



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ARQUITETURA E URBANISMO
CURSO DE DESIGN

DENISE PINHEIRO POMPEU

DESIGN++CÓDIGO:CÓDIGO COMO FERRAMENTA DE DESIGN

FORTALEZA

2018

DENISE PINHEIRO POMPEU

DESIGN++CÓDIGO:CÓDIGO COMO FERRAMENTA DE DESIGN

Monografia apresentada no curso de Design do Departamento de Arquitetura e Urbanismo da Universidade Federal do Ceará, como requisito parcial para obtenção do título de Bacharel em Design.

Orientador: Prof. Dr. Roberto Cesar Cavalcante Vieira.

FORTALEZA

2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

P851d Pompeu, Denise Pinheiro.

DESIGN++CÓDIGO : CÓDIGO COMO FERRAMENTA DE DESIGN / Denise Pinheiro
Pompeu. – 2018.
126 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Design, Fortaleza, 2018.

Orientação: Prof. Dr. Roberto Cesar Cavalcante Vieira.

1. ensino. 2. lógica de programação. 3. design paramétrico. 4. design generativo. 5. design. I. Título.

CDD 658.575

DENISE PINHEIRO POMPEU

DESIGN++CÓDIGO:CÓDIGO COMO FERRAMENTA DE DESIGN

Monografia apresentada no curso de Design do Departamento de Arquitetura e Urbanismo da Universidade Federal do Ceará, como requisito parcial para obtenção do título de Bacharel em Design.

Orientador: Prof. Dr. Roberto Cesar Cavalcante Vieira.

Aprovada em ___/___/_____. BANCA EXAMINADORA

Prof. Dr. Roberto Cesar Cavalcante Vieira. (Orientador)
Universidade Federal do Ceará (UFC)

Profa. Dra. Camila Barros
Universidade Federal do Ceará (UFC)

Prof. Dr. Paulo Jorge Alcobia Simões
Universidade Federal do Ceará (UFC)

Profa. Ma. Mara Franklin Bonates
Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

Sou grata a Deus por me dar forças e bom ânimo para para concluir esse trabalho tão importante na minha vida. Tenho enorme gratidão aos meus pais, por seu amor incondicional, por sempre me incentivarem e acreditarem em mim, sem eles isso não seria possível, assim como também a toda a minha família.

Sou grata à Universidade Federal do Ceará (UFC) pela oportunidade de fazer parte da instituição e levar para vida meu aprendizado e crescimento pessoal e profissional. Assim como sou grata a todo corpo docente, a banca de defesa e em especial ao meu orientador Professor Roberto Vieira pela sua dedicação, sensibilidade e entusiasmo, grata por sempre me incentivar a fazer o melhor e me auxiliar. Sou grata a Professora Camila Barros por seu incentivo durante o curso e por ter me auxiliado a chegar até aqui. Gratidão também a Professora Alécia, que me motivou com seu jeito alegre e simplicidade. Além disso, quero agradecer a Professora Lilu, Professora Cláudia e Professor Paulo por terem feito parte dessa minha caminhada.

Sou grata aos meu amigos que estiveram por perto me incentivando, auxiliando e apoiando, especialmente ao Levi que desde o começo da faculdade tem sido um amigo valioso e me apoiado sempre. Agradecer também a minha amiga Laís pelo seu apoio e amizade. Sou grata a Suelen por sua ajuda principalmente nesses momentos finais , como também as minhas amigas Júlia, Karol e Sibebe.

Durante o trabalho agradeço aqueles que contribuíram, Giovanna que desde o início trouxe bons insights, Lorraine que também foi uma voz amiga nesse trabalho, Josias por sua disponibilidade, e também à Luisa e Catarina . Assim como sou grata a todos os meus colegas da primeira turma de design.

Sou grata também ao programa Ciência sem Fronteiras por ter possibilitado uma experiência tão enriquecedora na minha vida. Sou grata ao Edelino pelo o seu auxílio e por toda a equipe do DAU, grata a todos.

“Projetar regras e instruções é uma parte intrínseca do processo de design.”

Ellen Lupton

RESUMO

Estudo sobre metodologia no ensino de programação e como técnicas de aprendizado podem ser aplicadas para potencializar esse ensino, a fim de criar um produto que possa aproximar design de programação. A pesquisa é constituída de estudos de caso a partir de questionários, testes metodológicos e observações feitas em *workshops* de ensino de programação. As conclusões serviram de base para elaboração de um material com proposta de tornar programação um assunto mais lúdico e pessoal.

Este estudo tem como intuito avaliar a maneira como se tem pensado o ensino de assuntos que não são de domínio do Design, nesse caso programação, em sala de aula, bem como analisar sua eficiência e testar materiais para a criação de um produto que torne o assunto mais acessível, sendo assim, uma proposta de material de estímulo para o aprendizado em programação voltada para design.

Palavras-chave: ensino; lógica de programação; design; construcionismo; design generativo.

ABSTRACT

Study about teaching methodology and other techniques and how the learning techniques can be applied to education, in the order to create a product that approaches Design and Programming. The research has case studies, developed on questionnaire, methodological tests and reflections produced on programming learning workshops. The results provide a base to create a material that would bring programming to a ludic and personal subject.

This study aims to evaluate how the teaching programming has been realize in the classroom. Besides that, it aims to analise the effience and to test materials of differentes teaching programming methods, to create a product that turn the programminh subject more approachble to design.

LISTA DE FIGURAS

Figura 1 - Exemplo de instalação interativa.....	16
Figura 2 - Exemplos de materialização, utilizando gravação a laser e caneta adaptada à fresadora CNC.....	17
Figura 3 - Exemplos de aplicações generativas de design	18
Figura 4 - Diagrama de <i>design cycle</i> por Vijay Vaishnavi e Bill Kuechler (2004)...	23
Figura 5 - Modo de preparo de bolo fofo.....	30
Figura 6 - Exemplo de parâmetro de número de segmentos.....	32
Figura 7 - Trabalho de Enzo Henze: <i>Red Ambush</i>	33
Figura 8 - Exemplo de processo de criação de código de design generativo.....	35
Figura 9 - Diagrama do processo de design generativo.....	36
Figura 10 - COP15 - software de identidade generativa.....	37
Figura 11 - COP15 - exemplos de resultados de identidade generativa.....	38
Figura 12 - Exemplo de letras generativas feitas por Hyper Glu.....	39
Figura 13 - Logo do MIT Media LAB e variações do logo feita para cada um dos componentes do laboratório.....	40
Figura 14 - Exemplo de aplicação da identidade.....	40
Figura 15 - Processo de criação de da arte generativa a partir da fotografia do Reitor Martins Filho.....	41
Figura 16 - Instalação " <i>Onskebronn</i> " em Berlim.....	42
Figura 17 - Visualização de dados por Jer Thorp.....	43
Figura 18 - interface do LOGO.....	48
Figura 19 - Exemplo de script de jogo de ping-pong feito no Scratch.....	49
Figura 20 - Kit com os componentes para montagem de robôs do Lego Mindstorm.....	50
Figura 21 - Exemplo de robô de corrida montado com componentes do kit Lego Mindstorm.....	50
Figura 22 - Bloco programável conectado ao computador.....	51

Figura 23 - O método de ciclo de aprendizagem 5E.....	53
Figura 24 - Representação do processo de funcionamento do coração humano feito por Paul Klee.....	55
Figura 25 - Decomposição do processo de construção de tipografia generativa feita na oficina Type and Code.....	56
Figura 26 - Instalação de esferas e processo de construção.....	58
Figura 27 - Resultado de exercício proposto na oficina de Tipografia Generativa.....	61
Figura 28 - Processo de ensino de oficina de Type and Code	63
Figura 29 - Folha para aplicação do pseudocódigo e desenho do processo.....	64
Figura 30 - Aplicação do pseudocódigo e desenho do processo.....	64
Figura 31 - Alunos aplicando metodologia testada.....	65
Figura 32 - Oficina Type and Code.....	65
Figura 33 - Exemplo de aplicação do teste com pseudo-código.....	66
Figura 34 - Kit de arduino para iniciantes.....	71
Figura 35 - Ícones representando os componentes e metáfora sobre resistência elétrica.....	72
Figura 36 - Livro <i>Generative Design</i>	73
Figura 37 - Livro <i>Design by Numbers</i>	74
Figura 38 - Livro <i>Processing for Visual Artists</i>	75
Figura 39 - Fotos do livro <i>Form and Code</i>	76
Figura 40 - Livro <i>Processing: a programming handbook for visual designers and artists</i>	77
Figura 41 - Processo de uso do livro de receitas <i>Cook This Page</i>	78
Figura 42 - Capa do livro <i>Escola de Design</i>	79
Figura 43 - Imagens da oficina do código do projeto "Design por Mulheres"	85
Figura 44 - Protótipo de material de ensino apresentado em oficina na UFC.....	85
Figura 45 - Painel de referências.....	88
Figura 46 Perguntas do formulário sobre preferências de design.....	89

Figura 47 - Fotos dos estudantes de design apresentados no livro.....	91
Figura 48 - Códigos selecionados para o livro.....	92
Figura 49 - Códigos apresentado na exposição "Design por Mulheres".....	93
Figura 50 - à esquerda Reitor da UFC Henry Campos mostrando códigos feitos no curso de Design. A direita Aluno Josias Nascimento no Circuito UFC-Arte.....	93
Figura 51 - Diagrama de associação de perguntas com características do código	94
Figura 52 - Códigos do livro categorizados.....	95
Figura 53 - Diagrama de associação das perguntas com parâmetros do código.	96
Figura 54 - Conjunto de ícones das respostas elaborados para o livro.....	97
Figura 55 - Conjunto de ícones dos códigos elaborados para o livro.....	97
Figura 56 - Família tipográficas utilizadas no livro.....	98
Figura 57 - Padrão cromático utilizado no livro.....	99
Figura 58 - Aplicação dos códigos com processamento de imagens.....	100
Figura 59 - Capa do livro Design++Código.....	100
Figura 60 - Folha de rosto do livro.....	102
Figura 61 - Introdução do livro - tópico "O que?".....	103
Figura 62 - Introdução do livro - tópico "Quem? Onde?".....	103
Figura 63 - Introdução do livro - tópico "O que você prefere?".....	104
Figura 64 - Relação dos parâmetros com as características do código.....	105
Figura 65 - Códigos categorizados.	105
Figura 66 - Relação das respostas com os parâmetros.....	106
Figura 67 - Abertura do capítulo de códigos.....	107
Figura 68 - Página de apresentação do código "Gritador".....	107
Figura 69 - Página de aplicação do código "Gritador".....	108
Figura 70 - Página de apresentação do código "Design por mulheres".....	108
Figura 71 - Página de aplicação do código "Design por Mulheres".....	109
Figura 72 - Página de aplicação do código "Design por Mulheres".....	109
Figura 73 - Livro em página de aplicação de código.....	110

LISTA DE GRÁFICOS

Gráfico 1 - Formulário: nível de receio em aprender programação.....	59
Gráfico 2 - Formulário: nível de conhecimento em programação.....	60
Gráfico 3 - Formulário: interesse em aprender programação.....	61

LISTA DE TABELAS

Tabela 1 - Cronograma de pesquisa.....	25
Tabela 2 - Parâmetros da análise de similares.....	80
Tabela 3 - Cronograma de Projeto.....	83

SUMÁRIO

1. INTRODUÇÃO	16
1.1. Justificativa	19
1.2. Problema	20
1.3. Objetivos	21
1.3.1. Geral	21
1.3.2. Específicos	21
1.4. Cenário	21
1.5. Metodologia	22
1.6. Estrutura	26
2. RELAÇÃO DO DESIGN COM PROGRAMAÇÃO	28
2.1. Introdução	28
2.2. Conceitos de linguagem de programação	28
2.3. O que é algoritmo	31
2.4. Aplicação no design	32
2.4.1. Dificuldades com o código para designers	44
2.5. Conclusões	45
3. TEORIAS PEDAGÓGICAS E FERRAMENTAS DE APRENDIZAGEM	46
3.1. Introdução	46
3.2. Construcionismo	47
3.3. Modelo de aprendizado	53
3.4. Design como ferramenta para facilitar o ensino	54
3.5. Conclusão	58
4. ESTUDO DE CASO	59
4.1. Introdução	59
4.2. Questionário de perfil do curso	60
4.3. Oficina Type and Code	62
4.3.1. Descrição e realização do pseudocódigo	63
4.3.2. Análise de metodologia aplicada na oficina	67
4.3.3. Análise do formulário	69
4.4. Conclusão	70
5. ANÁLISES DE SIMILARES	71
5.1. Avaliação dos parâmetros dos livros	81
6. DIRETRIZES DE PROJETO	83
7. PROJETO	85
	14

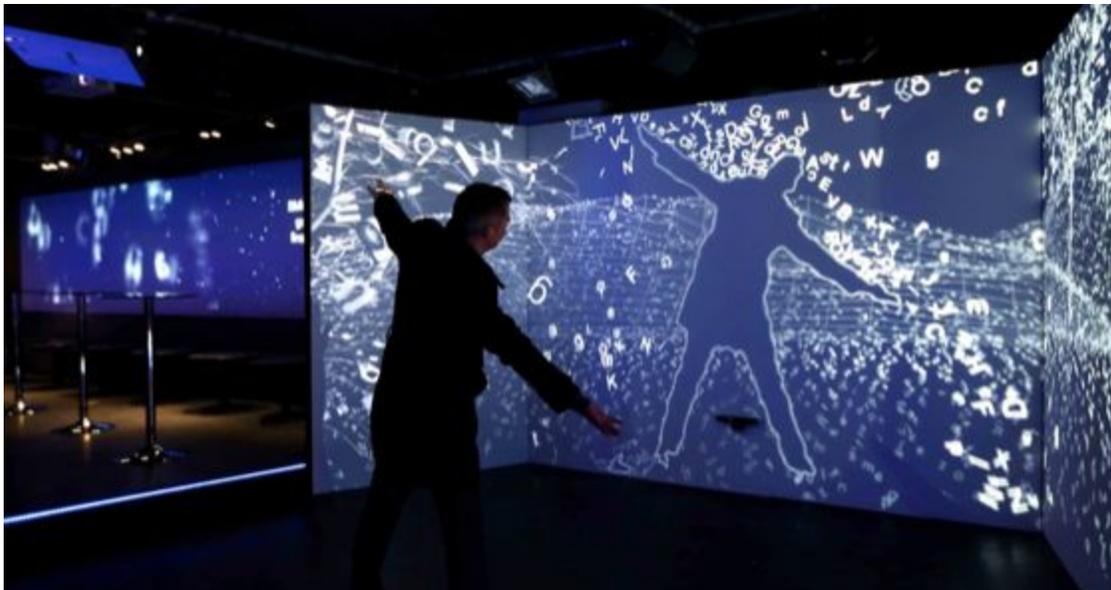
7.1. Introdução	85
7.2. Oficina	85
7.3. Reavaliação dos requisitos	87
7.3.1 Aprendizado	87
7.3.2 Tema	88
7.4. Processo	88
7.4.1 Referências	88
7.4.2 Levantamento de dados	90
7.4.3. Seleção de fotografias	91
7.4.4. Seleção dos códigos	92
7.4.5 Associação dos parâmetros	94
7.5. Elementos visuais	97
7.5.1 Ícones	97
7.5.2 Tipografia	98
7.5.3 Padrão Cromático	99
7.5.4 Aplicação dos códigos	100
7.5.5 Formato	101
7.6. Livro	102
7.6.1 Capa	102
7.6.2 Introdução	103
7.6.3 Explicação	105
7.6.3 Códigos	107
8. CONCLUSÃO	112
9. REFERÊNCIAS	113
10. ANEXOS	117
10.1. Formulário de perfil de curso	117
10.2. Formulário de feedback da oficina 1	122
10.3. Formulário de feedback da oficina 2	125

1. INTRODUÇÃO

O avanço das tecnologias vem trazendo novas possibilidades a designers e artistas no mundo todo. Com a utilização de sistemas computacionais torna-se possível a criação de imagens e formas generativas, além disso com possibilidade de interação. No design é possível aplicar essa tecnologia para gerar novos resultados em tipografia, identidade visual, instalações, artes visuais, visualização de dados, entre outros.

No Brasil, iniciativas como o Museu do Amanhã e Museu da Língua Portuguesa fizeram uso de instalações interativas, possibilitando inovações entre arte e tecnologia, com a criação de espaços interativos, lúdicos e experimentais. A figura 1.1 exemplifica uma instalação interativa com a captura dos gestos realizados pelo visitante e interação com elementos virtuais .

Figura 1 – Exemplo de instalação interativa.



Fonte - Projeto feito pelo estúdio de mídia Blow Factory¹ para BMW

O advento da fabricação digital por meio de fresadoras com controle

¹ Disponível em: <<http://blowfactory.com/bmw-interactive-installation>>. Acessado em 01 Jul. 2017.

numérico computadorizado (CNC), corte a laser e impressão 3D também possibilitam uma nova forma de criação e materialização. Redes open source propagam trabalho colaborativo e compartilhamento de informações.

Figura 2 - Exemplos de materialização, utilizando gravação a laser e caneta adaptada à fresadora CNC.



Fonte - Foto de Roberto Vieira.

Para que essa exploração de novas tecnologias ocorra há a necessidade de um contato mais direto do designer com a programação de computadores, com o intuito de gerar uma conversa reflexiva no processo de design, numa relação de “ver, mudar e ver” (Shon, 1992), ou seja, o designer modifica os resultados do código, num ciclo de experimentações até chegar em um projeto interessante, inovador e complexo.

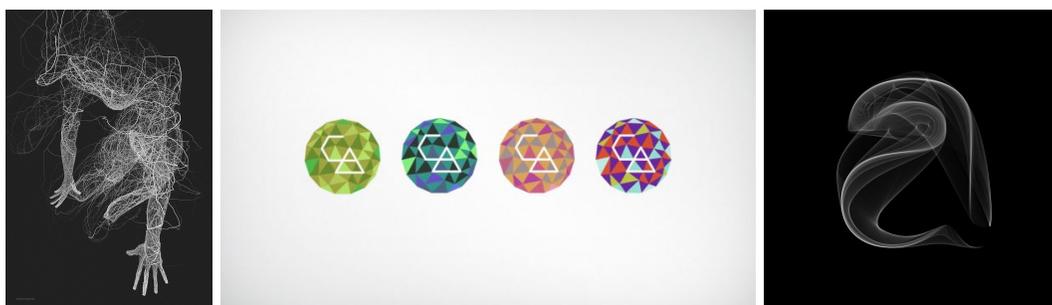
Para tanto, é necessário haver um nível básico de compreensão de linguagem de programação.

De acordo com Bohnacker (2012) o uso de programas como adobe creative suite, autoCAD, 3ds Max, tornaram ferramentas de design mais eficientes, mas não transformaram substancialmente o processo de design. Isso é diferente no uso de programação, segundo o mesmo autor, "design generativo é diferente de métodos convencionais, o processo de design é único e resulta fundamentalmente em novas

possibilidades"² (2012, p. 460). O designer não pensa no desenho mas nas regras que resultarão nele, o desafio está em formalizar suas ideias no código.

A figura 3 coloca exemplos de arte generativa de Janusz Jurek, logo generativo do site Creative Applications e a tipografia generativa do Sanrun Mining Co.

Figura 3 - exemplos de aplicações generativas de design.



Fonte- sites this is colossal³, brz comunicação⁴ e behance⁵

1.1. Justificativa

Segundo Flusser o designer é considerado um profissional que aproxima técnica e arte. Porém, no contexto atual existe uma dissociação entre programação e artes visuais. Universidades separam estudantes de arte e de tecnologia, sem integrar as capacidades de ambos, porém a convergência entre essas áreas pode ocasionar novas formas de pensar e de criar. Flusser descreve o papel de mediador entre o científico e artístico que cabe ao designer:

A cultura moderna (...) fez uma separação brusca entre o mundo das artes e o

² Generative design is different from conventional methods; the design process is unique and fundamentally results in new possibilities.

³ Disponível em:

<<http://www.thisiscolossal.com/2015/08/generative-illustrations-of-the-human-form-by-janusz-jurek/>>. Acessado em 01 Jul. 2017.

⁴ Disponível em: <<http://www.brzcomunicacao.com.br/design-processing/>>. Acessado em 01 Jul. 2017.

⁵ Disponível em:

<https://www.behance.net/gallery/22718045/Sanrun-Mining-Co?utm_content=buffera8b41&utm_medium=social&utm_source=pinterest.com&utm_campaign=buffer>. Acessado em 01 Jul. 2017.

mundo da técnica e das máquinas, de modo que a cultura se dividiu em dois ramos estranhos entre si: por um lado, o ramo científico, quantificável, duro, e por outro o ramo estético, qualificador, brando. Essa separação desastrosa começou a se tornar insustentável no final do século XIX. A palavra design entrou nessa brecha como uma espécie de ponte entre esses dois mundos. E isso foi possível porque essa palavra exprime a conexão interna entre técnica e arte. E por isso design significa aproximadamente aquele lugar em que arte e técnica (e, conseqüentemente, pensamentos, valorativo e científico) caminham juntas, com pesos equivalentes, tornando possível uma nova forma de cultura. (FLUSSER, 2007, p. 183-184)

No atual momento histórico existe uma tendência ao ensino de lógica de programação para crianças. Na cidade de Fortaleza existem algumas escolas voltadas pra isso, como a Gênio Azul , *Happy Code*, e *Super Geeks*. Porém a geração que está na universidade não teve contato com esse conteúdo na sua formação de base, ou seja próximas gerações poderão ter mais facilidade em desenvolver um pensamento computacional do que a atual geração universitária. Com isso surgem questões de como criar uma ferramenta que possa estimular o aprendizado desses conteúdos a pessoas que não o tiveram em sua formação de base.

Existe muita informação disponível e desconexa e, o papel do design, na atual conjectura, pode se dar pela gestão desta complexidade (MORAES, 2010) " torna-se, então necessário, para o design atual, dentro do cenário de complexidade existente, valer-se de novas ferramentas, instrumentos e metodologias para a compreensão e gestão da complexidade contemporânea.(MORAES, 2010, p.13)". Dessa forma o papel do design no contexto desta pesquisa é de criar uma ferramenta para a gestão do conhecimento abstrato de lógica de programação aplicada ao design.

O projeto resultante desta pesquisa tem como intuito propor um método de aprendizado de programação para designers, para que se possa dialogar com esse tipo de linguagem e também criar condições para se dar um passo adiante na

relação com programação, propondo uma utilização criativa do código, com o objetivo de aproximar arte de tecnologia para resultados mais experimentais.

Este trabalho propõe-se a investigar como essa relação entre design e programação pode existir e ser efetiva, considerando até que ponto o aluno deve se aprofundar, quais os conceitos que devem ser estudados, de que forma devem ser passados, enfim, como um profissional ou estudante de artes visuais pode se relacionar com código e criar a partir dele.

1.2. Problema

Segundo Maeda (2004, p.101) código não é uma ferramenta e sim um novo material de expressão. Dessa forma, ao se conceber um projeto utilizando programação, o designer tem a possibilidade de utilizar o código como material. Há, segundo Shon (1992), uma conversa reflexiva entre o designer e o material usado. Desse modo, uma relação mais direta do designer com o código pode gerar possibilidades no processo de design. A pergunta desta pesquisa é: como é possível estimular o aprendizado de lógica de programação para estudantes de design que não tiveram acesso a esse conhecimento na sua formação escolar?

1.3. Objetivos

Esta pesquisa tem como intuito analisar as dificuldades de aprendizado em programação de alunos de design. A partir disso, levantar e analisar dados sobre métodos complementares de ensino, como também criar parâmetros para fundamentar o projeto de uma ferramenta que estimule o aprendizado de programação no contexto universitário.

1.3.1. Geral

Propor uma ferramenta que estimule o ensino de lógica de programação no contexto universitário que possibilite um aprendizado contextualizado e atrativo a

alunos que não tiveram acesso a conceitos de programação em sua formação básica.

1.3.2. Específicos

- Compreender como a lógica de programação se relaciona com design.
- Descrever as dificuldades de aprendizado em programação de alunos de design
- Elencar ferramentas complementares de ensino de lógica de programação.
- Criar requisitos de como deve ser a ferramenta, de estímulo ao aprendizado de código, proposta e como seus conteúdos devem ser organizados.
- Apresentar metodologias pedagógicas que facilitam o ensino de conteúdos abstratos, como programação, música, matemática.

1.4. Cenário

Com base na justificativa e nos objetivos foram pensadas alguns cenários para a ferramenta proposta.

Cenário 1: Primeiramente foi pensado um *plug-in* para processing que auxiliaria em tempo real o aluno a construir o código, isso poderia ocorrer por meio de animações, esquemas visuais, exemplos etc. Essa opção daria um visão mais clara de como as ações eram executadas e do que poderia ser feito em seguida, sendo um guia para montagem do código.

Cenário 2: Com uma pesquisa de ferramentas tangíveis de ensino de programação, foi suposta a ideia de um jogo físico com ensino de lógica de programação, que abordaria conceitos básicos para um entendimento primário de estruturas.

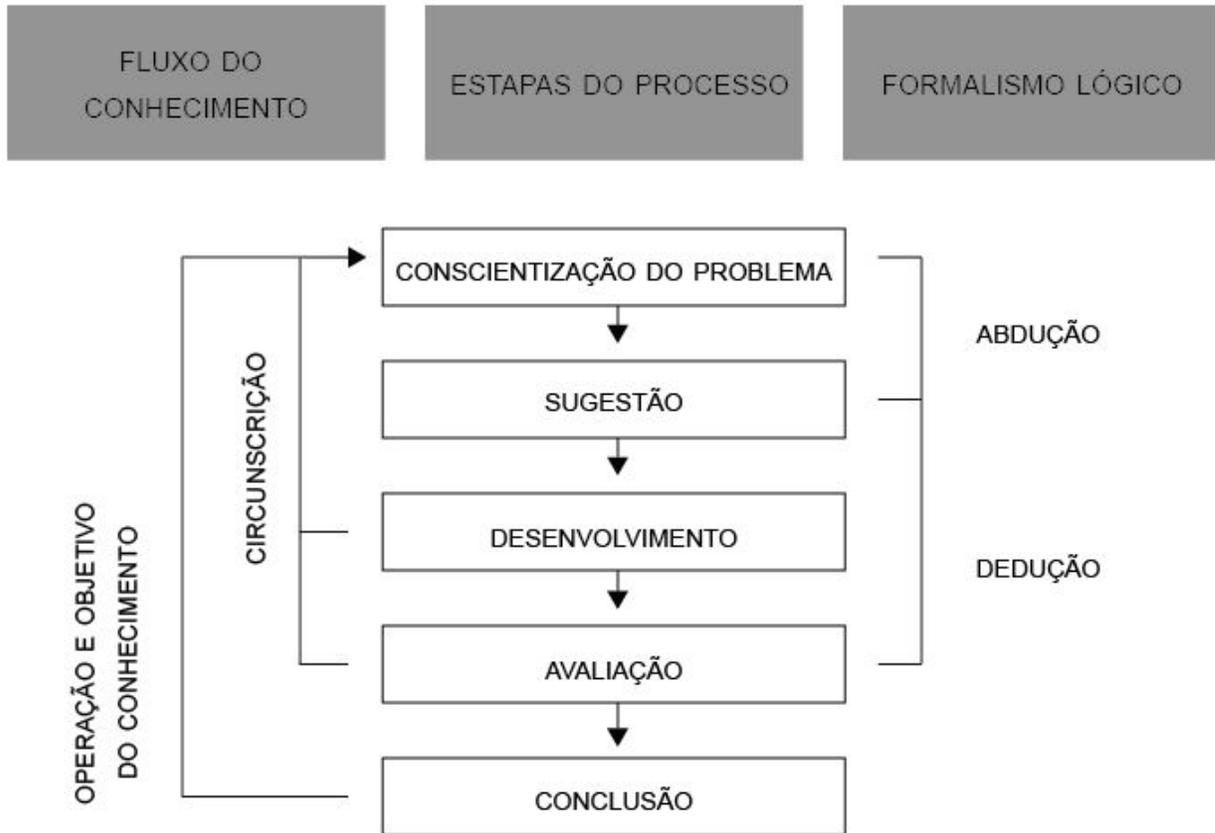
Cenário 3: Se supôs também o projeto de uma instalação para ensinar os conteúdos de forma interativa, como um tabuleiro virtual interativo. Seria interessante por ter uma abordagem mais participativa ter potencial de ser desenvolvida de forma mais visual

Cenário 4: A partir de método utilizando papel e caneta durante oficina de ensino de programação, foi pensando em expandir essa aplicação para outros métodos, o que acarretaria em uma coletânea de técnicas de aprendizagem de lógica de programação. Juntamente com isso, foi pensado em utilizar projetos interessantes feitos no curso e decompor seu processo de construção para que outras pessoas possam construir-los também. Dessa forma, o cenário 4 se trata de um livro com uma coletânea de métodos aplicados a projetos de design, com uma abordagem lúdica e visual.

1.5. Metodologia

Esta pesquisa se baseou em metodologia científica e projetual de Design Science Research (DSR). Essa metodologia surgiu a partir da lacuna no meio científico com pesquisas que tem como objetivo "prescrever soluções e métodos para resolver determinado problema ou projetar um novo artefato" (DRESH et al, 2015, p.52) O intuito é aproximar teoria e prática no meio científico. A presente pesquisa foi conduzida a partir do processo descrito por Vijay Vaishnavi e Bill Kuechler em 2004, proposto como *design cycle*.

Figura 4- Diagrama de *design cycle* por Vijay Vaishnavi e Bill Kuechler (2004)



Fonte- DRESCH et al, 2015, p.79 adaptado

Na figura 4 observa-se que o processo se inicia com a conscientização do problema, nesta fase é onde se identifica e compreende o problema, também se avalia a performance necessária para o sistema em estudo. Em seguida, elaboram-se possíveis soluções para o problema definido, isso ocorre através da criatividade e de conhecimentos prévios do pesquisador, logo, se trata de um pensamento abduutivo. Em seguida, há a fase de desenvolvimento, onde um dos artefatos propostos nas hipóteses será desenvolvido com o intuito de solucionar o problema (DRESH et al, 2015).

Se os resultados forem adequados para a solução do problema o processo continua e segue para a fase de avaliação. Porém, se os resultados não forem adequados durante a fase de desenvolvimento ou de avaliação o pesquisador pode

voltar a fase inicial e recomeçar o ciclo. Isso caracteriza um processo de circunscrição, que é útil para mostrar as decisões tomadas pelo pesquisador até chegar a um resultado final. Por fim, na conclusão são apresentados os resultados obtidos, se for observado que a própria conscientização do problema não foi suficiente para obter resultados adequados o ciclo pode começar novamente e a pesquisa contribui revelando lacunas existentes na teoria desenvolvida (DRESH et al, 2015).

No contexto desta pesquisa, este método foi aplicado da seguinte forma: na conscientização do problema, foi feita pesquisa bibliográfica da relação entre design e programação e uma busca de ferramentas de ensino de programação e teorias pedagógicas de aprendizado que pudessem auxiliar no processo de ensino de lógica de programação. Concomitantemente foram elaborados os objetivos da pesquisa.

No decorrer da elaboração do problema surgiram possíveis soluções, a partir desses dados foi escolhida a ideia mais adequada para ser desenvolvida e testada. O desenvolvimento ocorreu de duas formas: através da aplicação de questionários, um para analisar o perfil de alunos do curso de design da Universidade Federal do Ceará (UFC) em relação à programação e verificar se havia interesse e quais as dificuldades. Outro questionário como aplicação de um formulário para analisar resultados obtidos em um teste de ferramenta de ensino aplicado em oficinas de tipografia generativa no curso, com o intuito de investigar os resultados da ferramenta.

A partir dos resultados obtidos, optou-se por aprofundar o desenvolvimento da ideia testada através da análise de similares. Nesta fase ocorreu a análise de metodologias de ensino em livros de programação e ainda de outros assuntos para se compreender como este conteúdo é ensinado e quais os pontos fortes e fracos de cada publicação para a criação de requisitos do projeto proposto nessa pesquisa. Na tabela 1 está descrito o cronograma da pesquisa.

Tabela 1 - Cronograma de pesquisa

Cronograma de Pesquisa											
	Abril		Maio				Junho				
Objetivos											
Estado da arte											
Fundamentação teórica											
elaboração de hipótese											
estudo de caso											
levantamento de dados											
Conclusão											
Revisão											

■ conscientização do problema
 ■ sugestão
 ■ desenvolvimento
 ■ avaliação

Fonte - criada pela autora

1.6. Estrutura

Este trabalho é constituído de dois estudos, um da relação de design com código e outro do processo de aprendizado deste. No segundo capítulo busca-se compreender como design e código se relacionam para expandir as possibilidades de processos e onde se encontram as dificuldades de assimilação para iniciantes.

O terceiro capítulo aborda o processo de aprendizagem, onde são apresentados métodos para tornar o ensino de lógica de programação, um conteúdo abstrato, em algo mais prático e atrativo visualmente. Tais métodos têm o objetivo de criar estratégias para deixar o conteúdo mais concreto com o apoio de estratégias construcionistas e construtivistas e também utilizar uma abordagem visual de processo, descrevendo também o papel do designer como gestor de complexidade.

No quarto capítulo, descreve-se o estudo de caso feito com alunos do curso de Design da Universidade Federal do Ceará (UFC), o qual se baseou em um teste de método de aprendizado de lógica de programação aplicado em oficina de

tipografia generativa na Universidade Federal do Ceará (UFC), concomitante a isso foi feito um levantamento de dados com alunos de design da universidade para compreender quais aspectos precisam ser melhorados para que o aprendizado possa ser mais efetivo.

Por fim, no quinto capítulo foi feita uma análise de metodologia de livros e materiais focando nas estratégias de ensino utilizadas, para, a partir disso, elaborar uma lista de parâmetros que guiem a elaboração da ferramenta proposta por esta pesquisa.

2. RELAÇÃO DO DESIGN COM PROGRAMAÇÃO

2.1. Introdução

Este capítulo tem como objetivos apresentar informações relacionados às linguagens de programação, definir o conceito de algoritmo e definir as relações do designer com o código e quais as visões e aplicações da programação na perspectiva do design. Também serão apresentadas categorias de aplicações com processos que criam abordagens criativas baseadas em código e por fim quais as dificuldades nessa relação de design com a programação.

2.2. Conceitos de linguagem de programação

A linguagem de programação existe como um intermédio entre a linguagem binária do computador e a linguagem humana. É necessário que as linguagens de programação sejam diferentes da linguagem humana, para que o computador possa interpretar informações sem ambiguidade.

Escrever em uma linguagem humana permite que o autor utilize a ambiguidade das palavras e tenha grande flexibilidade para construir frases. Essas técnicas permitem múltiplas interpretações de um único texto e dão a cada autor uma voz única. Cada programa de computador também revela o estilo do seu autor, mas existe muito menos espaço para ambiguidade.⁶ (REAS; FRY, 2007, p. 17)

Cada linguagem de programação tem um conjunto de termos e regras próprios para sua utilização, como uma gramática, que constitui a sintaxe da linguagem. Cada tipo de linguagem propicia diferentes modos de uso, de acordo com

⁶ *Writing in a human language allows the author to utilize the ambiguity of words and to have great flexibility in constructing phrases. These techniques allow multiple interpretations of a single text and give each author a unique voice. Each computer program also reveals the style of its author, but there is far less room for ambiguity.*

seus *affordances*⁷ e limitações, culminando em diferentes resultados (REAS; MCWILLIAMS, 2010).

Existem vários tipos de linguagem de programação com paradigmas e aplicações específicas. Programação procedural e orientada a objetos são exemplos de paradigmas que possuem estruturação do código diferenciados. A primeira possui uma execução mais sequencial baseada em instruções lineares e funções. A segunda possui definições de objetos com atributos e funções específicas que se relacionam de forma não linear. Quanto às aplicações, as linguagens podem ser utilizadas para formatação (ex: html para criação de sites), consultas a banco de dados (ex: SQL), desenvolvimento de programas (ex: C, C++, Java, etc.), entre outras.

Existem diversas iniciativas com o objetivo de tornar o pensamento da lógica computacional mais acessível para pessoas que não são profissionais da área de computação. O LOGO e o Scratch desenvolvidos no MIT, são exemplos dessas iniciativas para crianças, com abordagem pedagógica baseada no construcionismo, pois possuem interação aluno-objeto mediada por uma linguagem de programação (VALENTE, 1997), e tem linguagem simples e direta. Também, para adultos, há programas com aplicações relacionadas à arte, baseados em abordagem visual de linguagem de programação, como o Pure Data para som, o Max para arte generativa em geral e o App Inventor para criação de aplicativos. O sistema visual destes programas baseado em blocos que se conectam, Reas (2010) explica a analogia dessas linguagens com cabos.

[...] foram influenciados pela maneira como som é construído usando cabos de conexão ligados a sintetizadores analógicos. Cabos virtuais, representados como linhas na tela, são usados para conectar módulos de programação juntos para definir o software. Linguagens de programação visual facilitam a geração de imagens e sons com filtros, mas elas são frequentemente incômodas para escrever códigos longos e complicados. Por exemplo, O programa Max é

⁷ Affordance, segundo Donald A. Norman em *Emotional Design*, é o atributo de um objeto que convida a realizar determinada ação, um exemplo é um botão de mouse que encoraja à ação de clicar.

escrito na linguagem de texto C++, não em uma linguagem visual de programação .⁸ (REAS; MCWILLIAMS, 2010, p.19)

O presente trabalho pretende utilizar a linguagem processing para desenvolvimento de projetos de design. *Processing* é uma linguagem e um programa criados por Ben fry e Casey Reas em 2001 e continua se desenvolvendo até o período atual. Sua sintaxe, termos e regras, é similar à linguagem Java. Segundo Bohnacker (2012, p.168) o principal objetivo do *Processing* é proporcionar acesso simples à programação para as pessoas orientadas visualmente. O seu funcionamento se baseia na criação de "*sketches*", traduzido como rascunhos. Isso se dá pelo fato dele proporcionar visualização rápida de qualquer alteração no código. Outro ponto importante é que existem várias bibliotecas de livre acesso com códigos que executam funções específicas e podem incrementar um código existente.

Este programa não tem o intuito de ser utilizado para a execução de um projeto já determinado, mas a criação através do programa, o utilizando como uma linguagem de expressão (REAS; MCWILLIAMS, 2010). Reas e Mcwilliams explicam a diferença dessas duas aplicações na citação abaixo.

O uso do software nas artes pode ser separado em suas categorias: produção e concepção. Na primeira categoria, o computador é usado para produzir uma forma preconcebida; na segunda, o computador participa do desenvolvimento da forma. [...] nós estamos primeiramente interessados no último. (É importante notar que essa distinção não implica um julgamento de valor mas impacta nos tipos de formas que são criadas.)⁹(REAS; MCWILLIAMS, 2010, P 25).

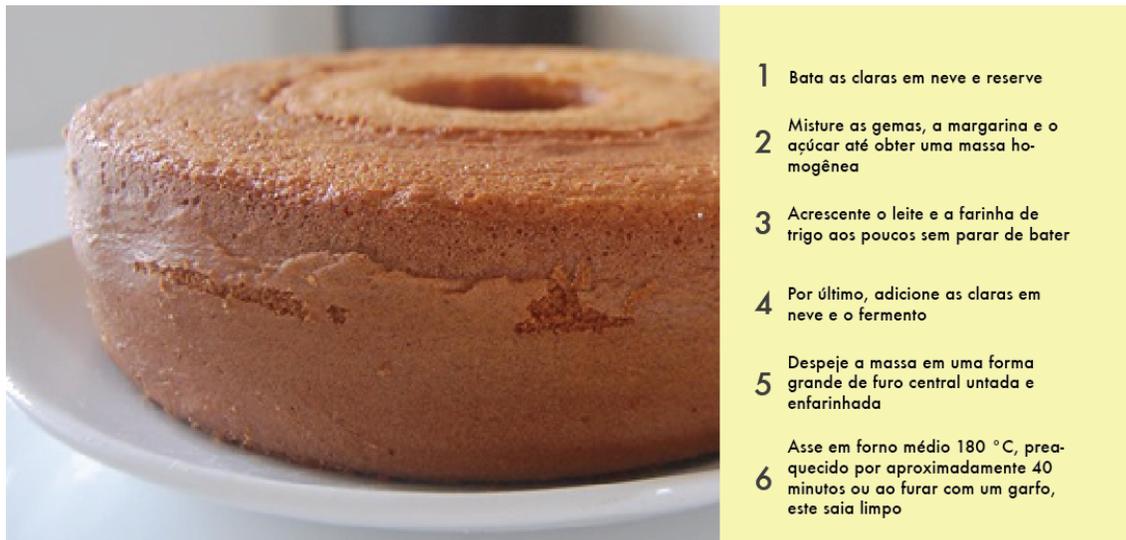
⁸ *were influenced by the way sounds are constructed using patch cables attached to analog synthesizers. Virtual cables, represented as lines on screen, are used to connect programming modules together to define the software. Visual programming languages make it easy to generate and filter images and sounds, but they are often too cumbersome for writing long, complicated programs. For example, the Max program is written in the text programming language C++, not a visual programming language.*

⁹ *[...]The use of software in the arts can be separated into two categories: production and conception. In the first category, the computer is used to produce a preconceived form; in the second, the computer participates in the development of the form. [...] we're primarily interested in the latter. (It is important to*

2.3. O que é algoritmo

Algoritmo é uma sequência de passos bem definidos para resolver um problema, ou seja instruções. Uma receita de bolo pode ser classificada como um algoritmo por ser uma sequência de instruções. A figura 2.2 ilustra um bolo e sua receita, como exemplo de algoritmo.

Figura 5 – modo de preparo de bolo fofo



Fonte – elaborada pela a autora, conteúdo de site de receitas¹⁰

Reas (2010) define quatro qualidades dos algoritmos: existem várias maneiras de escrever um algoritmo; para escrevê-lo é necessário ter os conhecimentos prévios que serão utilizados na sua construção; um algoritmo requer decisões de onde começar, para onde ir, etc; um algoritmo complexo deve ser quebrado em partes modulares, assim como uma receita de bolo é quebradas em passos.

note that this distinction does not imply a value judgment but does impact the types of forms that are created.

¹⁰ Disponível em: < <http://www.tudogostoso.com.br/receita/29124-bolo-simples.html> >. Acessado em 01 Jul. 2017.

2.4. Aplicação no design

A relação da linguagem de programação no design pode se dar desde a criação de um projeto a ser executado por um desenvolvedor, até ser parte do processo de design. Das duas formas é interessante que o designer tenha conhecimentos básicos de programação para saber o que pode ser feito a partir da programação de computadores e criar a partir dela.

A correlação de design com código acontece mesmo antes do uso de um ambiente software de programação, ao utilizar scripts, em programas como photoshop, illustrator, after effects etc., para automatizar ou criar uma lógica de ação, o designer já está pensando computacionalmente. Esse pensamento também pode ocorrer na criação de interfaces, onde é interessante conhecer os limites e potenciais do desenvolvimento daquilo que se produz visualmente.

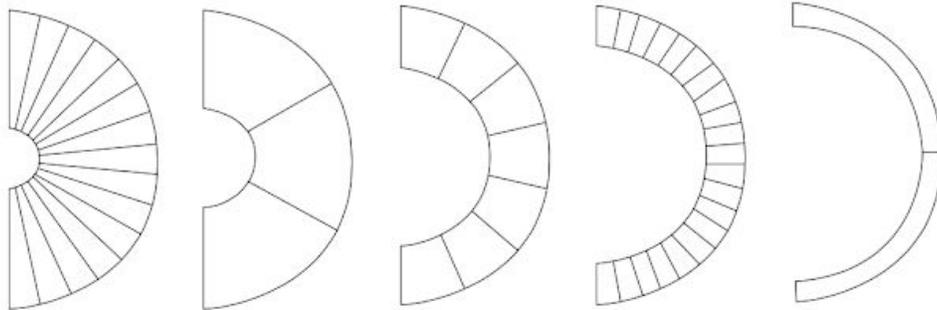
No contexto desta pesquisa, propõem-se outras formas de lidar com programação para criação de artes visuais, através de design paramétrico e design generativo.

No design paramétrico, o designer não faz escolhas voltado para o objeto final, mas sim cria uma matriz de possibilidades com inúmeros resultados diferentes. Reas descreve parâmetro como

[...] valor que tem um efeito no resultado de um processo. Isso pode ser algo tão direto quanto a quantidade de açúcar em uma receita (...). Parâmetros descrevem, codificam e quantificam as opções e as limitações em jogo em um sistema. Uma limitação comum pode ser o orçamento disponível para um projeto, enquanto uma opção de configuração pode ser o controle de cor, tamanho, densidade ou material.(2010, pag.95)¹¹

¹¹ [...]value that has an effect on the output of a process. This could be something as straightforward as the amount of sugar in a recipe(...). In the context of architecture and design, parameters describe, encode, and quantify the options and constraints at play in a system. A common constraint might be the budget available for a project, while a configuration option might control color, size, density, or material.

Figura 6 – Exemplo de parâmetro de número de segmentos.



Fonte –REAS,2010

Design generativo é um processo de criação onde não há interação direta entre o designer e o objeto, o processo ocorre através de um sistema generativo (FISHER, 2001). Fisher define um sistema generativo como "uma configuração baseada em definições abstratas de possíveis variações de design capazes de exibir ou produzir produtos de design (ou elementos de produtos de design)."¹² (2001).

Também está ligado à aleatoriedade criando resultados únicos, de acordo com o professor Celestino Soddu "Design generativo é um processo morfogenético que usa algoritmos como um sistema não-linear para resultados únicos infinitos e sem repetição executados por um ideia-código, como na natureza." ¹³(1992)

¹² *a set-up based on abstract definitions of possible design variations capable of displaying or producing design products (or elements of design products)*

¹³ *Generative Design is a morphogenetic process using algorithms structured as not-linear systems for endless unique and un-repeatable results performed by an idea-code, as in Nature.* Disponível em: < http://www.soddu.it/index_lab.htm >. Acessado em 01 Jul. 2017.

Figura 7 - trabalho de Enzo Henze: *Red Ambush*



Fonte - *Fast Co Design*¹⁴

Um exemplo de design generativo é o trabalho *Red Ambush* de Enzo Henze, onde há regras de algoritmo para que o computador execute o desenho seguindo padrões orgânicos e gerando alta complexidade.

O processo de design usa de conhecimento tácito, ou seja, conhecimento que não pode ser descrito e ocorre na prática. Visto isso, há uma “conversa” entre o designer e o objeto projetado, numa relação de “ver, mudar e ver”¹⁵. Segundo Schön (1992), cada alteração no objeto resulta em mudanças em diferentes domínios, que são analisadas pelo designer. Ao observar padrões é feito um julgamento se a mudança, no contexto geral, foi positiva ou não. Assim, o designer descobre consequências da mudança para além do que foi testado e vai organizando o objeto e sua complexidade.

Pode-se observar que o código tem potencial de criação, quando o designer o utiliza como um material, com características próprias pois: lida com grande número

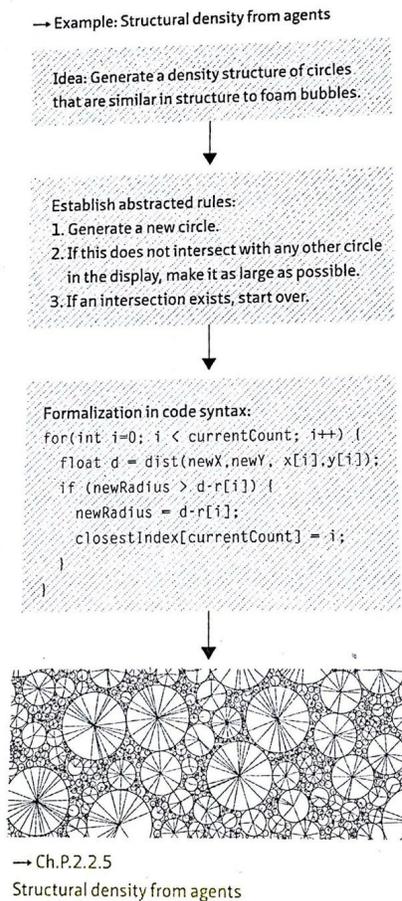
¹⁴ Disponível em: < <https://www.fastcodesign.com/1670547/9-groundbreaking-examples-of-generative-design> > . Acessado em 01 Jul. 2017.

¹⁵ “*Seeing-moving-seeing*” em tradução livre.

de informações, gera aumento de complexidade, parametriza características do objeto, automatiza comandos, têm fatores de aleatoriedade etc. A partir disso, o processo e os resultados mudam.

Segundo Bohnacker (2012), o primeiro passo é transformar uma ideia vaga em uma receita, dividir em pequenos passos, definir as regras abstratas, formalizar na sintaxe do código e obter o resultado visual. Na figura 8 ele descreve um exemplo que começa com a ideia vaga "gerar uma estrutura de densidade de círculos", estabelece as regras abstratas "1. Desenhe um círculo" " 2. Se esse círculo não tiver interseção com nenhum outro no display, faça esse círculo o maior o possível"," 3. Se esse círculo tiver interseção com outro, comece de novo". Então formaliza as regras no código e obtém os resultados visuais.

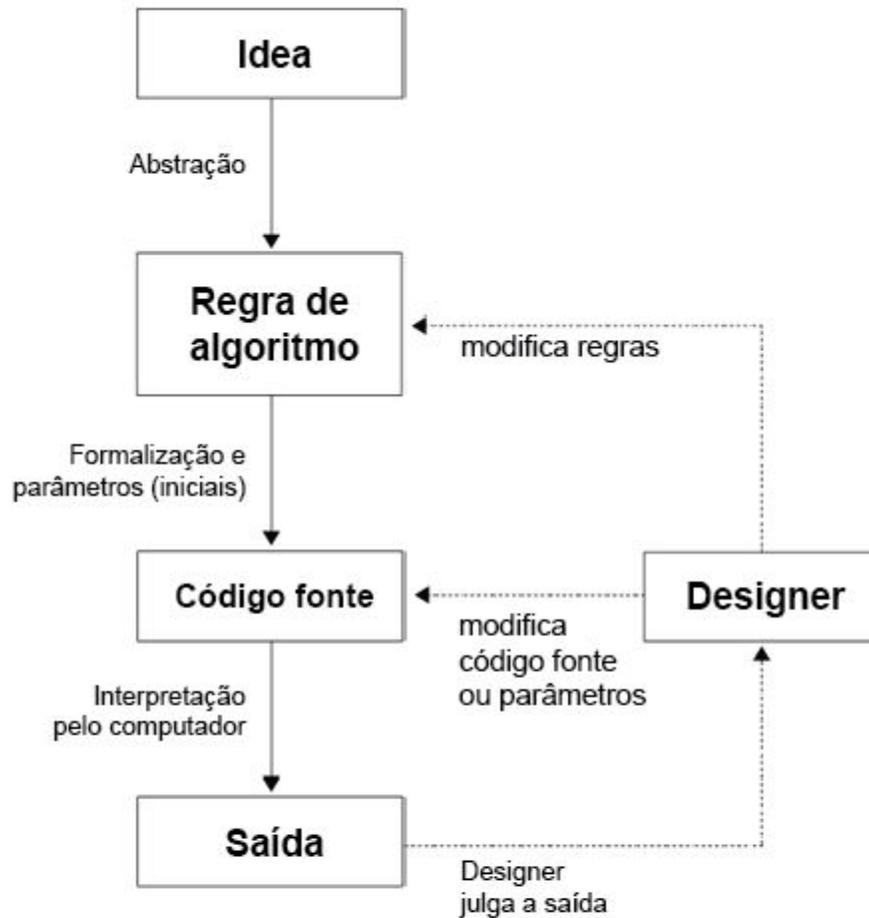
Figura 8 - exemplo de processo de criação de código de design generativo.



Fonte - retirado de BOHNACKER, 2012, p.461

A partir dos resultados visuais as imagens devem ser analisadas e interpretadas pelo designer para que possam ser melhoradas. Através de cada interação entre o designer e o resultado o refinamento vai aumentando para se chegar ao resultado final. A figura 9 descreve o processo de design generativo e a atuação do designer no processo: criando regras, julgando dos resultados e intervindo no código.

Figura 9 – diagrama do processo de design generativo



Fonte – BOHNACKER, 2012 p.461, adaptado.

No design generativo existem as possibilidades de "emergência", "simulação" e "ferramenta". O conceito de emergência trata de criar resultados que não são óbvios e predeterminados onde através de regras simples há resultados altamente complexos. Por sua vez, a simulação de processos naturais é o uso de

fórmulas de física, por exemplo, para imitar comportamentos naturais virtualmente. (2012).

A "ferramenta" se trata da possibilidade do designer criar seus próprios programas, amplia o universo de possibilidades de criação, explorando novos caminhos que não poderiam ser executados usando um software existente. Mesmo ferramentas simples podem tornar um projeto bastante complexo e com resultados inovadores. A figura 10 ilustra um exemplo de software desenvolvido pelo o estúdio de design *Okdeluxe* de Londres com parceria com estúdio de design de interação. Os controles do software permitem diferentes configurações, possibilitam salvar 10 estados e criam uma animação entre eles, baseando-se em regras de campo de força e agrupamento.

Figura 10 – COP15 - software de identidade generativa



Fonte – site da okdeluxe. ¹⁶

¹⁶ Disponível em: < <http://www.okdeluxe.co.uk/cop15/> >. Acessado em 25 Jun. 2017.

Figura 11 – COP15 - exemplos de resultados de identidade generativa



Fonte – site da okdeluxe.¹⁷

Existem diversas aplicações no design generativo "[...] design generativo já é usado em vários campos, como base para visualização de informação, obra de arte, instalações de mídia, modelos arquitetônicos, video clips, fontes e até mesmo design de lâmpadas."¹⁸ (BOHNACKER, 2012, p.464).

A seguir serão apresentadas algumas áreas de aplicação no campo do Design generativo que serão contempladas na proposta do livro *CodeBook*.

Tipografia Generativa

A figura 12 tem exemplo de tipografia generativa criada no processing por Hyper Glu chamada "*Wires Sparse*", "Eu criei um sistema autônomo que toma suas próprias decisões artísticas."¹⁹ O resultado foi gerado através de padrões matemáticos e linguagem de programação e Consiste em 5000 letras, dígitos e símbolos.

¹⁷ Disponível em: < <http://www.okdeluxe.co.uk/cop15/> >. Acessado em 25 Jun. 2017.

¹⁸ Generative design is already used in many fields, as the basis for information visualization, artwork, media installations, architectural models, video clips, fonts, and even generative lamp design.

¹⁹ I create autonomous systems that take their own artistic decisions. Disponível em: < <http://theinspirationgrid.com/generative-typography-project-by-hyper-glu/> >. Acessado em 25 Jun. 2017.

Figura 12 – exemplo de letras generativas feitas por Hyper Glu.



Fonte – site the inspiration grid²⁰.

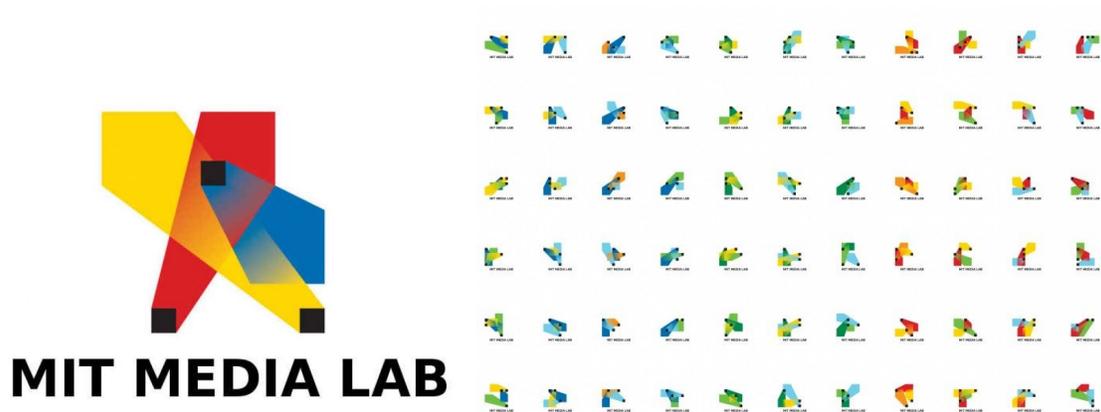
Identidade generativa

Identidade do MIT Media LAB para os 25 anos do laboratório, construída no Processing a partir de processo generativo, cada membro do laboratório recebeu o uma versão própria do logo e também o código fonte para reproduzi-la, expressando diversidade do laboratório que abrange diferentes áreas do conhecimento, contando com engenheiros, designer, artistas e cientistas. O logo tem três focos de luz fazendo uma interseção que podem ser organizados em 40.000 formas e com 12 combinações de cores diferentes através do algoritmo. O projeto foi desenvolvido pelos designers E Roon Kang e Richard The, segundo Richard "cada uma das três formas representa uma contribuição individual, a forma resultante representa o resultado desse processo: uma constante redefinição do que mídia e tecnologia significam hoje."²¹

²⁰ Disponível em: < <http://theinspirationgrid.com/generative-typography-project-by-hyper-glu/> >. Acessado em 25 Jun. 2017.

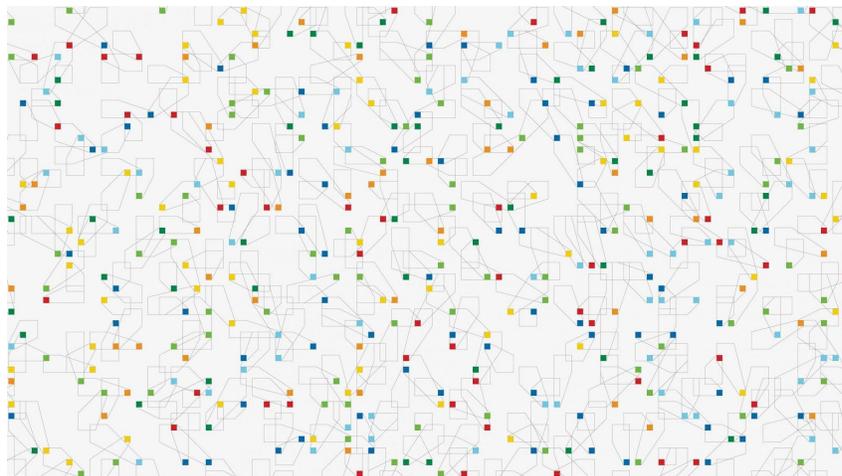
²¹ *Each of the three shapes stands for one individual's contribution, the resulting shape represents the outcome of this process: A constant redefinition of what media and technology means today.* Disponível em: < <http://www.rt80.net/medialab> > Acessado em 25 Jun. 2017.

Figura 13 – Logo do MIT Media LAB e variações do logo feita para cada um dos componentes do laboratório.



Fonte –site grapheine²²

Figura 14 - exemplo de aplicação da identidade



Fonte - site rt80²³

²²Disponível em: < <https://www.grapheine.com/en/branding-en/nouvelle-identite-visuelle-generative-bordeaux-metropole> > Acessado em 25 Jun. 2017.

²³ Disponível em: < <http://www.rt80.net/medialab> > Acessado em 25 Jun. 2017.

Arte generativa

Peça produzida a partir de fotografia do Reitor Martins Filho da Universidade Federal do Ceará pelo Laboratório de Fotografia Tátil, feita do processo de manipulação de fotografia com criação aleatória de pontos no processing e executada na máquina fresadora da UFC. Esta peça foi incorporada ao acervo do memorial UFC.

Figura 15 - processo de criação de da arte generativa a partir da fotografia do Reitor Martins Filho.

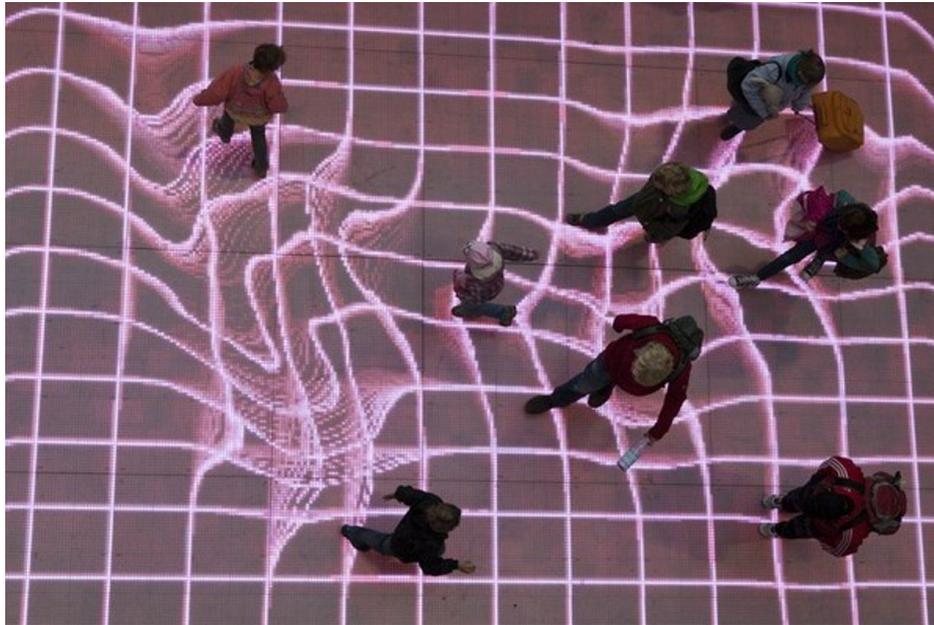


Fonte - Laboratório de Fotografia Tátil da UFC.

Instalação Interativa

A figura 16 demonstra exemplo de instalação interativa criada por German Sven Beyer, a instalação é uma instalação de LED que responde em tempo real aos passos das pessoas e se chama "Onskebronn", e foi apresentada na estação de trem *Hauptbahnhof* em Berlim.

Figura 16 - Instalação "Onskebronn" em Berlim.



Fonte - site *my modern met*.²⁴

Visualização de informações

Segundo o economista chefe do Google Hal Varian "A habilidade de tomar dados - ser capaz de entender, processar, extrair valor disso, visualizar, comunicar - isso vai ser uma habilidade de enorme importância nas próximas décadas [...]"²⁵

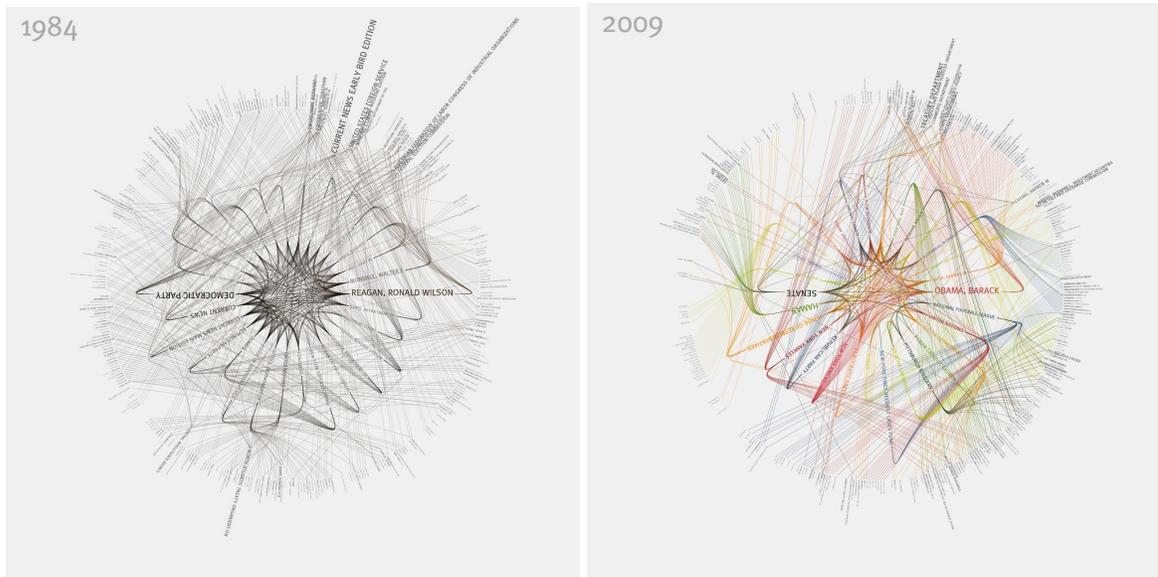
Um exemplo de visualização de dados está na figura 17 está um série criada por Jer Thorp que mostra as relações entre os assuntos mais citados no jornal *Times* em anos diferentes.

²⁴ Disponível em: < <http://mymodernmet.com/trippy-digital-pavement-at/> >. Acessado em 01 Jul. 2017.

²⁵ " The ability to take data – to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it's going to be a hugely important skill in the next decades [...]" Disponível em: < <https://flowingdata.com/2009/02/25/googles-chief-economist-hal-varian-on-statistics-and-data/> >.

Acessado em 01 Jul. 2017.

Figura 17 - Visualização de dados por Jer Thorp



Fonte - site newhive²⁶

2.4.1. Dificuldades com o código para designers

Por não se tratar de uma linguagem natural, a linguagem de programação precisa ser formalizada, sequenciada, enumerada em uma lógica diferente de uma linguagem humana, o que gera dificuldades de compreensão por pessoas que não são da área de computação. Aprender programação para designers pode ser um grande obstáculo, pois tradicionalmente não há grande conexão entre as duas áreas.

"Não deve ser esquecido que designers que programam ainda são a absoluta minoria, Isso tem razões históricas e culturais - universidades e faculdades, por exemplo, não tem a tradicionalmente proporcionado treinamento em ambos

²⁶ Disponível em: < http://newhive.com/samssong/2015_04_26-1 >. Acessado em 01 Jul. 2017.

design e programação, apesar disso, essa separação rigorosa está sendo abolida em algumas situações." ²⁷(BOHNACKER, 2012, p.465)

Resnick (2009) cita alguns dos problemas na educação tradicional de código para iniciantes: não conseguem entender a relação entre os elementos, ou seja, a sintaxe de programação; o aprendizado de código não é conectado com os interesses dos alunos e o seu contexto; o contexto de aprendizado não dá abertura para o entendimento de erros e de exploração maior quando o código funciona.

2.5. Conclusões

De acordo com o que foi analisado conclui-se que existem diversas possibilidades de aplicações no design com o uso de programação. Também nota-se que existe uma abordagem própria para pensar processo de design generativo com programação. Para compreender melhor como esse aprendizado deve existir serão abordados no próximo capítulo métodos de ensino que podem ser aplicados neste contexto.

²⁷ *It should not be forgotten, however, that designers who program are still the absolute minority. This has historical and cultural reasons- universities and colleges, for example, have not traditionally provided training in both design and programming, although this strict separation is being abolished in some situations.*

3. TEORIAS PEDAGÓGICAS E FERRAMENTAS DE APRENDIZAGEM

3.1. Introdução

Este capítulo aborda teorias e práticas pedagógicas. A partir desta pesquisa, fazendo-se uso de metodologias construtivistas e construcionistas criam-se parâmetros para a proposição de um ensino de código para designers.

O método de ensino adotado por cursos de ciência da computação passa por um processo lento com o objetivo de construir uma base sólida focada nas estruturas lógicas. Para tanto, são utilizados pequenos exercícios onde os resultados práticos não tem grande importância. Ao final do curso os alunos são capazes de desenvolver qualquer algoritmo com facilidade devido ao grande número de disciplinas e à progressão de complexidade ao longo delas.

Com os avanços tecnológicos, os cursos de artes visuais sentiram a necessidade de se apropriar desse conhecimento para obter resultados inovadores a partir de processos complexos e generativos gerados por computadores. No entanto este conhecimento exige tempo de maturação e exercício. Para o designer, o processo de aprendizado tradicional torna-se inviável, tanto pelo tempo necessário, como pelo fator motivacional de não ter resultados práticos no curto prazo. A formação do designer não exige o conhecimento profundo em programação, mas é desejável que seja um conhecedor da lógica de programação básica para ser capaz de propor soluções utilizando-se desse pensamento computacional. Para tanto, é necessária uma reflexão quanto ao método de ensino e necessidades desse público.

Para que o ensino da lógica de programação para iniciantes possa ocorrer é necessário, primeiramente, o desenvolvimento do pensamento computacional, Wing define "Pensamento computacional é processo de pensamento envolvendo formular um problema e expressar sua solução de uma forma que o computador- humano ou máquina - possa efetivamente realizar" ²⁸(WING, 2006). Este pode ser desenvolvido

²⁸ Computational Thinking is the thought processes involved in formulating a problem and expressing its solution in a way that a computer—human or machine—can effectively carry out.

através de diversas técnicas, com ou sem o uso do computador. A essência do pensamento computacional é a abstração e automatização. Grover define abstração como “redução de informações e detalhes para se concentrar em conceitos relevantes para a compreensão e resolução de problemas” (Grover, 2013).

Considerando o designer como um profissional que se utiliza de conhecimento tácito para projetar (Schön, 1992), é interessante que a abordagem pedagógica para alunos de design também se desenvolva de maneira prática. Com o objetivo de propor metodologias de ensino mais participativas que possam auxiliar os alunos a terem um aprendizado mais efetivo serão abordadas a seguir teorias pedagógicas que servirão de base para a proposta do projeto.

3.2. Construcionismo

A teoria construcionista de Papert nasce como uma vertente do construtivismo, este, por sua vez, foi baseado na teoria de psicogenética de Piaget, sobre desenvolvimento infantil e adaptado para teorias pedagógicas. A teoria construtivista se trata, segundo Wilson, de "uma visão filosófica sobre como entendemos ou sabemos" (1996). Seu principal atributo é que a partir das interações com o ambiente ocorre o aprendizado, segundo Newby "A perspectiva construtivista descreve o aprendizado como uma mudança no significado construída a partir da experiência ." ²⁹(1996, *apud* TAM, 2000, pag.02). O construtivismo também propõe o aprendizado de forma social, onde dentro de grupos se encontra espaço para troca e validação de conhecimentos.

O construtivismo também aborda o conceito de conflito cognitivo como um estímulo para o aprendizado (1996). O conflito acontece quando algo que não pode ser entendido pelo sujeito causa uma perturbação nas estruturas conceituais existentes, ou seja, seu conhecimento prévio. Esta perturbação gera a necessidade

²⁹ The constructivist perspective describes learning as a change in meaning constructed from experience

de uma acomodação, ou seja, criação ou adaptação de estruturas conceituais para estabelecer um novo equilíbrio, de acordo com Von Glasersfeld (1989).

Em síntese, existem sete valores construtivistas primários: colaboração, autonomia pessoal, generatividade, refletividade, engajamento ativo, relevância pessoal e pluralidade (Lebow, 1993, *apud* TAM, p.6). Há também princípios que devem guiar o trabalho de um professor construtivista, entre elas está em estimular criação de teste de hipóteses pelos alunos e proporcionar que os estudante construam metáforas e relações (Brooks and Brooks , 1993, *apud* TAM, p. 5-6).

O construcionismo é fundamentado em conceitos construtivistas, ambos se baseiam em construir estruturas de conhecimento, porém o construcionismo tem como principal característica adquirir conhecimento a partir da construção efetiva de algo e avaliação dos resultados, ou seja, aprender fazendo.(PAPERT; HAREL,1991) e é baseada em experimentar, elaborar hipóteses, testar e observar resultados, (GIBBONS, 2016).

O construcionismo tem três características principais, construir conhecimento utilizando o computador como intermediário, adquirir conhecimento efetivamente construindo algo e que o resultado seja de algo de interesse do aluno, tornando assim o processo de construção motivante (VALENTE, 1997).

Na busca de ferramentas que facilitem o entendimento da lógica de programação o MIT (*Massachusetts Institute of Technology*) mostrou diversas iniciativas nesta área, principalmente para crianças. O LOGO, Scratch e o LEGO Mindstorms são projetos que se baseiam no construcionismo de Papert e foram elaborados em seu laboratório *Media Lab* no MIT, contando também com parcerias.

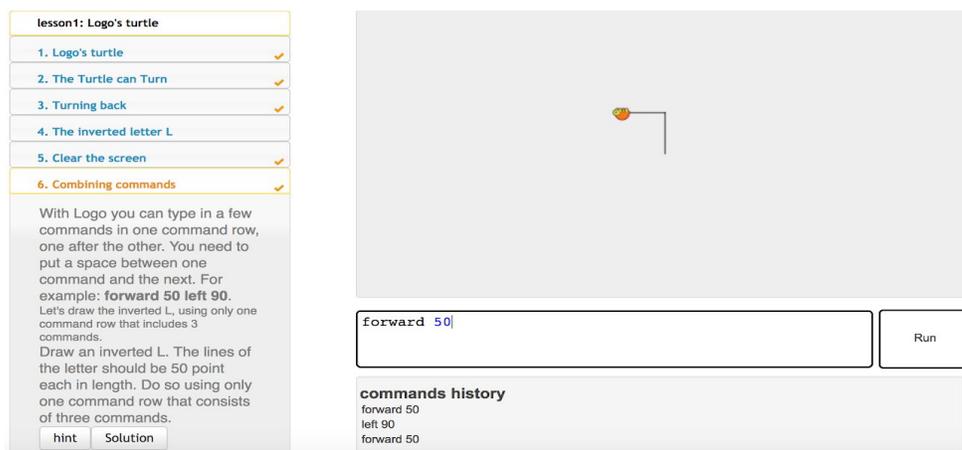
Logo

Os primeiros experimentos para aprendizado de lógica de programação para crianças *Media Lab* no MIT foram com o LOGO em 1967. Versões iniciais de LOGO utilizavam uma tartaruga robótica, depois ele foi desenvolvido virtualmente.

[...] Logo é um linguagem desenvolvida para crianças pequenas que não estudaram geometria; Permite que o usuário codifique uma forma apenas com o conhecimento de ângulos e da distinção entre esquerda e direita. Nesse programa, a criança desenha linhas direcionando o caminho de uma tartaruga numa tela. A criança imagina que ele ou ela é uma tartaruga e que move a frente, dobra à direita, move a frente, move a direita de novo e em seguida move a frente para completar um triângulo. (...) Logo promove exploração, além disso, LOGO encoraja o programador a executar o código mentalmente, que é uma habilidade útil para se desenvolver alfabetização procedural.(2010, pag.19)³⁰

A figura 18 ilustra uma tela do Logo com os comandos necessários para gerar a figura desenhada pela tartaruga.

Figura 18 – interface do LOGO



Fonte – elaborada pela autora

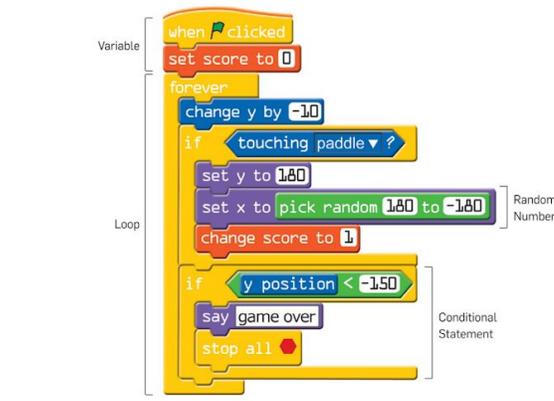
³⁰ LOGO is a language developed for young children who have not yet studied geometry; it allows the user to code a shape with only an understanding of angles and the difference between left and right. In this program, the child draws lines by directing the path of a turtle on screen. The child imagines that he or she is the turtle and moves forward, turns to the right, moves forward, turns to the right again, and then moves forward to complete the triangle. (...) LOGO fosters exploration. Additionally, LOGO encourages the programmer to run the code mentally, which is a useful skill for developing procedural literacy.

Scratch

O Scratch foi desenvolvido mais tarde no mesmo laboratório do LOGO, em 2005, e se trata de um sistema baseado em blocos de arrastar e soltar na interface. Constitui-se de blocos que representam visualmente partes do código do programa. Esses blocos adicionam forma, cor e funções. Existe nessa proposta uma grande relação de *affordance*, pois os blocos só se encaixam onde pode haver uma interação, evitando erros de sintaxe, um dos motivos pelo qual há maior facilidade de aprendizado nesse sistema do que no de linhas de texto (SHAPIRO; AHRENS, 2016).

A figura 19 ilustra um código construído com o software scratch. Nesta imagem é possível verificar o sistema de encaixes entre os comandos.

Figura 19 – Exemplo de script de jogo de ping-pong feito no Scratch



Fonte – RESNICK, 2009

LEGO Mindstorms

LEGO mindstorms foi desenvolvido no *Media Lab* em parceria com a LEGO e possui uma abordagem pedagógica baseada na robótica para o ensino de

programação. A figura 20 ilustra o kit com os componentes para a montagem de robôs e a figura 21 exemplifica um robô de corrida montado com o kit.

Figura 20 – Kit com os componentes para montagem de robôs do Lego Mindstorm



Fonte - Site do LEGO³¹

Figura 21 – Exemplo de robô de corrida montado com componentes do kit Lego Mindstorm



Fonte - Site do LEGO³²

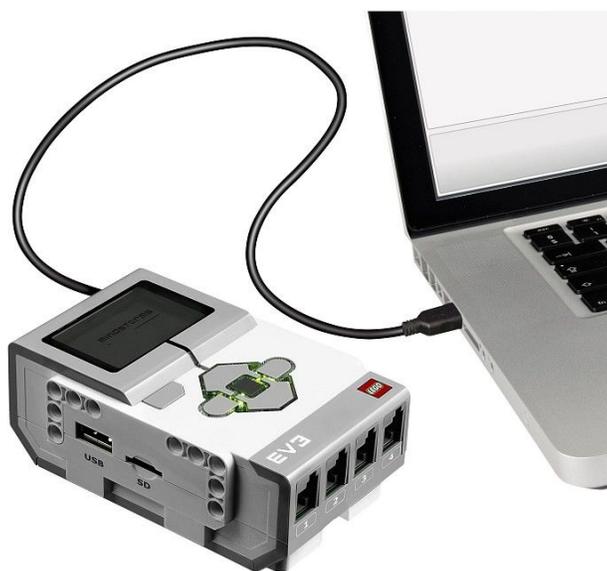
³¹ Disponível em: < <https://www.lego.com/en-us/mindstorms/products/mindstorms-ev3-31313> >. Acessado em 01 Jul. 2017.

³² Disponível em: < <https://www.lego.com/en-us/mindstorms/videos> >. Acessado em 01 Jul. 2017.

O Kit funciona combinando seus elementos com um bloco programável (ver na figura 22), onde este é conectado ao computador para a criação do código que ele executará no robô. A linguagem do código é a Robomind, que é mais simples do que a linguagem java. Podem ser adicionados sensores e motores ao robô possibilitando que eles andem, falem, segurem, pensem e lancem elementos.

Este produto é utilizado em escolas e universidades para ensinar lógica de programação e tem obtido sucesso na motivação dos alunos por ser uma maneira prática e desafiadora de aprender. Vahldick et al. (2009) atesta que a aplicação do Kit, no primeiro semestre do curso de Ciências da Computação da FURB, gerou grande interesse por parte dos alunos a partir da utilização dos robôs, e maior compreensão de estruturas de controle comparada ao semestre anterior. Essa abordagem aumentou o sucesso na formulação de soluções devido à abordagem prática com maior fator motivacional.

Figura 22 – Bloco programável conectado ao computador



Fonte - Site do klotsipood³³

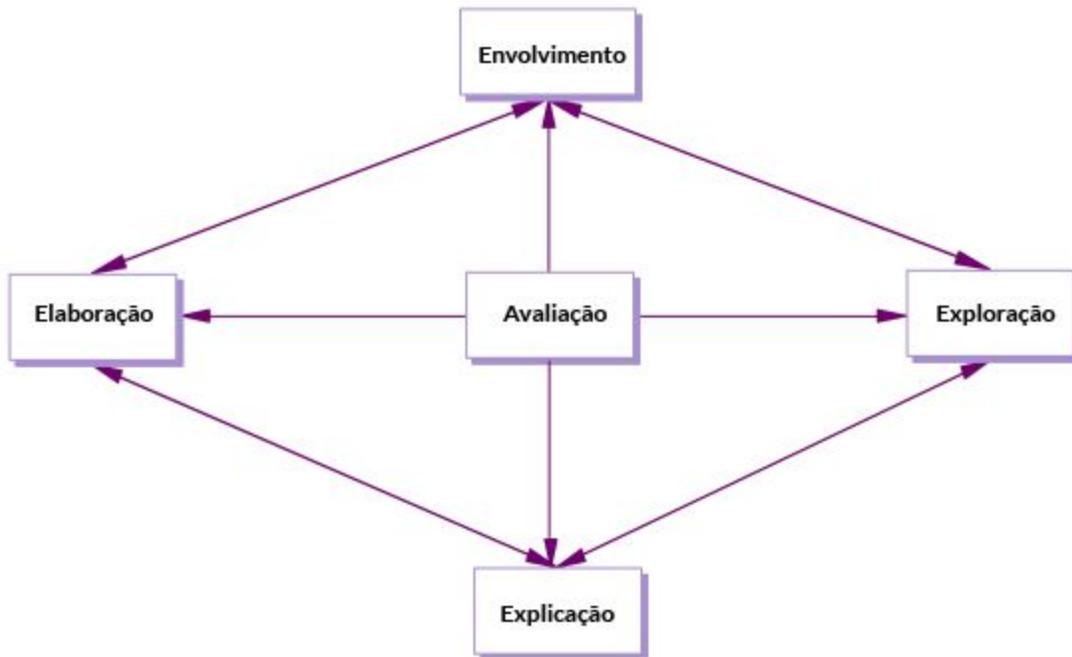
³³ Disponível em: < https://www.klotsipood.ee/index.php?id=lego-rotallietas&product_id=2274 >. Acessado em 01 Jul. 2017.

3.3. Modelo de aprendizado

A aplicação de conceitos construtivistas pode se dar através do modelo de ciclo de aprendizagem 5E de Roger Bybee , que é baseado em psicologia cognitiva e na teoria de aprendizado construtivista para ensinar ciência, podemos analisar o esquema do modelo na figura abaixo.

O ciclo começa com "Envolvimento", onde se desperta o interesse do aluno no assunto e revela o que ele já sabe sobre o conteúdo. "Exploração" é a fase de elaboração de hipóteses, onde o aluno observa, questiona, testa, momento em que ele é mais ativo no processo. Em "Explicação" são apresentados os conceitos que foram explorados anteriormente clarificando dúvidas causadas pela exploração. Aqui o professor passa a ser agente principal do processo. O ciclo continua com "Elaboração", quando são aplicados os conceitos aprendidos, desenvolvendo melhor o que tinha sido explorado. E por fim, "Avaliação" é a etapa que ocorre durante o processo e é apresentada de um modo alternativo, observando materiais que evidenciam o aprendizado do aluno, como mapas conceituais, modelos físicos etc., porém também deve haver uma forma tradicional de avaliar o aprendizado, como prova, redação etc. (DURAN; DURAN 2004)

Figura 23- O método de ciclo de aprendizagem 5E



Fonte- DURAN; DURAN 2004, p. 52 (adaptado) .

3.4. Design como ferramenta para facilitar o ensino

Para Moraes (2010) o designer se encarrega de criar inter-relações em informações desconexas pois "ao atuarem em cenários múltiplos, fluidos e dinâmicos lidam de igual forma com o excesso de informações disponíveis (MORAES, 2010, p.13)". Para que isso ocorra o design entra como gestor da complexidade e tem o papel no contexto de criar ferramentas, instrumentos métodos que ajudem na compreensão da complexidade.

Essa gestão da complexidade pode ocorrer através do conceito de interface entre o usuário e o conteúdo, é papel do design construir essa relação, considerando, dessa forma, o design como uma ferramenta para o metabolismo cognoscitivo³⁴ da informação (BONSIEPE, 2000). Gui Bonsiepe (2000) defende que o conhecimento deve ser comunicado e compartilhado, o design entra como mediação para que isso ocorra fazendo com que o conhecimento deixe de ser abstrato e passe a ser acessível e experimentável através de interface criada para a apresentação do conteúdo, ou seja, cria-se um intermédio entre o conteúdo abstrato e o leitor. (2000).

Os designers devem localizar-se exatamente nesse ponto, pois têm – ou supõe-se que tenham – a experiência de sintetizar a complexidade do saber e ajudar a apresentar a informação desenhando a interface entre a fonte de informação, os dados e o leitor. (BONSIEPE, 2000, sem página definida)

Esse intermédio do conteúdo e do leitor que o design propicia, pode ser utilizado no processo para adquirir conhecimento, este ocorre ao se transformar dados crus em informação, por conseguinte, informação em conhecimento. Este processo ocorre quando "o usuário internaliza, interpreta e utiliza a informação, ou seja, traduz a informação em ação." (BONSIEPE, 2000). Dessa forma, a prática da ação constitui o conhecimento.

Para compreender a relação do processo de conhecimento e o pensamento visual considera-se a percepção visual como cognição, segundo Arnheim (1969). O autor afirma que "as operações cognitivas chamadas pensamento não são privilégio de processos mentais acima ou além da percepção mas ingredientes essenciais da própria percepção"³⁵(ARNHEIM, 1969, p.13), o autor define essa cognição como "Exploração ativa, seleção, compreensão de elementos essenciais, simplificação, abstração, análise e síntese, conclusão, correção, comparação, resolução de

³⁴ Cognição é o ato ou processo da aquisição do conhecimento que se dá através da percepção, da atenção, associação, memória, raciocínio, juízo, imaginação, pensamento e linguagem. (wikipedia)

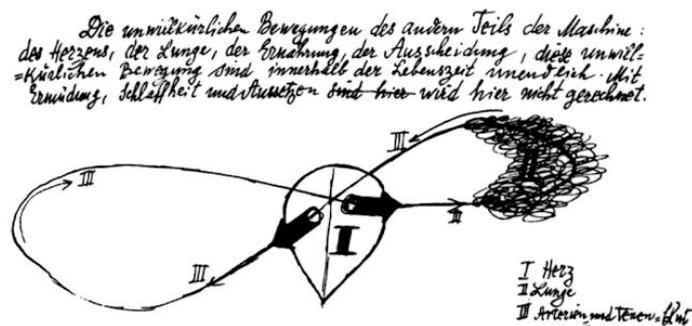
³⁵ the cognitive operations called thinking are not privilege of mental processes above or beyond perception but essential ingredients of perception itself.

problemas, além de combinar, separar, colocar em contexto" ³⁶(ARNHEIM, 1969, p.13).

Arnheim (1969) descreve que ilustrações ensinam ao transformar um processo complexo numa elaboração de princípios simples. Este processo de representação está relacionado com a capacidade de encontrar analogias, para isso é necessário encontrar um "valor funcional" incorporado nos objetos representados. ARNHEIM (1969) cita como exemplo Leonardo da Vinci " Ele viu cada músculo, osso, ou tendão como moldado para o seu propósito, e os representou como uma ferramenta. Ele usou relações espaciais, com o intuito de mostrar conexões de função." (ARNHEIM,1969, p.313).

O autor coloca como exemplo de representação de processo um diagrama do funcionamento do coração, este é um processo complexo que pode ter seus princípios do funcionamento apresentados de maneira simples como no diagrama criado por Paul Klee, "Toda forma foi radicalmente reduzida para a representação mais simples do processo básico"³⁷(ARNHEIM,1969, p.306). Depois do aluno aprender os princípios básicos, pode partir para um entendimento mais complexo, pois já teve entendimento das estruturas como num *template* (ARNHEIM,1969). Na figura 24 está o diagrama de Paul Klee.

Figura 24- Representação do processo de funcionamento do coração humano feito pro Paul Klee



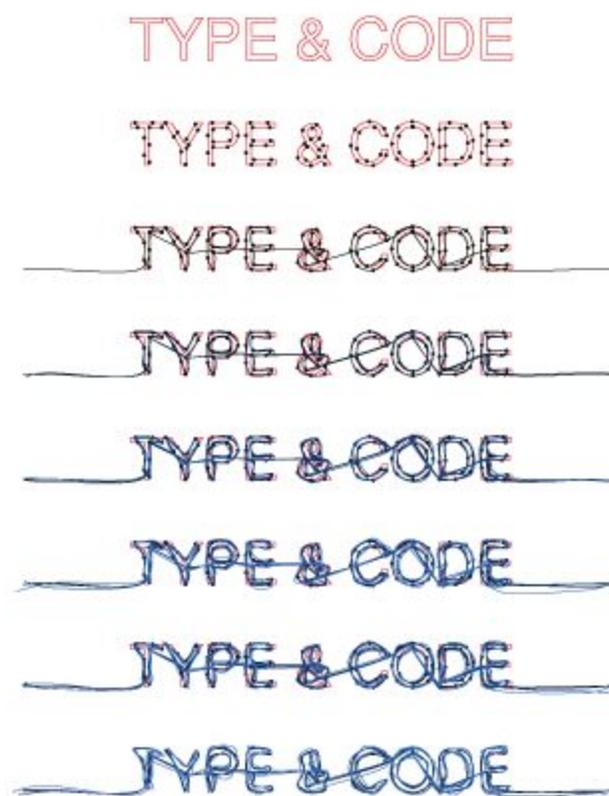
Fonte - ARNHEIM,1969

³⁶ active exploration, selection, grasping of essentials, simplification, abstraction, analysis and synthesis, completion, correction, comparison, problem solving, as well as combining, separating, putting in context.

³⁷ All shape has been radically reduced to the simplest representation of the basic process

Esse tipo de abordagem pode ser utilizada para representar princípios da lógica de programação, como forma de entender suas estruturas e lógica de construção. Além disso, pode propor relação entre o resultado visual e seu processo de construção. Possibilitando o aluno exercitar seu pensamento visual para deduzir as etapas de um algoritmo, decodificando formas e processos, como no exemplo da figura 25 de decomposição do processo de construção de tipografia generativa feita em oficina de programação aplicada à design na UFC.

Figura 25 - Decomposição do processo de construção de tipografia generativa feita na oficina Type and Code.



Fonte - tipografia generativa criada pelo professor Roberto Vieira.

A partir disso pode decompor elementos e estruturar padrões relacionando-se ativamente com o conteúdo. Este tipo de atividade está associada

ao desenvolvimento do pensamento visual, que está aliado a capacidade de enxergar padrões.

O pensamento visual necessita, mais amplamente, da capacidade de ver formas visuais como imagens de padrões de forças que estão subjacentes à nossa existência - o funcionamento de mentes, corpos ou máquinas, a estrutura de sociedades ou ideais.³⁸ (ARNHEIM, 1969, 315).

3.5. Conclusão

Esse tipo de abordagem visual do problema aliada a uma visão mais prática de ensino, baseada em conceitos construcionistas, pode criar uma nova forma de ensinar a lógica de programação, com uma maior atratividade para designers e resultar em um aprendizado mais efetivo.

³⁸ Visual thinking calls, more broadly, for the ability to see visual shapes as images of patterns of forces that underlie our existence - the functioning of minds, of bodies or machines, the structure of societies or ideas.

4. ESTUDO DE CASO

4.1. Introdução

No contexto do curso de design da UFC, no início da implementação da disciplina de design computacional em 2015 não haviam alunos interessados o suficiente para a sua realização. Porém foi desenvolvida uma instalação de esferas com manipulação de imagens no processing e com materialização realizada por meio de máquina fresadora (ver na figura 26) que despertou o interesse dos alunos para esse assunto. Com isso o número de matriculados em 2016.1 foi de 9 com 3 desistências. Com esses alunos foram gerados mais resultados visuais que deram mais visibilidade para programação gerando um aumento de matriculados em 2016.2 para 21 com 6 desistências. Isso demonstra que o interesse dos alunos pode ser despertado e desenvolvido, para isso é interessante promover ações para diminuir as barreiras em relação ao assunto.

Figura 26 -Instalação de esferas e processo de construção.



Fonte - Laboratório de Fotografia Tátil da UFC

4.2. Questionário de perfil do curso

Para entender o perfil do aluno do curso de Design da Universidade Federal do Ceará foi aplicado um formulário com 26 participantes, alunos do curso de design. O resultado se deu majoritariamente com alunos que estão cursando do meio para o fim do curso, ou seja, alunos que já tiveram contato inicial com programação na disciplina de Projeto Gráfico 3 no aprendizado de HTML, o formulário completo está em anexo. Foi observado que o receio em aprender programação está acima da média, como pode ser visto no gráfico 1.

Gráfico 1 - Formulário: nível de receio em aprender programação



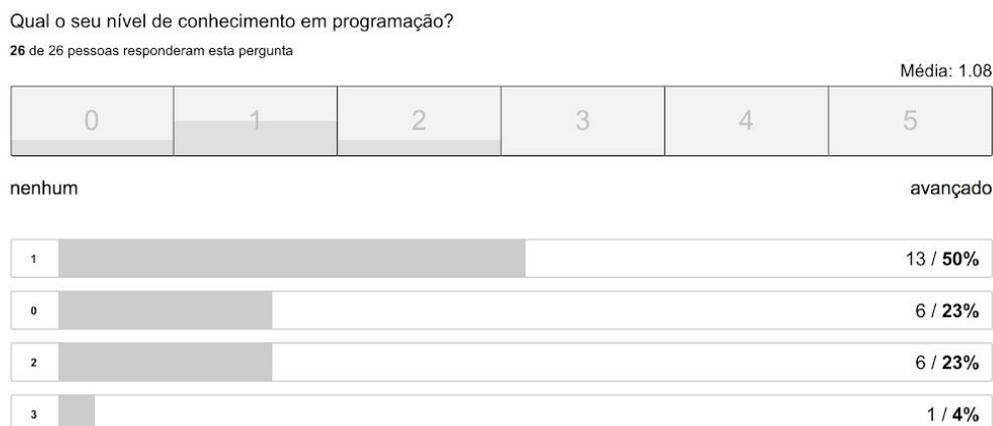
Fonte - resultados gerados em TypeForm

O nível de conhecimento foi descrito em 50% das respostas como 1, sendo considerado conhecimento básico. Porém através de outras respostas se observa que esse conhecimento na maioria dos casos se dá a partir da disciplina de Projeto Gráfico 3, no aprendizado de HTML que não se enquadra na categoria de design generativo. A maioria dos alunos também afirmou ter participado de projetos no curso onde seria interessante saber programar. Os softwares utilizados pelas

pessoas com conhecimento são: Processing, Rhinoceros 3D com *Grasshopper*, Dreamweaver, Sublime e Muse.

Em relação ao aprendizado as respostas se resumem em dificuldades em: memorizar e entender comandos, utilizar pensamento lógico e falta de motivação, prática e fontes de ensino. As sugestões de melhorias propostas foram em essência: metodologia focada em aplicações de design e não no código em si; exercícios que demonstram na prática o princípio que está sendo aplicado; simplificar comandos e códigos; sugestão automática de possíveis comandos que podem ser utilizados e tornar a experiência divertida para o usuário.

Gráfico 2 - Formulário: nível de conhecimento em programação



Fonte - resultados gerados em TypeForm

Com relação ao interesse médio em aprender programação para criação de artes visuais dentro da amostra houve 77% de respostas positivas com 20 alunos. A motivação para saber programar foi descrita no geral como para a ampliação de possibilidades e alguns alunos disseram achar resultados de design feitos com programação interessantes. Dentro do interesse de aplicações de programação em design o que apresentou maior resultado foi logo cambiante, seguido por arte generativa e depois tipografia generativa (itens propostos no formulário), também houveram sugestões de criação de objetos generativos, instalações, sites e aplicativos. Os participantes que não têm interesse justificaram sua escolha como

preferindo focar na interface e não acharem relevante para um designer saber programar.

Gráfico 3 - Formulário: interesse em aprender programação

Você gostaria de aprender programação para criação de artes visuais?

26 de 26 pessoas responderam esta pergunta

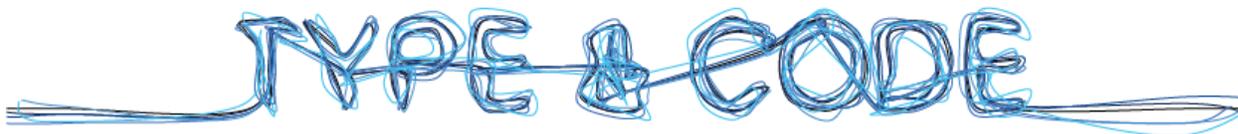


Fonte - resultados gerados em TypeForm

Conclui-se com esse formulário que existe interesse em aprender programação aplicada a design, porém um receio alto em aprendê-la.

4.3. Oficina *Type and Code*

Figura 27 - Resultado de exercício proposto na oficina de Tipografia Generativa



Fonte - imagem concebida pelo professor Roberto Vieira

Atualmente o ensino de lógica de programação no curso de Design da UFC acontece por meio de aulas expositivas e montagem de códigos ensinados na aula, apesar de já haver uma proposta de pensar o código aplicado a design os alunos ainda demonstram dificuldades em compreender o código. Dessa forma, foi testada uma metodologia alternativa para ensino de lógica de programação.

Com base nos estudos de metodologia pedagógica, foi aplicada uma metodologia alternativa de ensino durante oficina de tipografia generativa para iniciantes no curso de Design da Universidade Federal do Ceará. O público foi majoritariamente de alunos do curso, porém profissionais e alunos de outros cursos relacionados com design, como arquitetura e sistemas e mídias digitais, também participaram em menor escala da oficina.

As ferramentas utilizadas foram um teste de pseudocódigo manual, desenho do processo e um formulário para receber feedback. O teste era constituído da criação de um pseudocódigo, ou seja, escrever o algoritmo em português para entender as etapas de forma mais clara, traduzindo o que estava acontecendo virtualmente e, em seguida, desenhar o processo a mão. Após a oficina os alunos responderam um formulário sobre a efetividade do processo.

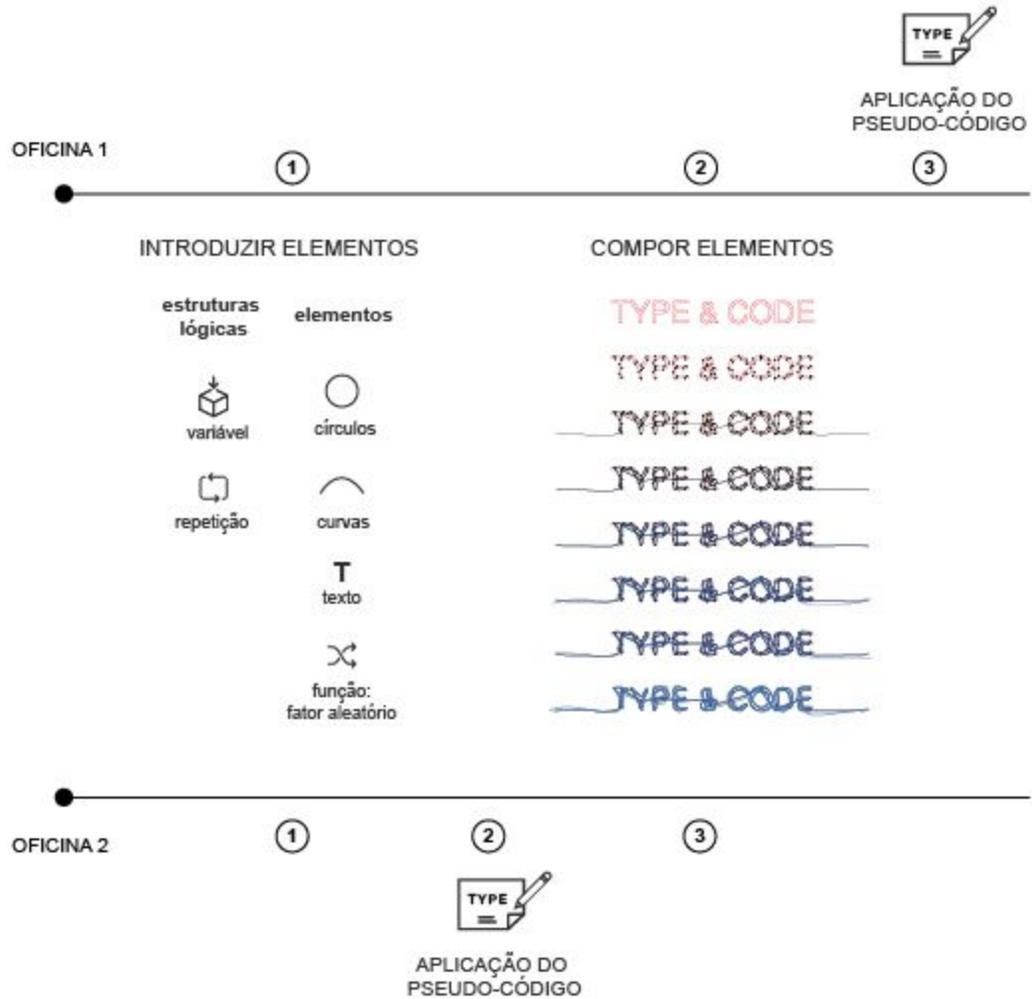
Na primeira oficina foi utilizado o método tradicional de aula expositiva e depois aplicado o teste para analisar se os alunos sentiam maior facilidade do que na forma tradicional. E na segunda oficina o teste foi aplicado no começo para ser um guia na construção do código. Na figura 28 se descreve o esquema de como foram estruturadas as oficinas.

4.3.1. Descrição e realização do pseudocódigo

O primeiro passo descrito no pseudocódigo era "- Escrever TYPE" e os seguintes seriam descritos pelos próprios alunos, que seriam de uma forma geral:

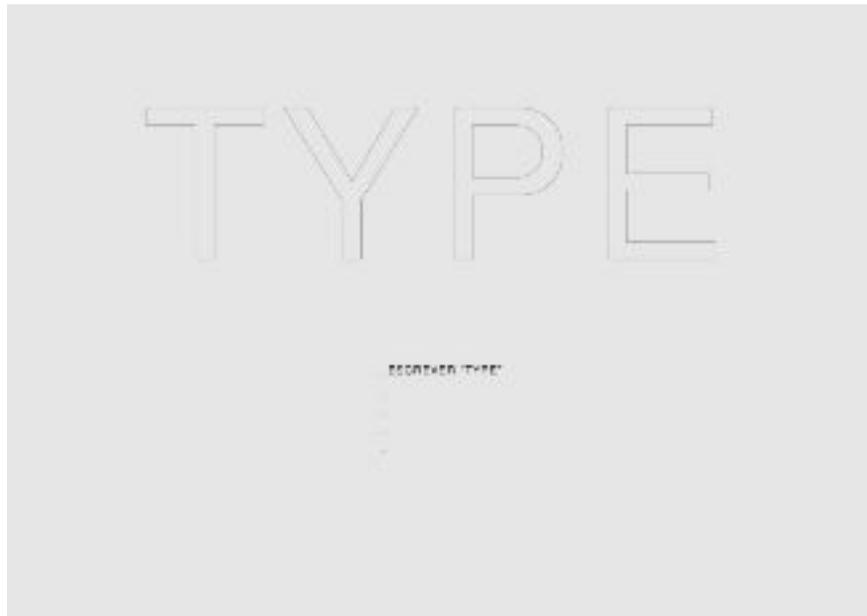
- Dividir em pontos equidistantes
- Traçar linha com fator de perturbação por todos os pontos
- Repetir 5 vezes

Figura 28 – Processo de ensino de oficina de Type and Code



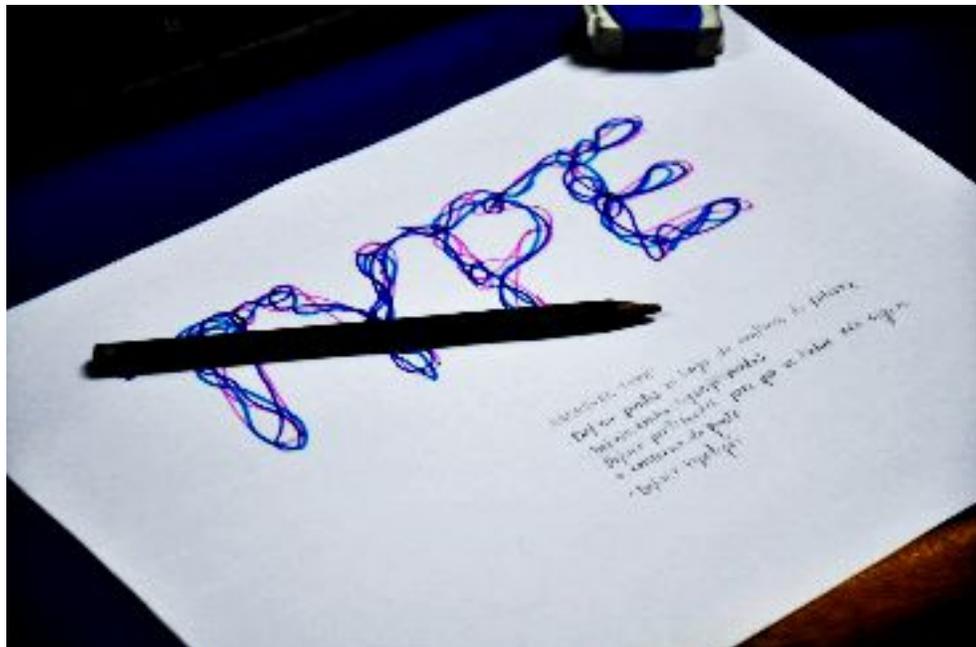
Fonte – elaborada pela autora

Figura 29 – Folha para aplicação do pseudocódigo e desenho do processo



Fonte – elaborada pela autora

Figura 30 – Aplicação do pseudocódigo e desenho do processo



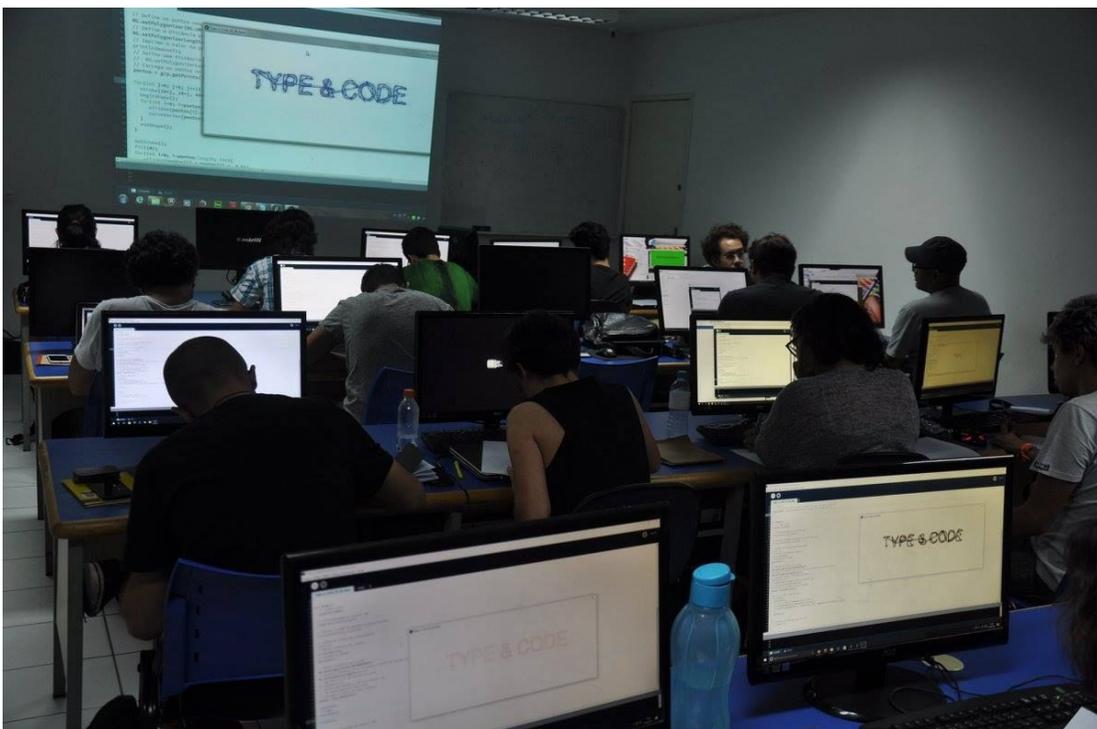
Fonte – foto feita pelo professor Roberto Vieira

Figura 31 – Alunos aplicando metodologia testada



Fonte – fotos feita pelo professor Roberto Vieira

Figura 32 – Oficina Type and Code



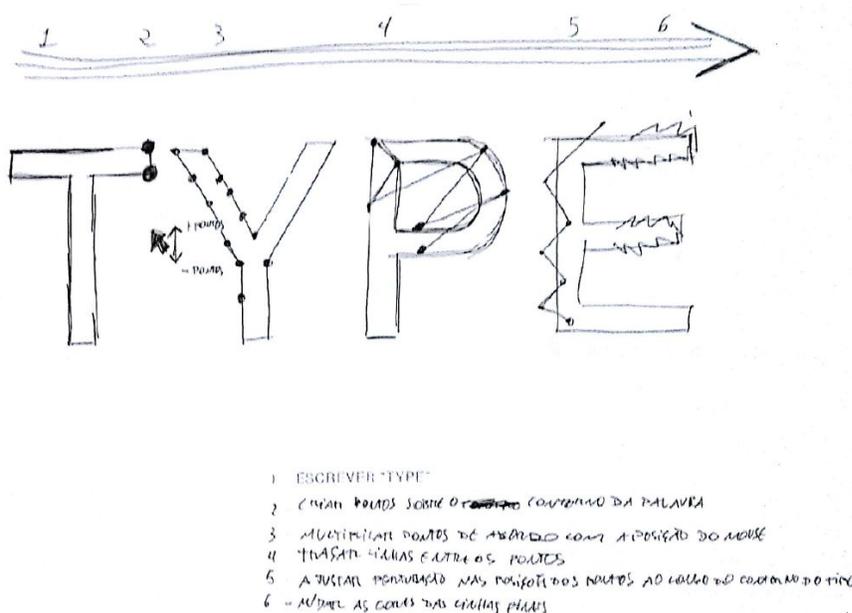
Fonte –foto feita pelo professor Roberto Vieira

4.3.2. Análise de metodologia aplicada na oficina

A descrição por pseudocódigo foi similar de uma forma geral para todos, variando no nível de detalhamento. Porém, notaram-se algumas abordagens diferentes a seguir:

- Utilização de metáforas com ferramentas conhecidas: "Ligação de pontos utilizando 1 curva única bézier de forma sequencial (caneta do illustrator)"
- Alguns alunos dividiram o passo a passo também visualmente, aplicando um comando para cada parte do esquema visual.

Figura 33 – Exemplo de aplicação do teste com pseudo-código



Fonte – criado pela autora

- Alguns alunos incluíram passos anteriores como "abrir setup" e passos de transição como "limpar tela".

No referente ao momento de usar o pseudocódigo, observou-se que na segunda oficina houve necessidade de consultar o pseudocódigo também na conclusão do projeto, para verificar se o que tinha sido estruturado no pseudocódigo tinha sido executado no código e de que forma.

Construtivismo e aplicação do pseudocódigo:

Baecker descreve "O comportamento de um programa não pode ser descrito por um desenho estático, se requer uma sequência dinâmica."³⁹ (1998, pag.1). Dessa forma o processo foi descrito e desenhado a mão pelos alunos como forma de "executar" essa sequência dinâmica para o entendimento do código.

O teste manual do pseudocódigo apresenta algumas características construcionistas, através da criação de uma hipótese elaborada no papel e testada no programa. A ferramenta Processing possibilita visualizar os resultados do que está sendo feito durante o processo. Dessa forma, testes são feitos sucessivamente para se chegar a um resultado. Isso está relacionado com o processo construtivista de realizar múltiplas vezes um experimento até criar uma teoria pessoal que resulta em conhecimento (Gibbons, 2016).

A partir do momento em que o aluno elabora o pseudocódigo já demonstra qual o seu nível de conhecimento sobre o tema, pois nele é necessário se prever a construção do código virtual, confrontando o aluno com a lógica do que ele irá executar.

O contexto do conteúdo, de fundamental importância para teoria construcionista, também foi pensado de forma a ser relevante para os alunos, estes criaram uma tipografia generativa própria. Dessa forma, visando um objetivo claro, pessoal e contextualizado com o curso de Design.

³⁹ a program's behaviour cannot be described by a static drawing; it requires a dynamic sequence.

O uso do processo manual aliado ao virtual criou um contraste que possibilitou enxergar o problema de ângulos distintos e através de materiais diferentes. Essa combinação possibilita a criação de relações, assim, se criam metáforas, relacionando um novo conhecimento com um já existente.

4.3.3. Análise do formulário

A partir de formulário, em anexo, pode-se observar as implicações do uso do pseudocódigo na primeira oficina: 73% disseram que o uso de pseudocódigo facilitou o entendimento, e 82% afirmaram que seria melhor utilizado se fosse aplicado no início da oficina. Com relação às dificuldades de aprendizado foram observados os seguintes tópicos:

- Entender as hierarquias do código;
- entender a lógica das ações;
- acompanhar o que estava sendo executado pelo professor;
- as nomenclaturas e suas funções;
- apreender aquilo que foi dito na oficina.

Também foram sugeridas melhorias para o processo de aprendizagem:

- Estudar a documentação da linguagem para entender melhor as funções prontas;
- explicação da lógica da linguagem;
- maior tempo de estudo;
- linkar as técnicas manuais utilizadas (pseudocódigo) com virtuais, entender o que se precisa manualmente e executar no código;
- explicar as regras e exceções do código (ex: sempre terminar com ponto e vírgula),
- visualizar as etapas de construção do código,

- explicar as funções que serão utilizadas e o que significa a linha dos algoritmos no código.

4.4. Conclusão

Essa oficina resultou na definição do projeto de pesquisa por livro de metodologias de aprendizado de código aplicado para design, a partir do resultado positivo do experimento. No capítulo seguinte serão analisados as metodologias de ensino utilizadas em livros.

5. ANÁLISES DE SIMILARES

Neste capítulo são analisados processos metodológicos de ensino através de livros com objetivo de compreender como programação tem sido ensinada de forma visual e utilizando pensamento sistêmico. Os primeiros cinco exemplos são livros relacionados à programação, os dois finais têm temáticas diferentes, porém metodologias interessantes que podem agregar nos parâmetros para a criação do livros proposto por essa pesquisa.

Os parâmetros para a análise serão voltados para a didática do livro, observando a maneira como os conteúdos são passados, serão:

- **Analogia:** uso de metáforas
- **Clareza:** simplicidade da explicação do processo
- **Categorias:** organização do conteúdo de forma coerente
- **Apresentação:** utilização de imagens interessantes, diagramação bem estruturada
- **Inovação:** criação inovadora no sentido didático
- **Interação:** pode se classificar como virtual ou manual
- **Estrutura:** uso de esquemas visuais
- **Atrativo para iniciantes:** conteúdos facilitados e interessantes
- **Metodologia:** metodologia utilizada para ensinar os processos

Arduino Projects Book

O "Arduino Projects Book" é um livro manual que faz parte do Kit para iniciantes de arduino. Esse formato gera alta interação porque o produto tem a teoria e prática juntas. Há grande quantidade de metáforas e esquemas visuais (ver na figura 35), juntamente com isso o livro cria ícones para representar os componentes que serão utilizados em cada projeto e mostra o tempo de execução, nível de dificuldade e projetos que podem incrementar em projetos anteriores - indo do mais simples para o mais complexo. Dessa forma o livro tem uma ótima abordagem para iniciantes. Seu ponto fraco é a categorização de projetos sem aplicações práticas para design.

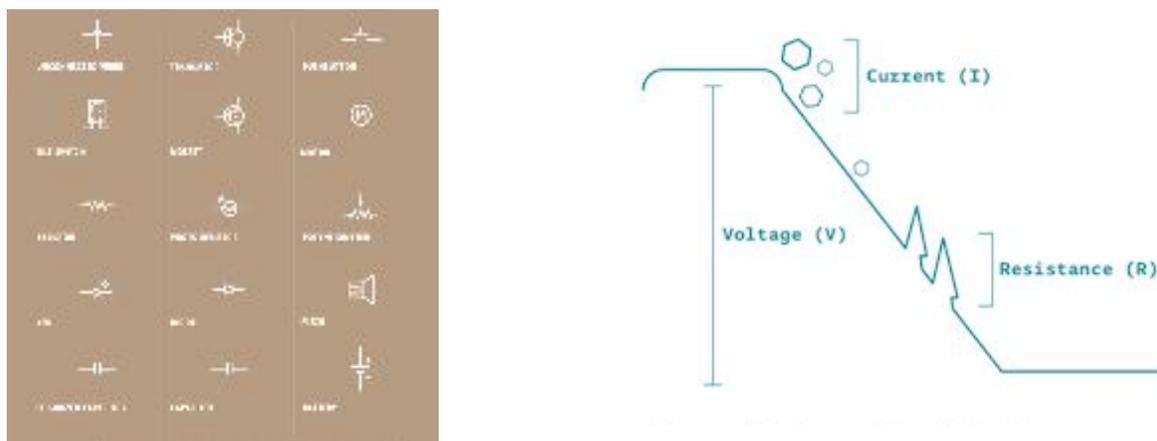
Figura 34 – Kit de arduino para iniciantes



Fonte – *arduino starter kits*⁴⁰

⁴⁰ Disponível em: <<https://www.arduinostarterkits.com/>>. Acessado em 01 Jul. 2017.

Figura 35 – ícones representando os componentes e metáfora sobre resistência elétrica.



Discover: analog Input, using the serial monitor

Time: **45 MINUTES**

Builds on projects: **1, 2**

Level: ■■■■■

Fonte - Retirado do Kit de arduino para iniciantes

Generative design: Visualize, Program, and Create with Processing

O livro "*Generative design visualize program and create with processing*" do Hartmut Bohnacker, Benedikt Gross, Julia Laub, é um livro de aprendizado de processing voltado para aplicação artística. O livro tem projetos visualmente interessantes, apresenta os projetos de forma interessante e esquemática, emprega um passo a passo bem estruturado e ilustrado. O livro usa do recurso de ter dois tipos de papéis diferentes para diferenciar a categoria de projetos da de ensino, utilizando o tipo de material como informação. Há também uma versão online dos códigos ensinados no livro, proporcionando maior interação. Sua desvantagem é categorização de projetos muito voltada para elementos básicos de design, como cor, imagem, tipografia. Poderia ser mais aplicado como identidade generativa, logo

cambiante etc. para ser um guia mais prático.

Figura 36 - Livro *Generative Design*



Fonte – Site da Amazon⁴¹ e benedikt-gross⁴²

Design by Numbers

Livro de John Maeda, professor do MIT que defende o aprendizado de código por designers e artistas. É um livro de ensino de uma linguagem de programação criada por Maeda voltada para seu público alvo. O livro apresenta esquemas visuais detalhados de cada pedaço de código, como se fossem amostras. A sua interação se dá pelo livro servir para ensinar uma linguagem voltada para designers que está disponível online. Suas desvantagens se constituem de não apresentar exemplos visuais muito interessantes, ter uma abordagem visual muito matemática. Outro ponto é sua categorização de projetos é incoerente, mostrando

⁴¹ Disponível em:

<<https://www.amazon.com/Generative-Design-Visualize-Program-Processing/dp/1616890770> >.

Acessado em 01 Jul. 2017.

⁴² Disponível em: <<http://benedikt-gross.de/log/2009/09/generative-gestaltung/> >. Acessado em 01 Jul. 2017.

tanto de elementos de design que serão explorados, como linha e ponto, como estruturas de lógica de programação.

Figura 37 - Livro *Design by Numbers*



Figura - à esquerda: site Maeda Studio⁴³ à direita: site da Amazon⁴⁴

Processing for Visual Artists: How to Create Expressive Images and Interactive Art

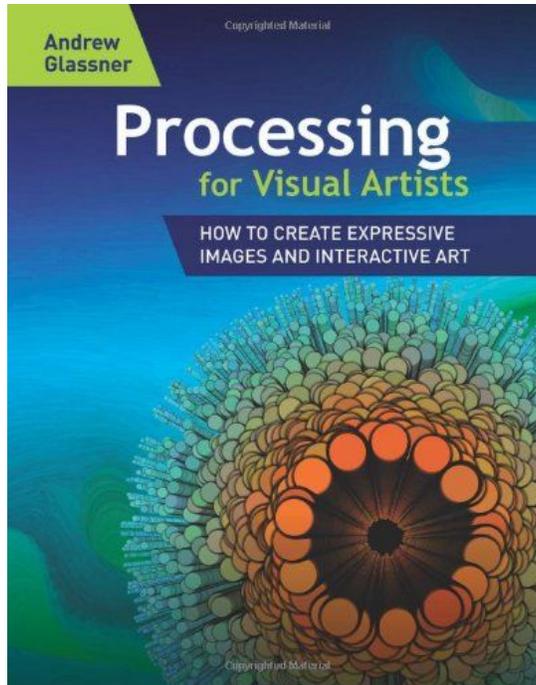
Livro de Andrew Glassner para o aprendizado de processing voltado para artistas e designers, usa uma abordagem mais tradicional de ensino. O livro é muito voltado para parte técnica, contém muito texto e não tem boa exploração visual.

⁴³ Disponível em: <<http://maedastudio.com/1999/dbn/index.php>>. Acessado em 01 Jul. 2017.

⁴⁴ Disponível em:

<https://www.amazon.com/Design-Numbers-John-Maeda/dp/0262133547/ref=sr_1_556?s=books&ie=UTF8&qid=1391267798&sr=1-556&keywords=design#reader_0262133547>. Acessado em 01 Jul. 2017.

Figura 38 - Livro *Processing for Visual Artists*



Fonte - site do *Processing*⁴⁵

Form and Code

Livro de Casey Reas, um dos precursores do *Processing*, aluno de John Maeda, defende o ensino de código para designers e artistas. O livro tem a proposta de explorar territórios possíveis com programação em artes visuais, explicando conceitos e a lógica de ação e não ensinar a programar. O livro apresenta diversos esquemas, tem muitos exemplos visuais interessantes. Um ponto diferente dele para os outros são suas analogias com objetos da realidade, como esculturas, prédios etc. Tem códigos disponíveis online criando maior interação. Uma desvantagem é

⁴⁵ Disponível em:

<<https://www.amazon.co.uk/d/cka/Processing-Visual-Artists-Create-Expressive-Images-Interactive/1568817169>>. Acessado em 01 Jul. 2017.

que sua categorização de projetos é muito abrangente e não tem aplicações muito práticas.

Figura 39 - Fotos do livro *Form and Code*



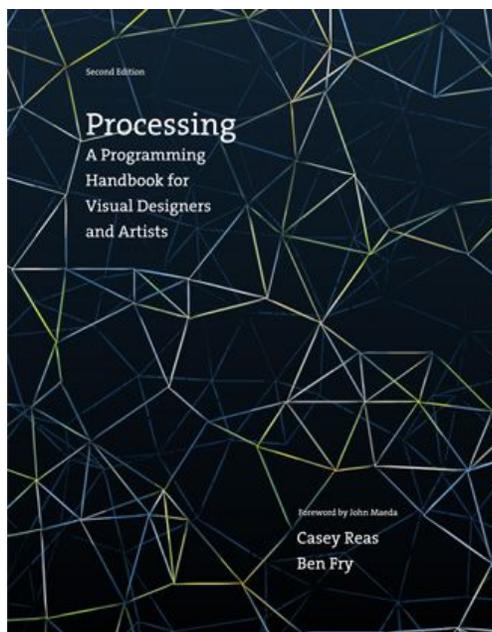
Fonte- site Form and Code⁴⁶

Processing: a programming handbook for visual designers and artists

Livro de Casey Reas e Ben Fry, os dois criadores do *Processing*. Diferente do *Form and Code* é um livro de ensino de programação. O livro é voltado para iniciantes e foca em aspectos básicos que são desenvolvidos gradualmente. Há esquemas com passo a passo colocando exemplos de código como amostras, sua categorização é prática e organizada pois se divide em elementos básicos e aplicações práticas em design. A interação se dá por disponibilizar os códigos usados online. Sua desvantagem é que há grande volume de texto comparado com as imagens.

⁴⁶ Disponível em: <<http://formandcode.com/>>. Acessado em 01 Jul. 2017.

Figura 40- Livro *Processing: a programming handbook for visual designers and artists*



Fonte - site do Processing⁴⁷

Ikea - *Cook this page*

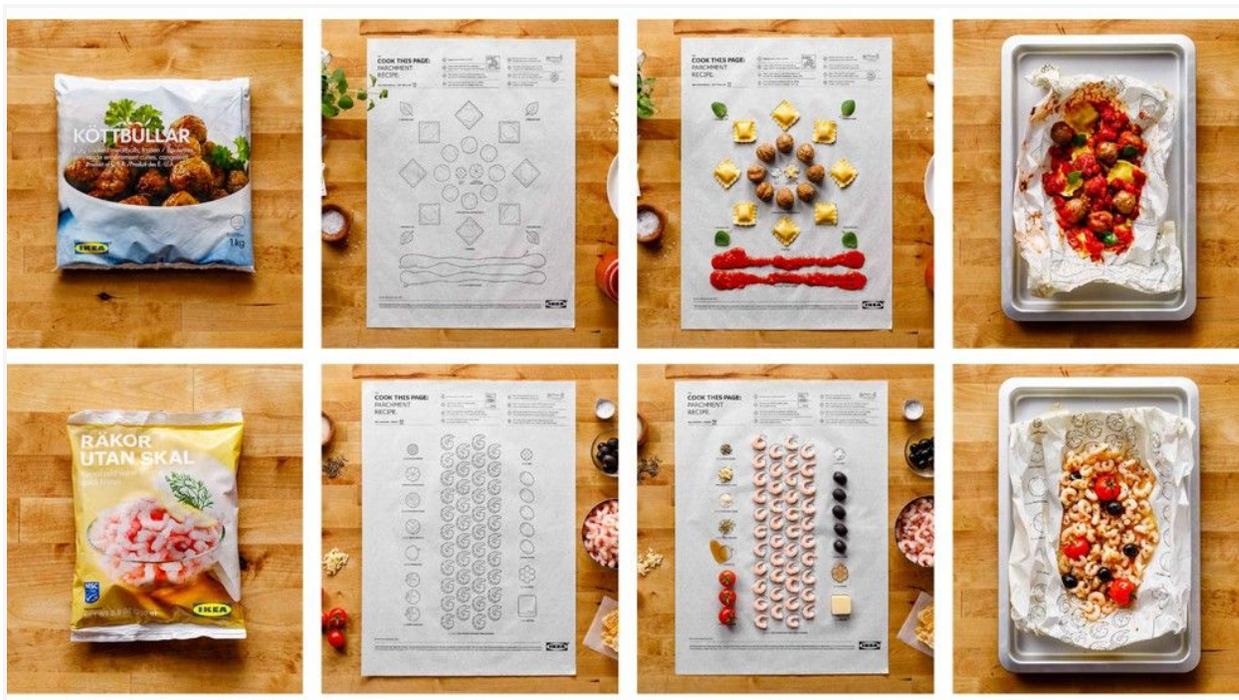
Este projeto é uma peça gráfica de livro de receitas, foi contemplada na análise de similares por ter um sistema visual de processo muito interessante. Um livro de receitas também pode ser considerado uma analogia ao um livro de ensino de código, porque ambos têm "ingredientes" e são divididos em passo-a-passo, assim como a receita de bolo é similar a um algoritmo.

⁴⁷ Disponível em: <<https://processing.org/handbook/>>. Acessado em 01 Jul. 2017.

Este projeto foi desenvolvido pelo estúdio de design Leo Burnett para a Ikea, o intuito de mostrar que cozinhar pode ser simples e divertido, seguindo o conceito da IKEA de vender móveis desmontados com manuais simples que tornam o processo de montar como um pequeno desafio.

O processo de cozinhar se torna algo simplificado e visualizado instantaneamente. O projeto também é prático porque tem a opção de comprar os ingredientes prontos para as receitas do livro, funcionando como um kit. A sua interatividade se dá no uso da página para informar e para embrulhar a receita no forno, é ao mesmo tempo informativo e funcional. Tem uma ótima abordagem para iniciantes, porém tem uso limitado a poucas receitas.

Figura 41 - Processo de uso do livro de receitas *Cook This Page*

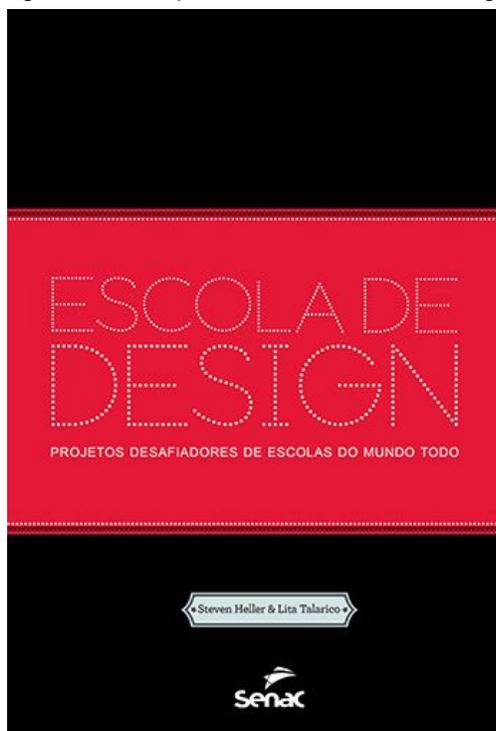


Fonte – foto de Leo Burnett

Escola de design: Projetos Desafiadores de Escolas no Mundo Todo

Escola de design é um livro de Steven Heller e Lita Talarico. Trata-se de uma coletânea de diversos projetos de design. Contém trabalhos interessantes e bem apresentados visualmente. O livro aborda a perspectiva do aluno e do professor, buscando uma metodologia de ensino de design mais engajadora, os trabalhos são apresentados de forma bem simples, como em um passo a passo, são expostos a partir de categorias como "enunciado", "meta", "produção", "atividade" e "entrega" dependendo do tipo de projeto, formulando assim a metodologia utilizada em cada projeto. Além disso, possui projetos questionadores e sociais.

Figura 42 - Capa do livro Escola de Design



Fonte- site da Editora Senac SP⁴⁸

⁴⁸ Disponível em:

<<http://www.editorasenacsp.com.br/portal/produto.do?appAction=vwProdutoDetalhe&idProduto=22356>>.
Acessado em 01 Jul. 2017.

5.1. Avaliação dos parâmetros dos livros

Tabela 2- Parâmetros da análise de similares

Análise de similares									
	analogia	clareza	categorias	*apres.	estrutura	inovação	interação	*atrativo	metodologia
Arduino Projects Book	X	X		X	X	X	X	X	
Generative Design		X		X	X		X	X	
Design by numbers		X			X		X	X	
Processing for Visual Artists									
Form and Code	X	X		X	X		X	X	
Processing		X	X		X		X	X	
Ikea Cook this Page		X		X	X	X	X	X	
Escola de Design		X		X	X				X

*apres: apresentação

*atrativo: atrativo para iniciantes

Fonte- criada pela autora

De acordo com a análise, foram observados os seguintes padrões que podem ser parâmetros para a produção do livro proposto nesta pesquisa, estão divididos em contexto, ou seja, o modo de utilizar o livro, e estrutura, como o conteúdo será abordado.

Contexto

- O livro como parte de um kit, ou seja, ter um material para se aplicar os conceitos dados.
- Utilização de mais de uma plataforma para abordar o conteúdo, como ao disponibilizar versão online para dos exemplos dados no livro.

Estrutura

- Trabalhos apresentados de forma simples, como em um passo a passo, divididos em categorias, como no exemplo do livro "Escola de Design" "enunciado", "meta", "produção", "atividade" e "entrega", dependendo do tipo de projeto. Mostra assim, a metodologia por trás do projeto. Categorização organizada em duas partes: elementos básicos de lógica de programação e aplicações práticas em design.
- Utilização de metáforas e esquemas para explicar conceitos e visualizar processos.
- Mostrar os pedaços de código como amostras, como ao colocar seu resultado ao lado de cada ação de programação.
- Mostra tempo de execução e nível de dificuldade de cada projeto.
- Contextualizar o conteúdo ao fazer comparações com objetos reais, como esculturas, prédios etc.
- Predomínio da linguagem visual
- Utilização de ícones para facilitar o entendimento.
- Projetos que podem incrementar em projetos anteriores, ou seja, vai do mais simples para o mais complexo.
- Exploração material do livro para sinalizar conteúdos diferentes, por exemplo através do tipo de papel.
- Organização do conteúdo por categorias de aplicações práticas para designers (ex: arte generativa, tipografia generativa, logo cambiante, etc)

6. DIRETRIZES DE PROJETO

Dos 4 cenários iniciais foi escolhido o de criação de um livro de metodologias, a outros analisados e foi concluído que o cenário de elaboração de um plug in necessitaria de grande entendimento de programação para ser executado. A instalação e o jogo iriam ser muito básicos e possivelmente pouco efetivos nesse caso, a instalação precisaria de diversas peças para contemplar o tema.

Dessa forma, baseado nos dados de análise de similares, se observou uma lacuna em um livro que contemplasse ao mesmo tempo: boa abordagem visual, com esquemas e metáforas, exemplos interessantes; metodologia de ensino de projetos; e categorias práticas direcionadas para designers. Dessa forma o livro se propõe a ser uma coletânea de projetos de design com metodologia de ensino dos projetos e aplicações práticas, totalmente voltado para designers, com proposta de metodologia de estímulo do ensino. Nessa medida, também se propõe uma metodologia de uso do livro em sala de aula.

A proposta é um livro de metodologias alternativas ao aprendizado tradicional de lógica de programação, ao aplicar pensamento sistêmico e visual, contextualizar os projetos com a realidade e propor interações com os usuários. O livro se baseia em princípios construcionistas, e abordará a lógica de programação em projetos de design. As categorias abordadas serão identidade generativa, tipografia generativa, arte generativa, processamentos de imagem e instalação interativa.

Para estimular o entendimento dos designers sobre programação será criado um livro que estabelecerá relações com a sintaxe do código. Dessa forma, o projeto se propõe a criar uma nova perspectiva do problema e pensar programação de uma forma manual. Entender como um processo manual tem relação com um processo executado por um algoritmo, ao transformar o abstrato em concreto através de um contato mais direto com o processo, deixando mais próxima a relação do mundo físico com o mundo virtual.

Para a formulação dos conteúdos do livro serão realizadas oficinas de programação para testar as metodologias dos projetos propostos. Estes serão trabalhos realizados por alunos dentro da faculdade e outros trabalhos serão propostos e testados em oficinas.

A metodologia de projeto continua como *Design Science Research* (DSR) por se tratar de uma metodologia científica e projetual. A presente pesquisa já contemplou a fase de conscientização do problema, logo, a fase projetual começa na etapa de desenvolvimento. Inicializa com levantamento de projetos existentes, proposição de projetos complementares e organização de conteúdos. A partir disso serão sugeridas categorias dos projetos, organização do código e desenvolvimento de metodologias de ensino. Esses conteúdos serão testados através das oficinas de programação, seus resultados avaliados, assim, conclui-se o projeto com a diagramação do livro.

Tabela 3 - Cronograma de projeto

CRONOGRAMA DE PROJETO					
	MÊS 1	MÊS 2	MÊS 3	MÊS 4	MÊS 5
Levantamento de projetos existentes					
Proposição dos projetos complementares					
Organização dos projetos					
Separação dos projetos em categorias					
Organização do código					
Desenvolvimento das metodologias					
Testes em oficinas					
Avaliação					
Diagramação do livro					

■ sugestão
 ■ desenvolvimento
 ■ avaliação

Fonte - criada pela autora

7. PROJETO

7.1. Introdução

Na fase projetual foi feito um teste de material visual explicativo de código em uma oficina de programação. A partir da avaliação desse teste, foi definido que, para o desenvolvimento do projeto, as diretrizes pré-estabelecidas precisariam ser adaptadas para um novo contexto, onde o objetivo do projeto não tinha mais como foco criar um livro de ensino de programação, mas sim estimular o interesse pelo aprendizado de código voltado para designers.

O livro se constitui de uma coleção de códigos feitos na disciplina de Design Computacional, do curso de design da Universidade Federal do Ceará, por alunos e pelo professor Dr. Roberto Vieira. A partir desses códigos, são definidos parâmetros relacionados com as respostas dos alunos sobre temas referentes ao universo do Design (tipografia, cor, estilo visual, ferramenta etc), gerando, dessa forma, um código personalizado para cada aluno.

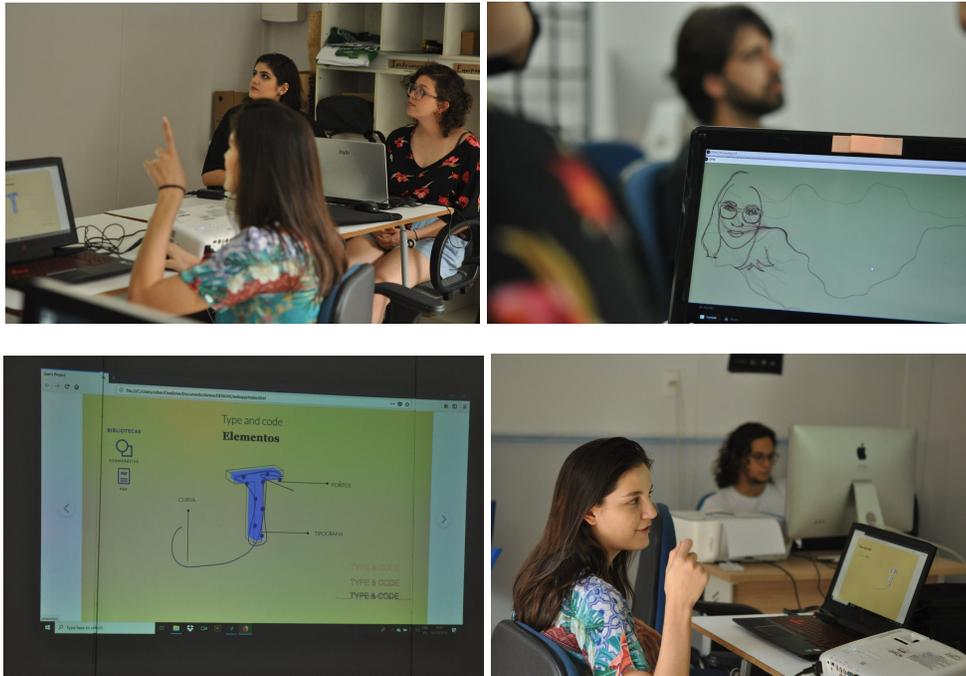
7.2. Oficina

Seguindo o processo metodológico de *Design Science Research*, foram testadas ferramentas complementares de aprendizagem de programação em oficinas. Na fase projetual foi criado um mapa visual de códigos, com diagramas de processo.

Na figura 44 é mostrado o material apresentado na oficina (Figura 43), realizada na UFC, de ensino do algoritmo do projeto "Design por Mulheres"⁴⁹. A partir disso foram definidos novos requisitos para a criação do material final.

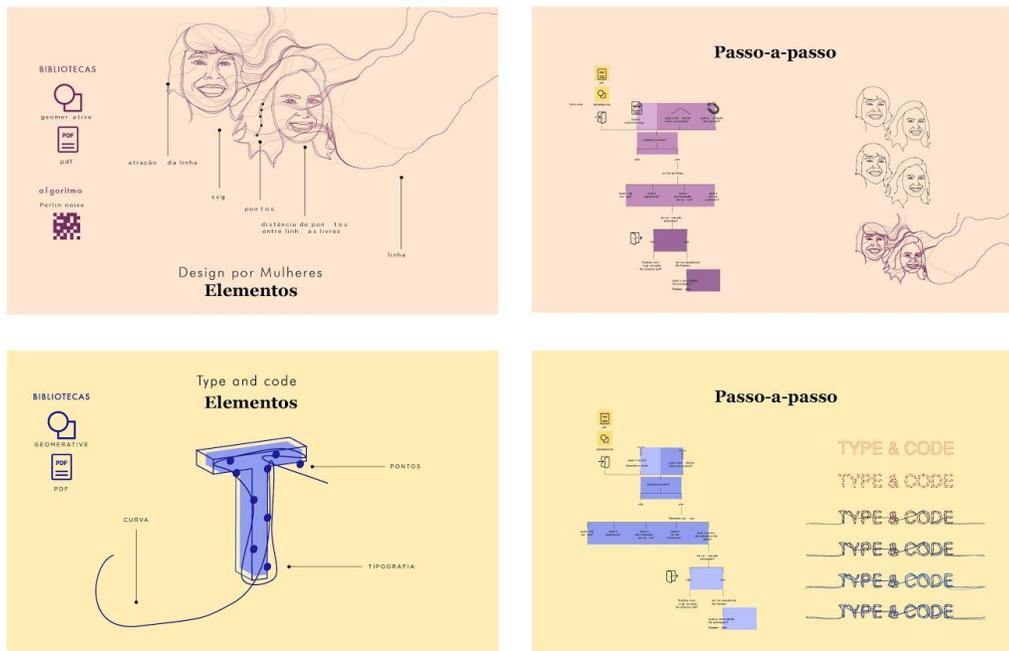
⁴⁹ Exposição realizada no Museu de Arte da Universidade Federal do Ceará (MAUC), para promover a produção feita por mulheres designers, idealizado por alunas mulheres do curso de Design da UFC

Figura 43 - Imagens da oficina do código do projeto "Design por Mulheres"



Fonte- Fotografias por Roberto Vieira e Denise Pompeu

Figura 44 - Protótipo de material de ensino apresentado em oficina na UFC.



Fonte- Elaborado pela autora, 2018.

7.3. Reavaliação dos requisitos

Com base no processo realizado nas oficinas, concluiu-se que o aprendizado de código é algo que demanda tempo e estudo, desta forma, um material visual de ensino não seria o suficiente para atender, completamente, a esta demanda. Além disso, seriam necessários conhecimentos prévios para o entendimento do material, e isso o tornaria menos atrativo para alunos que estivessem em um primeiro contato com programação.

Dessa forma foram revisados os requisitos de projeto para atender uma nova necessidade, não de ensinar programação, mas sim de estimular a aproximação de código e design. O intuito é mostrar que design pode se relacionar com programação de códigos e que o designer pode fazer parte desse processo.

Outro fator a ser levado em consideração é que a exposição de exemplos práticos de código, em instalações, vídeos interativos etc, pode levar os alunos de design a terem mais interesse em aprender programação. Observou-se que houve um aumento no número de matrículas na disciplina de Design Computacional do curso de Design, a partir da exposição de trabalhos feitos com algoritmos, conforme dados apresentados na seção 4.1. Dessa maneira, o livro conta com exemplos práticos de códigos de design aplicados com alunos do curso.

7.3.1 Aprendizado

O foco do livro é ser um atrativo para alunos de design e passar informações sobre código de uma forma descontraída, dessa forma, incentivar uma experiência com programação sem trazer um conteúdo de ensino. Isso pode despertar o desejo de aprender e iniciar um contato com programação. Newby (1996) explica que " a perspectiva construtivista descreve o aprendizado como uma mudança no significado construída a partir da experiência ." ⁵⁰(1996, *apud* TAM, 2000, pag.02). Dessa forma o

⁵⁰ The constructivist perspective describes learning as a change in meaning constructed from experience

livro se propõe a mostrar na prática o que ocorre ao escolher determinado parâmetro, como isso afeta o algoritmo.

7.3.2 Tema

Resnick (2009) cita que um dos problemas na educação tradicional de código para iniciantes é não estar conectado ao interesse dos alunos. Assim, o primeiro ponto do projeto se trata de abordar um tema que seja relevante para os estudantes de design.

A partir disso, foi proposto um livro que pudesse aproximar o aluno do código ao tratar o tema de uma forma pessoal, ao fazer perguntas sobre suas preferências de design, abordando tipografia, estilos, cores, ferramentas etc.

7.4. Processo

7.4.1 Referências

Na figura 45 é apresentado um painel de referências utilizadas no projeto. São livros com linguagem clara e visual através do uso de ícones. Além disso, um projeto que inspirou a abordagem atual da pesquisa foi o *SELFBOOK*⁵¹, que faz processamento de imagens através de perguntas, associando parâmetros utilizando formas geométricas.

Essa referência ajudou a alinhar ideias que já vinham sendo desenvolvidas em oficinas e trabalhos práticos para motivação dos alunos do curso de Design com a utilização de processamento de imagens.

⁵¹ Disponível em: <<https://www.behance.net/gallery/63063911/SELFBOOK-photobook-project> >. Acessado em 29 Nov 2018.

Figura 45 - Painel de referências



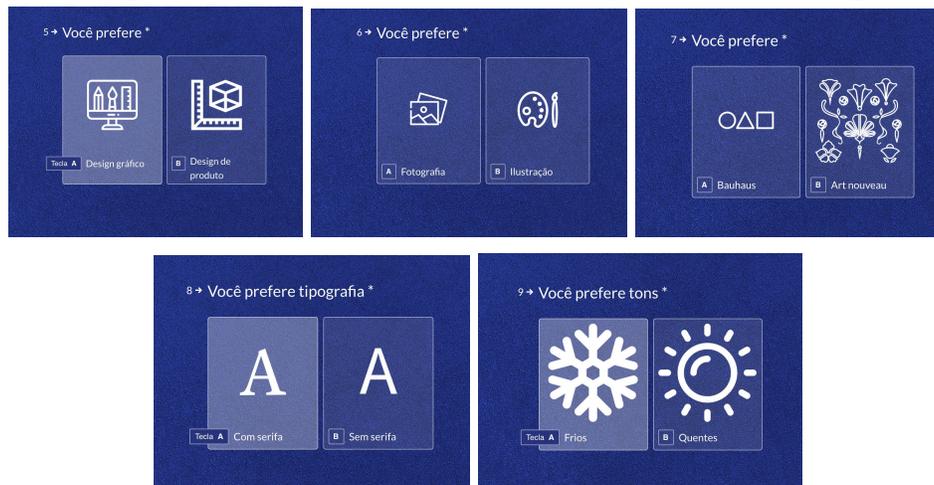
Fonte- elaborado pela autora, 2018.

7.4.2 Levantamento de dados

Um formulário para coletar as respostas usadas no livro, foi criado e publicado no grupo oficial do curso de design na rede social *Facebook* e respondido por 23 pessoas. O formulário foi composto de perguntas do universo de design, a seguir.

- Você prefere design gráfico ou design de produto?
- Você prefere fotografia ou ilustração?
- Você prefere Bauhaus ou Art Nouveau?
- Você prefere tipografia com serifa ou sem serifa?
- Você prefere tons frios ou quentes?

Figura 46 - Perguntas do formulário sobre preferências de design.



Fonte- Site do *Type and Form*.

7.4.3. Seleção de fotografias

Para a geração dos algoritmos personalizados foram utilizadas retratos dos próprios alunos do curso de Design a partir do formulário proposto. Desta forma, o intuito é gerar um reconhecimento deles no código, para que possam se perceber como agentes do processo, onde suas escolhas interferem no resultado final e isso é apresentado em suas próprias imagens.

Foram analisados os resultados com maior variação de respostas e selecionadas fotos destes alunos para serem aplicados os códigos selecionados.

Figura 47 - Fotos dos estudantes de design apresentados no livro.



Fonte- Elaborado pela autora, 2018.

7.4.4. Seleção dos códigos

Foram selecionados códigos produzidos na disciplina de Design Computacional da UFC, elaborados pelos próprios alunos, sendo eles, Josias Nascimento, Denise Pompeu, Levi Holanda, Lorraine Sampaio e Giovanna Leite, como também pelos professores Roberto Vieira e Eugênio Moreira. Eles funcionam com processamento de imagem, ou seja, a partir da leitura de pixels o algoritmo armazena as informações e gera a forma que foi designada. Esse processo é

definido por Bohnacker (2012) como gráfico a partir de valor de pixel, descrito a seguir.

"[...] cada pixel é reduzido ao valor de sua cor. Esses valores modulam os parâmetros de design como rotação, largura, altura e área. O pixel é completamente substituído por uma nova representação gráfica e o retrato se torna um tanto abstrato⁵²." (BOHNACKER, 2012, p.302)

Esses algoritmos abrangem linhas, tipografia, formas geométricas e pictogramas. Alguns deles já estiveram em exposições na UFC, como o código feito para a exposição "Design por Mulheres" e também o de "Curva única" e o "Retrato codificado" que estiveram no Circuito UFC-Arte.

Figura 48 - Códigos selecionados para o livro.



Fonte- Elaborada pela autora, 2018.

⁵² "[...] each pixel is reduced to its color value. These values modulate design parameters such as rotation, width, height, and area. The pixel is completely replaced by a new graphic representation, and the portrait becomes somewhat abstract." (BOHNACKER, 2012, p.302)

Figura 49 - Códigos apresentados na exposição "Design por Mulheres".



Fonte- Foto Roberto Vieira, 2018.

Figura 50 - à esquerda Reitor da UFC Henry Campos mostrando códigos feitos no curso de Design. A direita Aluno Josias Nascimento no Circuito UFC-Arte.



Fonte- Foto Roberto Vieira, 2016.

7.4.5 Associação dos parâmetros

Para que o processamento de imagem fosse feito, foi criada uma lista de parâmetros⁵³ dos algoritmos selecionados, como: quantidade, espessura,

⁵³ Reas descreve parâmetro como "[...] valor que tem um efeito no resultado de um processo. Isso pode ser algo tão direto quanto a quantidade de açúcar em uma receita (...). Parâmetros descrevem, codificam

espaçamento, cores e perturbação⁵⁴ dos elementos.

Dessa forma, as respostas dos alunos foram associadas a esses parâmetros para gerar resultados. As perguntas de preferência sobre tipo de design, estilo visual e tipografia definem qual código será usado para cada estudante, como pode se observar na figura 51 . Para realizar essas associações foi levado em consideração a densidade, tipo de linha - curva ou reta - e variação dos elementos no código.

Figura 51 - Diagrama de associação de perguntas com características do código.

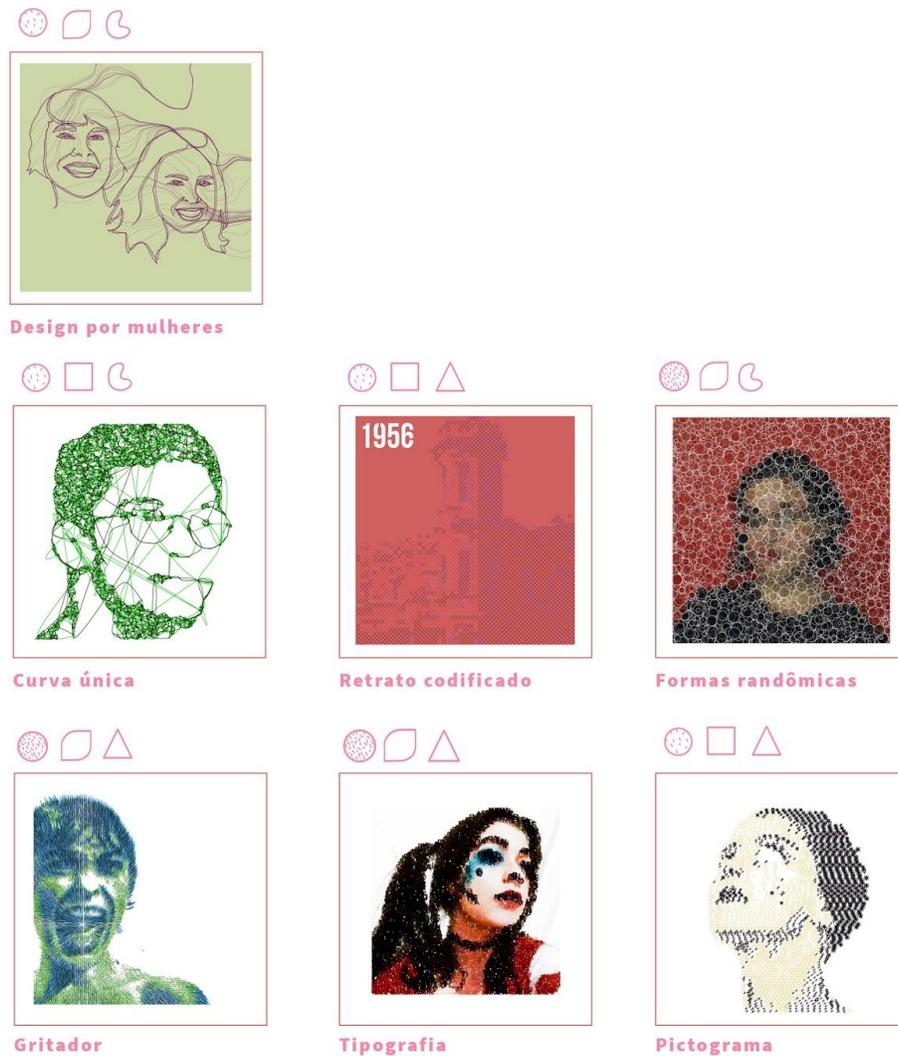


Fonte- Elaborado pela autora, 2018.

e quantificam as opções e as limitações em jogo em um sistema. Uma limitação comum pode ser o orçamento disponível para um projeto, enquanto uma opção de configuração pode ser o controle de cor, tamanho, densidade ou material. (2010, pag.95)"

⁵⁴ Perturbação no código está associada a aleatoriedade dos seus elementos.

Figura 52 - Códigos do livro categorizados.



Fonte - Elaborado pela autora, 2018.

No caso da alteração dos parâmetros individuais de cada código, houve uma associação entre as respostas de preferência de cores e tipo de ferramenta, definindo assim, respectivamente, a paleta utilizada e o valor da perturbação no algoritmo.

Figura 53 - Diagrama de associação das perguntas com parâmetros do código.



Fonte - Elaborado pela autora, 2018.

7.5. Elementos visuais

7.5.1 Ícones

A linguagem do livro está alinhada com o seu objetivo de tornar um conteúdo abstrato algo mais acessível, dessa forma, é composta por ícones, buscando exprimir conteúdos e ideias de maneira visual, simples e direta.

A partir das perguntas do formulário foi criado um conjunto de ícones para representá-las, e assim, são aplicados no livro mostrando as escolhas dos alunos ao lado de suas respectivas fotos. Foi, também, necessário criar ícones que pudessem representar características dos códigos para categorizá-los, como mostrado na figura 55.

Figura 54 - Conjunto de ícones das respostas elaborados para o livro.



Fonte - Elaborado pela autora, 2018.

Figura 55 - Conjunto de ícones dos códigos elaborados para o livro.



Fonte - Elaborado pela autora, 2018.

7.5.2 Tipografia

A tipografia de texto é a Source Sans Pro, escolhida por ter grande variedade de estilos e ser um fonte leve e de boa leitura, mas também podendo ser utilizada em destaques quando em pesos maiores.

Para gerar contraste de tipografias, visto que o projeto aborda relações opostas para gerar resultados, foi escolhida a Bitter, uma tipografia com serifa, que foi utilizada para os títulos.

Figura 56 - Família tipográficas utilizadas no livro.

Source Sans Pro		Bitter
Extra Light	Semibold	<i>Italic</i>
<i>Extra Light Italic</i>	Semibold Italic	Bold
Light	Bold	Bold Italic
<i>Light Italic</i>	Bold Italic	
Regular	Black	
<i>Italic</i>	Black Italic	

Fonte - Elaborado pela autora, 2018.

7.5.3 Padrão Cromático

Seguindo o estilo do projeto foram utilizadas cores que tivessem contraste entre si, foi criada uma paleta com quatro cores. O azul foi a cor predominante por remeter a tecnologia, sendo uma cor fria e séria; para compor com ele foi definida uma cor quente, o rosa. Para fazer um contraponto com o azul e rosa, foi escolhida a cor amarela juntamente com um cinza chumbo. Dessa forma estabelecendo as duas paletas de cores que seriam utilizadas nos códigos e também no livro.

Quando a resposta do aluno era de preferência por tons frios, foi utilizada a paleta azul e rosa, dando destaque ao azul. No caso da escolha ser por tons quentes, se aplica a paleta amarela e cinza com predominância do amarelo.

Figura 57 - Padrão cromático utilizado no livro.

	c96 m94 y1 k0 r 51 g57 b148 #333994		c3 m22 y100 k0 r 247 g196 b19 #f7c413
	c0 m59 y10 k0 r 243 g135 b168 #f387a8		c69 m61 y61 k52 r 57 g58 b58 #393a3a

Fonte - Elaborado pela autora, 2018.

7.5.4 Aplicação dos códigos

Cada aluno teve seu código personalizado, aplicando as regras definidas no processo. Utilizou-se a paleta criada no projeto gráfico, mas também com emprego de elementos nativos do algoritmo. Na figura a seguir são apresentados os resultados.

Figura 58 - Aplicação dos códigos com processamento de imagens.



Fonte- Elaborado pela autora, 2018.

7.5.5 Formato

Foi definido o formato de livro, por ser utilizado como um registro da produção de código no curso de design da UFC. Seu formato é quadrado, medindo 21 cm x 21 cm, impresso em papel couché fosco de gramatura 115g.

O projeto tem também a possibilidade de ser estendido para outros formatos como aplicativo, site ou mesmo cartazes, por exemplo. Ao ser aplicado em mídia

digital se tornaria um produto interativo, onde o aluno pudesse escolher e ver o resultado automaticamente e então poderia explorar as possibilidades do código.

7.6. Livro

7.6.1 Capa

Figura 59 - Capa do livro Design++Código.



Fonte - Elaborado pela autora, 2018.

Figura 60 - Folha de rosto do livro.

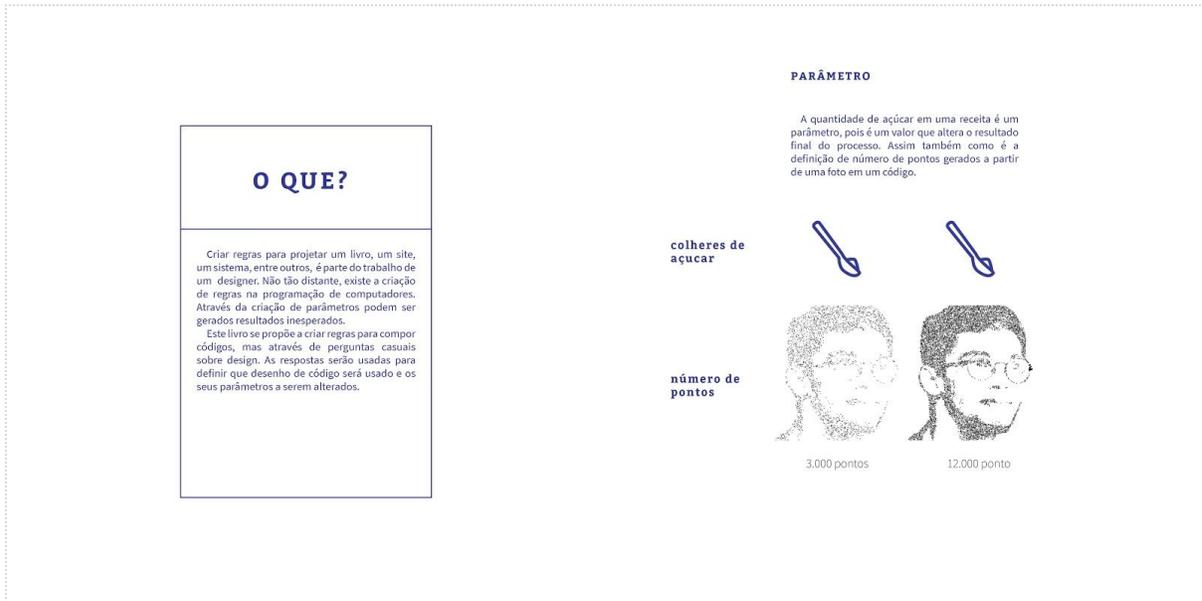


Fonte - Elaborado pela autora, 2018.

7.6.2 Introdução

Nessa parte inicial são explicados conceitos básicos para o entendimento do livro e como ele funciona, numa linguagem próxima do leitor. A primeira parte explica também o que é um parâmetro, já que esse conceito norteia todo o projeto, fazendo uma analogia com colheres de açúcar numa receita de bolo. A apresentação continua ao se responder as perguntas "O que?" "Quem?" "Onde?" e concluindo ao questionar o leitor "E você o que prefere?".

Figura 61 - Introdução do livro - tópico "O que?".



Fonte - Elaborado pela autora, 2018.

Figura 62 - Introdução do livro - tópico "Quem? Onde?".



Fonte - Elaborado pela autora, 2018.

Figura 63 - Introdução do livro - tópico "O que você prefere?".



Fonte - Elaborado pela autora, 2018.

7.6.3 Explicação

Em seguida são apresentados os parâmetros e suas associações, através de ícones e textos explicativos. Se apresenta a categorização dos códigos e mostra os parâmetros a serem alterados.

Figura 64 - Relação dos parâmetros com as características do código.

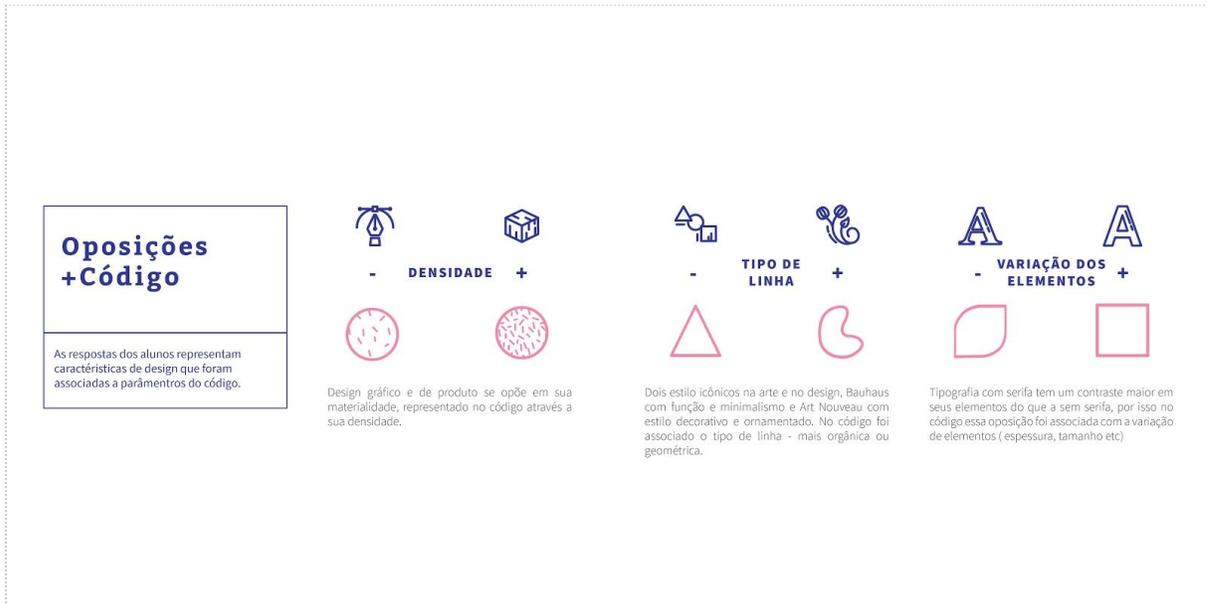
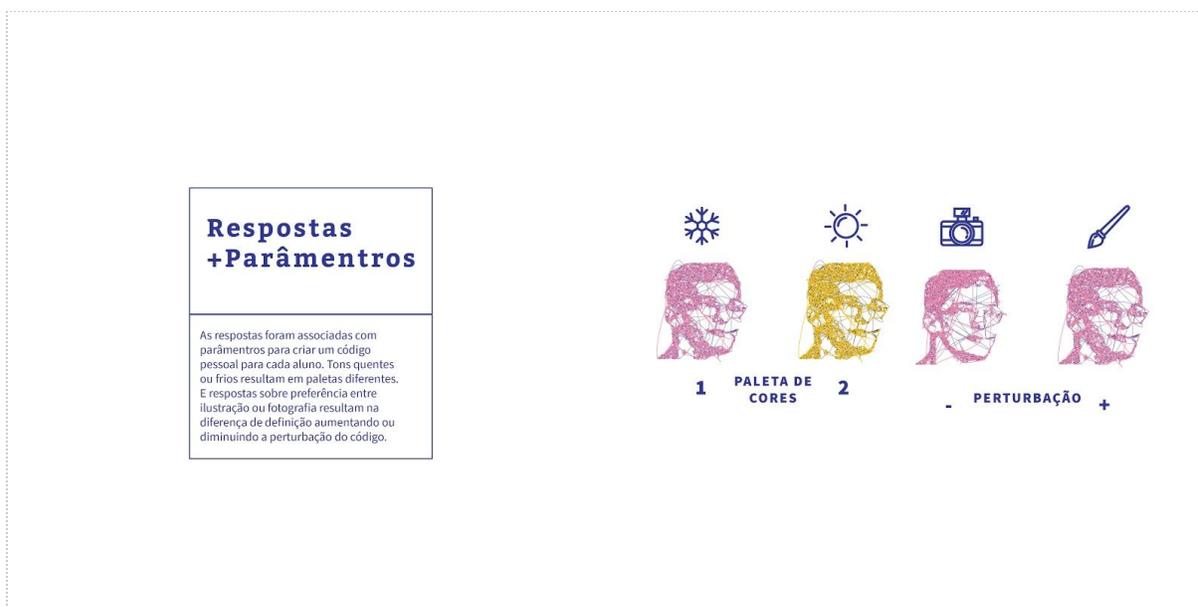


Figura 66 - Relação das respostas com os parâmetros.



Fonte - Elaborado pela autora, 2018.

7.6.3 Códigos

A terceira parte do livro mostra os os códigos. Primeiramente, apresenta o código original, seu autor, suas aplicações, como também um pequeno texto explicativo sobre o funcionamento dele, ao lado dos parâmetros que podem ser alterados.

Em seguida, são apresentadas as aplicações personalizadas do algoritmo com os estudantes, juntamente com seu nome e os ícones das respostas de suas preferências de design. Acima do ícone há a resposta escrita e abaixo o parâmetro ao qual ele está associado.

De acordo com os parâmetros houve códigos nos quais mais de uma aluno foi vinculado de acordo com suas respostas, possibilitando assim uma comparação naquele mesmo código.

Figura 67 - Abertura do capítulo de códigos.



Fonte - Elaborado pela autora, 2018.

Figura 68 - Página de apresentação do código "Gritador".

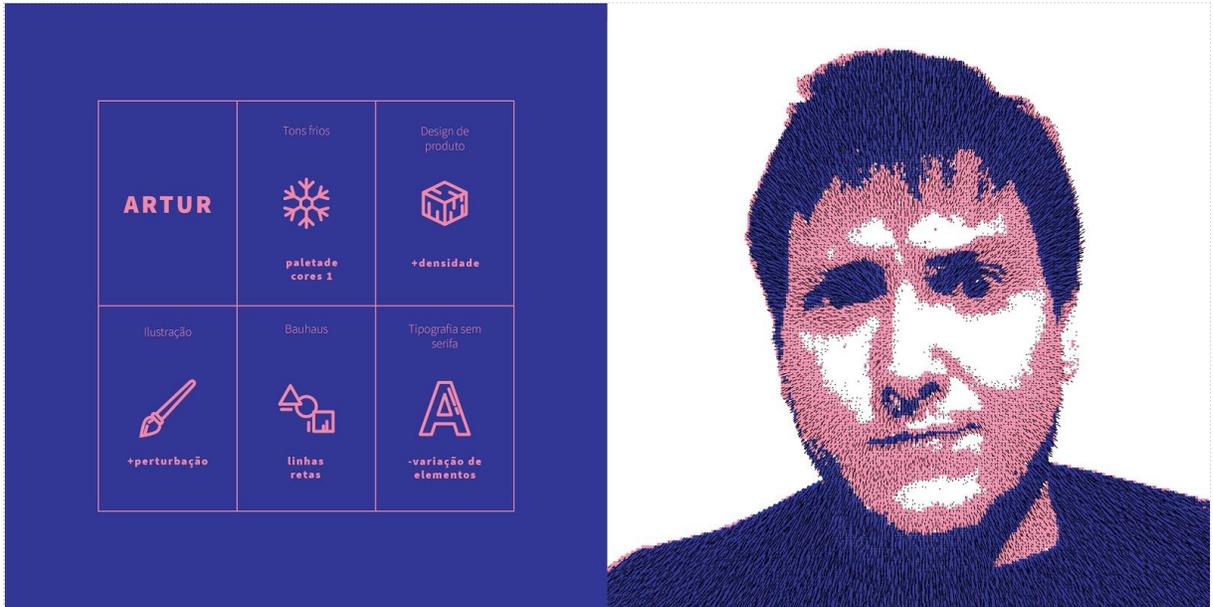
CÓDIGO GRITADOR
EUGÊNIO MOREIRA

Este algoritmo tem como objetivo principal potencializar características semânticas de imagens de grito, criando um efeito de linhas que irradiam de um foco escolhido pelo usuário.

PARÂMETROS DO CÓDIGO	
Perturbação	Espaçamento das linhas
Espessura das linhas	Comprimento das linhas
Cores	

Fonte - Elaborado pela autora, 2018.

Figura 69 - Página de aplicação do código "Gritador".



Fonte - Elaborado pela autora.

Figura 70 - Página de apresentação do código "Design por mulheres".



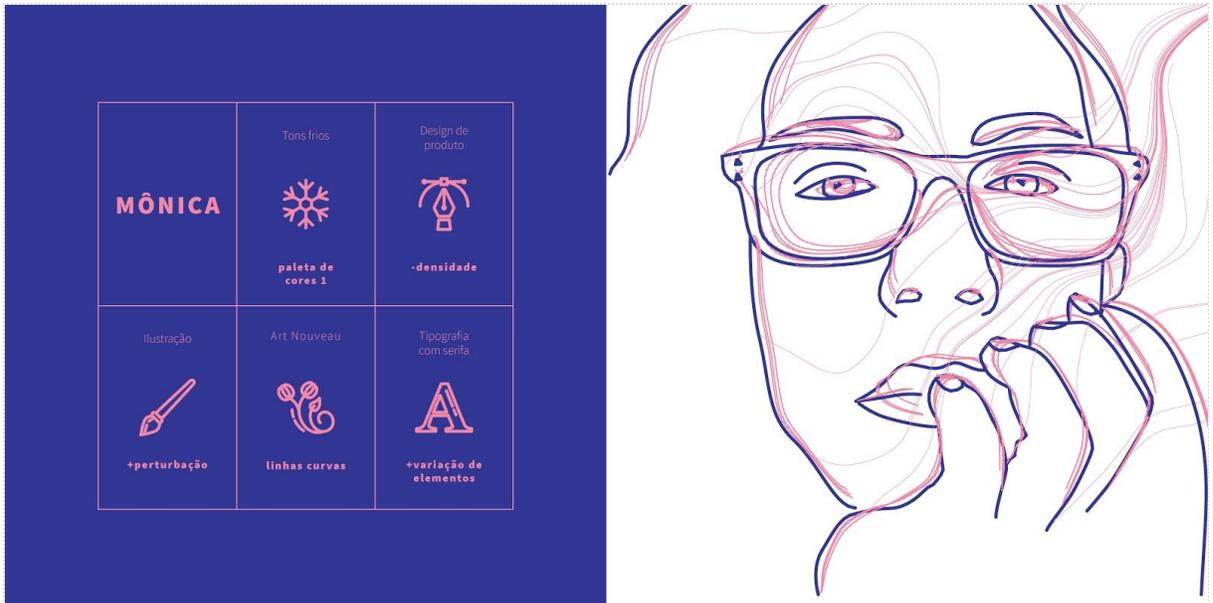
Fonte - Elaborado pela autora.

Figura 71 - Página de aplicação do código "Design por Mulheres".



Fonte - Elaborado pela autora, 2018.

Figura 72 - Página de aplicação do código "Design por Mulheres".



Fonte - Elaborado pela autora, 2018.

8. CONCLUSÃO

Dentro do contexto dessa pesquisa, conclui-se que a relação direta do designer com o código é acompanhada de dificuldades, como as observadas através de formulário aplicado em oficina de Tipografia Generativa no curso de design da UFC, que demonstrou falta de compreensão de sintaxe e de funções da programação por parte dos designers. Por outro lado, também foi observada a motivação dos alunos no uso do código como forma de expandir possibilidades aplicadas ao design. Ou seja, existe uma busca desse conhecimento, entretanto há, dificuldades de assimilá-lo.

Esse trabalho surge como reflexão dos problemas percorridos sobre aprendizado de programação, bem como análise de como essa relação entre design e programação de computadores pode acontecer para ser efetiva. Para isso foram feitos testes de métodos complementares de ensino e pesquisa bibliográfica que auxiliaram a compreender como, por meio do design, pode-se gerar estratégias para estimular o ensino e assim intervir na área da educação de programação.

A partir da metodologia de Design Science Research foi desenvolvido um livro que propõe uma experiência com o conceito de parâmetros associado a individualidade dos alunos, referente às suas preferências no universo de design, gerando um código personalizado para cada estudante. Ao fazer isso, são exploradas as possibilidades existentes na parametrização num código, e cria-se um reconhecimento do aluno com o resultado visual de suas escolhas.

Dessa forma, aliar a linguagem visual, reconhecimento pessoal e parâmetros abrem espaço para experimentos digitais e possibilitam que os alunos tenham acesso a novas experiências e explorem outras possibilidades no âmbito da arte, computação e design. Esta proposta tem o intuito de contribuir para reflexão de como ter um ensino mais motivador nesta área e potencializar seus resultados.

9. REFERÊNCIAS

Reas, Casey, and Chandler McWilliams. **Form+ Code: in design, art, and architecture.** Princeton Architectural Press, 2010.

BOHNACKER, H. **Generative design: visualize, program, and create with processing.** New York: Princeton Architectural Press, 2012.

Maeda, John. **Creative code.** London: Thames & Hudson, 2004.

Schön, Donald A. **Designing as reflective conversation with the materials of a design situation.** *Knowledge-based systems* 5.1 (1992): 3-14.

Mateas, Michael. **Procedural literacy: Educating the new media practitioner.** *On the Horizon* 13.2 (2005): 101-111.

Reas, Casey, and Ben Fry. **Processing: a programming handbook for visual designers and artists.** No. 6812. Mit Press, 2007.

Resnick, Mitchel, et al. **Scratch: programming for all."** *Communications of the ACM* 52.11 (2009): 60-67.

Tam, Maureen. **Constructivism, instructional design, and technology: Implications for transforming distance learning.** *Educational Technology & Society* 3.2 (2000): 50-60.

Gibbons, Andrew S., and Matthew B. Langton. **The application of layer theory to design: the control layer.** *Journal of Computing in Higher Education* 28.2 (2016): 97-135.

Kerren, Andreas, and John T. Stasko. **Algorithm animation** *Software Visualization*. Springer Berlin Heidelberg, 2002. 1-15.

Shapiro, R. Benjamin, and Matthew Ahrens. **Beyond blocks: Syntax and semantics.** *Communications of the ACM* 59.5 (2016): 39-41.

Peppers, Ken, et al. **A design science research methodology for information systems research.** *Journal of management information systems* 24.3 (2007): 45-77.

Vahldick, Adilson, et al. **O uso do Lego Mindstorms no apoio ao Ensino de Programação de Computadores.** *XV Workshop de Educação em Computação, Bento Gonçalves, RS*. 2009.

Wilson, Brent Gayle. **Constructivist learning environments: Case studies in instructional design.** Educational Technology, 1996.

Von Glasersfeld, Ernst. **Cognition, construction of knowledge, and teaching.** *Synthese* 80.1 (1989): 121-140.

Lebow, David. **Constructivist values for instructional systems design: Five principles toward a new mindset.** *Educational technology research and development* 41.3 (1993): 4-16.

Bonsiepe, Gui. **O design como ferramenta para o metabolismo cognoscitivo. Da produção à apresentação do conhecimento.** *Documento preparado para el Simposio Internacional de Investigación de Diseño Industrial" Ricerca+ Design. Politécnico de Milão: Maio. 2000.*

Arnheim, Rudolf. **Visual thinking.** Univ of California Press, 1969.

Wiley, David A. **Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy.** 2003.

Duran, Lena Ballone, and Emilio Duran. **The 5E Instructional Model: A Learning Cycle Approach for Inquiry-Based Science Teaching.** *Science Education Review* 3.2 (2004): 49-58.

Papert, Seymour, and Idit Harel. **Situating constructionism.** *Constructionism* 36.2 (1991): 1-11.

FLUSSER, V. **O mundo codificado: por uma filosofia do design e da comunicação.** Org. Rafael Cardoso. Trad. Raquel Abi-Sâmara. São Paulo: Cosac Naify, 2007.

MORAES, Dijon de. 2010. **Metaprojeto: o design do design.** São Paulo: Blucher.

Valente, José Armando. **Informática na educação: instrucionismo x construcionismo.** *Manuscrito não publicado, NIED: UNICAMP (1997).*

DRESCH, Aline et al. **Design Science Research.** Porto Alegre: Bookman, 2015.

Wing, Jeannette M. **Computational thinking.** *Communications of the ACM* 49.3 (2006): 33-35.

Grover, Shuchi, and Roy Pea. **Computational thinking in K–12: A review of the state of the field.** *Educational Researcher* 42.1 (2013): 38-43.

Baecker, Ronald. **Sorting out sorting: A case study of software visualization for teaching computer science.** *Software visualization: Programming as a multimedia experience* 1 (1998): 369-381.

10. ANEXOS

10.1. Formulário de perfil de curso

#	Em que semestre do curso você está?	Existiu algum projeto de design que você sentiu necessidade de saber programar ou usar programação? Como seria aplicada programação?	Qual o seu nível de receio em aprender programação?	Qual o seu nível de conhecimento em programação?	Você gostaria de aprender programação para criação de artes visuais?	{{(answer_54999022)}}; por quê?	tipografia generativa	logo combiante	arte generativa	nenhum	Other
1ae04ac0d3233fa0bc8b0c2dcd649732	7	Sim, através da logo combiante.	4	1	1	Porque proporciona uma gama de possibilidades a mais. E pode trazer uma riqueza maior ao projeto.		logo combiante	arte generativa		
214807460d8acf7e30e31935bf37e44	7	Seria interessante saber programar o básico para desenvolver sites e aplicativos, fazer com que o que desenvolvemos conceitualmente saísse do projeto e fosse concretizado.	4	0	1	Para conhecer novas possibilidades... apesar de acreditar que não seja estritamente necessário.		logo combiante			
e8e5b96c0b523ea43ca29724d2acecd	8	sim, para desenvolvimento de web/ aplicativo. Seria um nível basico de programação, mas como um dialogo com desenvolvedores.	4	2	1	Novas possibilidades criativas			arte generativa		
6f91ce4227ec18b0dbf60583e0a365d5	7	PG3	0	2	0	Não vale a pena				nenhum	
e0f92698b7d810f2b20ebbe6ba023626	7	não necessidade, mas seria um caminho interessante a tomar pra logos cambiantes, estudos responsivos etc.	2	1	1	pra aumentar as possibilidades de criação	tipografia generativa	logo combiante	arte generativa		
b57c462f9bf955c15a4609cbdeccc728	5	Ainda não	3	0	1	Para trabalhar de forma mais autônoma sem precisar pedir ajuda a algum programador		logo combiante	arte generativa		
88c8b05390e4716d747935634d446779	8	Sim	3	1	1	Porque tenho curiosidade e ampliaria minha área de conhecimento	tipografia generativa	logo combiante	arte generativa		
be9250b06d192b35459aaccca6bbc9fd	5	Não. As necessidades que eu possuíam podiam ser supridas por softwares automáticos.	1	2	1	Esse aprendizado apoiaria as possibilidades de construção de soluções para projetos.	tipografia generativa	logo combiante	arte generativa		Criação de objetos inteligentes, objetos generativos (Michael Hansmeyer: Building unimagina-ble shapes) etc
998e5b0461cd90546191b38dccc9c941	8	Não	4	1	1	Uma ferramenta importante e em ascensão.	tipografia generativa	logo combiante			
b3a49cd6e1584d42207099c7b6d81b41	7	Projetos no 5 e 6 semestre exigiram um nível de programação que não tínhamos conseguido suprir com a disciplina de pg3. E programação não é algo que conseguimos aprender em optativas de outros cursos pois são optativas que exigem pré requisitos que não temos. Trabalhos de web design são exigidos esse conhecimento que não nós é fornecido.	0	2	1	Programação não é uma obrigação do designer saber mas é uma ferramenta de crescimento para o seu trabalho.	tipografia generativa	logo combiante			
8bed3fd125d156ad131a04b3a02e73b	7	Ainda nenhum projeto senti necessidade, mas pode ser que precise para o TCC	4	1	1	É um diferencial esse tipo de conhecimento		logo combiante	arte generativa		

e404b604 8b8c75f1f 514ee770 7da314d	7	Como a maioria dos meus projetos abordam uma ótica tecnológica, e hoje se aproximam do vies da performance, sempre senti a necessidade do uso de princípios lógicos e da programação. Posso dizer que desde o segundo semestre quando foi sediado o SIGRADJ 2012 tenho feito uso ou do raciocínio ou da prática propriamente dita.	0	3	1	Primeiramente para associar essa visão aos meus projetos. Acredito que seria um acréscimo válido para as áreas gráficas, e porque também acredito que todo conhecimento é passível de ser utilizado pelo design.	tipografia generativa	logo combiante	arte generativa			produtos performáticos e interativos
5ea1332dc 83a1f22be 7acd73af 9ab17	7	Sim, em uma cadeira de PG3. Mas foi só o básico	3	1	1	Acho interessante		logo combiante	arte generativa			
aca80729 5d61e9e3 75ce0bc54 dab3079	7	não	4	0	0	Não me interesso por programação						nenhum
c698773e b9a944a0 a72c5c162 728494e	5	Desenvolvimento de sites na cadeira de PG, HTML	4	1	1	Porque eu acho muito incrível isso de desenvolver coisas a partir de códigos e tais, eu não entendo muito como funciona mas os resultados que eu já vi são bastante interessantes		logo combiante	arte generativa			
89be960cc cc964b314 09bc7bc9f 8ac97	7	Sim, instalação interativa museológica com o uso de processing e visão computacional	2	0	1	Para ampliar meus conhecimentos						site e instalações
645b9b76 38a097e8 7fe510bf4 56236c1	5	Sim, na programação do site em pg3	3	2	1	Poderia vir a ser uma ferramenta muito útil futuramente em algum projeto	tipografia generativa	logo combiante				
e1e1db00 96498ce8 bd967ebf0 2b8841f	7	Sim. Para sites e interfaces relacionadas a web	3	1	0	Porque é chato						nenhum
ab8c50bb 0b34bcf03 1031b6c2c 251d19	5	Sim. Fazer aplicativos de jogos.	4	0	0	Eu prefiro que o foco seja na experiência e na interface.						nenhum
50d602edf 9d8690d8 3393a0bc d5fca66	5	sim	4	0	0	Porque não sou programador. Sou designer!						nenhum
779674d6f cc9103870 f53db6e66 7e5cb	5	Sim. O projeto que estou fazendo nesse semestre: um jogo virtual usando html, etc.	0	1	1	Para facilitar a criação de mockups mais adequados com o resultado é, consequentemente, repassar melhor o que preciso para uma pessoa que fará isso.	tipografia generativa	logo combiante	arte generativa			
7620ce3c3 d42b9963 38f7280be 254d72	5	O professor solicitou um site para apresentação de um projeto, foi bem complicado pra mim por que não sabia nada de programação e tive que aprender na marra.	2	1	1	Com certeza, por que a gente vive numa época em que o mundo todo gira em torno do digital, do online, então para profissionais da nossa área, quanto mais nos especializarmos nesse sentido, melhor.						Sites, aplicativos ... etc
7f7779abc 490963ab 62c7e936 5b00b1a	5	PG3. Fazer o jogo / site	3	1	0	Quero aprender a parte prática e técnica aplicada a uma interface, seja jogo, app ou site		logo combiante				nenhum
5cf8cc337 e5912e9d 545024a9 988bb0e	3	Sou bolsista de um projeto de extensão na Fisioterapia e fui requerida para fazer um aplicativo para celular	1	1	1	Me identifico com a área e acredito que, aqui em Fortaleza, ela é ainda pouco explorada.	tipografia generativa	logo combiante	arte generativa			
8cab525f4 7baa2068 c58bf8aae 4ebfd1	5	Sim, a aplicação seria numa instalação interativa.	0	1	1	É um recurso que cria formas interessantes.		logo combiante	arte generativa			
8d918056 e1cb3005 888abb85 a8f4e16a	7	Interface PG4	2	2	1	Cuniosidade		logo combiante	arte generativa			

Rhinos 3D + Grasshopper	Conhecer a linguagem	Exercícios que demonstrem na prática o princípio que está sendo aplicado	2017-07-09 14:06:10	2017-07-09 14:11:01	ced1fda64f
Não	Sim. Identificar os códigos	Um programa mais intuitivo	2017-07-09 14:17:44	2017-07-09 14:20:14	f1d7758e33
	todas as dificuldades kk	não conheço programação o suficiente para entender de algo que possa facilitar	2017-07-09 14:42:00	2017-07-09 14:46:03	2804:7f7:e
Muse	Todas possíveis kkk	Não sei	2017-07-09 14:47:47	2017-07-09 14:52:35	2804:7f7:e
Não	Sim, não possuo conhecimento à respeito dos softwares.		2017-07-09 14:53:25	2017-07-09 14:59:48	698b6ef4ca
Não	Sim, tenho dificuldade em memorizar os comandos ou de encontrar onde estão os erros quando surgem ao longo do projeto	Não tenho ideia, talvez tornar de alguma forma uma experiência divertida para o usuário (?)	2017-07-09 15:49:58	2017-07-09 15:53:56	56956f0ef1
	Sim. Não entendo a linguagem.	Uso de códigos mais visuais	2017-07-09 16:04:09	2017-07-09 16:05:37	b2bb859170
dreamwave	Aprender os códigos.		2017-07-09 16:01:28	2017-07-09 16:06:34	a78f523fb9
Não	Não sei programar	Fazer uma graduação em Computação ou engenharia de softwares.	2017-07-09 16:53:13	2017-07-09 16:55:38	a78f523fb9
Sublime	Só consigo entender o básico: html/CSS. Não tenho muita facilidade com js e php.	Tornar mais simples a metodologia.	2017-07-09 16:54:00	2017-07-09 16:58:52	3bb6f7d7a1
Não sei nem se é programar, mas o Muse é um amorzinho <3 Grasshopper eu te odeio	Entender a lógica da coisa	mais aulas práticas durante as aulas	2017-07-09 16:43:53	2017-07-09 17:02:08	e58e60f5e2
			2017-07-09 17:17:14	2017-07-09 17:20:12	2804:7f7:e
nenhum	Sim, me exige um pensamento lógico que tenho dificuldade em usar	Coisas que fujam do convencional e estereótipo em programação	2017-07-09 17:16:00	2017-07-09 17:20:27	2804:7f7:e
	Sim, entender a localização e a função de cada tag.		2017-07-09 19:34:30	2017-07-09 20:35:37	a0666a2072
			2017-07-09 21:22:27	2017-07-09 21:23:47	05a4a15a3d

Existe algum software de sua preferência para programar? Se sim, qual e por quê?	Você tem dificuldade para programar? Quais?	Você sugere algo que poderia facilitar o aprendizado em programação?	Start Date (UTC)	Submit Date (UTC)	Network ID
Já usei o processing, mas não tenho software de preferência	Sim, acho muito trabalhoso	Talvez alguma metodologia que propiciasse códigos de base para determinadas tarefas de design, já que o foco seria design e não programação de código em si.	2017-07-01 18:12:19	2017-07-01 18:15:28	3b396bb4cc
			2017-07-01 18:13:10	2017-07-01 18:17:50	a29a5362b6
	SIM, a falta de pratica e o excesso de desinformação	entender as possibilidades acho uma coisa e também uma maior rede de contato entre artistas e designer que programam. Para separar mais a barreira desenvolvimento versus design.	2017-07-01 20:30:43	2017-07-01 20:35:04	f5777b0c7d
Processing. Facilidade	Falta de vontade	Algo que não deixe o aprendizado maçante	2017-07-01 23:19:50	2017-07-01 23:21:55	8bc4973572
nope	do pouco que tentei, acho difícil memorizar e entender o funcionamento dos comandos		2017-07-04 14:03:02	2017-07-04 14:05:58	2c514f358e
Não conheço nenhum, ainda não cursei PG3	Nunca tentei		2017-07-08 21:05:03	2017-07-08 21:07:37	7ef829fcoe
Não, porque não conheço nenhum.	Sim, a série de combinações de códigos	Simplificação de comandos e códigos	2017-07-09 00:49:50	2017-07-09 00:57:00	2804:14:c:d
Não	Decorar os comandos	Um "app" para programar que usa tecnologia preditiva para sugerir os possíveis comandos que você vai utilizar em seguida.	2017-07-09 00:50:42	2017-07-09 01:02:37	9e4238dc5f
Não	Trabalhar com números para criar visualidade	Não sei	2017-07-09 13:53:08	2017-07-09 13:55:35	2804:d4b:3
Não	No pouco que eu tive contato com o aprendizado de PG3 eu não tive dificuldades. Acho que a maior dificuldade é achar onde aprender.	Acho que por ser alguém complexo exigira um nível de projeto bastante avançado para conseguir simplificar esse processo.	2017-07-09 13:49:00	2017-07-09 13:56:14	50299b309f
Não	Conseguir aprender a base da programação	Nunca parei pra pensar nisso	2017-07-09 13:53:03	2017-07-09 13:56:33	5209ef3627

10.2. Formulário de feedback da oficina 1

#	Qual o seu nome?	Quais foram as suas dificuldades para entender o código?	O tempo foi suficiente para assimilar o código?	Escrever o pseudo-código no papel contribuiu para assimilar a lógica de programação?	Other	Se o pseudo-código fosse aplicado no início da oficina facilitaria o entendimento?	arte generativa	logo cambiante	instalações interativas	Other	Algo mais poderia facilitar o entendimento do código para você?	Start Date (UTC)	Submit Date (UTC)	Network ID
ca536682741a099c3603915e03cf8e95	Michael Almeida	Poucas, tenho certa experiência em programação.	1			Por possuir experiência com programação, o pseudocódigo não ajudou muito.	1				Estudar um pouco mais a documentação da linguagem para entender quais as funções prontas.	2017-06-07 11:14:07	2017-06-07 11:19:40	91488f550a
b763b3adc3932c65f7d80d697d605187	Renata Pinheiro de Almeida	As hierarquias do código	1	sim		1		logo cambiante			Uma maior explicação no que diz respeito à lógica da linguagem dos códigos.	2017-06-07 11:20:10	2017-06-07 11:25:02	ce648da465
8ad3de471655769a09201cd850a804d5	Breno morais	Decorar o que fazer	0	sim		1		logo cambiante			Acredito que o tempo	2017-06-07 11:22:10	2017-06-07 11:40:10	d42890035e
0f38dd5ce7d9eec7763b192b55e87764	Adriano Gomes	A lógica das operações. Pq estou executando cada ação?	0	sim		1	arte generativa	logo cambiante	instalações interativas		Desmistificar a lógica da programação. A que se refere os comandos. Linkar tudo com as técnicas manuais. Tipo, pra desenhar um círculo eu preciso ter raio, diâmetro, etc etc. Então no código isso é raio, isso é diâmetro etc etc	2017-06-07 11:26:27	2017-06-07 11:43:19	c18d1656d9
ec7f1879605606be53a41ds588e85931	Isabelle Medeiros da Cunha	As nomenclaturas e saber o que cada uma fazia	0	não		1	arte generativa	logo cambiante			Saber o que cada código significa e saber as "regras", tipo, não existe maiúsculas, fechar sempre com ; exceto em tais ocasiões etc	2017-06-07 11:46:15	2017-06-07 11:50:35	2804:d4b:2

8e507569 ed63b280 728401ab 015c110e	Roderico Neto	Entender o que o curveVert ex fazia	0	sim	1	arte generativa	logo cambiante	instalações interativas		Acho que precisava de mais tempo mesmo. Para primeiro mostrar como o código funciona, estando pronto e visualizan do as etapas, e depois fazer do zero, com mais tempo hehe.	2017-06- 07 12:41:42	2017-06- 07 12:44:50	9c7edd5f3 b
c40089e5a 40249057 434caa4e6 c70c81	Adriano da Costa Silva	houvem funções desconhec idas por mim, foi bem fácil entender o que estava sendo feito. Toda a parte do código que acompanh amos na oficina, pelo menos. Já a parte que recebemos produzida, por motivos de	1	sim	1	arte generativa	logo cambiante	instalações interativas		introdução ao processing ou a outros programas que fossem utilizados em outras oficinas. A parte inicial na qual o prof. Roberto falou sobre duas funções simples que seriam usadas e os momentos em que ele	2017-06- 15 02:38:01	2017-06- 15 02:53:29	707c7fa8b e
52ce4b92 e61d42caa 3a77fcfb 8130b	Gabriel da Silva Ferreira	Como eu nunca tive experiência com programação, eu acabei esquecendo o algumas coisas rapidamente.	1	sim	1	arte generativa		instalações interativas			2017-06- 15 19:31:47	2017-06- 15 19:34:42	2804:7f7e
f6cb2ff58c 39bab693 275231f1f 5e471	MARLEY FARIAS	ENTENDER O POSICION AMENTO	1	sim	0	arte generativa	logo cambiante			SERIA ÓTIMO UMA OPTATIVA COM MAIS TEMPO PARA ESTE ASSUNTO	2017-06- 18 21:58:12	2017-06- 18 22:00:16	b59b0f78d b
46c232c16 cb5b34c94 606d6aa5 d8dcde	Diogo Torres	As dificuldades foram principalmente relacionadas ao conhecimento mais teórico contido no código	1	sim	1	arte generativa	logo cambiante	instalações interativas	Design pra web	talvez um conhecimento prévio das lógicas de programação para maior assimilação	2017-06- 18 22:52:08	2017-06- 18 23:06:12	b5edce2d 4b
38798dc9f ceed4389 75a7b96a 559a06d	Rebeka Samyrra	a falta de conhecimento do assunto, do que cada função significa e tal	1		Um pouco sim, mas queria ter entendido mais de sintaxe, do que é objeto, variável e tal.	0	arte generativa			Sim, explicar como funciona o método de escrita, de sintaxe da programação etc.	2017-06- 24 17:38:23	2017-06- 24 17:45:21	c6129676 2a

10.3. Formulário de feedback da oficina 2

#	Qual o seu nome?	Quais foram as suas dificuldades para entender o código?	O tempo foi suficiente para assimilar o código?	Escrever o pseudo-código no papel antes de escrever o código contribuiu durante o processo?	Se sim, qual foi a contribuição do pseudo-código?	Que tipo de aplicações entre design e programação você gostaria de aprender?
27320831 381b601a 385d3080 ea4fd295	Adriano da Costa Silva	Embora houvesse funções desconhecidas por mim, foi bem fácil entender o que estava sendo feito. Toda a parte do código que acompanhamos na oficina, pelo menos. Já a parte que recebemos produzida, por motivos de economia de tempo, ficou um pouco de lado.	0	1	Ele cria um passo a passo daquilo que você quer fazer e, na hora de codificar, é só saber as funções que o programa pede	Logo cambiante
68304526 72787101 e4725a75 17684a1e	Lucas	A quantidade de dados me deixou confuso algumas vezes	1	1	Consegui criar etapas que ajudaram a pensar melhor as partes do código	Logo cambiante

Other	Algo mais poderia facilitar o entendimento do código para você?	Start Date (UTC)	Submit Date (UTC)	Network ID
	Uma introdução ao processing ou a outros programas que fossem utilizados em outras oficinas. A parte inicial na qual o prof. Roberto falou sobre duas funções simples que seriam usadas e os momentos em que ele explicava o que a linha do algoritmo representava foram ótimos para o entendimento de maneira geral	2017-06-25 18:27:56	2017-06-25 18:32:07	707c7fa8be
	Mais oficinas e com tempo maior :)	2017-06-26 15:16:44	2017-06-26 15:18:51	dcbc420f9d

