, é possível criar círculos vermelhos e carimbar informações, como a data do cartaz, usando teclas do teclado.

Após analisar os níveis de complexidade das três possibilidades de atividades desenvolvidas junto aos orientadores desta pesquisa, foi decidido que o cartaz 03 se tratava de um exercício mais adequado para iniciantes na programação criativa, e que caso necessário, novos tópicos poderiam ser adicionados em cima do exercício base, complementando assim eventuais interesses dos alunos que não fossem contemplados. A partir da atividade decidida foi desenvolvida uma apresentação em PowerPoint que serviu para a introdução da atividade e foi dividida da seguinte forma:

O que é código e quais são os tipos de código;

Qual a função das diferentes linguagens (CSS, HTML e JavaScript) na construção de uma aplicação;

O que é Processing e porque vamos utilizar o Processing no modo JavaScript;

O que podemos fazer com programação criativa;

Introdução sobre a atividade que vamos realizar.

No dia da visita, finalizada a apresentação, os alunos foram convidados a abrir o Processing em seus computadores e a acompanhar a explicação oral enquanto programávamos em tempo real o exercício mostrado. O objetivo era que cada aluno seguisse a explicação, adicionando um pouco de sua personalidade ao longo do processo, alterando lógicas, estilos ou até mesmo propondo alterações que não tivessem sido previstas. Uma dessas modificações não antecipadas foi o interesse em adicionar imagens nos cartazes, pedido esse que foi prontamente atendido pelo monitor, que logo ensinou aos participantes como guardar imagens em variáveis, para que fossem utilizadas da forma desejada pelos alunos.

Figura 18 - Aluno experimenta o uso de imagens em seu cartaz



Fonte: Arquivo dos autores, 2021.

Figura 19 - O bolsista ministra o workshop Programa Cartaz no Laboratório de Interfaces Físicas Experimentais - LIFE



Fonte: Arquivo de Durão, 2021.

No fim da atividade os alunos foram convidados a preencher um breve formulário para o levantamento de opiniões sobre o minicurso, sobre o exercício, perguntas sobre a experiência prévia do participante com programação, uma enquete sobre a dificuldade de aprendizagem dos tópicos tratados e uma pergunta de caráter qualitativo pedindo uma análise do aluno sobre a relevância da atividade.

Sobre as respostas do formulário é importante destacar algumas informações coletadas: na atividade, todos os participantes eram alunos do curso de design com habilitação em mídia digital e possuíam pouca ou muita experiência com programação, sendo que 60% dos participantes sentiram alguma dificuldade no exercício proposto. Essas dificuldades relatadas foram: “Improvisar” e “Basicamente coisas relacionadas a números”, indicando que os exercícios com objetivos livres podem ser uma barreira para alguns alunos que têm mais dificuldade com a expressão livre e que a matemática segue sendo uma grande dificuldade. Sobre a pergunta “Você acha que programação pode te ajudar como designer?” Houve unanimidade nas respostas indicando que para os participantes a programação tem sim valor no processo de criação do designer, com destaque para a seguinte resposta de um dos participantes que sintetiza bem o objetivo desta pesquisa: “Sim, bastante, dá pra mesclar muito bem os dois mundos”.

12 Conclusões

A programação criativa surge como uma alternativa que possibilita uma nova forma de expressão para designers no contexto digital. Segundo Abreu (2020), “Ao fazer uso de algoritmos para gerar elementos digitais e artefatos artísticos expressivos, não necessariamente funcionais, a codificação criativa desempenha uma atividade poética estética.”

Nesse sentido, incentivar e discutir o ensino de programação criativa para estudantes de design é uma forma de apoiar uma nova geração de designers de todas as especialidades, cujos projetos podem ser facilitados ou potencializados com o uso da programação criativa.

A partir da análise das informações levantadas ao longo da pesquisa, foi possível compreender melhor o contexto da programação criativa e seu uso e ensino no Design, tanto no cenário nacional quanto internacional, bem como as diferentes frentes ligadas diretamente ao ensino de programação criativa em todo o Brasil. A etapa seguinte foi a organização e fichamento de diferentes propostas de ensino de programação criativa, coletando assim um panorama sobre as possibilidades de ensino deste tema para estudantes de design, sempre seguindo a metodologia de prática reflexiva de Donald Schön. Com os dados levantados, foi possível organizar uma atividade de curta duração para coletar depoimentos e aprendizados acerca da prática educacional, sendo a etapa seguinte a formulação de mais atividades de média e curta duração para turmas maiores e mais diversas, no que se refere às habilitações dos participantes, sempre seguindo e coletando feedbacks para incremento das atividades.

13 Referências

ABREU, Igor. **Imaginação como método em MULTIVERSO**; Gisela Andrade (org). Rio de Janeiro, RJ: BARLEU EDIÇÕES, 2020

BONELLI, João. **Prática Reflexiva em Interfaces Físicas: uma proposta de ensino-aprendizagem de design de interação**. Tese (doutorado)—Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Artes e Design, 2016

MAEDA, John. **Design by Numbers**. Cambridge: MIT Press, 1999.

MENABREA, Luigi. **Notions sur la machine analytique de M. Charles Babbage**. Bibliothèque Universelle de Genève, nouvelle série 41 pp352–76, 1842.

OLIVEIRA JUNIOR, Carlos de. **Encontros digitais: a construção de uma plataforma digital para comunidade de programação criativa brasileira**. Dissertação (Programa de Mestrado Profissional em Gestão da Economia Criativa) - Escola Superior de Propaganda e Marketing, [Rio de Janeiro], 2021.

PEREIRA, Augusto dos S.; BONELLI, João e NOVAES, Luiza (orientadores). **Experimentação da linguagem de programação Lua em ambiente de ensino-aprendizagem de programação**



criativa para designers. XXVI Seminário de Iniciação Científica e Tecnológica PUC-Rio. Rio de Janeiro: PUC-Rio, 2018.

PEREIRA, Augusto dos S.; BONELLI, João e NOVAES, Luiza (orientadores). **Programação Criativa: Ensino de Linguagens de Programação para Designers.** Relatórios de Iniciação Científica PUC-Rio. Rio de Janeiro: PUC-Rio, 2019 e 2020.

PINHEIRO, Mauro. **Design de interação e computação pervasiva: um estudo sobre mecanismos atencionais e sistemas de informação ambiente.** Tese de Doutorado, Programa de Pós-graduação em Design, PUC-Rio, 2011.

RANOYA, Guilherme. **PROJETO Futuro do Design.** [S. l.], 2021. Disponível em: <https://www.ranoya.com/public/futuro/index.php>. Acesso em: 19 ago. 2021.

REAS, Casey e FRY, Ben. **Processing: a programming handbook for visual designers and artists.** Cambridge: MIT Press, 2007.

SCHÖN, Donald. **Educando o Profissional Reflexivo: Um novo design para o ensino e a aprendizagem.** 1. ed. [S.l.]: ARTMED, 2000.

SCHÖN, Donald. **The Reflective Practitioner: How Professionals Think in Action.** New York: Basic Books, 1983

WHITAKER, João. **Tinta Solta.** Trabalho de Conclusão de Curso - TCC, Bacharelado em Design. Rio de Janeiro: PUC-Rio, 2014.