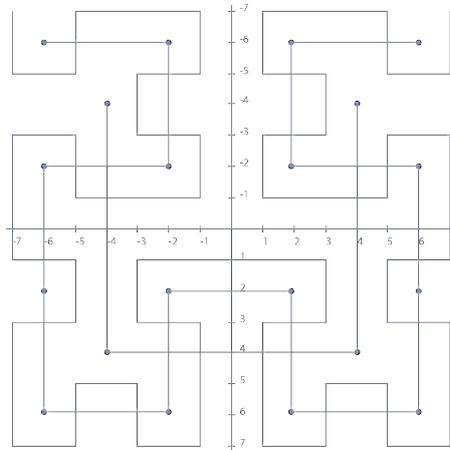


UNIVERSIDADE DE BRASÍLIA
INSTITUTO DE ARTES
PROGRAMA DE PÓS-GRADUAÇÃO EM ARTE E TECNOLOGIA



**DESENHO *SCRIPTIVO*: PROGRAMAÇÃO GRÁFICA APLICADA À
TRANSFORMAÇÃO DE TEXTO EM IMAGEM.**

por
FERNANDO AQUINO MARTINS

Dissertação submetida à avaliação como requisito parcial
para a obtenção do grau de Mestre em Arte e Tecnologia

Prof. Dr. Rogério Câmara
Orientador

Brasília, maio de 2014

**DESENHO *SCRIPTIVO*: PROGRAMAÇÃO GRÁFICA APLICADA À
TRANSFORMAÇÃO DE TEXTO EM IMAGEM.**

UNIVERSIDADE DE BRASÍLIA
INSTITUTO DE ARTES
PROGRAMA DE PÓS-GRADUAÇÃO EM ARTE E TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA - UnB
INSTITUTO DE ARTES - IdA
PROGRAMA DE PÓS-GRADUAÇÃO EM ARTE E TECNOLOGIA – PPG-Arte

AGRADECIMENTOS

Jesus Cristo_ O Caminho, A Verdade e A Vida

SUMÁRIO

LISTA DE FIGURAS	07
LISTA DE TABELAS	08
LISTA DE CÓDIGOS	08
RESUMO	09
CAPÍTULO 1 — INTRODUÇÃO	10
1.1 Considerações iniciais	10
1.2 Caracterização do problema	11
1.3 Objetivo	12
1.4 Justificativa	12
1.5 Metodologia da pesquisa	13
1.6 Organização da dissertação	15
CAPÍTULO 2 — O DESENHO <i>SCRIPTIVO</i>	16
2.1 Desenho e <i>script</i>	18
2.2 Construção de um desenho a partir de um texto	21
2.3 Comunicação gráfica de ideias	24
2.4 Máquina (sem rodas)	29
Descrição A	30
Descrição B	31
2.5 O desenho contido em uma fórmula	35
2.6 Uma geometria sem distância	41
2.7 Recursão e repetição	45
CAPÍTULO 3 — DETALHES DE IMPLEMENTAÇÃO	54
3.1 Arquitetura cliente/servidor	58
3.2 Desenhando no lado cliente	61
3.3 Desenho vetorial para <i>web</i>	69
3.4 Comparação entre <i>SVG</i> e <i>canvas</i>	73
3.5 Desenho no lado servidor	75

3.5.1	Construção de Polígonos	78
3.6	Desenho distribuído	82
3.6.1	AJAX	85
3.6.2	CGI	87
3.6.3	Ambiente de hospedagem	89
 CAPÍTULO 4 — PROCESSO DE TRABALHO		92
 CAPÍTULO 5 — CONCLUSÃO		105
 BIBLIOGRAFIA		108

LISTA DE FIGURAS

Figura 1 — Geometria descritiva de Dürer. -----	22
Figura 2 — Construção do triângulo retângulo partindo de uma reta simples. -----	23
Figura 3 — Uma possível ontologia para a geometria. -----	26
Figura 4 — Sequência de construção e desconstrução. Lógica natural? -----	26
Figura 5 — The Half Triangle, de Raymond Lull. -----	29
Figura 6 — Triângulo da inteligência <i>web</i> . -----	32
Figura 7 — Comparação das operações de uma empilhadeira. -----	35
Figura 8 — Exemplo de desenho contido em uma fórmula. -----	36
Figura 9 — Triângulo de Ouro Renascentista. -----	37
Figura 10 — Plano coordenado para <i>web</i> . -----	38
Figura 11 — Manfred Mohr. "The cubic limit series". -----	44
Figura 12 — Uma imagem recursiva. -----	45
Figura 13 — Curvas de Hilbert de ordem 1, 2, 3. -----	46
Figura 14 — Demonstração de um desenho construído a partir de uma regra. -----	47
Figura 15 — Demonstração de um padrão recursivo possível, mas restrito. -----	47
Figura 16 — Acerca do Ritornelo, A máquina de gorjear. -----	48
Figura 17 — Curvas de Hilbert de 4ª ordem no plano coordenado. -----	50
Figura 18 — Curvas de Hilbert de ordem $H_1... H_5$. -----	51
Figura 19 — Cubos de cor em unidades RGB e CMYK. -----	55
Figura 20 — Diagrama de formato de imagens. -----	57
Figura 21 — Arquitetura cliente/servidor -----	59
Figura 22 — Arquitetura cliente/servidor -----	60
Figura 23 — Desenho do lado cliente com o elemento <i>canvas</i> HTML 5 e <i>JavaScript</i> . -----	62
Figura 24 — Coordenadas do contexto 2D de <i>canvas</i> -----	62
Figura 25 — Um quadrado escrito em <i>JavaScript</i> e plotado em <i>canvas</i> HTML5. -----	65
Figura 26 — Mecanismo. -----	66
Figura 27 — Curva de Sluse escrita em <i>JavaScript</i> . -----	67
Figura 28 — Alavancas baseadas nas condições do mecanismo. -----	68
Figura 29 — Estrutura de desenho vetorial com <i>SVG/XML</i> . -----	70
Figura 30 — Um quadrado vetorial escrito com <i>SVG/XML</i> . -----	72
Figura 31 — Um desenho recursivo vetorial escrito com <i>SVG/XML</i> . -----	73
Figura 32 — Análise de cenários para escolha entre <i>canvas</i> e <i>SVG</i> -----	74
Figura 33 — Desenho no lado do servidor plotado como <i>bitmap</i> com <i>GD/Perl</i> -----	76
Figura 34 — Um quadrado escrito em <i>Perl</i> . -----	78
Figura 35 — Curva "W" de 1ª ordem. Matriz de pontos sobre o plano. -----	82
Figura 36 — Lógica distributiva presente em uma Curva "W" de 2ª Ordem. -----	85
Figura 37 — Diagrama de uma requisição síncrona tradicional -----	86

Figura 38 — Diagrama de uma requisição assíncrona com AJAX	86
Figura 39 — Diagrama de uma seção CGI	88
Figura 40 — Curva “W” de 2ª Ordem. Método de trabalho com pixel.	89
Figura 41 — Ambiente de trabalho em um servidor de hospedagem.	91

LISTA DE TABELAS

Tabela 1 — NBR 8403 - Tipos de linhas e sua aplicação.	28
Tabela 2 — Gráfico e valores para $y = x^2$	39
Tabela 3 — Gráfico e valores para $x = y^2 - 2y$	39
Tabela 4 — Gráfico e valores para $y = 1/x$	40
Tabela 5 — Formatos de imagem e suas características.	56
Tabela 6 — Métodos e atributos da API de <i>canvas</i>	63
Tabela 7 — Elementos e atributos <i>SVG/XML</i>	70
Tabela 8 — Comparação entre desenhos feitos em <i>SVG</i> e <i>canvas</i>	75
Tabela 9 — Primitivas e métodos auxiliares de <i>GD Perl</i>	77
Tabela 10 — Verificação da política de mesma origem.	84
Tabela 11 — Métodos do objeto <i>XMLHttpRequest</i>	86
Tabela 12 — Variáveis de ambiente CGI.	87

LISTA DE CÓDIGOS

Código 1 — Curva de Sluse escrita em <i>JavaScript</i>	67
Código 2 — Desenho recursivo escrito em <i>SVG/XML</i>	72
Código 3 — Polígono escrito em <i>Perl</i>	81
Código 4 — Desenho distribuído.	90

RESUMO

Este trabalho se insere no campo da arte e tecnologia. Investiga a construção de imagens em ambientes virtuais e está fundamentado na disciplina desenho. O objetivo é desenvolver um conjunto de ferramentas e trabalhá-las pela óptica da programação gráfica, da filosofia e da linguagem, fornecendo assim um material relevante sobre leitura, manipulação e gravação de imagens ou sequência de imagens a partir de um texto. A metodologia seguiu os pressupostos da teoria aliada à prática, trazendo diversificados exemplos de metalinguagem entre a imagem e o texto e também demonstrações sobre geometria e tecnologia. A conclusão desse estudo sugere que o desenho é uma ferramenta de trabalho e ensino que pode ser operacionalizada por meio de algoritmos e estrutura de dados. Sendo uma atividade construtiva, é estratégico inserir em nosso dia-a-dia a problemática e a geometria das formas no espaço, assim como os fundamentos da gestão informativa e do processo de trabalho em arte.

Palavras-chave: Desenho, Programação Gráfica, Texto.

CAPÍTULO 1

Introdução

1.1 Considerações iniciais

A internet é um ambiente que sempre desafia a área de criação gráfica. Hoje, com o crescimento do volume de dados provindos das redes de comunicação, torna-se necessário a criação de novas dinâmicas para o processamento e a interação entre informação e imagem, visualização de dados ou mesmo interfaces dinâmicas. A maioria dos dispositivos ainda necessitam de uma saída gráfica para sua operacionalização. Câmeras fotográficas se popularizaram, celulares tiram fotos e postam instantaneamente o conteúdo nas redes sociais. Aplicações de análise de dados registram suas estatísticas em gráficos e geram tabelas para visualização de rendimentos, preços de produtos, estoques e controle de fluxos. Infográficos auxiliam o aprendizado, sintetizam estruturas de DNA, ensinam sobre cadeias alimentares, ontologias de espécies, produção e consumo de alimentos.

A cada dia, os programas tornam-se mais sensíveis ao contexto, ou seja, transformam-se de acordo com geolocalização, perfil e outros dados pessoais do cliente. As interfaces se diversificaram na interação com o sistema, sendo ativadas por meio de gestos, sons e toques. No espaço plano do ecrã é possível simular multidimensões de imagens, captadas pelas câmeras do usuário ou provindas de outras fontes distribuídas.

Atualmente, a área de computação gráfica (CG) está presente nos mais diversos segmentos: arte, medicina, diagnósticos, todas as áreas de engenharia (mecatrônica, civil, aeronáutica e as outras), meteorologia, marketing, segurança pública, turismo, processamento de dados, indústria, educação, tratamento de imagens, desenvolvimento motor, paisagismo, arquitetura, geografia, colheitas, astronomia, planejamento, por exemplo (Gianetti, 2006), (VIANA & MELO, 2008) e (DOMINGUES, 1997).

Basta pensarmos em nossas tarefas cotidianas, para se evidenciar o fato de que, a todo momento, estamos envoltos por imagens sintéticas e interativas. As operações de corte, cópia, tratamento e outras formas de interação digital, tornaram-se simples ferramentas do dia-a-dia, como se fossem garfos, martelos ou pincéis. A labuta sobre a imagem não é mais

restrita a um grupo privilegiado, como nos períodos do renascimento e do barroco, mas está destinada ao povo, milhões de pessoas, que manipulam, editam e compartilham suas experiências visuais. Por esse motivo, não há dúvida de que a nossa relação com a imagem está em processo de transformação.

1.2 Caracterização do problema

Como estudar o desenho na sociedade em rede? Como trabalhá-lo como disciplina, exata e tecnicamente precisa? Como desenvolver esse caráter técnico, que é uma das faces do desenho, considerando que está sendo esquecida dentro do campo artístico.

Este trabalho de mestrado se insere na linha de pesquisa Arte e Tecnologia, do Programa de Pós-Graduação em Arte – PPG-Arte, da Universidade de Brasília – UnB. A Arte é o local desta pesquisa. A disciplina histórica Desenho é o objeto.

Sendo o desenho planejamento, engenharia e conhecimento, além de outras práticas, como então, direcionar um estudo que satisfaça as prerrogativas de concentração e precisão técnica?

A Física se matematizou nos séculos XVI e XVII; a Lógica se matematizou nos séculos XIX e XX (SILVESTRE, 2011); e o desenho, para permanecer na era da informática, também terá que se matematizar. É difícil pensar um desenho digital sem a lógica matemática. O estudo da imagem na *web* está parado nas predefinições colocadas por aplicativos vários, com intuito de comércio e dominação. Nesse sentido, para trabalhar o desenho na *web*, é preciso partir para o código, pensar o suporte e as suas recursões.

Eis o problema aqui enfrentado. Trata-se de um passo que, com certeza, não será o primeiro nem o certo. O fato é que a imagem está mistificada em uma mágica incompreensível que fascina e controla. Então torna-se necessário se apoderar da imagem, conhecê-la. O método aqui apresentado é o conhecimento pelo desenho e a concentração dessas operações no plano coordenado.

1.3 Objetivo

O objetivo dessa dissertação é desenvolver um conjunto de ferramentas de desenho e trabalhá-las pela óptica da programação gráfica, da filosofia e da linguagem, fornecendo assim um conjunto prático e teórico sobre leitura, manipulação e gravação de imagens ou sequência de imagens a partir de um texto.

1.4 Justificativa

Atualmente existem várias abordagens que tratam a questão da Arte e da Tecnologia em suas mais diversas linguagens e procedimentos (VENTURELLI, 2008). Em relação ao desenho, essas perspectivas demonstram apenas procedimentos tradicionais da linguagem, estruturados em visualizações digitais. Há pouca inovação e poucos estudos que encaram as primitivas da tecnologia concatenadas com as primitivas da arte. Também, na arte em geral, é difícil encontrar pesquisas que procedem uma revisão histórica da disciplina Desenho e ainda há poucos estudos que avançam esse desenvolvimento no contexto das novas tecnologias (LAS-CASAS, 2008), (BORGES & JENSEN, 2014) e (SANTOS, 2000) .

Em face desse cenário, considera-se prioritário desenvolver o desenho pela via das ferramentas, para um ponto em que elas permitam ao povo investigar processos de um fenômeno natural e transferí-los para o mundo das técnicas. Há uma evidente necessidade, principalmente na sociedade brasileira, de desmistificar a imagem. É preciso revisar o trabalho e as ferramentas de manipulação da forma e do espaço. Eis a essência da disciplina Desenho. A revisão é necessária, mas é tarefa imensa que não pode ser vencida em uma dissertação de mestrado.

A experiência adquirida no trabalho e na manipulação de máquinas diversas, seja na indústria de lapidação de pedras, de beneficiamento de grãos e na própria computação, propiciou ao autor dessa dissertação o entendimento de que os mecanismos devem estar inseridos no seio das comunidades, assumindo um papel central no cotidiano dos trabalhadores. A máquina é o elemento que realiza a transformação da matéria prima, é o instrumento que permite ao operário a manipulação dos elementos da natureza e traz a dinâmica para o interior do trabalho. Isso podemos constatar com o advento da informática. Contudo, a tecnologia digital está se distanciando desse processo.

Justifica-se, portanto, a necessidade de estudar, sistematizar e construir ferramentas gráficas que se ajustem a nossa realidade, assim como um dia o fizeram países como Alemanha, Estados Unidos e Rússia. Não apenas adquirir *photoshops*, *dreamweavers* ou *autocads*, mas inserir no dia-a-dia a problemática e a geometria das formas no espaço, sendo este é o papel histórico da arte que agora ressurge com urgência, ou seja, tornar-se vanguarda no processo de renovação da educação. De certa forma, este trabalho aponta para essa direção.

1.5 Metodologia de pesquisa

Como é bastante comum no campo da arte e também da tecnologia, adotou-se como metodologia o procedimento da teoria aliada à prática, ou seja, a conceituação do objeto e depois sua construção. Primeiro, porque o Desenho é, em essência, uma aplicação teórica sobre o papel, assim como a tecnologia digital também pressupõe a implementação de um objeto em um ambiente virtual. Portanto, se entendeu como necessário o uso deste método em uma pesquisa sobre desenho digital implementado na internet.

Ademais, ao considerar o objetivo subjacente deste trabalho, que é a representação do espaço no plano, podemos considerar que esse movimento pressupõe a síntese da realidade no ecrã. Não há teoria que satisfaça esse pressuposto, é preciso praticar. O desenho é antes de tudo uma aplicação prática, estatuto natural que objetiva o delineamento de uma teoria e a sua demonstração no papel. Então, o protocolo de pesquisa foi a própria demonstração, podendo ser exemplificada com a consideração do geômetra e astrônomo alexandrino-grego Apolônio de Perga (c.262-190 a.C.) que considerava a validade de seus estudos "pelo simples fato de serem demonstrações". A prova de validade de um desenho é a sua própria execução.

Assim, a prática contou com aproximadamente 1000 horas de programação, somente para testar os objetos virtuais discutidos nesse trabalho. Os testes foram guiados por estudos de caso em trigonometria coordenados com a programação gráfica, notadamente em linguagens de *script*, como *Perl*, *PHP*, *SVG* e *JavaScript*. Tidas como ferramentas de trabalho, assim como pincéis, lápis e régua, intentou-se chegar a um conjunto de primitivas para desenho *scriptivo* que melhor representasse a coalescência entre texto e imagem.

Sendo o desenho uma linguagem que trabalha essencialmente com o risco sobre o suporte, todo conteúdo foi agregado sob essa constante. Por exemplo, a curva de Hilbert de 4ª ordem apresentada na *Figura 17* foi ordenada como uma sequência de coordenadas, algo que pode muito bem ser entendido como um movimento de imagens no plano, muito utilizado em vídeos e animações. Mas esse fator não está descrito explicitamente, visto que implicaria um detalhamento adicional que fugiria aos objetivos do trabalho. Por isso, optou-se por trabalhar cada exemplo como objeto e, ao conjunto de objetos, a sua explicação; uma tradição da disciplina desenho, praticado por (Vitruvius, 2007), (Vignola, 1562), (Dürer, 1525), (Da Vinci, 1478-1519) e (Artobolevski, 1976).

No primeiro momento foi feita a revisão da literatura e a definição dos parâmetros de apresentação deste estudo. Os parâmetros foram: demonstração pelo método geométrico, descrição das construções pela via filosófica e implementação dos resultados no domínio *www.desenhoscriptivo.com.br*. A cadência do texto foi considerada prioridade, visto ser assunto de relativa complexidade e com forte tendência ao enfado. Por isso, as descrições quantitativas dos elementos foram suprimidas e em seu lugar foi ressaltada a modularidade e qualidade das operações para o simples entendimento, considerando que um objeto explica o outro, numa estrutura que compreende mais uma lemniscata do que uma exponencial.

Essas definições somente foram compreendidas após a fase da qualificação, onde a banca apontou repetidamente a ausência de uma forma textual bem definida, além de outras questões como a falta de foco e a quantidade excessiva de dados. Então, a primeira atitude foi a eliminação de 80% do texto apresentado na qualificação e a reconstrução do objeto a partir do campo prioritário da programação gráfica. O *Capítulo 3* foi completamente adicionado e o *Capítulo 4* totalmente reformulado.

Por fim, a metodologia foi desenvolvida como uma tentativa de se realizar uma metateoria, pois assim exigiu o objeto, de forma que o texto construísse a imagem e a imagem fosse proposição para o texto. Antes de mais nada, trata-se de um trabalho de desenho, disciplina fundamental para o exercício do conhecimento com vistas à compreensão do universo que nos cerca.

1.6 Organização da dissertação

A presente dissertação é constituída por 4 capítulos, ou seja, a Introdução e outros 3 capítulos cujos conteúdos estão descritos a seguir.

O *Capítulo 2* apresenta as características do que se convencionou chamar Desenho *Scriptivo*. A abordagem é feita com base numa geometria filosófica que pretende traçar uma ontologia entre desenho e tecnologia. Trata-se de um objeto baseado em sete métodos: definição, construção, comunicação, máquina, fórmula, distância e repetição. Os elementos vão sendo apresentados em metalinguagem — texto e imagem — até se chegar à *Curva de Hilbert de 4ª ordem no plano coordenado*, considerado um elemento síntese dos conceitos trabalhados no capítulo.

O *Capítulo 3* trata dos detalhes de implementação e descrição de ferramentas específicas de programação gráfica. Aqui, a preocupação principal foi a de fornecer um conjunto significativo para o Desenho *Scriptivo*. O processo construtivo foi o desenho distribuído, ou seja, um desenho interativo composto de partes oriundas de diversas origens e montadas em um único lugar. Optou-se por 3 ferramentas de desenho e 2 ferramentas de comunicação. As ferramentas de desenho foram *canvas HTML5* e *JavaScript*, para um desenho do lado do cliente; *SVG/XML* para um desenho vetorial na *web*; e *GD Perl* para um desenho do lado do servidor. As ferramentas de comunicação escolhidas foram *AJAX* e *CGI*.

No *Capítulo 4*, que trata do processo de trabalho, foram apresentados 10 Desenhos *scriptivos* como proposição. Os desenhos deste capítulo são implementações que representam a síntese conjuntiva desta dissertação, pois são compostos por diversos elementos de linguagem como número, título, desenho, descrição e fórmula, perfeitamente concatenados, servindo como prova e exemplo de transformação dinâmica entre textos, regras e imagens. Os exemplos foram rigorosamente produzidos segundo critérios semânticos, estruturais e construtivos. Funcionam como uma série padronizada de exemplos que compartilham de uma filosofia-geométrica para o Desenho *scriptivo*.

CAPÍTULO 2

O desenho *scriptivo*

Um texto sobre a história da linguagem desenho, ou uma ontologia sobre o ato de desenhar, não poderia ser construído sem uma menção especial à linguagem escrita. Alguns exemplos, surgidos no encontro entre indígenas e europeus, como o dos índios tupiniquins, que não conseguiam reproduzir seus desenhos corporais com lápis e papel, uma vez que estavam acostumados com a pintura no espaço flexível e tridimensional do corpo humano. Outro exemplo, a menção indígena “*ilhuicactlapaltlachia*” para cor do céu, que não significa uma cor específica (azul ou alaranjado), mas algo que contém uma variável dependente de tempo, espaço e percepção. Também a observação feita por Jacques Derrida (1973) sobre a denominação Nhambiquara para o ato de escrever, “*iekariukedjutu*”, isto é, “fazer riscos”. Estes exemplos indígenas são claros quanto a coalescência ou cisão entre desenho e escrita.

De fato, a construção de imagens a partir de textos é uma atividade cheia de desafios e incongruências, algo que, dito de forma espontânea, pode ser colocado em paralelo com o problema da quadratura do círculo, um clássico da antiguidade. Parece mesmo não ser possível a resolução da questão, mas somente uma aproximação.

Afinal, o que é uma imagem? O que é um texto? É possível descrever todas as imagens, é necessário ver para ler? Seria possível um cego aprender a desenhar com base na relação xy de um plano coordenado? É possível criar um algoritmo que reproduz uma pintura em camadas?

Imagens geradas por fórmulas matemáticas, ou mesmo descritas por textos, trazem implícitas esta relação com a realidade e o mundo físico. Não há separação nesse sentido. Portanto, fazer transições entre imagens e textos não é uma medida simples, uma vez que explora diversos limites, entre redes, semânticas, percepções e lógicas. É um trabalho que exige engenho e arte.

Assim como nas ciências exatas, que descrevem processos químicos ou eventos físicos do mundo, o estudo da imagem também pode deduzir sobre formas, cores e camadas da realidade perceptiva. Trata-se de uma lógica que mensura o espaço e o ser humano numa projeção ortogonal no papel.

É bastante esclarecedor o fato de Picasso possuir memória muscular sobre as pinceladas produzidas pelos pintores clássicos que estudava. Capacidade adquirida pela repetição exaustiva das obras vistas em museus de arte em que visitava. Sendo assim, a tentativa de concatenar o escopo da tecnologia com a problemática da arte deve considerar o fator humano, e não apenas descrever sobre processos e rotinas de computação. Eis o objeto e o mote para essa pesquisa, que é feita no campo teórico e prático da arte e da tecnologia.

Não se pretende neste capítulo encerrar estes assuntos nem definir uma semântica intacta sobre essa relação verificadamente dinâmica. O que se oferece aqui é um subconjunto de possibilidades referentes ao desenho construído a partir de um texto. Os exemplos apresentados foram escolhidos tendo em vista a possibilidade simbólica de modular uma construção. Alguns trabalhos apontam nesse sentido, em que a linguagem computacional é compreendida como uma metalinguagem e que concatenam estética, algoritmos e estrutura de dados (NUNES, 2008).

Assim, é provável que a esfera filosófica e geométrica destes estudos encontrem um campo fértil para desenvolvimento futuro, com possibilidade de construção dos objetos e proposições aqui apresentados. A opção foi a de trabalhar dinamicamente o desenho escrito pela arte, a tecnologia e a filosofia. Trata-se da construção de uma metalinguagem onde o texto descreve a imagem e a imagem demonstra a óptica do texto. A verificação de validade, ou sua possível prova, é a exemplificação simples do problema em termos de programação gráfica, feita no *Capítulo 3*. Assim, o conceito aliado à descrição das ferramentas apresenta um conjunto teórico-prático completo, algo bastante comum nos trabalhos e pesquisas realizados no campo da arte.

2.1 Desenho e *script*

Desenho é a ciência do risco sobre o suporte. Considere o suporte como a delimitação de um espaço em branco e o risco como a definição de uma sintaxe. Sendo ferramenta, o desenho satisfaz inúmeras aplicações da atividade humana, desde a compreensão territorial até a construção gráfica de ideias. Tentar traçar uma ontologia sobre esse conjunto demonstra que, após o risco, o suporte passa a ter significado, tornando-se imagem ou proposição. De fato, em acordo com Nuno Crespo, Ludwig Wittgenstein apresenta a imagem [*Bild*] como "uma espécie de desenho ou ilustração que se pode utilizar no interior de uma proposição" (CRESPO, 2008, p.115).

A obra portuguesa *Desenho Técnico Moderno*, dos autores Arlindo Silva, Carlos Tavares Ribeiro, João Dias e Luís Sousa (SILVA [et al.], 2006), desenvolve um conjunto de metodologias e procedimentos necessários à pesquisa e ao desenvolvimento do desenho técnico. Os autores separam e diferenciam o desenho técnico do desenho livre. Classificam o primeiro como algo que "deve ser perfeitamente perceptível e sem ambiguidades na forma como descreve determinado objeto" (*idem*) e o segundo como podendo ter, "para diferentes indivíduos, várias interpretações e significados do mesmo objeto" (*ibdem*). Apesar dessa diferenciação ser dicotômica e até restritiva, possui um aspecto didático próprio para o alargamento do tema, pois mostra o desenho como uma ferramenta prática e contextual, de um lado, ética e perceptiva, de outro.

Constitui-se na mais antiga forma de registro e comunicação de informação e, embora tenha conhecido mais mudanças, quanto ao modo de produção e de apresentação, do que as mudanças tecnológicas verificadas ao longo da história, nunca foi substituída efetivamente por nenhuma outra. (SILVA [et al.], 2006, pág.3).

Sendo tão antiga quanto a escrita, a investigação sobre desenho pode ter tido diferentes estágios ao longo da história, indo desde a larga utilização no dia a dia, até o esquecimento geral e utilização somente em círculos restritos. O fato é que o desenho permaneceu. Trouxe em sua ontologia uma capacidade de

adaptação extraordinária e pode mesmo ser considerado como uma necessidade humana essencial. Nesse sentido amplo, como um modelo ou como um quadro vivo, o desenho é uma ferramenta do mesmo gênero que a escrita, uma espécie de mapa constituído pela abstração ou pelo pensamento [*Abbildung*]. Seja em uma descrição da realidade, feita através de uma figuração, ou em uma lógica, que constitui e dá sentido à imagem, o desenho parte de elementos da realidade perceptiva e, portanto, está sujeito às transformações de natureza humana e representativa.

2.202 A figuração representa uma situação possível no espaço lógico. 2.203 A figuração contém a possibilidade da situação, a qual ela representa. 2.21 A figuração concorda ou não com a realidade, é correta ou incorreta, verdadeira ou falsa. 2.22 A figuração representa o que representa, independentemente de sua verdade ou falsidade, por meio da forma da afiguração. 2.221 O que a figuração representa é o seu sentido. 2.222 Na concordância ou na discordância de seu sentido com a realidade consiste sua verdade ou sua falsidade. 2.223 Para reconhecer se uma figuração é verdadeira ou falsa devemos compará-la com a realidade. 2.224 Não é possível reconhecer apenas pela figuração se ela é verdadeira ou falsa. 2.225 Não existe uma figuração *a priori* verdadeira. 3 Pensamento é a figuração lógica dos fatos. (WITTGENSTEIN, 1968, p.60).

A figuração — imagem ou proposição — está à mercê de uma comparação com o real, onde o resultado obtido por esse paralelo alcança concordância ou discordância com a realidade. Trata-se de um processo ortogonal. A figuração projeta algo, uma situação ou um objeto. A descrição então se transforma em uma pergunta sobre “onde encontrar mais ferramentas de análise, como construí-las?” Não existe uma ideia inicial de verdade no elemento figuração, mas o fato é que a “figura” terá o seu enfrentamento com a realidade. Não é possível mensurar a dimensão de um livro raro somente pela sua característica de raridade, é necessário prová-lo.

A palavra *scriptivo* é um neologismo criado pela importação do vocábulo inglês *script*, que significa escrita. — Uma antropofagia. Refere-se também às linguagens de programação, interpretadas e de extensão, que servem para

controlar as funcionalidades de um programa de computador. Uma interface de escrita, escritura, escrituração e lógica textual.

(*Script*) é uma linguagem de programação usada para manipular, personalizar e automatizar as funcionalidades de um sistema existente. Em tais sistemas, as funcionalidades já se encontram disponíveis por meio de uma interface de usuário e a linguagem de *script* provê um mecanismo para acessá-las. Dessa forma, o sistema existente oferece um ambiente de hospedagem para objetos e funcionalidades que complementa os recursos da linguagem de *script*. (SAMY, 2010, p.29)

Assim, forma-se desenho *scriptivo*. O desenho relacionado à escrita, a escrita escrevendo o desenho. Texto e figura, escrita e geometria, forma e conteúdo, imagem e proposição. Pode ser colocado em paralelo com a teoria da representação pictórica, no qual Wittgenstein sugere que a imagem, ou figuração [*Bilt*], pode funcionar como proposição. Em certo sentido, uma proposição necessita de uma imagem, algo para dizer o que acontece em determinada instância da situação, sendo mesmo uma condição interna para as proposições: "só no contexto da linguagem as imagens podem ou não ter sentido e é só no contexto de uma proposição que uma imagem pode representar o que acontece."(CRESPO, 2008, p.117).

“Quando escrevi ‘uma proposição [Satz] é uma imagem lógica de um facto’, eu queria dizer que posso inserir uma imagem, literalmente um desenho, numa proposição e depois continuar a minha proposição. Podia assim usar uma imagem como uma proposição. Como é que isto é possível? A resposta reza assim: porque, de um certo ponto de vista, ambas concordam [übereinstimmen], e a este comum [Gemeinsam] chamo imagem.” (WITTGENSTEIN *apud* CRESPO, 2008, p.114).

Vale lembrar, em acordo com CRESPO (2008), que a palavra *Satz* [Proposição] possui uma riqueza de significados que não é encontrando correlato na língua portuguesa. Significa "proposição", "frase", "afirmação", "princípio", "axioma". Neste sentido, quando Wittgenstein utiliza a palavra *Satz* refere-se não só às proposições da lógica, mas também

a frases enunciativas, predicativas, declarativas ou interrogativas. De certa forma, na afirmação de que “ambas concordam”, Wittgenstein utiliza o desenho como texto e o texto como desenho, imagem lógica dos fatos.

Este é o sentido da junção desenho *scriptivo*, ou seja, a forma de concatenar texto e imagem, como proposição e investigação sobre as consonâncias e cisões entre a geometria e a escrita, a forma e o conteúdo, o risco e o suporte. O que interessa é a oportunidade histórica, obtida pela implementação das ferramentas digitais, de retirar esse tipo de trabalho do papel e torná-lo prático.

2.2 Construção de um desenho a partir de um texto

A palavra está em paralelo com a geometria, sendo ambas receptáculos lógicos — significantes à espera do significado — ordenados segundo sua natureza particular, a palavra pelo conteúdo e a geometria pela forma. A geometria somente pode ser investigada pelo nosso próprio campo visual. "Uma geometria depois é transformada em conhecimento e investigação sobre o modo de organização dos objetos, das coisas: por isso a geometria nos é dada e nenhum dos seus elementos está escondido, tudo está já presente nas possibilidades humanas de visão" (CRESPO, 2008, p.100).

A geometria do espaço visual é-nos dada, i.e. para encontrá-la não necessitamos de uma investigação sobre factos escondidos até agora. Não é uma investigação no sentido em que o é uma investigação em física ou em psicologia. E, no entanto, pode dizer-se que ainda não conhecemos essa geometria. Essa geometria é gramática e sua investigação é uma investigação gramatical. (WITTGENSTEIN, 2003, p.145).

A geometria descritiva de Dürer da *Figura 1* não pode exercer, por si só, nenhuma consequência no espaço físico, sendo mesmo algo inventado, uma linguagem. No entanto, é possível que a investigação de seu conteúdo ofereça subsídios para a implementação de ferramentas e maquinarias utilitárias no mundo. Trata-se de um apoderamento do espaço pelo plano. Investigar gramaticalmente essa geometria é trazer luz à ela, torná-la utilizável.

Não se trata de uma predefinição sobre a utilidade ou o conteúdo do objeto. Naturalmente, suas características e propriedades vão sendo povoadas de atributos e métodos retirados do espaço concreto, tornando-a útil. Os métodos podem ser construídos por pares nome/valor, enquanto as características são progressivamente descobertas pela realização em estudos sobre o próprio objeto.

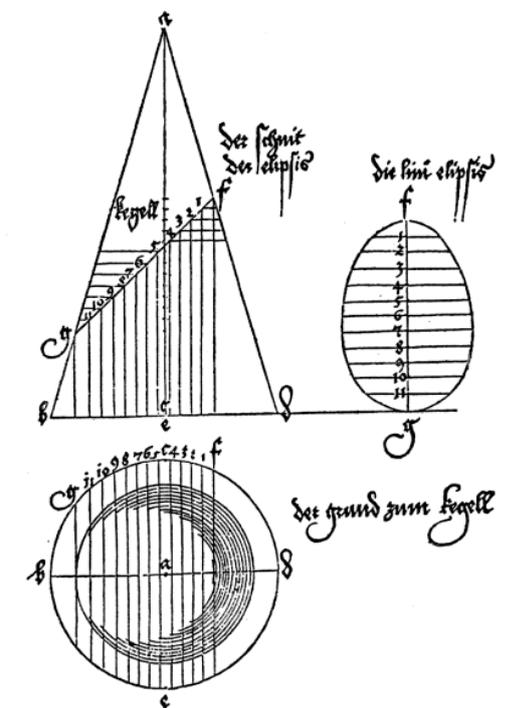


Figura 1 — Geometria descriptiva de Dürer.

Suponha, por exemplo, uma reta qualquer. Enumere dois pontos sobre esta reta, designados por A e B (Figura 2). Com centro em A e raio AB, trace uma circunferência. Com centro em B e raio BA, trace outra circunferência. Ache o ponto D, descrito pela intercessão entre as duas circunferências. Ache o ponto C e feche o triângulo retângulo pitagórico. Nessa construção, a reta CB denominada α é a hipotenusa; a reta BD é a hipotenusa dividida por 2, enquanto BDC é o ângulo reto (90°), DCB (30°) e CBD (60°).

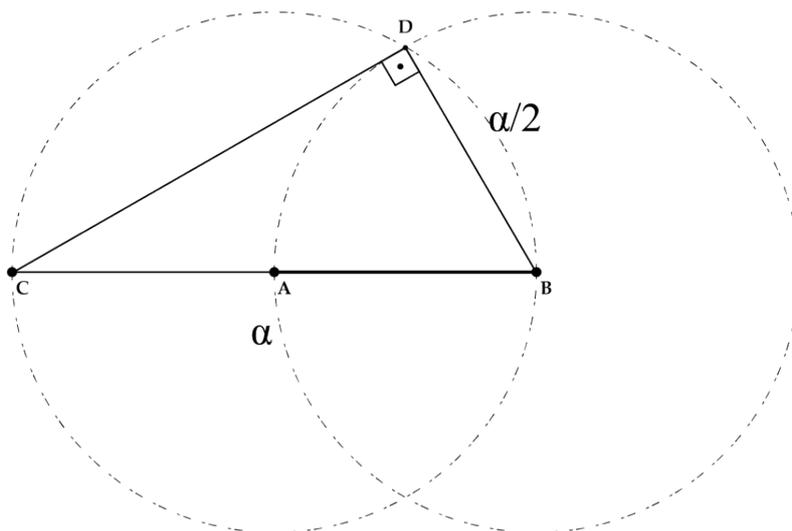


Figura 2 — Construção do triângulo retângulo partindo de uma reta simples.

Simplificadamente, podemos dizer que, no exemplo acima, o papel é o suporte que contextualiza a linguagem, tornando inseparável a forma do conteúdo. O desenho é a ação que possibilita o risco linguístico ou mesmo a representação da realidade, enquanto o texto explicativo perfaz a lógica referida na disciplina matemática. Trata-se de uma geometria pelo texto. "A ligação entre 'linguagem e realidade' é feita por meio de definições de palavra e estas pertencem à gramática, de modo que a linguagem permanece autocontida e autônoma. (WITTGENSTEIN, 2003, p.69)".

O que está em jogo não é a elegância da proposição, mas a relevância dessa construção na mediação entre linguagem e realidade, ou seja, é possível implementar essa solução agora, com compasso e régua — neste papel — e trazê-la para o mundo concreto?

Inventar uma linguagem poderia significar inventar um instrumento para um propósito particular, com base nas leis da natureza (ou compativelmente com elas); mas também tem o outro sentido, análogo àquele em que falamos da invenção de um jogo. No caso, estou enunciando algo a respeito da gramática da palavra “linguagem”, ligando-a com a gramática da palavra “inventar” (WITTGENSTEIN, 2003, p.145).

A geometria diz respeito ao que se encontra "diante de nós", dos nossos olhos. A geometria pode tornar o espaço um instrumento esquemático, ortogonal, sendo um modo de traçar o mundo e delimitar o que nele existe. Já o texto subsiste como representação das coisas pela linguagem, do fato pela linguagem. O texto é o que torna a geometria legível, uma vez que a estrutura como objetos e como coisas ordenadas entre si. Cria-se um conjunto de regras e coordenadas espaciais em que todas as imagens e proposições serão estruturadas. Cria-se a partir desse conjunto, tornando a palavra proposição recursiva, destinada ao mesmo paradigma e à mesma utilização. Com esse procedimento, o desenho pode então ser lido com universalidade e precisão.

2.3 Comunicação gráfica de ideias

As transformações trazidas pelas tecnologias da informação e comunicação, sejam elas físicas ou lógicas, tornaram a imagem digital uma prática cotidiana. Também abriu caminho para a mecanização dos processos visuais e trouxe a computação gráfica avançada para o âmbito informal. As interfaces, cada vez mais, são construídas com base nos parâmetros pessoais do usuário. Os aplicativos tornaram-se úteis, autocomunicativos e atraentes. Suas lógicas internas vão tecendo métodos de avaliação sobre as necessidades de progressão da própria ferramenta, como se fosse um autoaprendizado.

As tecnologias de interface integradas em sistemas audiovisuais digitais permitem, atualmente, uma série de investigações sobre a inter-relação entre realidade-virtualidade-observador-entorno. Ao contrário das imagens analógicas, os dados digitais permitem a variabilidade e a manipulação dos parâmetros de informação que constituem a representação. (GIANNETTI, 2006, p.125)

De acordo com Cláudia Giannetti, "devemos entender a comunicação como um processo artificial baseado em modelos, ferramentas e instrumentos que manipulam símbolos em nível de códigos." (GIANNETTI, 2006, p.60). As informações podem conter significados que se referem a outros símbolos ou

fenômenos concretos. Novos meios de informação podem gerar resultados imprevisíveis durante o processo de comunicação, mas a matriz provavelmente será formada por dados e informações existentes. A matriz poderá ser a estrutura semântica do objeto.

Por exemplo, poderíamos tomar a demonstração expressa na *Figura 3* como um começo possível para a geometria, uma síntese da passagem do espaço para o plano ou mesmo um traço deflagrado sobre o assunto? Seria esse um processo de comunicação da geometria, baseado em modelos e ferramentas?

Suponha uma seleção aleatória de gravetos debaixo de uma árvore onde, indistintamente, relacionamos três gravetos de tamanhos iguais, ordenando e simplificando um passo a passo desse conjunto. É um claro processo de evolução da forma que, intercalado com a *Figura 2*, prevê a transformação do graveto em reta e do triângulo em roda. É o domínio do terreno, sua medição e geologia. Esse tipo de demonstração, que utiliza volutas e classificações, foi muito utilizada na história do desenho, notadamente no *Códice Atlântico (1478-1519)*, de Leonardo da Vinci, e nas *Regras das cinco ordens da arquitetura (1562)*, de Giacomo Vignola.

Mas, voltando a nossa ontologia, será que poderíamos fugir da lógica trigonométrica descrita? Por que, desde Euclides, o método triangular prevaleceu? Veja bem, se trocássemos os valores, para 1 ou 2 gravetos, não haveria elementos eficientes para representar o espaço real. Se optássemos para quatro ou cinco gravetos, haveria excesso de dados e, ademais, se multiplicarmos o triângulo por 2, é possível chegar ao quadrado. Ao multiplicar por 3, outros polígonos surgem. Eis a hegemonia do triângulo.

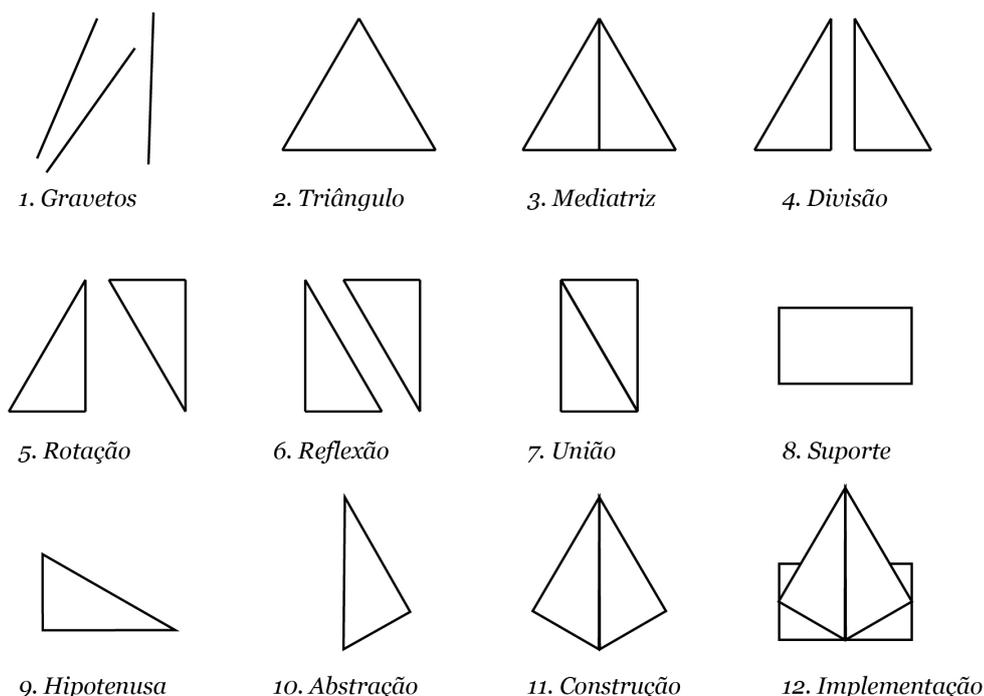


Figura 3 — Uma possível ontologia para a geometria.

Seria a demonstração da *Figura 3* uma invenção? É uma imagem que concorda com a realidade? Ou poderíamos enunciar sua dimensão de realidade, tendo como prova o seu próprio procedimento demonstrativo?

O que há por traz destas formas, números e palavras? Se retiramos as formas e os números, ficamos com diversas palavras: Gravetos, Triângulo, Mediatriz, Divisão, Rotação, Reflexão, União, Suporte, Hipotenusa, Abstração, Construção, Implementação. Palavras que, a princípio, estão desprovidas de contexto e significação. Por outro lado, se conservamos somente os números (1,2,3,4,5,6,7,8,9,10,11,12), obtemos outros tantos signos, também dependentes de significação contextual. Mas, estranhamente, a sequência geométrica da *Figura 4* conserva em si uma certa “lógica natural”, independente de explicações, onde três gravetos causam um lapso significativo, cuja significação atinge nossos sentidos lógicos naturais.



Figura 4 — Sequência de construção e desconstrução. Lógica natural?

Obviamente, a tal “lógica natural” é contestável. Imagine que determinada sociedade desconheça o conceito de *6.Reflexão* no plano ou mesmo a ideia de *5.Rotação* no plano. Sim, pois é no plano que se dá a leitura dessa geometria. Para compreendê-la é necessário mais que simplesmente “ver”, pois nela há mais lógica que visualidade, ou seja, dificilmente obteríamos o conjunto *8.Hipotenusa* sem a ideia de *3.Mediatriz*, ou mesmo *10.Abstração* sem *5.Rotação*.

Para Luhmann, o código é por si um veículo de comunicação e, como tal, regula a totalidade dos processos e canaliza a informação para as redes de relações sociais. Porém, como uma espécie de “operador”, o código desempenha uma atividade basicamente funcional, neutra e desprovida de valores adicionais. (GIANNETTI, 2006, p.64)

O método clássico de demonstração geométrica, por si só, não é suficiente para comunicar as ideias dessa pesquisa. Isso porque, a programação gráfica para web exige o estabelecimento de outros assuntos, como protocolos de comunicação, funções, algoritmos, *strings*, objetos, declarações e estruturas, circuitos integrados, recursão, combinatória, eventos, banco de dados, etc. Obviamente, a maioria desses assuntos não serão tratados aqui, mas apenas uma síntese representativa de uma concatenação entre as primitivas da arte com as primitivas da tecnologia.

Por fim, a comunicação gráfica em desenho *scriptivo* pode partir de elementos tão simples como o conjunto de regras descrito na *Tabela 1*, a *NBR 8403*, que normaliza a aplicação de linhas em desenho técnico. A comunidade que adota esse conjunto de regras constrói para si um mecanismo de comunicação gráfica de ideias. Não existe para a teoria, mas para a prática, em um processo de trabalho que objetiva a identificação de um ofício, um serviço ou uma companhia. Estão estritamente relacionados ao contexto e ao suporte de trabalho que dá vazão ao modelo esquemático e paramétrico da construção. E essa é a qualidade da internet como ferramenta. Pelo processo de comunicação, pela ação do recorte e da cópia — compartilhar — o observador faz nascer o entorno da imagem, seu significado e contexto.

Linha	Denominação	Aplicação geral (ver Figuras 1a, 1b e outras)
A 	Contínua larga	A1 contornos visíveis A2 arestas visíveis
B 	Contínua estreita	B1 linhas de interseção imaginárias B2 linhas de cotas B3 linhas auxiliares B4 linhas de chamadas B5 hachuras B6 contornos de seções rebatidas na própria vista B7 linhas de centros curtas
C 	Contínua estreita a mão livre (1)	C1 limites de vistas ou cortes parciais ou interrompidas se o limite não coincidir com linhas traço e ponto (ver Figura 1c)
D 	Contínua estreita em zigue-zague (1)	D1 esta linha destina-se a desenhos confeccionados por máquinas (ver Figura 1d)
E 	Tracejada larga (1)	E1 contornos não visíveis E2 arestas não visíveis
F 	Tracejada estreita (1)	F1 contornos não visíveis F2 arestas não visíveis
G 	Traço e ponto estreita	G1 linhas de centro G2 linhas de simetrias G3 trajetórias
H 	Traço e ponto estreita, larga nas extremidades e na mudança de direção	H1 planos de cortes
J 	Traço e ponto larga	J1 indicação das linhas ou superfícies com indicação especial
K 	Traço dois pontos estreita	K1 contornos de peças adjacentes K2 posição limite de peças móveis K3 linhas de centro de gravidade K4 cantos antes da conformação (ver Figura 1f) K5 detalhes situados antes do plano de corte (ver Figura 1e)

(1) Se existirem duas alternativas em um mesmo desenho, só deve ser aplicada uma opção.

Tabela 1 — NBR 8403 - Tipos de linhas e sua aplicação. Um conjunto de regras de desenho.

2.4 Máquina (sem rodas)

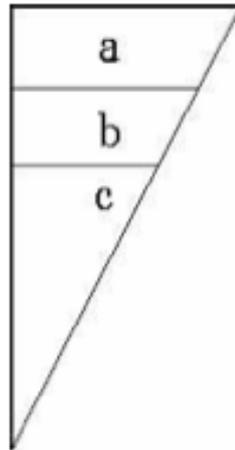


Figura 5 — The Half Triangle, de Raymond Lull.

De acordo com Lúcia Santaella (1997), "a palavra máquina se refere a uma estrutura material ou imaterial, aplicando-se a qualquer construção ou organização cujas partes estão de tal modo conectadas e inter-relacionadas que, ao serem colocadas em movimento, o trabalho é realizado como uma unidade." Máquinas são aplicadas na transmissão de diferentes tipos de força e movimento, como observa-se na *Figura 5*, onde a ordenação textual alcança sentido de sequência dentro de uma estrutura geométrica.

No artigo denominado *O homem e as máquinas (1997)*, a autora mapeia três níveis de máquinas que se relacionam com o humano: o nível muscular-motor, o nível sensorio e o nível cerebral. Interessa-nos principalmente os níveis que mais se aproximam do processamento de símbolos.

A diferença entre um dispositivo, por mais extremamente complexo que seja, e um computador digital, visto como uma variante da máquina de Turing, está no fato de que o computador não é simplesmente uma complicada rede de impulsos elétricos, nem apenas um dispositivo que caminha por meio de estados distintos como um autômato de estados finitos, mas é um dispositivo que processa símbolos. (SANTAELLA, 1997, p.39)

Como exemplo, dois objetos serão aqui concatenados para análise — a arte combinatória e a máquina (sem rodas) — perfazendo um paralelo simbólico entre algoritmos e estrutura de dados. O primeiro, sendo analisado a partir de um fragmento do livro *Ars Magna. The Ultimate General Art*, datado de 1305 e escrito por Raymond Lull (1305), que enumera dez regras gerais que se aplicam a todas as perguntas.

O segundo, com o manuscrito *De la numération Binaire*, datado em 15 de março de 1679, de Gottfried Leibniz, que descreve a possibilidade construtiva de uma máquina lógica, uma espécie de computador. Os dois textos em conjunto demonstram que o processamento simbólico pode ser estritamente maquínico, algo que em nossos dias têm se tornado banal, se considerarmos a constante manipulação de dados na chamada era da *big data*.

O termo *big data* é recente e diz respeito não somente ao grande volume de dados ordenados, mas também ao processo de trabalho necessário para tornar todo conjunto relevante. O processo em *big data* simplifica-se em quatro fases: coleta, armazenamento, análise e acesso aos dados¹. A transformação da informação em conhecimento é feita com análise de ligação semântica e a automatização do processo leva em conta a interação do usuário com o sistema. Esse tema é amplo e não será aprofundado, bastando a compreensão de que o conjunto de dados que não contém uma leitura semântica adequada corre o risco de se tornar uma informação ineficiente ou mesmo inútil.

Descrição A: Raymond Lull, As Regras.

"Estas regras são dez questões gerais que se aplicam a todas as perguntas. No processo de investigação, cada regra, de acordo com sua essência natural, esclarece cores e exhibe o assunto para o intelecto [...] As dez perguntas são gerais e assim são as suas espécies.

Regra B faz a pergunta - "Se?".

Regra C, ou a regra de quiddade, ou "O que uma coisa é em si". Essência.

¹ Palestra online assistida pelo instituto IGTI em 27 de junho de 2013. <http://www.institutogti.com.br/>

Regra D investiga a consistência material.

Regra E pergunta "porquê".

Regra F é sobre a quantidade.

Regra G é sobre a qualidade.

Regra H pergunta sobre o tempo.

Regra I investiga a natureza do locus perguntando "onde é".

Regra K (1) é sobre modalidade ou como as coisas existem.

Regra K (2) lida com a instrumentalidade."

(LULL, 1305. Ars Magna. The Ultimate General Art, tradução do autor)

Descrição B: Gottfried Leibniz, Máquina (sem rodas).

"Este tipo de cálculo pode também ser conseguido com uma máquina (sem rodas). Observe o seguinte para contruí-la facilmente e sem esforço. Disponha de uma caixa provida de orifícios que podem ser abertos ou fechados, dependendo da ocasião. Estes orifícios são espaços que correspondem a uma pequena roda com dois dentes.

Os furos abertos correspondem a 1. Os espaços fechados correspondem a 0. Através dos furos abertos caem, de pequenos cubos, as esferas nos canais; e nada através dos buracos fechados. A caixa é deslocada de coluna em coluna, como exigido pela multiplicação. Os canais representam as colunas e nenhuma bola deve sair de um canal para o outro, a menos que aconteça um movimento da máquina. Em seguida, todas as bolas rolam para o canal seguinte, umas sendo levadas e as outras caindo nos buracos. A regra é que uma única esfera cai em cada abertura. Tudo pode ser organizado, de forma que duas bolas para fora sempre estariam em conjunto e de outra forma não seriam removidas."

(LEIBNIZ, 1679. De la numération binaire, tradução do autor).

A máquina (sem rodas) de Leibniz não foi construída, mas talvez tenha sido o primeiro projeto feito para a concretização de uma lógica computacional. Uma estrutura de decisão maquínica onde esses dois textos em conjunto simbolizam a materialização da antiga ideia de confluência entre texto e imagem, ou mesmo da ideia filosófica que versava sobre a universalidade do método matemático, uma problemática que conduziu o trabalho de diversos estudiosos ao longo da história, dentre eles Raymond Lull (*Ars Magna*), Cornelius Agrippa (*De Incertitudine et Vanitate de Scientiarum et Artium*), Descartes (*Mathesis Universalis*), Adrianus Romanus (*Apologia pro Archimede*), Bernoulli (*Arte de La Suposición*) e Leibniz (*Characteristica Universalis ou, alfabeto do pensamento humano*) — em acordo com referência no estudo “Um estudo sobre Leibniz e a criação de um alfabeto do pensamento” (FRANZON & BRITO, 2012).

Na obra *Algorithms of the Intelligent Web*, os autores Haralambos Marmanis e Dmitry Babenko (MARMANIS & BABENKO, 2009) descrevem três ferramentas que são necessárias para a construção de uma *web* inteligente: o conteúdo agregado, as estruturas de referência e os algoritmos (*Figura 6*). O conteúdo agregado é um conjunto de dados específicos para que o objeto funcione. As estruturas referenciais fornecem interpretações semânticas e sintáticas sobre o conteúdo. Já os algoritmos são definidos tendo como base a estrutura referencial e a análise combinatória gerada sobre os dados.

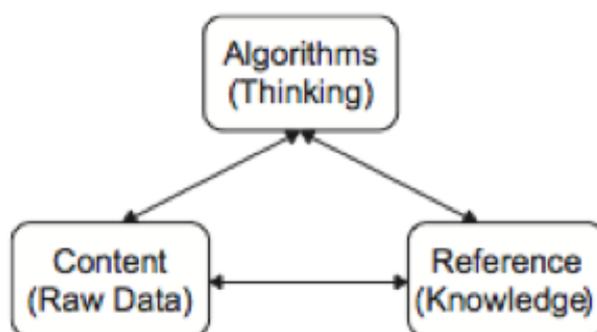


Figura 6 — Triângulo da inteligência: 3 ingredientes para uma aplicação inteligente: Conteúdo Agregado, Estruturas de Referência e Algoritmos (MARMANIS & BABENKO, 2009).

Com a dinâmica acima descrita, os conteúdos de forma modular, tornando eficiente as ligações entre algoritmos, os bancos de dados e a leitura semântica. Os autores citam o termo *folksonomia* como o conjunto de representações de um conhecimento sobre determinado objeto construído coletivamente.

Já o professor de informática Niklaus Wirth, em sua obra *Algoritmos e Estrutura de Dados*, demonstra como a estrutura de dados é inseparável da síntese de programas. "As decisões sobre estruturação de dados não podem ser tomadas sem o conhecimento dos algoritmos aplicados aos mesmos e que, vice-versa, a estrutura e a escolha dos algoritmos, em geral, dependem fortemente da estrutura de dados em que se baseiam" (WIRTH, 1999).

A dependência da escolha de um algoritmo quanto à estrutura de dados a ser processada — um fenômeno onipresente — é tão forte no caso da ordenação que os métodos utilizados são, em geral, classificados em duas categorias, a saber, ordenação de vetores e ordenação de arquivos (sequenciais). (WIRTH, 1999, p.50).

Retomando Santaella, "o processamento de listagens trouxe à tona as capacidades denotativas dos símbolos, definindo o processamento de símbolos de uma maneira tal que permitia a independência da estrutura fixa da máquina física subjacente."(SANTAELLA, 1997, pág.40). Exemplo claro é a máquina de Turing que, "diferentemente de uma máquina meramente física, Alan Turing inventou uma máquina teórica, cujos propósitos são essencialmente teóricos."(idem).

Trata-se de uma máquina que visava iluminar as noções de calculabilidade em geral, permitindo reduzir todos os métodos de cálculo a um conjunto subjacente, simples e básico de operações. No seu todo, essa máquina é composta por um certo número de estados, sendo capaz de ler símbolos localizados em quadrados numa fita infinita. Alguns quadrados podem estar vazios. As operações básicas são desempenhadas pela máquina em resposta a uma combinação de: (1) o estado em que a máquina está e (2) o símbolo que ela está lendo, naquele momento, no quadrado. (SANTAELLA, 1997, p.39)

Ainda de acordo com a autora, a máquina cerebral, diferentemente de uma máquina física, é uma ferramenta intelectual destinada ao processamento de símbolos, onde a própria noção de máquina está sendo substituída por um "agenciamento instável" de redes de interfaces. Ocorre que, com a capacidade computacional de transformar impulsos eletrônicos em informação, "não há signo que não possa ser absorvido, traduzido, manipulado e transformado." (*idem*).

Por ser, em si mesma, um princípio de interface, a codificação digital, com seus *bits* de imagens, textos sons, imbrica, em suas tramas, nosso pensamento e nossos sentidos. É o grande processador leve, móvel, maleável e inquebrantável. (SANTAELLA, 1997, p.41).

Assim, a internet é um espaço de luta e de guerra, uma espécie de máquina (sem rodas) movimentada por uma combinatória. O discurso flui entre ambientes lógicos diversos, ativado ou desativado pelo usuário na simples ação de passá-lo adiante ou ignorá-lo. Todos esses movimentos podem ser armazenados e combinados, modelados e medidos.

Exemplos como a espionagem industrial, os cavalos de tróia do século XXI e os ciberataques de todo tipo, faz da internet um local que “graças aos milagres da navegação espacial e do progresso técnico, conhecemos a extensão de nossos próprios recursos, assim como a dos de outros países” (SANTOS, 2007, p.32). Então, quais perguntas devem ser feitas para se chegar à compreensão sobre as regras dessa máquina (sem rodas)? Qual é o ponto em que ela se apoia?

Aqui será concatenado a descrição A com a descrição B, perfazendo as regras para uma máquina (sem rodas) — Regra B, traz a condição essencial "se", ou seja, se a *web* existe enquanto modalidade de algo pré-existente ou seria de fato algo novo, uma máquina nova? — Regra C, qual é a essência dessa máquina, é virtual ou real, está para a lógica ou para a natureza? — Regra D, é uma máquina com ou sem rodas, ou seja, está em nossa motricidade, no corpo, ou tende somente à abstração, ao conceito? — Regra E, historicamente, quando

sobreveio a ideia de uma rede e porque esse espaço se tornou indispensável para essa sociedade? — Regra F, qual a sua quantidade, quantos objetos, quantos arquivos possui. Há mais textos produzidos por humanos para criar a máquina ou mais máquinas escrevendo os textos para estruturar as ações do humano? (*Figura 7*) — Regra G, a qualidade da interação com os textos, como organizá-los e reescrevê-los, com que dinâmica, com quais parâmetros? Regra — H, como a noção de tempo se modifica quando estamos conectados, o tempo online retorna? — Regra I, o *locus* da internet, onde está a internet? Na terra, nas nuvens? — Regra K(1) e K(2), como as coisas existem neste espaço (movimento, tempo, mecânica, semântica), com que matrizes estamos lidando e quais as especificações desse instrumento?

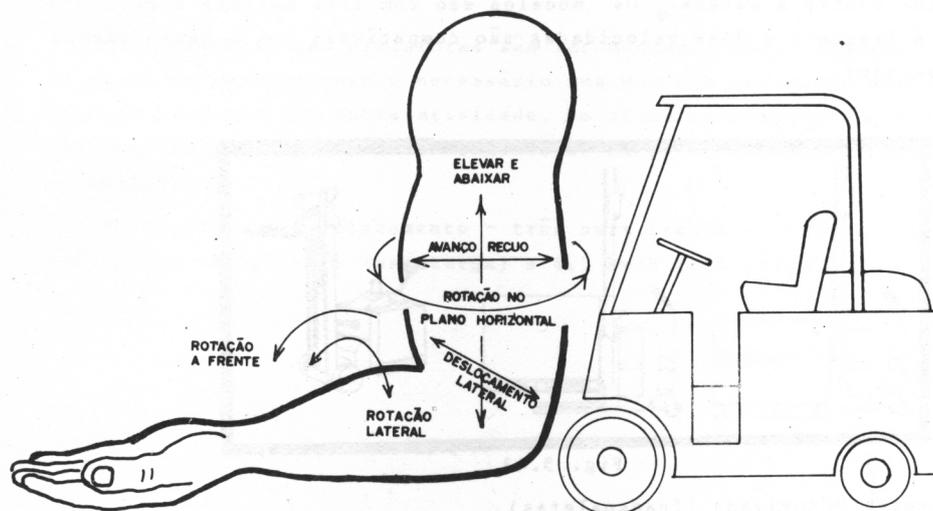


Figura 7 — Comparação das operações de uma empilhadeira e o braço e a mão humana para movimentar materiais (MOURA, 1975).

2.5 O desenho contido em uma fórmula

Jean Dhombres (1996), em entrevista temática sobre “O que é uma forma?”, trabalha com a questão da imagem contida em uma fórmula (*Figura 8*). Os termos com que Dhombres apresenta o modo particular de expressar o movimento do pensamento humano para o plano coordenado, ou mesmo para o espaço em branco, traduz-se em uma

dinâmica compreensiva que tenta ser direta mas não imediata, ou seja, pretende atingir uma profundidade do que está expressa em nossa forma de enumerar e quantificar os elementos do mundo, entretanto, sem deixar de ser crítico quanto ao procedimento controverso com o qual o executamos.

Esse é um modo particular de expressar o movimento do procedimento científico: passar para duas dimensões, geometrizar pelo plano o que está no espaço é efetuar uma simplificação. (DHOMBRES, 1996, p.25).

Pois é no local onde já estamos, e onde sempre estivemos, no movimento da simplificação e da síntese, que nos apoderamos do espaço. As confrontações e as guerras obtidas por meio dessas atitudes de dominação não retiram a camada gloriosa com a qual o ser humano se reveste ao pensar na impossibilidade de ter feito o risco sobre o suporte, na caverna ou no papel. A forma concreta da linguagem no espaço físico quase sempre se sobrepõe à possibilidade lógica de se fazer um movimento no papel. Não há erro possível no espaço concreto, há apenas a efetividade ou não de sua permanência.

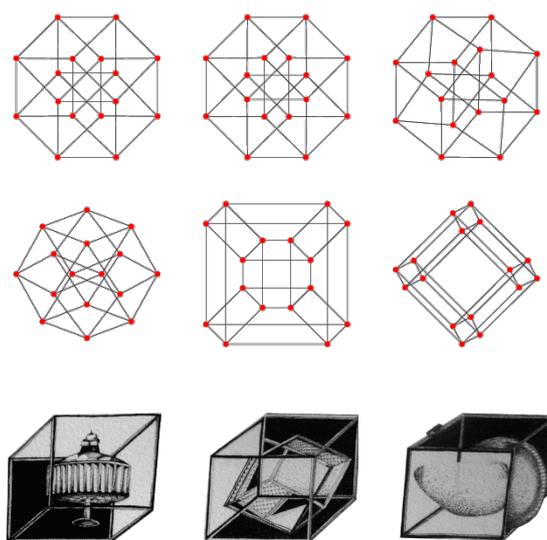


Figura 8 — Exemplo de desenho contido em uma fórmula.

Por ser linguagens que trabalham com representações de espaço e tempo utilizando técnicas e métodos próprios, escrever e desenhar requerem qualquer delimitação analítica, ou seja, torna-se necessário trabalhar sob a perspectiva de

uma metateoria, com um ponto de vista que simplifique, pelo plano, o espaço e o tempo, tornando-os processo. Perfazendo uma escrita geométrica, primeiramente podemos dizer que o desenho é o elemento raiz, composto por duas subárvores — o risco e o suporte — que estruturam uma dialética entre a forma e o conteúdo. A definição sobre o método de trabalho nesse conjunto, ou o que ele representa, constitui a sua efetividade textual.

A imagem está contida em uma fórmula simplesmente enquanto é representação contínua de um certo número de pontos tomados em uma sucessão, de um certo tipo de dados empíricos organizados por um movimento pré-programado: de certo modo, a máquina nivela os dados. E há obtenção de uma imagem porque há programação de uma forma, no sentido algébrico — pré-desenho de alguma coisa. (DHOMBRES, 1996, p.25).

Por outro lado, somente passar para o plano as definições apreendidas do espaço não efetiva qualquer modificação sobre esse espaço em um nível considerável de precisão e reprodução, pois não há nele padrão algum, o plano está disperso e a fórmula não retorna para a realidade palpável. Observando a *Figura 9* podemos perceber o que significa estipular uma fórmula para o espaço. O triângulo de ouro renascentista, um passo fundamental para a geometria moderna é a suspensão do ângulo reto (90°). No caso, a hipotenusa é colocada em perpendicularidade com o solo, algo até certo ponto inovador. Quase uma reinvenção da roda, esse movimento alça o vôo dos objetos futuros.



Figura 9 — Triângulo de Ouro Renascentista e a perpendicularidade da hipotenusa com o solo.

A passagem para o plano coordenado (sistema cartesiano) é considerado outro movimento fundamental para o apoderamento do espaço pelo plano. Nessa gramática, o plano é associado aos números reais, com os pontos nesse mesmo plano colocados com pares de números reais a partir de um sistema retangular de coordenadas, que consiste em dois eixos perpendiculares coordenados. Neste sistema, os eixos dividem o plano em 4 regiões ou quadrantes, numerados em sentido anti-horário.

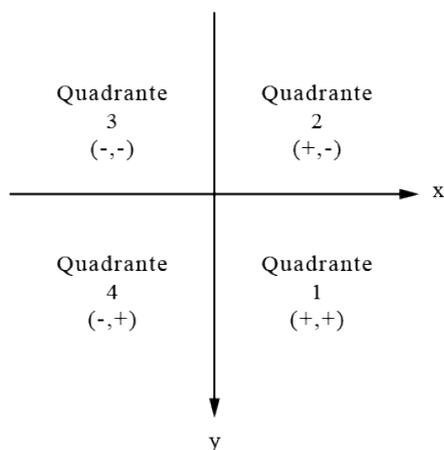


Figura 10 — Plano coordenado para web.

O sistema cartesiano para *web* (Figura 10) é semelhante ao matemático, com a diferença da origem do ponto $(0,0)$. Na matemática, o quadrante das coordenadas positivas se origina no canto inferior esquerdo, enquanto na *web* a origem é o canto superior esquerdo (ou área de renderização), com a coordenada positiva horizontal x da esquerda para a direita e a vertical y de cima para baixo (SAMY,2012, p.34).

A correspondência entre pontos de um plano e pares ordenados de números reais torna possível visualizar equações algébricas como curvas geométricas e, inversamente, representar curvas geométricas por equações algébricas. (ANTON, 2000, p.A17).

Para exemplificar, em acordo com Howard ANTON (2000), suponha um sistema de coordenadas xy e uma equação que envolve duas variáveis x e y .

$6x - 4y = 10$; $y = x^2$ (Tabela 2); $x = y^2 - 2y$ (Tabela 3); ou $y = 1/x$ (Tabela 4).

A definição de uma solução para essas equações pode ser um par ordenado (a,b) , cujas coordenadas, quando substituídas x por a e y por b , satisfaçam a equação apresentada. Neste exemplo, o par ordenado $(3,2)$ é uma solução para a equação $6x - 4y = 10$, pois a equação está satisfeita por $x = 3$ e $y = 2$. Já o par ordenado $(2,0)$ não é uma solução para essa equação, pois não se satisfaz por $x = 2$ e $y = 0$. Abaixo, três gráficos desenhados a partir de dados gerados pela solução das respectivas fórmulas.

x	$y = x^2$	(x,y)
0	0	(0,0)
1	1	(1,1)
2	4	(2,4)
3	9	(3,9)
-1	1	(-1,1)
-2	4	(-2,4)
-3	9	(-3,9)

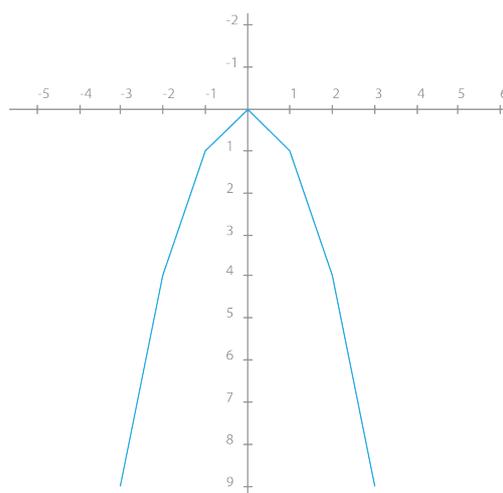


Tabela 2 — Gráfico e valores para $y = x^2$, baseado em ANTON (2000, p.A18).

y	$x = y^2 - 2y$	(x,y)
-2	8	(8,-2)
-1	3	(3,-1)
0	0	(0,0)
1	-1	(-1,1)
2	0	(0,2)
3	3	(3,3)
4	8	(8,4)

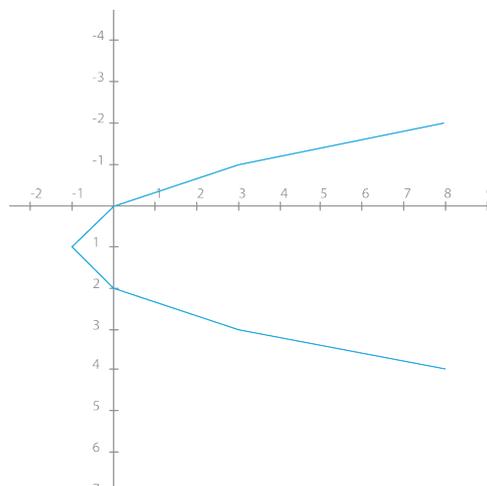


Tabela 3 — Gráfico e valores para $x = y^2 - 2y$, baseado em ANTON (2000, p.A19).

x	y = 1/x	(x,y)
1/3	3	(1/3,3)
1/2	2	(1/2,2)
1	1	(1,1)
2	1/2	(2,1/2)
3	1/3	(3,1/3)
-1/3	-3	(-1/3,-3)
-1/2	-2	(-1/2,-2)
-1	-1	(-1,-1)
-2	-1/2	(-2,-1/2)
-3	-1/3	(-3,-1/3)

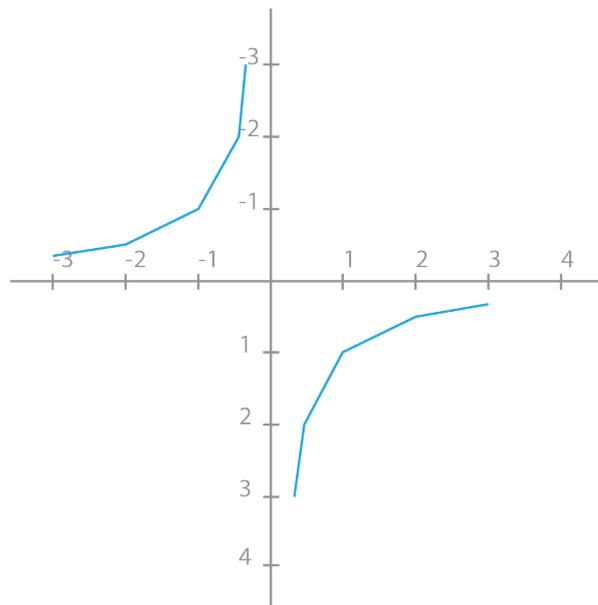


Tabela 4 — Gráfico e valores para $y = 1/x$, baseado em ANTON (2000, p.A19).

Temos então a seguinte definição: "O conjunto de todas as soluções de uma equação em x e y é chamado *conjunto da solução* da equação, e o conjunto de todos os pontos no plano xy cujas coordenadas são elementos do conjunto da solução é chamado de *gráfico da equação*." (ANTON, 2000, p.a17). O autor afirma que identificar a forma exata de um gráfico é um tema muito importante para o cálculo. Ou, visto pela perspectiva do desenho *scriptivo*, "O que está em questão não é somente um procedimento que simplifica, mas também um processo de prejuízo que avança: apoderamo-nos do espaço por meio do plano." (DHOMBRES, 1996, p.25).

2.6 Uma geometria sem distância

Na segunda seção da obra *Gramática Filosófica*, Ludwig Wittgenstein (2003) menciona uma certa “geometria na qual não houvesse algo como distância”. Sob o título — A generalidade. Parte II (Sobre a Lógica e a Matemática) — o texto destaca e analisa a seguinte proposição:

A proposição “O círculo está no quadrado” é, em certo sentido, independente da atribuição de uma posição particular (em certo sentido, é totalmente desligada). (WITTGENSTEIN, 2003, p.199)

Para Wittgenstein (1968), os objetos são os elementos mais simples em que se pode decompor o mundo. Sua configuração “forma o estado das coisas” (§2.0272) e se ligam uns aos outros como “elos de uma cadeia”(§2.03). O objeto sustenta uma estrutura lógica da imagem, como suporte e risco. “O objeto é o fixo, o subsistente; a configuração é o mutável, o instável”(§2.0271).

Um objeto depende de um espaço. Não haveria lugar para os objetos ideais, hierarquizados ou instituídos por qualquer especialidade. Essa ideia é chamada de *atomismo lógico* de Wittgenstein, sendo mesmo tradição entre leitores e comentadores do Tratado Lógico-Filosófico. “A diferença entre as coisas são determinadas não pela sua substância mas pelo seu lugar no espaço lógico e, logo, pelo seu lugar na estrutura lógica da proposição.” (CRESPO, 2008, p.87).

Dois objetos de mesma forma lógica – abstraindo suas propriedades externas – se diferenciam um do outro apenas por serem distintos. (WITTGENSTEIN, 1968, §2.0233)

Eis o ponto que permite compreender a ideia de uma geometria sem distância, onde “cada coisa (Ding) está como que num espaço (Raum) de possíveis estados de coisas possíveis. Posso pensar este espaço vazio, mas não a coisa sem o espaço”(§2.013). Nessa geometria, as relações de posição são ignoradas, as distâncias elásticas e as extensões sem importância. Um exemplo

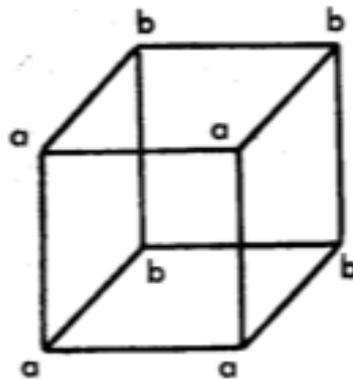
visual. Para o autor, as imagens  e  seriam iguais (WITTGENSTEIN, 2003, p.201). "A forma é a possibilidade da estrutura" (§2.033).

A distinção interior/exterior é uma necessidade instrumental e metodológica em que ao interior correspondem as condições de possibilidade da existência de objectos no espaço e ao exterior as posições que esses objectos ocupam na realidade. Aquilo que aqui interessa são as condições de possibilidade da existência de objectos no espaço: uma oposição que permite opôr a uma geometria do campo de visão (poder-se-ia talvez dizer humana), uma geometria da física ou das ciências da natureza. (CRESPO, 2008, p.99)

No caso da proposição "Há um círculo no quadrado" as relações de distância, métrica, extensão e posição são desconsideradas, sem importância ou mesmo não são mencionadas. Não importa o tamanho do círculo ou as extensões do quadrado, visto que isso poderia ser averiguado em um segundo momento, no acesso a suas particularidades. O caso geral não faz nada mais do que segurar (colar) o círculo no quadrado. "O modo pelo qual os objetos se vinculam no estado de coisas constitui a estrutura do estado de coisas" (§2.032). Dentro do quadrado, ao círculo é concedida "completa liberdade", uma variável geométrica.

Onde quer que o retalho (círculo) esteja no quadrado... significa "contanto que esteja no quadrado..." e, no caso, tudo o que se quer dizer é a liberdade (falta de restrição) no quadrado, não um conjunto de posições. (WITTGENSTEIN, 2003, p.201)

“As regras que governam as variáveis não são as mesmas que governam seus valores particulares.” (WITTGENSTEIN, 2003, p.199). Mas como trabalhar essa lógica em um *script* de desenho?



Perceber um complexo quer dizer perceber que suas partes constituintes estão em relação entre si de um certo modo. Isto também explica por que é possível ver a figura de duas maneiras como um cubo; e todos os fenômenos parecidos. Porquanto vemos realmente dois fatos diferentes. (Primeiro veja a partir dos vértices a , e só ligeiramente a partir de b ; a aparece na frente; e vice-versa). (WITTGENSTEIN, 1968, p.108 com figura)

Sendo essa criação gráfica baseada em recursões semânticas, os termos providos pela proposição geral, além de possuírem regras filosófico-gramaticais, devem trazer toda a multiplicidade dos casos particulares, o que nos leva à fórmula $(x).fx$, proposta por Wittgenstein, em que fa decorre de $(x).fx$.

(x).fx

fa decorrer de $(x).fx$ significa que a é mencionado em $(x).fx$? Sim, se a proposição geral é pretendida de tal maneira que a verificação consista em uma enumeração. (WITTGENSTEIN, 2003, p.200)

O artista gráfico Manfred Mohr, apresenta a imagem “*The cubic limit series*” (Figura 11), estruturada de maneira recursiva, onde demonstra 961 formas de se representar um cubo. Trata-se de uma proposição geral, enumerada em um segundo momento, no acesso a suas particularidades. Aqui, vemos a

forma como possibilidade da estrutura, descrita por 961 formas de representação da figura.

"Só se houver objetos, pode haver forma fixa no mundo" (§2.026). "O fixo, o subsistente e o objeto são um só" (§2.027). "No estado de coisas os objetos estão uns em relação aos outros de um modo determinado" (§2.031). Há um modo de representar os limites do cubo, e esta forma poderá ser encontrada na particularidade da proposição, em suas variáveis. Já a afirmação de Wirth (1999) quanto à característica das estruturas dinâmicas da informação demonstra que a própria estrutura de uma variável ou objeto pode ser alterada.

A característica de tais problemas reside no fato de que, durante o processamento, não apenas os valores de tais variáveis são alterados, como também o são as próprias estruturas das variáveis em questão. (WIRTH, 1999, p.141).

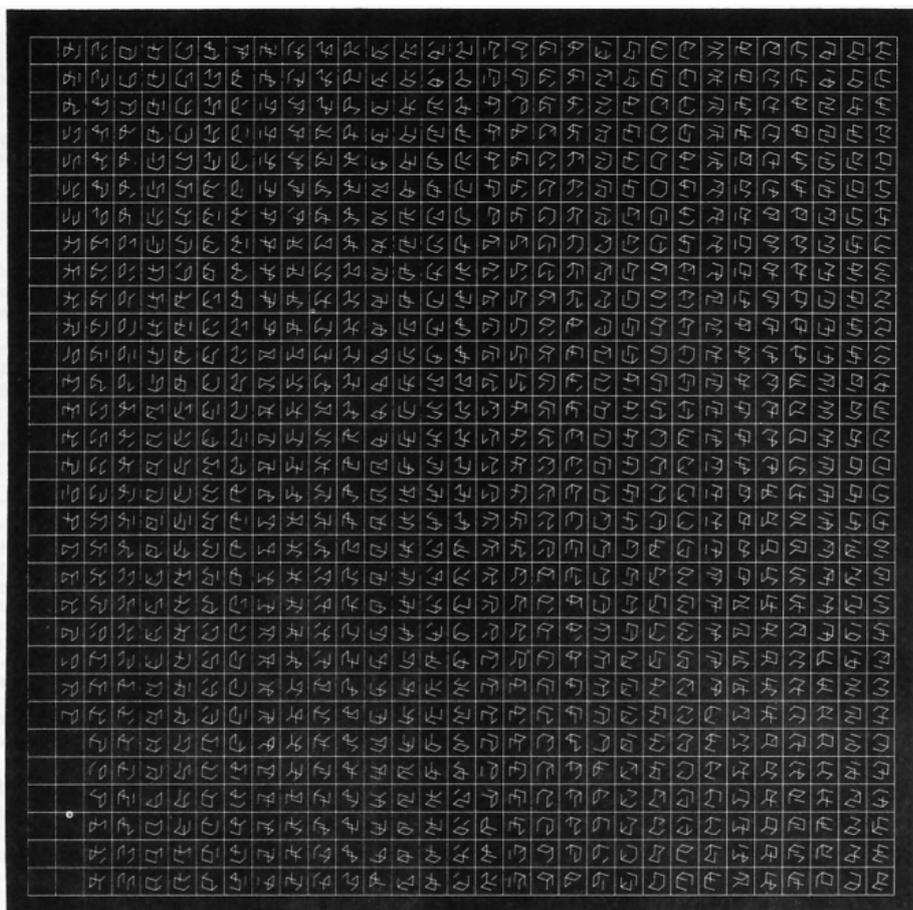


Figura 11 — Manfred Mohr. "The cubic limit series", 1976 - One of four works from ICCH/2

Podemos submeter a que comparações esta "geometria sem distâncias" com o desenho ou, como concatenar desenho e variáveis de forma elegante? O cubo é um limite, somente o cubo. As 961 formas podem ser lidas como um único cubo, o mesmo cubo. Não importam as suas particularidades, contanto que todas as possibilidades sejam esgotadas. As formas de resolução desse problema podem ser acessadas num segundo momento, observando até o lugar em que se encontram na matriz lógica criada por Mohr.

As regras tomam o lugar da visualidade. Não é possível memorizar todas as possibilidades da série, mas é possível reproduzi-las, bastando somente compreender a sua construção, ou seja, uma combinatória de 12 linhas unindo pontos. Obtém-se o par cubo e geometria sem distância, pois ao cubo também é dada completa liberdade, desde que esteja em concordância com a regra.

2.7 Recursão e repetição

Entende-se por recursão o procedimento onde um objeto é definido em termos de si próprio, como na *Figura 12*. De acordo com Niklaus Wirth (1999), o poder da técnica está na “possibilidade de se definir um conjunto infinito de objetos através de uma formulação finita”. Na prática, a recursividade é uma repetição acumulativa de uma função que chama e retorna seus próprios parâmetros, onde $A = A[B,A]$.

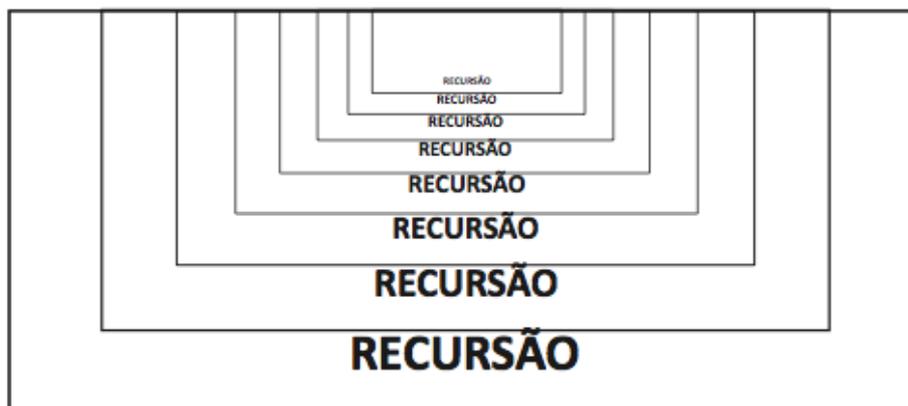


Figura 12 — Uma imagem recursiva.

Considera-se que o tema recursão também pertença ao campo da visualidade. Concatenada com a escrita, possibilita a transmissão de determinada informação aos conjuntos recursivos da imagem, surgindo assim uma sequência visual construída por uma fórmula algébrica, ou mesmo sequências semânticas estruturadas por regras.

Assim, a técnica da recursão é utilizada como ferramenta para a definição de um princípio de estruturação, a sequenciação ou a iteração. As sequências e as iterações são tão comuns que são usualmente consideradas como padrões fundamentais de estruturas e comportamentos. Deve-se, porém, ter sempre em mente que tais padrões podem ser alternativamente definidos por meio da recursão. Entretanto, o inverso não é verdadeiro, já que a recursão pode ser utilizada de modo elegante e eficaz na definição de estruturas muito mais sofisticadas. (WIRTH, 1999, p.165)

O padrão gráfico da *Figura 13* é baseado numa demonstração feita por Wirth (1999, p.114-117). Apresenta uma imagem construída por uma técnica recursiva.

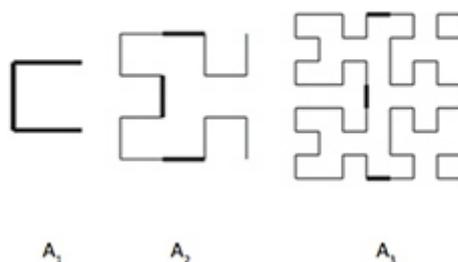


Figura 13 — Curvas de Hilbert de ordem 1, 2, 3. Padrão baseado em Wirth, 1999, p.115.

No exemplo de Wirth, pressupomos a *Figura 14*, pois a regra diz que A_1 deve ser construído por quatro elementos conectados por três retas. A_2 por quatro elementos de A_1 , com metade do tamanho, rotacionado adequadamente e conectado por mais três retas. A_3 por A_2 , e assim por diante.

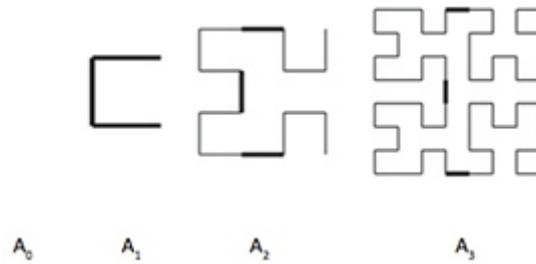


Figura 14 — Curvas de Hilbert de ordem 0, 1, 2, 3. Demonstração de um desenho construído à partir de uma regra.

Já o exemplo da *Figura 15* transfigura a noção de recursão e executa a demonstração com restrição repetitiva. O nó *a4* e *a5* nada trazem de novo para a proposição, sendo mesmo elementos de repetição e não de ritornelo.

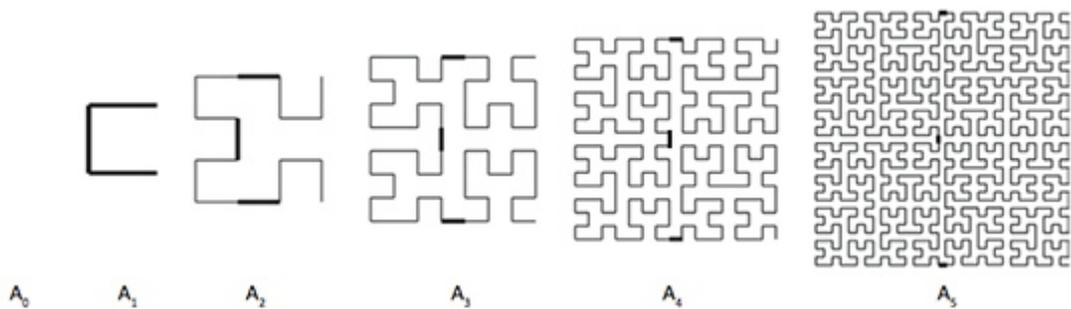
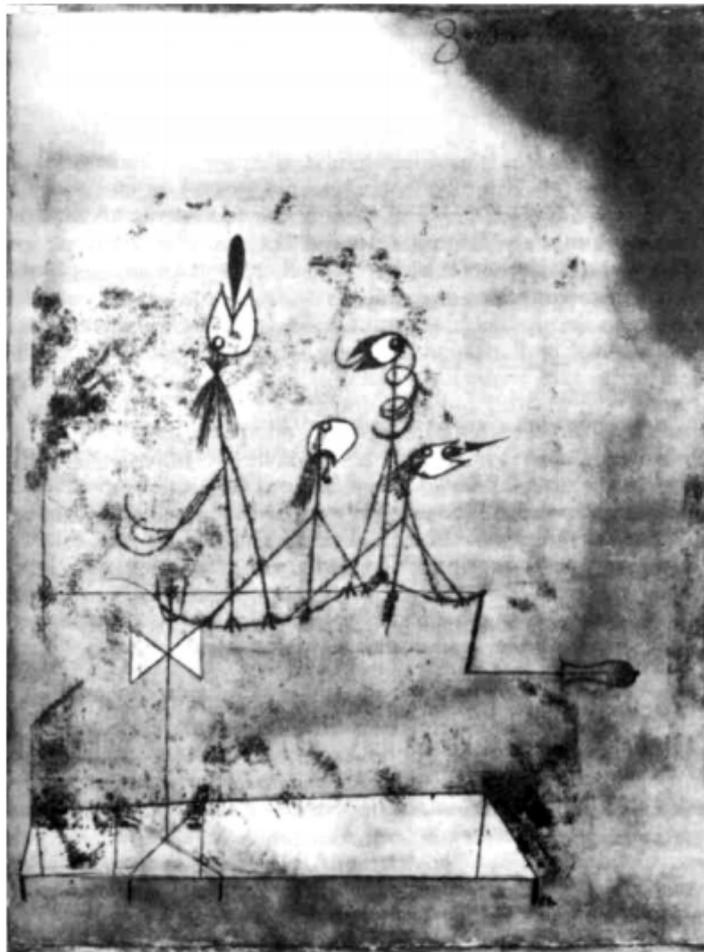


Figura 15 — Curvas de Hilbert de ordem 1, 2, 3 acrescido de *a4* e *a5*. Demonstração de um padrão recursivo possível, mas restrito.

A *Figura 16* é uma captura de tela da página "Acerca do Ritornelo", do livro *Mil Platôs* de Gilles Deleuze e Félix Guattari. Máquina de Gorjear, ou seja, máquina de canto, de sons melódiosos, como a música aquática — *Water Music* — de Händel. O ritornelo é o estribilho, um termo que descreve uma ação de retorno na música.



A máquina de gorjear

Figura 16 — Acerca do Ritornelo, A máquina de gorjear. Captura de tela do livro Mil Platôs vol.4. (DELEUZE & GUATTARI, 1997)

Para Deleuze e Guattari (1997), a noção de Ritornelo é direcionada para uma compreensão territorial, sendo agenciado em motivos e paisagens territoriais. A noção carrega os elementos de direção, dimensão e passagem.

Agora que consideramos expressões que têm seu conteúdo nelas mesmas, temos uma outra distinção: o rosto com seus correlatos visuais (olhos) remete à pintura; a voz remete à música com seus correlatos auditivos (a orelha é ela própria um ritornelo, ela tem a forma do ritornelo). A música é primeiro uma desterritorialização da voz, que se torna cada vez menos linguagem, assim como a pintura é uma desterritorialização do rosto. (DELEUZE & GUATTARI, 1997, pág.90).

O confronto desses conceitos direcionam a compreensão para outras paragens, sendo preciso assimilar o modo de construção de uma voluta barroca, por exemplo. Trata-se de uma forma envolvente, semelhante a uma curva cíclica, linha curva plana sempre em movimento, que se desenvolve em uma relação áurea de proporção. A construção desse modelo com compasso e régua pressupõe a descrição detalhada dos seus pontos. Eis a repetição controlada e tecnicamente precisa, como se fossem circuitos integrados por curvas e correntes elétricas.

Assim, outro caminho para esse padrão é o da *Figura 17*, onde A_4 é construído a partir dos três elementos anteriores, notadamente A_1 , A_2 e A_3 , os elementos que contém alguma informação visual. A inovação deste quarto passo está na sincronia dos três desenhos em lugar de três retas. Essas retas não são mais repetidas, mas alinhadas em camadas. O que acontece é que A_0 permuta a regra, prosseguindo assim a elegância da proposição. A figura também perspectiva planos sucessivos, além de exemplificar um circuito integrado descrito em um plano coordenado para *web*.

[...] sincronizadores de ritmos (a palavra "sincronizador" é ambígua, pois estes sincronizadores moleculares não procedem por medida igualizante ou homogeneizante, e operam de dentro, entre dois ritmos). Não seria a consolidação o nome terrestre da consistência? O agenciamento territorial é um consolidado de meio, um consolidado de espaço-tempo, de coexistência e de sucessão. O ritornelo opera com os três fatores. (DELEUZE & GUATTARI, 1997, pág. 124).

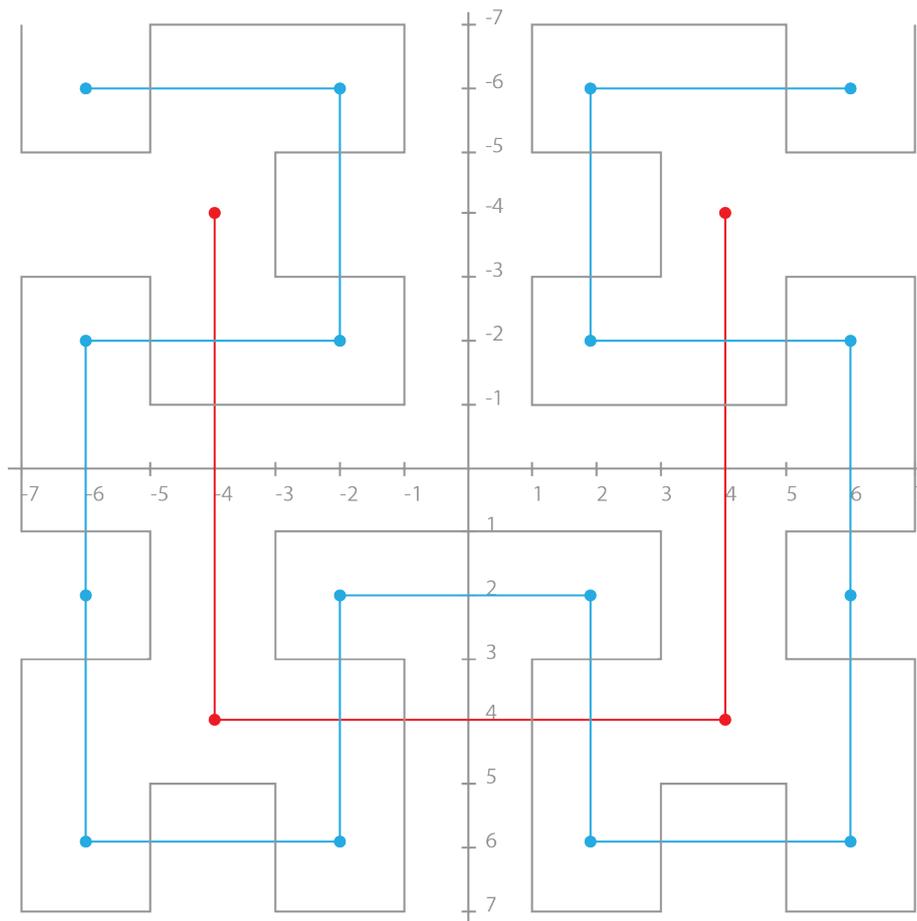


Figura 17 — Curvas de Hilbert de 4ª ordem no plano coordenado.

Abaixo, circuito integrado da Figura 17, enumeração dos pontos (x,y) no plano coordenado:

Curva 1ª Ordem $\Rightarrow (-4, -4) — (-4, 4) — (4, 4) — (4, -4)$;

Curva 2ª Ordem $\Rightarrow (-6, -6) — (-2, -6) — (-2, -2) — (-6, -2) — (-6, 2) — (-6, 6) — (-2, 6) — (-2, 2) — (2, 2) — (2, 6) — (6, 6) — (6, 2) — (6, -2) — (2, -2) — (2, -6) — (6, -6)$;

Curva de 3ª Ordem $\Rightarrow (-7, -7) — (-7, -5) — (-5, -5) — (-5, -7) — (-1, -7) — (-1, -5) — (-3, -5) — (-3, -3) — (-1, -3) — (-1, -1) — (-5, -1) — (-5, -3) — (-7, -3) — (-7, 1) — (-5, 1) — (-5, 3) — (-7, 3) — (-7, 7) — (-5, 7) — (-5, 5) — (-3, 5) — (-3, 7) — (-1, 7) — (-1, 3) — (-3, 3) — (-3, 1) — (3, 1) — (3, 3) — (1, 3) — (1, 7) — (3, 7) — (3, 5) — (5, 5) — (5, 7) — (7, 7) — (7, 3) — (5, 3) — (5, 1) — (7, 1) — (7, -3) — (5, -3) — (5, -1) — (1, -1) — (1, -3) — (3, -3) — (3, -5) — (1, -5) — (1, -7) — (5, -7) — (5, -5) — (7, -5) — (7, -7)$.

O padrão contido na *Figura 18* demonstra uma curva de Hilbert $H_1...H_5$, também baseada em Wirth (1999, p.117). H_5 opera uma multiplicação quantitativa do elemento de 4ª Ordem. Novamente, a elegância caminha no limiar do sucesso, entre a utilidade e a estética. É evidente o desgaste do recurso axial. As possibilidades estão limitadas, assim como estavam no momento apresentado pela *Figura 15*. Exemplificando com base em A_4 e A_5 , as proposições novamente parecem esgotadas.

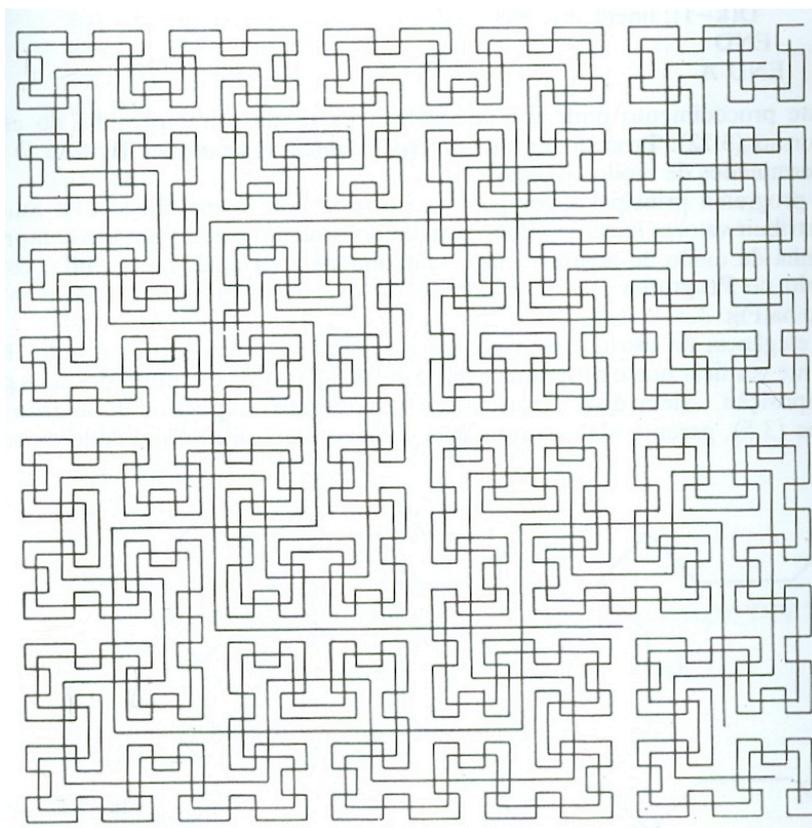


Figura 18 — Curvas de Hilbert de ordem $H_1...H_5$. Padrão retirado de WIRTH, 1999.

Nesse momento, o passo A_0 movimenta-se como variável, torna e define novas regras a serem seguidas, como em uma máquina (sem rodas). O trecho abaixo, retirado da obra *Tuttaméia* de João Guimarães Rosa, exemplifica esse momento, em que a estética defronta-se com o real. O objeto “faca” existe somente para satisfazer a lógica do texto. Enquanto o texto reduz o objeto à uma figuração vazia, como na permuta do elemento A_0 .

O nada é uma faca sem lâmina, da qual se tirou o cabo [...] porque a ideia do objeto “não existindo” é necessariamente a ideia do objeto “existindo”, acrescida da representação de uma exclusão desse objeto pela realidade atual tomada em bloco. (ROSA, 2001, p.32)

Quando procuramos refletir sobre o Desenho inserido nos campos da Arte e da Tecnologia, não podemos deixar de questionar sobre a contribuição que a disciplina trouxe para ambas as áreas no desenvolvimento de suas técnicas. O Desenho possibilita a reflexão teórica sobre a técnica ao estabelecer uma comunicação baseada num modelo bidimensional, ou seja, uma geometrização do que está no espaço pelo plano. Trata-se de um risco sobre o suporte ou mesmo uma escrita sobre o objeto. Tomado neste sentido, a técnica é “questionada” pela teoria, resultando na “tecnologia – isto é: a '*logia*' ou o '*saber*' da técnica – [que] só é possível depois do advento do conhecimento teórico.” (VARGAS, 1994, p.224).

A questão da técnica é especialmente importante para a civilização latina, sendo essencial para o desenvolvimento e mesmo uma questão de sobrevivência. É interessante recordar o jargão que os intelectuais bizantinos – em especial Demétrio Cidones – repetiam sobre nossa civilização. Cidones foi um tradutor grego das obras de Tomás de Aquino e incentivador da aproximação entre os povos, dizendo que “antes, ninguém conseguiu convencer o nosso povo de que existe alguma inteligência nos latinos, ou de que são capazes de elevar sua mente para outros temas que não o comércio, a navegação e a guerra” (CHADWICK e EVANS, 1987). Fico aqui tentando tirar-lhes a razão, mas com muita dificuldade.

O fato é que as três matérias acima são historicamente especialidades dos latinos, o que nos induz a aferir qual seria a dimensão logística e a esfera de influência dessas questões em nosso modo de pensar. Durante séculos houve um importante desenvolvimento e mesmo uma escola técnica latina fundamentada na armazenagem e no transporte de mercadorias por via marítima. Os romanos foram exímios empreiteiros, tendo construído estradas por todos os seus domínios, desenvolvido a metalurgia em larga escala e a construção monumental. Além disso, foram conquistadores que de certa forma absorviam as culturas dominadas sem maiores problemas. No mesmo passo, os

portugueses fizeram história e fundaram o Brasil com a mesma doutrina de navegação, comércio e guerra.

Na civilização grega clássica aparece duas novas formas de saber: a teoria (*theoreo*) e a técnica (*techné*). A teoria busca uma “forma” constante nas coisas sobre a qual é possível pensar logicamente e estabelecer um critério de verdade. Já a técnica é um saber-fazer transmitido de geração em geração ou registrado em tratados. É necessário saber que nem as técnicas nem a arte participavam dos critérios de verdade estabelecidos pela teoria, uma vez que elas “não implicam temas verdadeiros ou falsos; mas, seus resultados têm que ser eficientemente válidos. Elas têm como motivo realizar efetivamente aquilo a que se propõem.” (VARGAS, 1994, p.178). Essa mentalidade somente se modificou no período renascentista com a confluência entre técnica, arte e ciência, uma crença de que qualquer disciplina poderia ser aferida por conhecimentos científicos.

CAPÍTULO 3

Detalhes de implementação

A programação gráfica é utilizada para leitura, manipulação e gravação de imagens ou sequência de imagens a partir de um *script*. A principal diferença entre a técnica de programação gráfica e um aplicativo para desenhar é que o programador precisa especificar textualmente o desenho ao invés de manipular fisicamente outras interfaces (VERBRUGGEN, 2002). Desenhar via texto é, então, essencialmente uma atividade lógica. Trata-se de escrever o desenho. (SABOIA, 2001).

De acordo com Martien Verbruggen (2002), no livro *Graphic Programming with Perl*, a programação gráfica consiste em trabalhar com um limitado conjunto de objetos denominados primitivas de desenho, sendo elas: 1 — Suporte, 2 — Coordenadas, 3 — Objetos e 4 — Ferramentas.

1 — Suporte é o instrumento que fornece as variáveis de ambiente, permitindo a plotagem do desenho na tela. Há grande multiplicidade de suportes, sendo prezados módulos simples e independentes. É positivo trabalhar o suporte com eficiência e sem cair nas restrições impostas pelo equipamento. Essa capacidade de adaptação às circunstâncias é chamado desenho responsivo.

2 — Coordenadas são todas as medidas e métricas utilizadas pelo desenho. Seja para um algoritmo ou para dados de interface, a conversão de valores é o que possibilita maior versatilidade dos objetos sobre o ambiente.

3 — Objeto é qualquer elemento que possa ser manipulado. Independe de natureza. Um objeto pode, em relação a outro objeto, ter acoplagem forte ou fraca. Ou seja, pode depender totalmente de outros objetos, de modo a se tornar inseparável e depender do funcionamento desse objeto — acoplagem forte — ou funcionar e retornar um resultado independente de outro objeto — acoplagem fraca. Além disso, pode apresentar outras propriedades. Exemplos de objetos para desenho: linha, reta, curva, gps ou glonass, tela, navegador, objetos criados por programação, pixel, array, etc.

4 — Ferramentas são os arsenais de trabalho disponíveis para desenho. Exemplos de ferramentas: estruturas de repetição, diferentes qualidades de desenho, funcionalidades do cliente e do servidor, espaços de cor como na *Figura 19*, formatos de imagem, linguagens de programação, métodos de comunicação ou mesmo interfaces ou aplicativos prontos. Apesar de ser prática comum a escolha de ferramentas prontas (programas e aplicativos), é recomendável, principalmente no desenho *scriptivo*, criar novos módulos a partir do arsenal básico disponível.

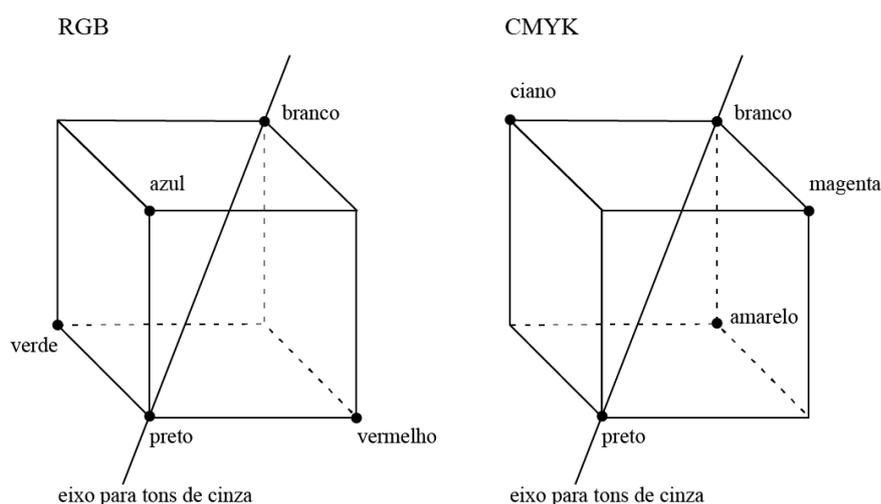


Figura 19 — Cubos de cor em unidades RGB e CMYK

Um exemplo de ferramenta é o espaço de cor. Pode-se dizer que um espaço de cor é o conjunto de todas as combinações possíveis feitas a partir de 3 cores primárias. Todas as cores que vemos podem ser representadas por uma composição de três outras cores, que são correlatos dos três tipos de sensores retinianos que utilizamos para perceber o espectro visual (VERBRUGGEN, 2002, p.7). O espaço de cor mais conhecido talvez seja a mistura de tintas com vermelho, amarelo e azul. No entanto, é possível chegar a outros espaços de cor igualmente úteis, como o *RGB*, *CMYK*, *HLS*, *YCbCr* ou *Grayscale*.

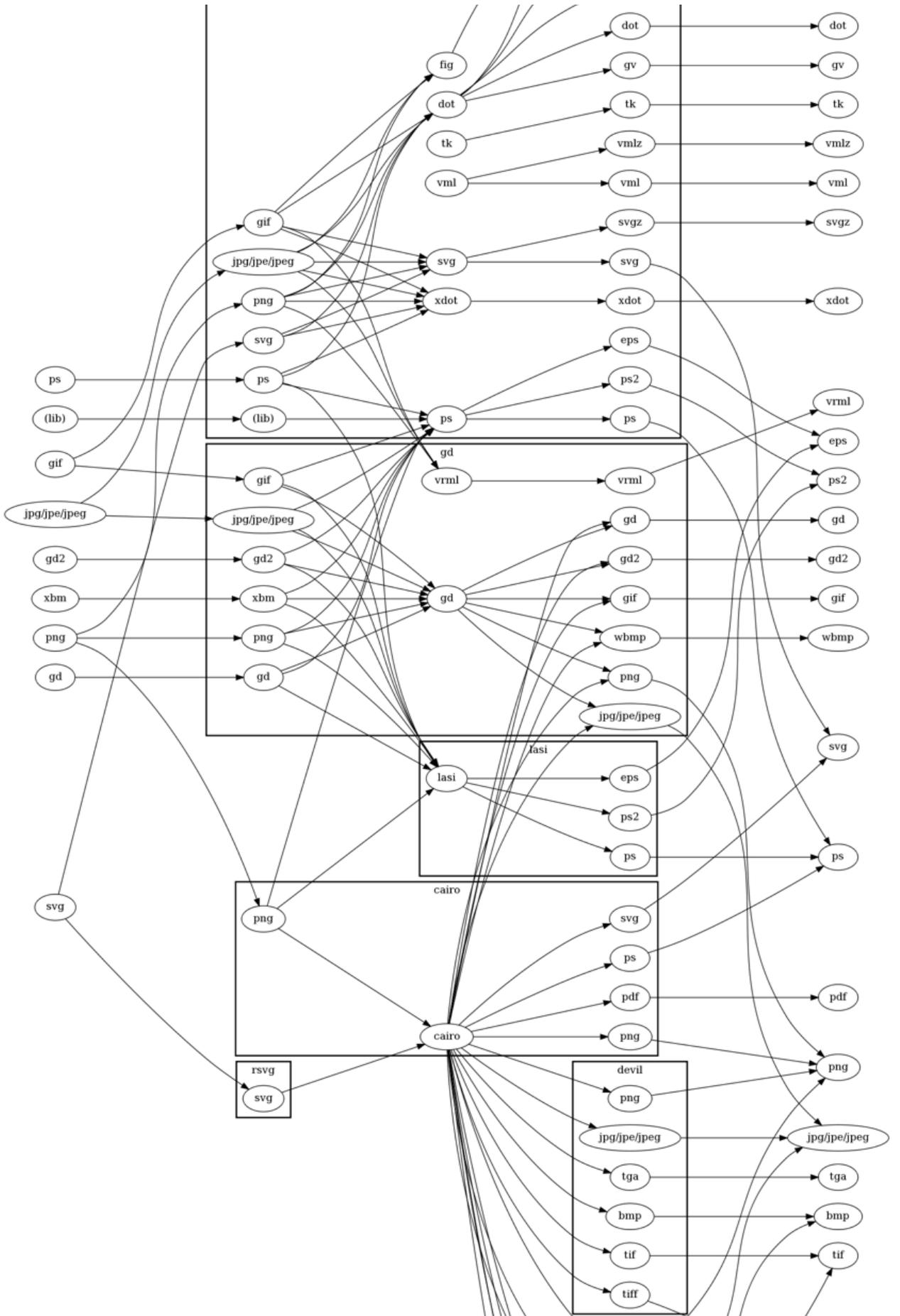
Outra ferramenta importante para o desenho é o formato de arquivo para gráficos (*Tabela 5*) e o diagrama de formatos de arquivos (*Figura 20*), que apresenta um breve exemplo da versatilidade de conversão dos formatos e a maneira como são representados os dados da imagem. Por fim, é importante

salientar que cada tarefa e cenário projetivo exige um formato apropriado de arquivo, sendo os formatos *PNG*, *JPEG* e *SVG* mais utilizados para a internet.

Formato	Tipo	Máxima profundidade de cor	Canal Alfa	Máximo tamanho de imagem	Múltiplas imagens	Ferramentas
<i>PNG</i>	<i>Bitmap</i>	8-bit <i>indexed</i> 16-bit <i>grayscale</i> 48-bit <i>RGB</i> +16-bit w/alfa	16 bit	2Gig x 2Gig	Não	GD, GD::Graph, ImageMagick, Gimp, pode ser incorporado em <i>SWF</i> , <i>SVG</i> , <i>canvas</i> .
<i>JPEG</i>	<i>Bitmap</i>	12-bit <i>grayscale</i> 36-bit <i>RGB</i> 32-bit <i>CMYK</i>	Não	64k x 64k	Não	ImageMagick, GD, Gimp, pode ser incorporado em <i>SWF</i> , <i>SVG</i> , <i>canvas</i> .
<i>GIF</i>	<i>Bitmap</i>	8-bit indexado	1-bit transparência	64k x 64k	Sim	ImageMagick, Gimp
<i>SVG</i>	Vetor	24-bit	Limitado pelo usuário	Limitado pelo usuário	Sim	ImageMagick, Gimp
<i>SWF</i>	Vetor	24-bit	8-bit	1Gig x 1Gig	Sim	Biblioteca Ming
<i>Post-Script</i>	Documento	Limitado pelo dispositivo de visualização	Não	Limitado pelo dispositivo de visualização	Sim	Módulo <i>PostScript</i> , ImageMagick, Gimp
<i>PDF</i>	Documento	Limitado pelo usuário	Limitado pelo usuário	Limitado pelo usuário	Sim	PDF::API2, ImageMagick
<i>TIFF</i>	<i>Bitmap</i>	32-bit, 48-bit	Limitado pelo usuário	4Gig	Sim	ImageMagick, GraphicsMagick, NetPBM, Gimp

Tabela 5 — Formatos de imagem e suas características. Formatos de arquivo baseado no original disponível em <http://helpx.adobe.com/br/photoshop/using/file-formats.html>.

Figura 20 — Diagrama de formato de imagens. Fonte: <http://www.grqphviz.org/doc/info/output.html>



3.1 Arquitetura cliente/servidor

Na internet, uma troca normal de informações entre aplicações é feita por duas partes: o cliente, que está operando o programa; e o servidor, que processa as requisições do cliente e retorna com as informações adequadas (VINEGAR & KOVALYOV, 2013) e (WALLACE, 2003).

A internet surgiu nos Estados Unidos, a partir da ARPAnet, um projeto desenvolvido na década de 1960 pela *Advanced Research Projects Agency Department of Defense* - ARPA, cujo objetivo era a troca de informações entre diversas pessoas, a partir das mesmas vias de comunicação. Projetada para operar sem um controle centralizado, inicialmente seu uso estava restrito às universidades e aos institutos de pesquisa. Em seguida utilizada pelas forças armadas e, posteriormente, para fins comerciais (DEITEL, 2002).

Paralelamente ao desenvolvimento da internet, organizações de todo mundo criaram redes para comunicações internas e entre organizações, diversificando a oferta de *softwares* e *hardwares* e dificultando a padronização da comunicação. Para resolver esse problema foram criados protocolos específicos de conexão entre redes, hoje conhecidos como *Transmission Control Protocol (TCP)* e *Internet Protocol (IP)* – TCP/IP.

A *world wide web (www)* é um sistema global de hipertexto que utiliza a internet como sistema de transporte de dados, uma mistura entre tecnologias de computação e comunicação que possibilita a localização e a visualização de documentos baseados em multimídia, como vídeos, textos, áudios e gráficos. A *web* atual, criada em 1989, na Organização Européia de Pesquisa Nuclear (CERN), é fruto de uma tradição que, desde a década de 1940, buscava associar diferentes fontes de informação, por meio da computação interativa. Logo se tornou veículo eficiente e dinâmico para troca de informações instantâneas, interligando por *hiperlinks* diferentes computadores pelo mundo. Isso permitiu à internet alcançar a escala global com que é conhecida atualmente, indispensável à indústria, saúde, cultura, transporte, logística, engenharia, arte, política, organizações e sociedade em geral (DEITEL, 2002).

Na *web*, a interação é feita por meio de um conjunto de métodos e cabeçalhos que permite a clientes e servidores trocarem informações de uma maneira uniforme e padronizada. A interação é dividida em 2 partes (VINEGAR & KOVALYOV, 2013). A primeira parte é o usuário que opera o navegador e a segunda é o servidor de conteúdo (Figura 21).

Quando o usuário clica em um botão, movimenta o ponteiro ou visita uma página, o navegador processa a informação e faz uma requisição ao servidor. Essa requisição pode ser trabalhada dinamicamente com aplicações do lado do servidor, que interpreta dados específicos de cada usuário, reconhecendo que alguns recursos devem ser manipulados de forma diferente.

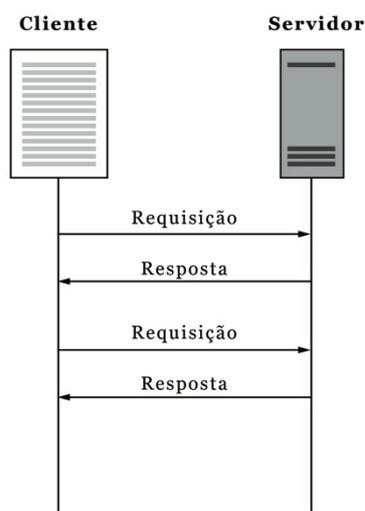


Figura 21 — Arquitetura cliente/servidor

Há diversos métodos, técnicas e restrições neste processo de comunicação. É possível obter desde informações sobre geolocalização, navegador *web* utilizado e conexão HTTP até informações sobre o ambiente, as circunstâncias e o entorno do cliente e do servidor. Ajustar esse entremeio comunicativo permite que as aplicações gerem conteúdo *web* dinâmico, tendo como base as informações fornecidas pelo próprio usuário. (VERBRUGGEN, 2002)

Para o escopo desse trabalho o que interessa são os métodos e as técnicas apropriadas para o trabalho de programação gráfica e desenho *scriptivo* na *web*. Cumpre

lembrar que há diversas linguagens e técnicas possíveis para se desenhar tanto no lado do cliente quanto no lado do servidor, passando por linguagens características para o transporte de dados, com desenho vetorial, misto ou mesmo o trabalho com um desenho distribuído e construído por vários usuários ao mesmo tempo. De forma geral, quase todas as linguagens de programação possuem um módulo gráfico, cabendo ao desenhista escolher a técnica mais adequada para cada projeto.

Escolheu-se 3 formas de desenhar com *scripts* no ambiente *web*, consideradas aqui como 3 qualidades de desenho *scriptivo* que serão detalhadas nas seções seguintes. Para desenho no lado do cliente, utilizou-se o *JavaScript*, juntamente com o elemento *canvas* do HTML5. Para o desenho de gráficos vetoriais foi utilizado o *Scalable Vector Graphics (SVG)*, que é baseado em *eXtensible Markup Language (XML)*. Já para desenhar do lado do servidor, foi utilizado o módulo *Drawn Graphics (GD)* da linguagem de programação *Perl*, que renderiza o desenho em um formato *bitmap*, como *.jpeg*, *.png* ou outro.

Após a análise e exemplificação destas três qualidades de desenho, veremos que a principal diferença entre eles não reside no resultado visual (como percebido na criação de um quadrado simples em cada linguagem) e sim no método e forma de comunicação de cada um dos gráficos com o ambiente cliente/servidor (*Figura 22*).

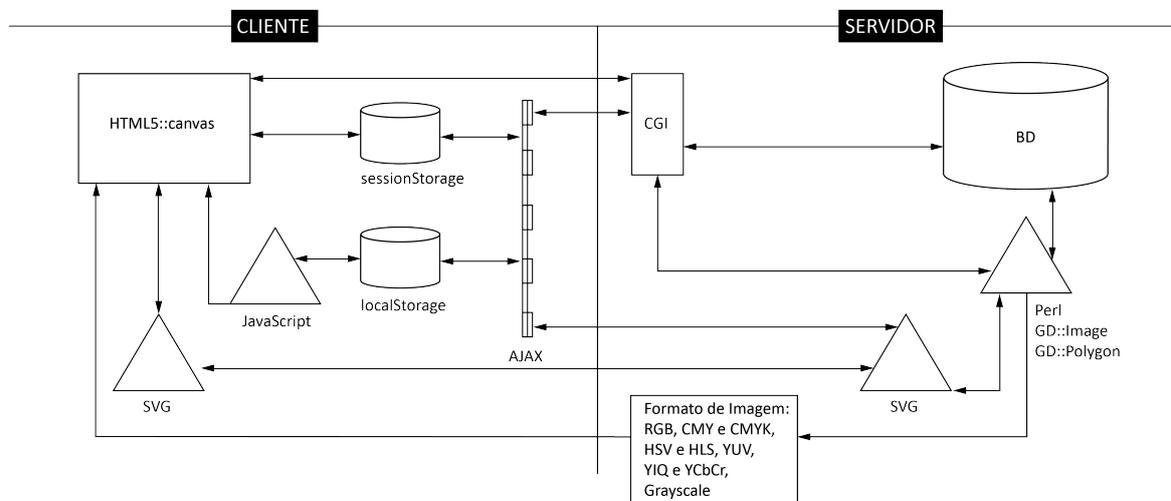


Figura 22 — Arquitetura cliente/servidor

3.2 Desenhando no lado cliente

O dispositivo de interação gráfica no lado cliente é ideal para a transmissão de sensações sensoriais e motoras para os usuários, pois possui um tempo de resposta próximo ao real. De forma geral, a construção de uma interface gráfica com interação multidimensional é feita no lado cliente da aplicação. A comunicação com o servidor é obtida com requisições assíncronas via texto e totalmente transparente para o usuário.

Em nosso exemplo, o desenho feito no lado cliente é um gráfico escrito em *JavaScript* e renderizado no elemento *canvas* do *HyperText Markup Language (HTML5)*. O *canvas* é uma tela de *bitmap* utilizada para renderização rápida de gráficos e outras imagens visuais. É dependente de resolução e pode ser utilizado para gerar gráficos dinâmicos em duas ou três dimensões². A tela é basicamente uma imagem baseada em pixel com uma *Application Programming Interface (API)* de desenho.

O elemento *canvas* destina-se a delimitar uma área para criação dinâmica de imagem, como gráficos estáticos, jogos e gráficos dinâmicos e imagens em geral criadas com linguagem de programação dinâmica. Todo trabalho de criação e animação é feito com *JavaScript*. (SILVA, Maurício Samy. *HTML 5 / Maurício Samy Silva*. -- São Paulo : Novatec Editora, 2011).

A *Figura 23* representa a dinâmica de criação de um desenho com a linguagem. A página do cliente faz a requisição do conteúdo para o servidor e este retorna com o arquivo *JavaScript*. A renderização acontece do lado do cliente dentro do elemento *canvas* do *HTML5*. A vantagem de trabalhar com essa técnica de desenho é a rapidez e flexibilidade para comportamentos dinâmicos, isso porque o desenho é construído no próprio dispositivo do usuário.

Um desenho é dinâmico se a sua forma muda sem ser alterado manualmente por um indivíduo ou se acompanha as interações e movimentos do usuário. "Escrever" o desenho no lado do cliente com o *JavaScript* permite a adição desse tipo de

² <http://msdn.microsoft.com/>
<http://msdn.microsoft.com/pt-br/library/dn151487.aspx>

comportamento para formas, textos e cores, de tal forma que torna possível a criação de jogos e animações complexas e leves.

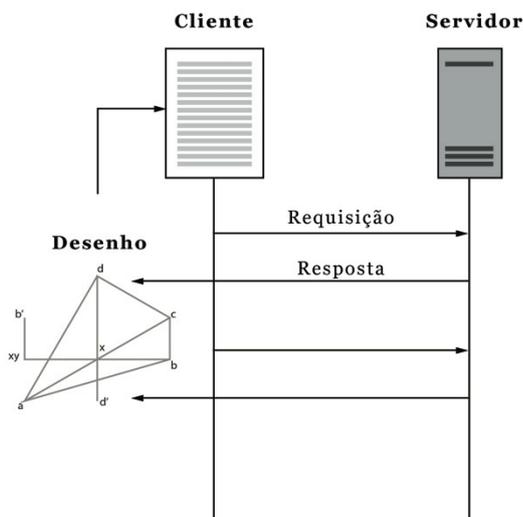


Figura 23 — Desenho do lado cliente com o elemento canvas HTML 5 e JavaScript.

A primeira versão de *JavaScript* foi lançada em 1995 pela Netscape e a Sun Microsystems (SILVA, 2011). As funcionalidades gerais da linguagem incluem a manipulação do navegador, do conteúdo e da apresentação de uma página *web*. Interagir com formulários, com outras linguagens dinâmicas e requisitar, receber e tratar dados de um servidor de conteúdo.

As coordenadas de *canvas* são definidas seguindo o sistema de eixos cartesianos *x* e *y*, cuja origem é o canto superior esquerdo da tela. As coordenadas *x* crescem da esquerda para a direita e *y* do topo para baixo (Figura 24). Já a Tabela 6 fornece uma gama de primitivas, métodos auxiliares, atributos, alocação de cores e textos contidos na API de *canvas*.

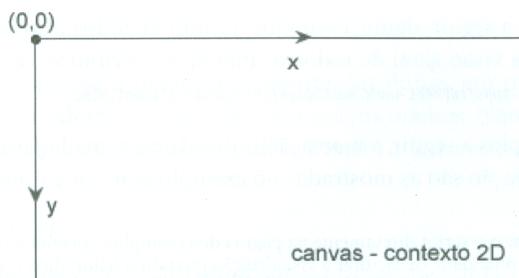


Figura 24 — Coordenadas do contexto 2D de canvas (SAMY, p.143).

strokeRect(x,y,l,h)	Contorno de um retângulo. x e y definem o canto superior esquerdo, enquanto l e h definem a largura e altura em pixel.
strokeStyle	Define a cor das linhas do elemento gráfico. Aceita espaços de cor HSL, HSLA, RGB, RGBA e keywords.
lineWidth	Espessura da linha em pixel.
fillRect(x,y,l,h)	Desenho de retângulo em cor sólida.
fillStyle	Define a cor de preenchimento do elemento gráfico.
clearRect(x,y,l,h)	Apaga uma área retangular em <i>canvas</i> .
shadow	Atributos para sombreamento: shadowColor, shadowOffsetX, shadowOffsetY, shadowBlur.
beginPath()	Resetar o path corrente e iniciar a construção de um novo path.
moveTo(x,y)	Posiciona a pena de desenho nas coordenadas x e y.
lineTo(x,y)	Desenha uma linha da posição que se encontra a pena até as coordenadas x e y.
stroke()	Aplica os estilos definidos pelos atributos de criação stroke.
arcTo(x1,y1,x2,y2,r)	Desenha um arco com início em x1 e y1, término nas coordenadas x2 e y2 e raio.
arc(x,y,r,anguloInicial,anguloFinal,sentido)	Desenha um arco com centro nas coordenadas x e y, raio em r, ângulos em radianos inicial e final e sentido horário ou anti-horário do ângulo.
fill()	Preenche com cor sólida os caminhos definidos pelos atributos de criação fill.
rect(x,y,l,h)	Desenha um retângulo.
bezierCurveTo(cp1x,cp1y,cp2x,cp2y,x,y)	Desenha curvas bezier a partir da inserção de pontos de controle.
closePath()	Fecha um caminho desenhado.
globalAlpha	Define uma transparência.

lineJoin	Define a forma como as linhas se unem: miter, bevel e round.
lineCap	Define o acabamento das extremidades das linhas.
createLinearGradient(x,y,x1,y1)	Cria um gradiente linear.
createRadialGradient(x,y,r,x1,y1,r1)	Cria um gradiente radial.
addColorStop(offset,cor)	Define a distribuição das cores do gradiente.
creatPattern(imagem,Repetição)	Insere uma imagem como preenchimento de um contorno.
GlobalCompositeOperation	Define 11 tipos de composição entre duas imagens: source-atop, source-in, source-over, source-out, destination-atop, destination-in, destination-over, destination-out, lighter, copy, xor.
scale(x,y)	Escalonamento x e y de todos os elementos inseridos em <i>canvas</i> .
translate(x,y)	Deslocamento x e y de todos os elementos inseridos em <i>canvas</i> .
rotate(ângulo)	Rotaciona todos os elementos contidos em <i>canvas</i> .
drawImage(imagem,dx,dy)	Insere uma imagem nas coordenadas especificadas.
drawImage(imagem,dx,dy,dl,dh)	Insere uma imagem nas coordenadas e redimensiona em pixel.
drawImage(imagem,sx,sy,sl,sh,dx,dy,dl,dh)	Recorta uma imagem a partir das coordenadas especificadas, insere e redimensiona.
font	Características usadas nos textos.
fillText(texto,x,y[,cMax])	Define o preenchimento do texto a ser inserido.
strokeText(texto,x,y[,cMax])	Define o contorno do texto a ser inserido.
textAlign	Define o alinhamento do texto a partir do ponto de inserção: start, end, left, right, center.
textBaseline	Define o alinhamento do texto a partir da base: top, hanging, alphabetic, middle, ideographic e bottom.

Tabela 6 — Métodos e atributos da API de *canvas*

Exemplos de desenho com *canvas/javascript*

```
strokeRect(120,70,200,200);
```



Figura 25 — Um quadrado escrito em JavaScript e plotado em canvas HTML5.

O exemplo a seguir apresenta um mecanismo de transformação retirado da obra "*Mecanismos en la tecnica moderna*" do professor e cientista Ivan Ivanovich Artobolevski, impressa na URSS em 1976. A tradução para o português, o tratamento da imagem e a montagem da fórmula correspondente foi realizadas pelo autor da dissertação. O item aqui apresentado foi selecionados dentre um conjunto de 2288 mecanismos contidos na obra de Artobolevski, que por sua vez estão subdivididos em 12 grupos básicos classificados segundo critérios estruturais e construtivos.

Mecanismo de alavanca e conexões de Artobolevski para transformar a circunferência na curva de Sluse.

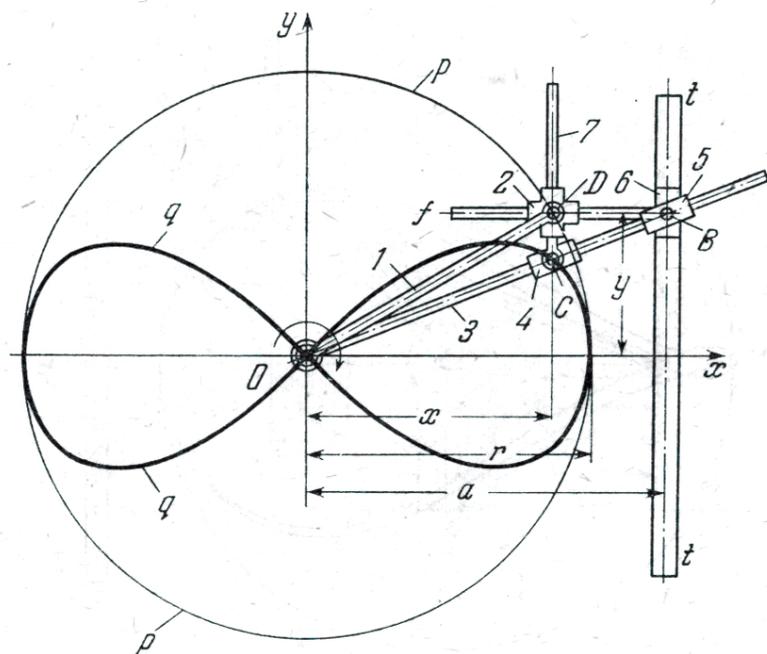


Figura 26 — Mecanismo. Imagem original de Artobolevski, 1977, pág.274.

[1219] - Os comprimentos dos elementos do mecanismo satisfazem a condição $OD = r$, onde r é o raio da circunferência $p - p$, que se transforma na curva de Sluse. O elemento 1, que gira ao redor do eixo fixo O , forma o par de rotação D com a corrediça cruceiforme 2, que possui os eixos perpendiculares. O elemento 3, que gira ao redor do eixo fixo O , formando pares de translação com a corrediça 4 e 5. A corrediça 5 forma o par de rotação B com a corrediça 6, que se desloca ao longo das guias $t - t$ que, por sua vez, possui o eixo paralelo a Oy . A corrediça 6, com a sua transversal Bf , cujo eixo é paralelo ao eixo Ox , forma um par de translação com a corrediça 2. O elemento 7 forma o par de rotação C com a corrediça 4 e um par de translação com a corrediça 2. Ao girar o elemento 1 ao redor do eixo O , o ponto D descreve a circunferência $p - p$, e o ponto C descreve a curva $q - q$ de Sluse, cuja equação é

$$x^4 + r^2x^2 + a^2y^2 = 0$$

onde a é a dimensão constante do mecanismo. (ARTOBOLEVSKI, 1976).

```

1.      /*Jesus Cristo_ O Caminho, A Verdade e A Vida*/
2.      var desenho, x, y, tempo = 0.0030, a = 0, repetição=320;
3.      canvas = document.getElementById('telaDeSaída');
4.      desenho = canvas.getContext('2d');
5.      desenho.translate(canvas.width / 2, canvas.height / 2);
6.
7.      function curvaDeSluse() {
8.      for(i = 0; i <repetição; i++){
9.      x = Math.cos((a * (Math.PI)));
10.     y = Math.sin((a * (Math.PI) / 180) + 30);
11.     a = a + tempo;
12.     desenho.beginPath();
13.     desenho.fillStyle = '#000000';
14.     desenho.arc(250*Math.cos((x)*Math.PI/(y)),
15.     180*Math.sin((x)*Math.PI/(y))*(Math.cos((x)*Math.PI/(y))), 7, 2*Math.PI, 0, true);
16.     desenho.fill();
17.     };};
18.     curvaDeSluse();

```

Código 1 — Curva de Sluse escrita em JavaScript.



Figura 27 — Curva de Sluse escrita em JavaScript e plotado em canvas HTML5. Fernando Aquino, 2014.

Acesso: <http://www.programaçãográfica.com.br>

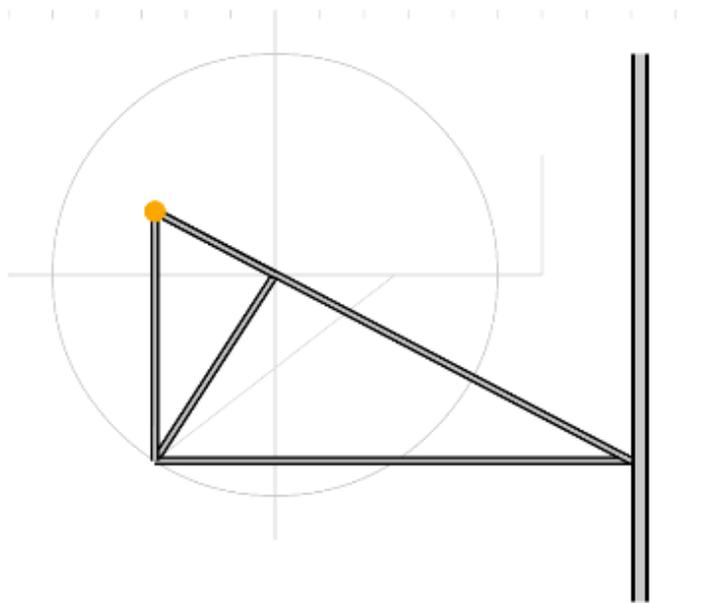


Figura 28 — Alavancas baseadas nas condições do mecanismo. A. Aquino, 2014.

Pseudocódigo: desenha a Curva de Sluse.

- Linha 1 — *Comentário;*
- Linha 2 — *Declaração das variáveis do desenho: tempo, repetição, xy, a;*
- Linhas 3 e 4 — *Chamada do suporte canvas, contexto 2D, telaDeSaída HTML5;*
- Linha 5 — *Divisão da tela por 2 e centralização do eixo coordenado;*
- Linha 7 a 18 — *Função curvaDeSluse();*
- Linha 8 a 18 — *Laço de repetição;*
- Linha 9 — *Variável trigonométrica x;*
- Linha 10 — *Variável trigonométrica y;*
- Linha 11 — *Variável de sequência temporal;*
- Linha 12 a 17 — *Construção do desenho;*
- Linha 13 — *Cor de preenchimento do desenho;*
- Linha 14 e 15 — *Traçado do arco. Coração do desenho. Cada uma das 320 repetições do laço desenha uma parte da curva de Sluse a partir da*

estrutura do arco, que é arc(x,y,raio,inícioDoArco,finalDoArco,sentido). Somente as posições x e y sofrem variações ao longo da repetição, sendo os outros métodos apenas para definir o formato e o raio do arco.

- Linha 16 — *Traçar desenho.*
- Linha 18 — *Executar função curvaDeSluse().*

3.3 Desenho vetorial para web

Dentre as três qualidades de desenho apresentadas, o desenho vetorial é a que menos apresenta perda da qualidade da imagem. Isso acontece porque o desenho vetorial não é representado por uma matriz de pixel fixa, mas por um conjunto de pontos e instruções para conectar esses pontos. Por esse motivo, os gráficos vetoriais são um recurso de imagem especial para o desenho *scriptivo*, visto que todas as instruções podem ser dinamizadas por linguagens de programação, inclusive via *JavaScript*.

A criação de gráficos vetoriais é feita no formato *SVG*, sigla em inglês que significa *Scalable Vector Graphics*. De início, pode-se diferenciar os tipos de gráficos baseados em vetor dos tipos baseados em *bitmap*. Quando redimensionados, os gráficos que adotam o formato *bitmap* perdem qualidade de imagem enquanto os gráficos vetoriais permanecem inalterados. Isso porque, como dito acima, as informações nos gráficos vetoriais estão contidas em vetores, isto é, linhas unindo pontos.

Gráficos *SVG* são descritos por vetores. Vetores são linhas (retas ou curvas) que unem dois pontos. Assim, para criar um gráfico vetorial, devemos descrever uma sequência de pontos formando um contorno aberto ou fechado e definir espessura e cor dos vetores, bem como cor e efeitos do preenchimento. (SILVA, Maurício Samy. Fundamentos da *SVG* / Maurício Samy Silva. -- São Paulo : Novatec Editora, 2012).

A forma encontrada para descrever esse tipo de gráfico foi com a utilização da linguagem de marcação XML, que é um protocolo para transporte e gerenciamento de informações. Uma linguagem de marcação trabalha com a informação incluída em um documento de forma a otimizar a sua utilização via identificação e relação entre as partes que a compõem. A XML foi desenvolvida em 1996 dentro da *World Wide Web*

Consortium (W3C) e é amplamente suportada para o intercâmbio de dados entre aplicações.

A força dessa técnica está em sua integração natural em páginas *web*. Suponha que um gráfico *SVG* se encontra no meio, entre o cliente e o servidor. Inicialmente, o cliente carrega o gráfico com todas as funcionalidades nele contidas (*Figura 27*). Qualquer alteração de estado nessas funcionalidades deverá ser requisitada ao servidor e o arquivo recarregado. Naturalmente essa característica pode ser utilizada positivamente, aproveitando a chamada ao servidor para acoplar funcionalidades ligadas ao novo contexto do usuário. A menos que seja realmente necessário, não se renderiza persistentemente um gráfico provindo do lado servidor.

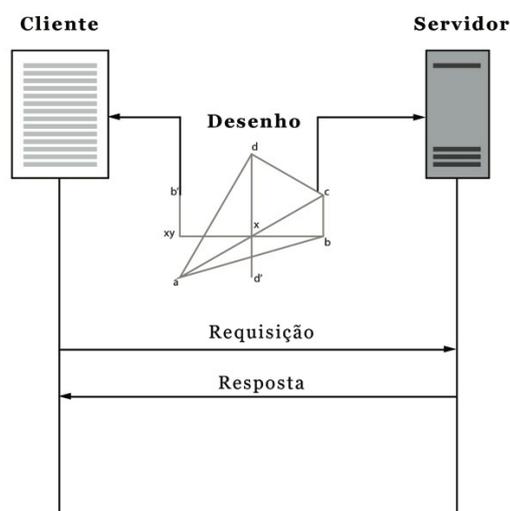


Figura 29 — Estrutura de desenho vetorial com SVG/XML.

A *Tabela 7* contém uma lista de elementos e atributos da *SVG*, bem como a forma de escrevê-los.

<code><rect x="10" y="10" width="100" height="200"/></code>	O elemento <i>rect</i> cria um retângulo nas posições <i>x</i> e <i>y</i> com as dimensões <i>width</i> x <i>height</i> .
<code><rect ... fill="#ffd700"/></code>	Adicionar o atributo <i>fill</i> define uma cor de preenchimento para o retângulo.
<code><rect ... rx="70" ry="50"/></code>	Os atributos <i>rx</i> e <i>ry</i> são utilizados para

<code><rect ... stroke="#000000" stroke-width="5"/></code>	arredondar as bordas de um retângulo. Definem a cor e a espessura de uma borda, respectivamente.
<code><rect ... fill-opacity="0.7" stroke-opacity="0.3"/></code>	Definem a opacidade de preenchimento e de borda. Os valores variam de 0 (transparência total) a 1 (cor totalmente opaca).
<code><line x1="7" y1="7" x2="40" y2="210" stroke="#ffd700" stroke-width="5"/></code>	O elemento <i>line</i> cria uma linha reta com início nas coordenadas <i>x1 y1</i> e termina nas coordenadas <i>x2 y2</i> . Os atributos <i>stroke</i> e <i>stroke-width</i> definem a cor e a espessura da linha.
<code><line ... stroke-linecap="square"/></code>	Esse atributo define o formato das extremidades de uma linha em <i>butt</i> , <i>square</i> e <i>round</i> .
<code><line ... stroke-dasharray="7,13,2,5,12,8"/></code>	Define linhas tracejadas a partir de uma sequência de números inteiros.
<code><circle cx="235" cy="123" r="77"/></code>	O elemento <i>circle</i> cria círculos e circunferências com centro nas coordenadas <i>cx cy</i> e raio <i>r</i> .
<code><ellipse cx="132" cy="55" rx="95" ry="33"/></code>	O elemento <i>ellipse</i> cria ellipses com centro nas coordenadas <i>cx cy</i> e raio horizontal <i>rx</i> e raio vertical <i>ry</i> .
<code><polygon points="70,30 120,30 290,75 120,315 120,315,70,30"/></code>	O elemento <i>polygon</i> cria polígonos fechados em qualquer formato com as coordenadas de cada um dos vértices definidas pelo atributo <i>point</i> .
<code><polyline stroke-width="#ffd700" stroke="#000000" points="30,30 60,120 60,30 150,150 270,70"/></code>	O elemento <i>polyline</i> cria uma forma aberta por linhas retas conectadas. Os vértices são definidos pelos pares de coordenadas <i>x</i> e <i>y</i> .

Tabela 7 — Elementos e atributos SVG/XML.

```
<rect x="120" y="70" width="100" height="100" stroke="#000000"/>
```

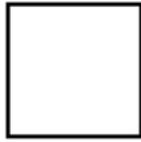


Figura 30 — Um quadrado vetorial escrito com SVG/XML.

```
/* Jesus Cristo _O Caminho, A Verdade e A Vida*/  
<?php  
//  
$points_count = 20;  
$points = array();  
for( $p=0; $p<$points_count; $p++ )  
{  
    $d = ( 60 / $points_count ) * $p;  
    $x = -15 + ( cos( deg2rad( $d ) ) * 25 );  
    $y = 25 + ( sin( deg2rad( $d ) ) * 5 );  
    $points [] = array( 'x' => $x, 'y' =>$y );  
}  
echo ("<?xml version=\"1.0\" standalone=\"no\"?>\n");  
?>  
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/TR/SVG/DTD/svg10.dtd">  
<svg style="shape-rendering:geometricPrecision;" viewBox="0 0 100 100"  
    xmlns:xlink="http://www.w3.org/1999/xlink"  
    xmlns="http://www.w3.org/2000/svg" preserveAspectRatio="xMidYMid meet">  
<?php  
foreach( $points as $start ) {  
    $sx = $start['x'];  
    $sy = $start['y'];  
    foreach ( $points as $end ) {  
        $ex = $end['x'];  
        $ey = $end['y'];  
    }  
?>  
<g id="r">
```

```

<!--path fill-rule="nonzero" style="fill:#000000;stroke:#FF0000;stroke-width:0.2"
      d="M<?php echo( $sx." ".$sy); ?> L<?php echo( $ex." ".$ey ); ?> Z"/-->
<circle fill-rule="nonzero" style="fill:none;stroke:#000000;stroke-width:0.02"
      cx="<?php echo( $sx )?>" cy="<?php echo( $ey )?>" r="20"/>
<!--animate attributeName="width" values="<?php echo( $sx )?>,50,200"
      dur="6s" repeatCount="indefinite"/-->
</g>
<?php
}}?>
</svg>

```

Código 2 — Desenho recursivo escrito em SVG/XML.

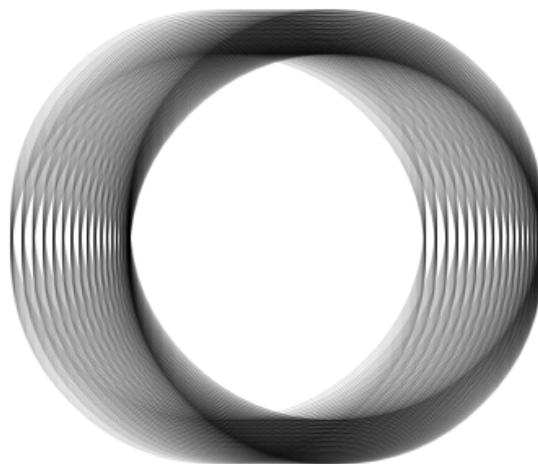


Figura 31 — Um desenho recursivo vetorial escrito com SVG/XML.

3.4 Comparação entre *SVG* e *canvas*

A comparação entre o elemento *canvas* e o *SVG* não está destinada para que se escolha uma tecnologia em detrimento da outra. Não há qualquer conclusão sobre qual é a melhor, pois o ideal é trabalhar com as duas em conjunto, obtendo uma interação dinâmica e versátil. O que se pretende aqui é ressaltar as potencialidades de ambas e incentivar um

desenvolvimento baseado em cenários, podendo mesmo incluir um terceiro elemento, que é o desenho em *Perl*, que roda no lado servidor e será estudado na Seção 3.6. A *Figura 30* ilustra uma análise de cenários entre as tecnologias.

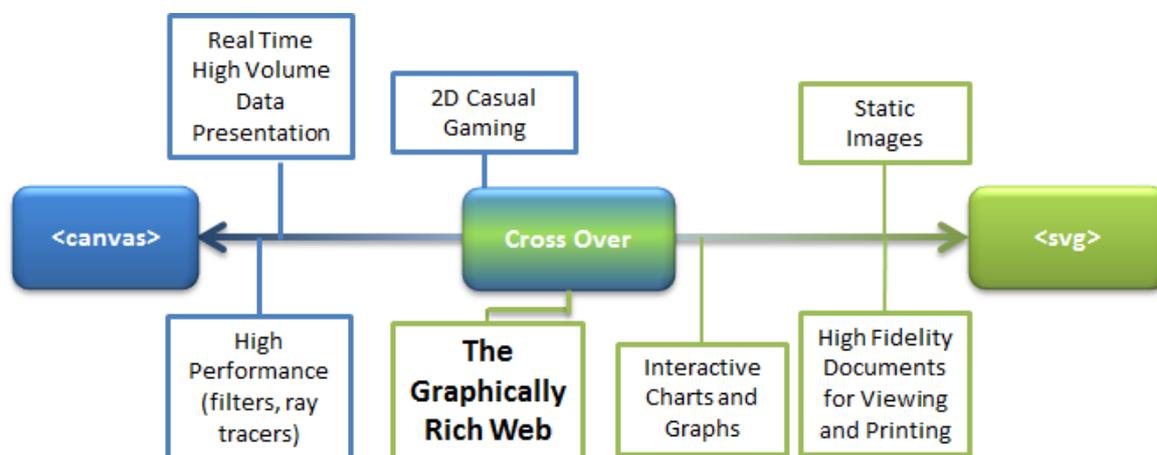


Figura 32 — Análise de cenários para escolha entre canvas e SVG (imagem de blogs.msdn.com)

O elemento *canvas* é uma imagem rastreada, baseada em pixel, sendo indicado para desenhar elementos gráficos complexos que exigem uma rápida renderização de imagem. Indicado para o trabalho de desenho interativo contido em uma fórmula e aplicações com grandes quantidades de dados, pois utiliza *JavaScript* para a construção. *JavaScript* é linguagem que roda no lado cliente e é atualmente a principal linguagem cliente-servidor para navegadores *web*. É recomendada para modulação de voz em *displays*, movimentos baseados na interação do usuário e apresentações em tempo real.

Já o desenho em *SVG* é vetorial, podendo ser redimensionado sem que se perca a qualidade. É indicado para gráficos, imagens estáticas para impressão e animações pré-definidas. É ideal para visualização de gráficos gerados a partir de dados existentes, pois transforma XML em *SVG*. É recomendado para organogramas interativos, fluxogramas, plantas de edifício, esquemas de engenharia, mapas e gráficos financeiros. É também utilizado em desenhos distribuídos por provedores de terceiros e podem ser gerados por inúmeros programas vetoriais e CAD presentes no mercado. A *Tabela 8* faz uma análise entre as duas tecnologias.

Análise entre desenhos feitos com *SVG* e com *canvas*

SVG

Baseados em modelos de objetos, seus elementos são semelhantes aos elementos HTML.

Vários elementos gráficos se tornam parte do Document Object Model (DOM).

Apresentação visual criada com marcação e modificada tanto por CSS quanto por programação de *script*.

Modelo de interação evento/usuário baseada em objeto no nível de elementos gráficos primitivos: linhas, retângulos, polígonos.

Marcação *SVG* e modelo de interação baseado no objeto que suporta acessibilidade aos seus elementos.

Canvas

Baseada em pixel, onde a tela é um elemento de imagem com uma API de desenho.

Elemento HTML simples semelhante ao `` em comportamento.

Apresentação visual criada e modificada por programação de *script*.

Modelo de interação evento/usuário dinâmico programados manualmente a partir de coordenadas e construção em tempo real.

API não suporta acessibilidade. Técnicas de marcação devem ser utilizadas juntamente com *canvas*.

Tabela 8 — Comparação entre desenhos feitos em SVG e canvas (baseado em <http://blogs.msdn.com/b/ie/archive/2011/04/22/thoughts-on-when-to-use-canvas-and-svg.aspx>)

3.5 Desenho no lado servidor

O desenho do lado do servidor aqui apresentado é feito com a biblioteca gráfica GD da linguagem de programação *Perl*. *Perl* é uma linguagem que roda do lado do servidor, isto é, dependente de uma máquina remota onde ficam hospedadas as funcionalidades necessárias para o funcionamento do programa (*Figura 31*). GD.pm é uma interface para biblioteca gráfica criada por Thomas Boutell's e possui diversos métodos de trabalho para manipulação de imagens, como controle de cor, comandos de desenho, métodos de saída de dados, comandos de transformação, canais alfa e métodos diversos.

Perl é uma das linguagens de programação *web* mais populares, isso devido a suas capacidades únicas e técnicas poderosas na manipulação de textos. É utilizada em aplicações de missão crítica em diversos setores, sendo estável e multiplataforma (perl.org.br). Está disponível sob a Licença Artística ou a *GNU General Public Licence (GPL)*.

Tal como as bibliotecas apresentadas nos itens anteriores, o módulo GD oferece um conjunto de primitivas de desenho e outros métodos auxiliares destinados à criação dinâmica de gráficos. Embora a saída do programa seja uma imagem *bitmap* (PNG, JPEG, etc.), o desenho feito no lado servidor tem como benefício o acesso irrestrito a todo tipo de aplicações e bancos de dados. Um recurso excelente para o desenho *scriptivo*.

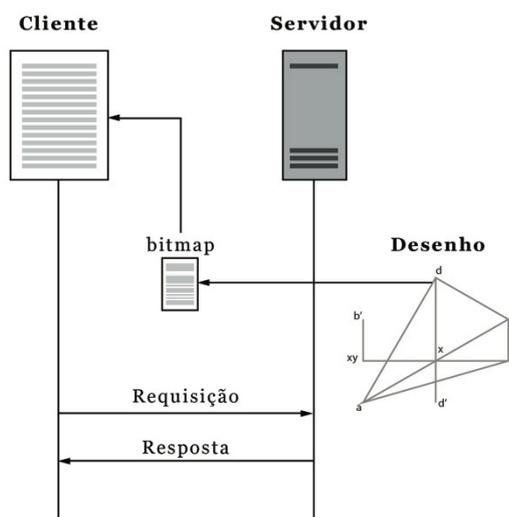


Figura 33 — Desenho no lado do servidor plotado como bitmap com GD/Perl

Uma ressalva importante é que, de acordo com Martien Verbruggen, *Perl* não é uma linguagem totalmente adequada para o processamento e manipulação de grandes volumes de imagens, trabalho que exige um grande processamento de números. No entanto, *Perl* é conhecida como "a fita adesiva da internet", ou seja, é utilizada para juntar algoritmos, pedaços de dados, ferramentas de texto, bibliotecas, APIs e outros recursos provindos das mais diversas fontes e linguagens de programação. É isso que torna *Perl* uma linguagem útil para a programação gráfica na *web*.

Por causa da capacidade de colagem do *Perl*, existem diversas interfaces para manipulação de bibliotecas gráficas e programas disponíveis em forma de módulos. Outros pacotes têm sido escritos para fazer o uso desses módulos e criar gráficos em um nível superior, como por exemplo para criar gráficos em pizza ou barras. (VERBRUGGEN, 2002).

A *Tabela 9* fornece um conjunto de primitivas e métodos auxiliares para desenhar com o módulo GD. Estão disponíveis alocação de cor, preenchimento, pincéis, caracteres, fontes com tamanho definido em pixels etc. Todos os métodos devem ser chamados com um objeto GD::image ou GD::font. Após a apresentação dessa tabela, será aprofundado o estudo da primitiva *polygon* e da criação de curvas com *polyline*. As informações aqui apresentadas foram colhidas no site do *Comprehensive Perl Archive Network* (CPAN) — <http://search.cpan.org/~lds/GD-2.52/GD.pm> — e no livro *Graphic Programming with Perl*, de Martien Verbruggen.

setPixel(x,y,color)	Define a cor do pixel especificado nas coordenadas.
line(x1,y1,x2,y2,color)	Desenha uma linha sólida entre os pontos especificados.
dashedLine(x1,y1,x2,y2,color)	Desenha uma linha tracejada entre os pontos especificados.
rectangle(x1,y1,x2,y2,color)	Desenha um retângulo a partir dos cantos especificados.
filledRectangle(x1,y1,x2,y2,color)	Desenha um retângulo preenchido a partir dos cantos especificados.
polygon(poly,color)	Desenha um polígono especificado em <i>poly</i> . Esse polígono é criado com a ferramenta <i>GD::Polygon::new()</i> .
polyline	Desenha uma polilinha, que é um polígono aberto.
filledPolygon(poly,color)	Desenha um polígono preenchido especificado em <i>poly</i> .
ellipse(cx,cy,width,height,color)	Desenha elipses com centro do arco em <i>cx</i> e <i>cy</i> , largura, altura e cor.
filledEllipse(cx,cy,width,height,color)	Desenha elipses preenchidas a partir das coordenadas especificadas.
arc(cx,cy,width,height,start,end,color)	Desenha arcos com centro especificado em <i>x</i> e <i>y</i> , largura, altura, começo e fim do ângulo.
filledArc(cx,cy,width,height,start,end,color [,arc_style])	Desenha arcos preenchidos com as especificações correspondentes.
fill(x,y,color)	Inundação de cores. Preenche os pixels especificados com a mesma cor.
fillToBorder(x,y,bcolor,color)	Inundação de cores. Preenche os pixels especificados com a mesma cor até encontrar um pixel de outra cor especificada.

setBrush(brush)	Definir o pincel a ser utilizado a partir de GD::Image object.
setStyle(color-list)	Definir um estilo de linha para a lista de cor especificada.

Tabela 9 — Primitivas e métodos auxiliares de GD Perl.

```
rectangle(120,70,220,170,$preto);
```



Figura 34 — Um quadrado escrito em Perl.

3.5.1 — Construção de Polígonos

Podemos destacar o método auxiliar *polygon(a,b)* destinado à construção e manipulação de polígonos. Um polígono (*poli* = muitos, *gono* = ângulos) é uma figura geométrica fechada e formada apenas por segmentos de reta. Os elementos de um polígono são: vértices, lados, ângulos internos, ângulos externos e diagonais. Os polígonos são classificados em função do número de lados que apresentam: triângulo (3); Quadrilátero (4); Pentágono (5); Hexágono (6); Heptágono (7); Octógono (8); enágono (9); decágono (10); pentadecágono (15); icoságono (20 ângulos).

Em GD *Perl*, os métodos de trabalho com polígonos são: criar, adicionar e recuperar pontos, alterar e deletar vértices, traçar linhas, aferir o comprimento e os vértices, definir limites, mapear, transformar, escalar, rotacionar e deslocar vértices ou todo o polígono.

Abaixo, a descrição detalhada para alguns métodos de criação e manipulação da primitiva *polygon* de *Perl*³.

Novo Polígono: cria um polígono vazio sem vértices.

```
$polígono = new GD::Polygon;
```

Adicionar: adiciona um ponto (x, y) ao polígono.

```
$polígono->addPt($x,$y);
```

Recuperar: recupera o ponto do vértice especificado.

```
($x,$y) = $polígono->getPt($index);
```

Alterar: altera o valor de um vértice existente.

```
$polígono->setPt($index,$x,$y);
```

Deletar: exclui o vértice especificado e retorna seu valor.

```
($x,$y) = $polígono->deletePt($index);
```

Excluir: exclui todos os vértices e restaura o polígono ao seu estado inicial.

```
$polígono->clear();
```

Traçar: desenhar a partir do vértice atual para um novo vértice.

```
$polígono->addPt($x,$y);  
$polígono->toPt($dx,$dy);
```

Comprimento: retorna o número de vértices do polígono.

³ <http://www.cpan.org/>.

```
$points = $polígono->length;
```

Vértices: retorna uma lista de todos os vértices do objeto polígono. Cada elemento da lista traz uma referência matricial de (x,y).

```
@vertices = $polígono->vertices;  
foreach $v (@vertices)  
print join(", ", @$v), "\n";
```

Limites: retorna o menor retângulo que circunscreve o polígono. O valor retorna um *array* contendo os lados esquerdo, superior, direito e inferior do retângulo.

```
($esquerda, $superior, $direita, $inferior) = $polígono->bounds;
```

Deslocamento: desloca todos os vértices do polígono no valor horizontal (\$dx) ou vertical (\$dy) especificados. Números positivos movem o polígono para baixo e para a direita.

```
$polígono->offset($dx, $dy);
```

Mapear: mapeia o polígono a partir de um retângulo de origem, move e redimensiona o polígono para uma posição equivalente em um retângulo de destino. Tanto os retângulos de origem quanto os de destino possuem os lados esquerdo, superior, direito e inferior.

```
$polígono->map($srcL, $srcT, $srcR, $srcB, $destL, $destT, $dstR, $destB);
```

Escalar: escala cada vértice do polígono pelos fatores (x,y).

```
$polígono->scale($sx, $sy);
```

Transformar: executa cada vértice do polígono por meio de uma matriz de transformação: \$sx e \$sy são fatores (x,y) de escala; \$rx e \$ry são fatores (x,y) de rotação; \$tx e \$ty são fatores (x,y) de deslocamento.

```
$polígono->transform($sx, $rx, $sy, $ry, $tx, $ty);
```

```

#!/usr/bin/perl
use cPanelUserConfig;
use GD;
my $im = new GD::Image(1500,1000);
my $branco = $im->colorAllocate(255,255,255);
my $preto = $im->colorAllocate(0,0,0);
my $x; my $y;
$im->transparent($white);
$im->interlaced('true');

$poly = new GD::Polygon;
    $poly->addPt(500,200);
    $poly->addPt(500,300);
    $poly->addPt(400,300);
    $poly->addPt(400,400);
    $poly->addPt(500,400);
    $poly->addPt(500,500);
    $poly->addPt(600,500);
    $poly->addPt(600,400);
    $poly->addPt(700,400);
    $poly->addPt(700,300);
    $poly->addPt(600,300);
    $poly->addPt(600,200);

for( $a = 0; $a < 12; ++$a){
    $g = $a;
    ($x,$y) = $poly->getPt($g);
    $im->string(gdGiantFont,61*$a,95,$x,$preto);
    $im->string(gdGiantFont,61*$a,145,$y,$preto); };
$im->polygon($poly,$preto);

binmode STDOUT;
print "Content-type:image/png\n\n";
print $im->png;

```

Código 3 — Polígono escrito em Perl.

500	500	400	400	500	500	600	600	700	700	600	600
200	300	300	400	400	500	500	400	400	300	300	200

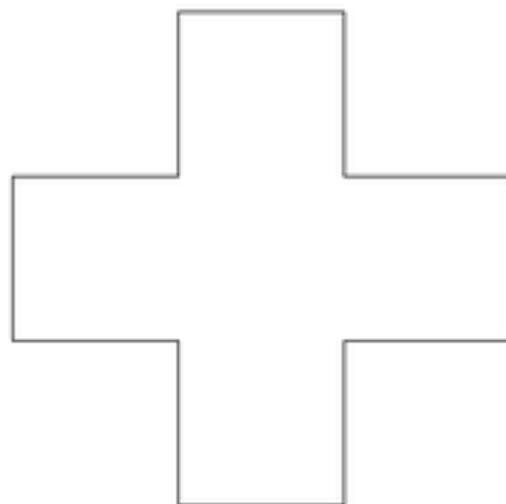


Figura 35 — Saída .PNG do programa do Código 3. Curva “W” de 1ª Ordem (WIRTH, 1999).
Matriz de pontos sobre o plano.

3.6 Desenho distribuído

A ideia de um desenho distribuído nada mais é do que *scripts* remotos servidos a partir de servidores de terceiros, que possam constituir uma imagem na *web* sem que esteja sob o controle de um único servidor de conteúdo. A questão resvala no assunto da computação distribuída e também sobre ambientes virtuais distribuídos, mas o interesse aqui é somente a construção de uma imagem que seja virtualmente constituída de diversas fontes. Como se fosse uma parede de azulejos, onde cada um procedesse de uma origem diferente, a imagem final é obtida pela constituição desse todo. Sobre a técnica *JavaScript* remoto, que é utilizada para o trabalho com o desenho distribuído, Ben Vinegar e Anton Kovalyov escrevem:

JavaScript remoto é código cliente independente, executado no *website* do provedor de conteúdo, mas servido a partir de um endereço *web* remoto. É usado na criação de aplicativos altamente

distribuídos, desde widgets sociais e rastreadores de análise de tráfego até aplicativos incorporados. (VINEGAR & KOVALYOV, 2013, pág.17).

Assim, os mecanismos estudados no *Capítulo 4* tornam-se aplicações de desenho que são executados em diferentes servidores e interligados pela internet. O procedimento tende a tornar esses mecanismos mais robustos, pois a possibilidade de se tornar remoto pressupõe um maior apuro nas interconexões das suas partes, resultando em saídas e entradas explícitas para dados e objetos, visto que empregará recursos de terceiros.

A aparente simplicidade da ideia acaba na primeira tentativa de implementação. De acordo com Ben Vinegar e Anton Kovalyov, no livro *JavaScript Remoto* (2013), há muito trabalho por traz de um aplicativo remoto. É necessário distribuir o código para os provedores de conteúdo, carregar bibliotecas e arquivos necessários para o funcionamento e por fim, passar informações de ambiente da página hospedeira, de modo a tornar possível a busca e o envio de dados para o destino correto.

[...] se considerarmos o desenvolvimento normal de aplicativos locais, os *scripts* remotos enfrentam desafios adicionais. Eles exigem que você execute seu código em ambientes de navegador desconhecidos, compartilhados e possivelmente hostis. (VINEGAR & KOVALYOV, 2013, pág.51).

A *Tabela 10* demonstra os resultados obtidos no compartilhamento de recursos entre origens. Nota-se que há três elementos para averiguação: protocolo, domínio e porta. Protocolos são conjuntos de instruções e diretrizes responsáveis pela comunicação entre cliente e servidor. Neste caso, os protocolos HTTP e HTTPS (segurança) são utilizados para padronizar a comunicação. As portas de conexão são recursos criados para facilitar a conexão entre dois computadores e geralmente possuem tarefas padrão. A porta 80 é responsável pela conexão HTTP. Acessar uma porta diferente requer a sua descrição no esdereço. Já o domínio é o endereço na internet que são transformados para endereços de IP por meio do Sistema de Nomes de Domínio (*Domain Name System - DNS*).

URL de origem	Resultado	Motivo
http://desenhoscriptivo.com.br/desenho/cg.pl	Sucesso	Mesmo protocolo, domínio e porta (80)
http://desenhoscriptivo.com.br	Sucesso	Mesmo protocolo, domínio e porta (80)
http://desenhoscriptivo.com.br:8080	Erro	Porta diferente
https://desenhoscriptivo.com.br	Erro	Protocolo diferente (https)
http://cg.desenhoscriptivo.com.br	Erro	Domínio diferente

Tabela 10 — Verificação da política de mesma origem contra <http://desenhoscriptivo.com.br>

Há diversas técnicas disponíveis para realizar a comunicação entre domínios e a coleta de dados de ambiente, como AJAX, CORS, JSONP, mensagens em iframes, *window.postMessage*, *web messaging*, *web socket* e *web workers*, *object*, CGI, etc. O estudo dessas técnicas foge ao escopo dessa dissertação e não serão aprofundadas. Apenas serão exemplificadas as técnicas AJAX — *Asynchronous JavaScript and XML* — para comunicação assíncrona cliente/servidor; e a CGI — *Common Gateway Interface* — que descreve um conjunto de protocolos para interação com o servidor. O esquema abaixo traça um paralelo entre estes dois métodos de comunicação de dados e as tecnologias de programação gráfica estudadas.

Cliente ⇔ Transporte ⇔ Servidor

AJAX ⇔ XML ⇔ CGI

JavaScript ⇔ SVG ⇔ Perl

O esquema aqui apresentado trabalha com uma lógica de desenho entre as instâncias do cliente e do servidor. A transferência de informações gráficas (coordenadas, traçados, cores, etc.) pode ser flexibilizada de acordo com o projeto em curso e armazenada textualmente. Há transferências de ambas as partes. Ressalta-se, por exemplo, que o desenho ou imagem processado no servidor pode obter um ganho de eficiência de até 50% de imagem original, ou seja, o desenho plotado na tela fica mais leve (*Figura 34*).

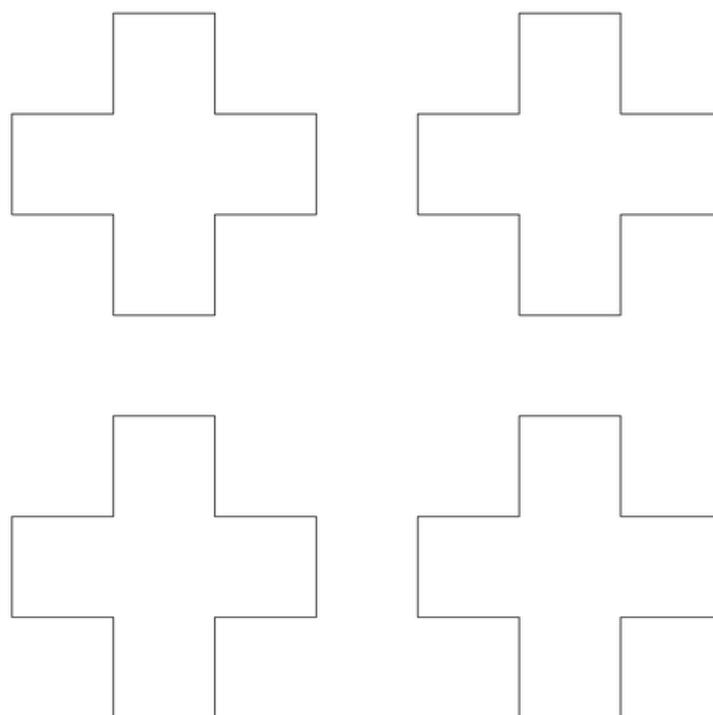


Figura 36 — Lógica distributiva presente em uma Curva “W” de 2ª Ordem.

3.6.1 AJAX

AJAX é uma técnica de carregamento assíncrono de dados (Figura 35 e 36). Não é uma tecnologia em si, mas um método de carregamento de dados em *JavaScript* que pode ser utilizado com diversas tecnologias, entre elas XHTML, CSS, DOM, XML, XSLT, etc. A técnica permite o tráfego eficiente de dados, com diminuição sensível no uso de banda, respostas mais rápidas e maior interatividade nas páginas, além de ser *open source*.

AJAX é a sigla em inglês para Asynchronous *JavaScript* and XML e trata-se de uma técnica de carregamento de conteúdos em uma página *web* com uso de *JavaScript* e XML, HTML, TXT, PHP, ASP, JSON ou qualquer linguagem de marcação ou programação capaz de ser recuperada de um servidor. (SILVA, Maurício Samy. *JavaScript: guia do programador* / Maurício Samy Silva. -- São Paulo : Novatec Editora, 2010).

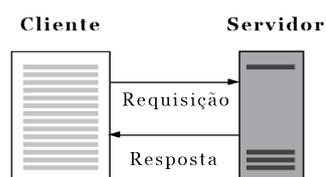


Figura 37 — Diagrama de uma requisição síncrona tradicional

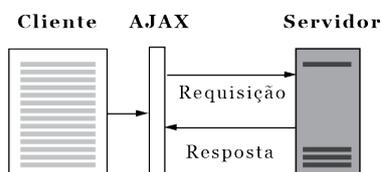


Figura 38 — Diagrama de uma requisição assíncrona com AJAX

XMLHttpRequest é o objeto responsável pelo funcionamento de AJAX e que utiliza o protocolo HTTP para transferir arquivos. HTTP são metadados descritivos sobre as requisições feitas pelo cliente ao servidor. A *Tabela 11* apresenta os métodos do objeto.

Métodos	Descrição
Open (método, URL, assinc, usuário, senha)	Destina-se a informar ao servidor o endereço do arquivo que está sendo requisitado pelo navegador. O argumento método pode ser GET, POST, CONNECT, DELETE, HEAD, OPTIONS POST, PUT, TRACE e TRACK. O endereço URL pode ser relativo ou absoluto. Os demais parâmetros são opcionais.
Send	Destina-se a iniciar uma requisição que já tenha sido definida pelo método <i>open</i> . Este método admite um parâmetro constituído do conjunto de dados a ser enviado ao servidor. Os dados a serem enviados podem ser codificados em diversos formatos, como <i>query string</i> , <i>DOM string</i> , etc.
setRequestHeader	Destina-se a enviar cabeçalhos HTTP para uma requisição AJAX.
getAllResponseHeaders	Destina-se a retornar um conjunto de cabeçalhos HTTP de uma requisição em forma de <i>string</i> .
getResponseHeader	Destina-se a retornar um determinado cabeçalho HTTP de uma requisição em forma de <i>string</i> .
abort	Cancela uma requisição em andamento.

Tabela 11 — Métodos do objeto *XMLHttpRequest* (baseado em SAMY, 2010).

3.6.2 CGI

A CGI, abreviação de *Common Gateway Interface*, é uma interface entre um servidor de internet e outro programa, geralmente um navegador *web* (Figura 37). Pode ser escrito em qualquer linguagem de programação e descreverá um conjunto de protocolos que irão interagir com o servidor e retornar uma saída, como documentos HTML ou imagens (ver Tabela 12). A principal vantagem desse método é a sua capacidade de criar páginas dinâmicas, tendo como base as informações de entrada do usuário diversas, como geolocalização, versão do sistema, dados inseridos via formulários, interação e escolhas particulares.

A Common Gateway Interface (CGI) descreve um conjunto de protocolos através dos quais as aplicações (*scripts* CGI) interagem com servidores *web* e (indiretamente) com clientes (ex. navegadores da *web*). Estes protocolos frequentemente são usados para gerar conteúdo da *web* dinâmico com base nos dados oferecidos pelo cliente. Uma página *web* é dinâmica se seu conteúdo muda sem ser alterado manualmente por uma pessoa... (DEITEL, 2002, p.224).

Variável de ambiente	Descrição
AUTH_TYPE	Método de autenticação usado para validar um usuário.
CONTENT_LENGTH	O comprimento dos dados (em <i>bytes</i> ou número de caracteres) passados ao programa CGI através da saída padrão.
CONTENT_TYPE	O tipo MIME dos dados enviados.
GATEWAY_INTERFACE	Versão CGI utilizada pelo servidor.
HTTP_ACCEPT	Lista dos tipos MIME que o cliente pode aceitar.
HTTP_USER_AGENT	Nome e versão do navegador utilizado pelo cliente.
QUERY_STRING	No caso de um FORM com o método GET, o conteúdo dos campos é atribuído aos pares do tipo nome=valor e então passado ao <i>script</i> .
REMOTE_ADDR	O endereço de IP do cliente.
REMOTE_HOST	Máquina que solicita a execução do <i>script</i> .

REMOTE_IDENT	O usuário fazendo a solicitação, a variável irá conter o nome do utilizador que está fazendo a requisição.
REMOTE_USER	<i>Login</i> do usuário, autenticado pelo servidor.
SCRIPT_NAME	Caminho e nome do <i>script</i> que está sendo executado (ex: /cgi/programa.cgi).
SERVER_NAME	Endereço IP ou nome da máquina do servidor.
SERVER_PORT	O número da porta pela qual o servidor atende às requisições.
SERVER_PROTOCOL	Versão do protocolo utilizado pelo servidor.
SERVER_SOFTWARE	Nome e versão do software do servidor que está respondendo os pedidos do cliente.

Tabela 12 — Variáveis de ambiente CGI (baseado em GUELICH, 2001 e <http://paginas.fe.up.pt/>).

A importância desse protocolo de comunicação para o desenho *scriptivo* pode ser visto de diferentes perspectivas, uma vez que é possível recolher informações passadas via entrada de usuário ou variáveis de ambiente, integrando os dados aos bancos de dados ou aplicações no lado servidor.

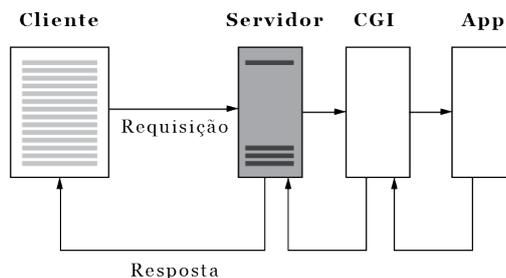


Figura 39 — Diagrama de uma seção CGI

Controlar as modificações de uma imagem distribuída exige o estabelecimento de regras explícitas, sendo o constrangimento geométrico um exemplo característico, mas importado de outras disciplinas. Essa técnica prevê o estabelecimento de regras e limitações aplicadas ao desenho que assegura a verificação das relações geométricas, de forma que a modificação de determinada parte do desenho conduza automaticamente a modificação de outros elementos. É um recurso ótimo para desenho *scriptivo*.

A *Figura 38* representa uma curva “W” de 2ª ordem, baseada em outro exemplo de Wirth. Pode ser lida em paralelo com a *Figura 34*. O movimento paramétrico definiu a fusão das 4 curvas “W” de 1ª ordem em um único polígono de 44 lados. Houve a transformação geométrica por aproximação. Essa forma também possibilita o trabalho de tratamento de imagens a partir da relação entre pixels. Para tanto, é necessário a criação de algoritmos que parametrizam as posições relativas das curvas “W” e depois trabalhar tratamentos diferenciados e especificados para cada uma dessas posições.

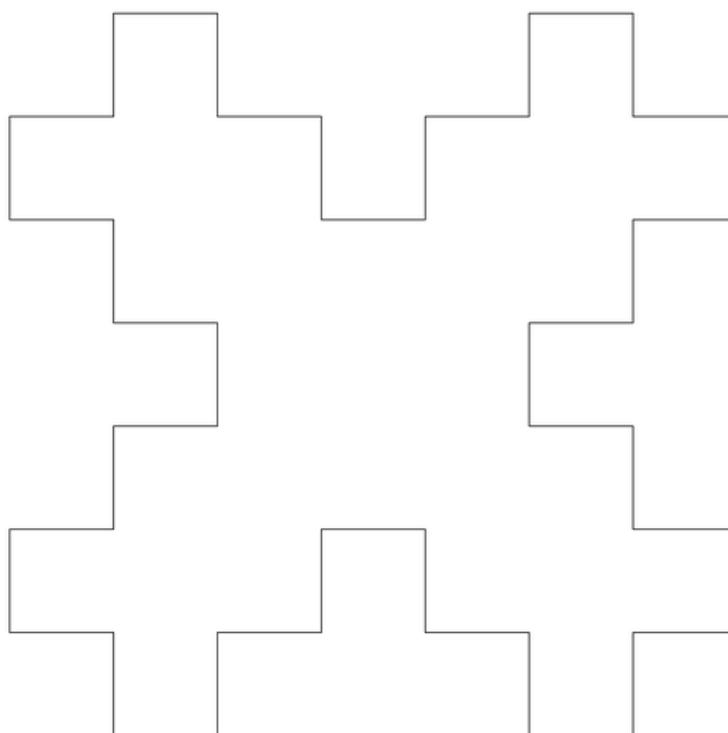


Figura 40 — Curva “W” de 2ª Ordem (WIRTH, 1999, p. 139). Método de trabalho com Pixel e reunião poligonal dos elementos da Figura 34.

3.6.3 Ambiente de hospedagem

As partes que compõem um desenho construído de forma distribuída devem trabalhar em cooperação, exigindo a troca de dados, adoção de protocolos de comunicação, variáveis e constantes de desenho. Em certo nível, não há nada de novo aqui. A relação cliente/servidor pode ser considerada uma ilustração de sistemas distribuídos, pois o

computador do cliente é especializado em obter e apresentar as páginas enquanto o computador do servidor é especializado em montar e disponibilizar as páginas na rede⁴.

A *Figura 39* mostra um ambiente de trabalho dentro de um servidor de hospedagem. A estrutura de pastas e arquivos é semelhante ao que encontramos em nossos computadores pessoais. Já o *Código 4* é uma marcação HTML que requisita 3 desenhos remotos, juntando todos na mesma página <http://www.desenhoscriptivo.com.br>. Os desenhos requisitados são os mesmos estudados neste capítulo: *JavaScript*, *SVG* e *Perl*. Os desenhos são requisitados das páginas <http://www.minaspadrãodesãojosé.com.br> e <http://www.programaçãográfica.com.br>. É um desenho construído a partir de diversas fontes, possibilitando a interação e troca de dados entre seus algoritmos.

```
<!-- Jesus Cristo_ O Caminho, A Verdade e A Vida -->
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Desenho Distribuído Web</title>
<link rel="stylesheet" type="text/css" href="estilos.css">
</head>

<body>
<?php // São José nós vos agradecemos.??>
<object id="perl" type="text/perl" data="http://xn--minaspadrodesojos-
gqbe8j.com.br/desenho5.pl">
<p>Conteúdo alternativo para navegadores que não suportam object</p>
</object>

<embed type="image/svg+xml" src="http://http://www.xn--programaogrifica-
xjb4a5e.com.br/bissetriz.svg"/>
<svg width="1600" height="1600">
<rect id="E" x="110" y="60" width="100" height="100" stroke="#000000" stroke-width="3"
fill="transparent"/>
</svg>
```

⁴ (<http://www.ime.usp.br/>).

```

<canvas id='ttt' onclick = "move(200 + a++); mud(200 + a++);" width='1600px'
height='1600px'>Seu navegador não suporta canvas HTML5</canvas>
<script defer src="http://desenhoscriptivo.com.br/formula2.js"></script>

<div><button type="button" id="botao-disparador">Inspecionar objeto Ajax</button></div>
<script type="text/javascript" src="ajaxtext.js"></script>

</body>
</html>

```

Código 4 — Código de um desenho distribuído hospedado em <http://www.desenhoscriptivo.com.br>.

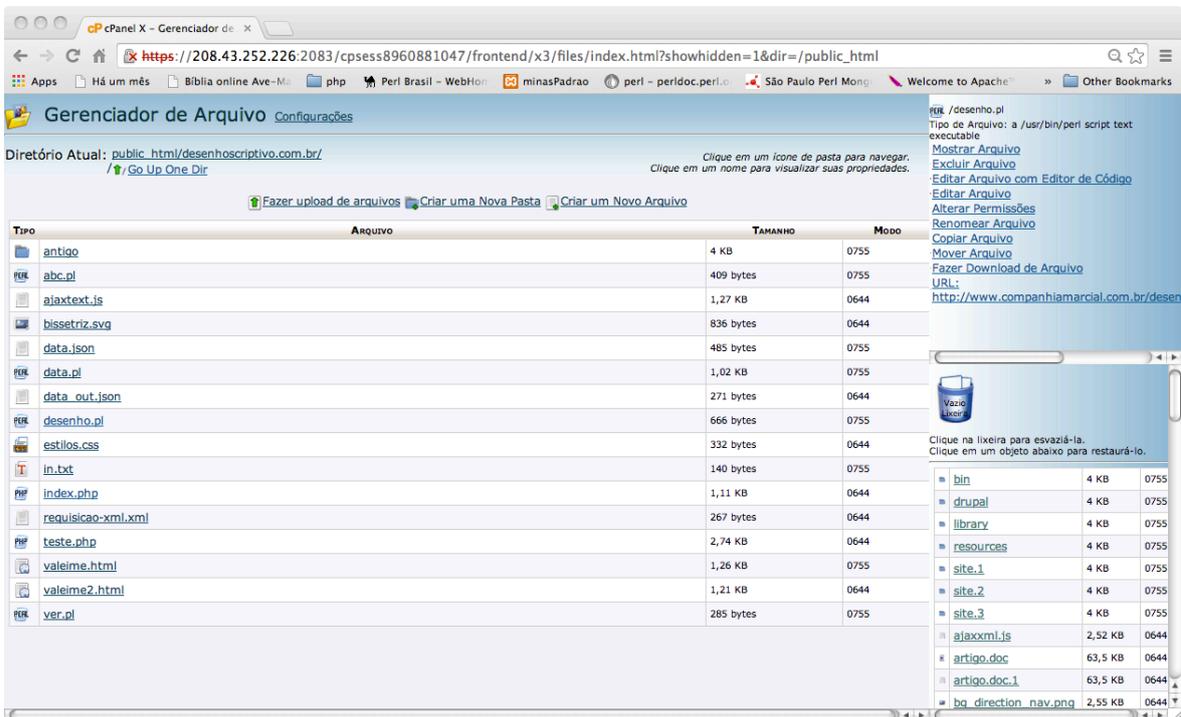


Figura 41 — Ambiente de trabalho em um servidor de hospedagem.

CAPÍTULO 4

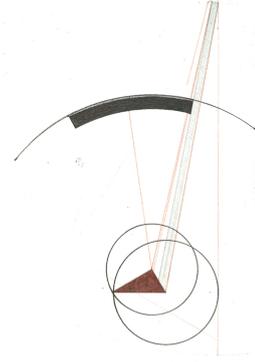
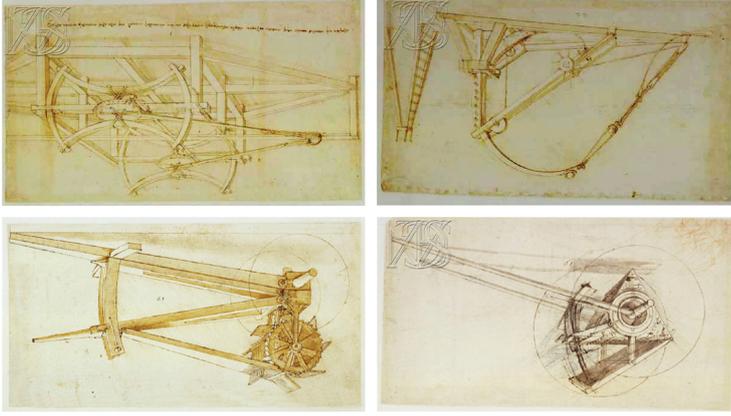
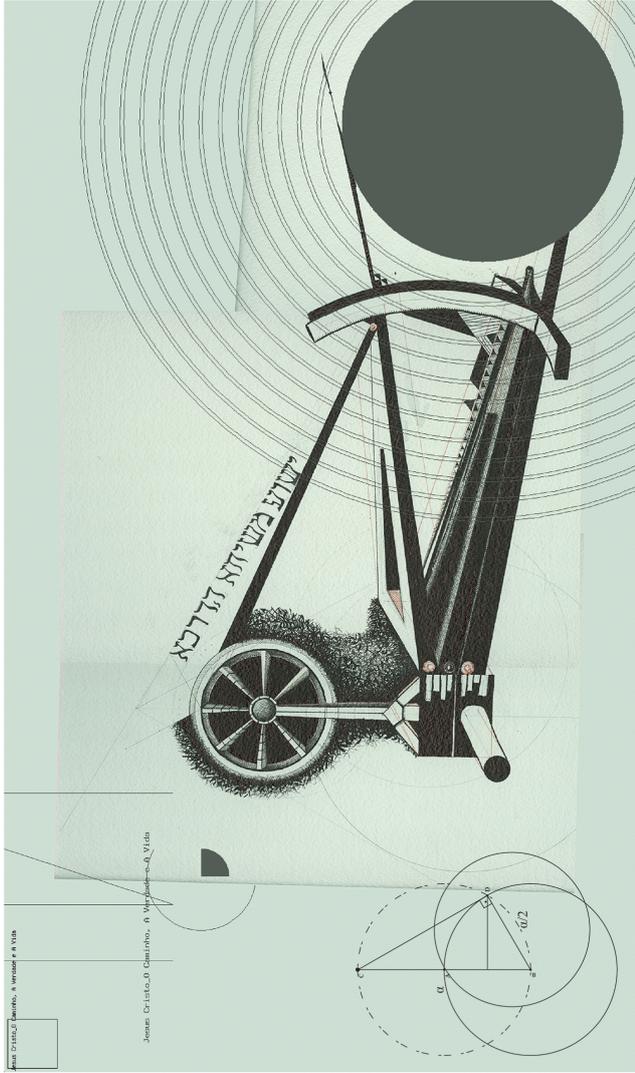
Processo de Trabalho

Depois de apresentar os esquemas conceituais para o desenho *scriptivo*, juntamente com os detalhes técnicos de implementação em linguagem de programação *script*, passo a demonstrar alguns trabalhos construídos a partir da concatenação entre texto e imagem. Os desenhos a seguir foram feitos utilizando os mais diversos recursos de linguagem, perfazendo um conjunto específico entre o desenho real e o desenho virtual, com elementos de geometria, texto, algoritmos, ambiente cliente/servidor, lápis, papel, digitalização, protocolos de comunicação, aplicações de imagem, vetorização, mapas de bits, régua, compasso, transferidor, pantógrafo, astrolábio, iteratividade, laços de repetição, regras, interação humano-máquina, fórmulas, etc. Trata-se de um conjunto dinâmico de desenho que dialoga com antigos tratados de desenho e outros códigos.

Os resultados aqui apresentados são aplicações de texto e imagem que trabalham a partir do estudo filosófico-geométrico feito no capítulo 2 bem como da pesquisa técnica realizada no capítulo 3. Podem ser lidos como uma série de desenho *scriptivo*, assim como é bastante usual a produção serializada dentro do campo das artes visuais. Os 10 desenhos *scriptivos* foram implementados e funcionam como padrão e base para trabalhos futuros.

A escolha desse material é justificada pela síntese conjuntiva que realiza. Os mecanismos utilizam elementos como número, título, desenho, descrição e fórmula que, concatenados, servem como prova e exemplo de transformação dinâmica entre texto e imagem. Adaptar esses mecanismos para *web* foi um primeiro passo para o desenvolvimento de outras formas de ação da arte dentro do espaço virtual. A seleção dos esquemas privilegiou os mecanismos que podem ser expandidos para outros trabalhos futuros, visto ser o ponto que melhor ressalta o desenho *scriptivo* aqui conceituado.

Máquina Lógica



A ciência moderna e as tecnologias contemporâneas possuem um aspecto higiênicé que elimina toda proximidade e vizinhança dos humanos e das coisas. Como resultado a fisionomia é encarcerada nos pressupostos do objeto universal, mentiroso e regrado. Ma o que somos sem nossas máquinas lógicas? Animais sem rodas. E o que podemos sem nossas regras semânticas? A caotologia! Outras fisionomias aparecerão por aqui e todo aparato de nossa lógica causará fascínio somente com o pressuposto ignorante das regras.

A nuvem é de fato uma máquina (sem rodas) que tem como regra a semântica nua. Nosso corpo será então abduzido pelas linguagens, num desejo descontrolado pela imagem, pelo virtual, onde nosso próprio caos

torna-se motivo de adivinhação mercadológica: universalidade fisionômica, ontologia pelo corpo móvel que adentrará nessa realidade ubíqua, fria e cínica. Nesse novo mundo, onde estará o olfato e o paladar? Dai, uma semântica que possui a pretensão de agir universalmente nada mais é do que um clichê. Ver Máquinas Lógicas de Leonardo Da Vinci à direita.

Noção, teoria, pressuposto, demonstração, declinação, etc. Serão entendidos como máquinas do pensamento e do discurso. Como? Bolhas significativas que operam pela combinatória, máquina (arco e flecha) da adivinhação "pré-colombiana", estruturas para arfete, para jogar objetos além da muralha, sem destruir a muralha, o muro da catapulta.

Perspectiva



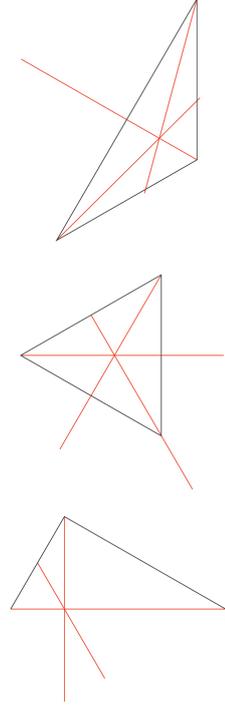
a) LISTA

- 0-Coração
- 1-Apoio
- 2-Intensificador

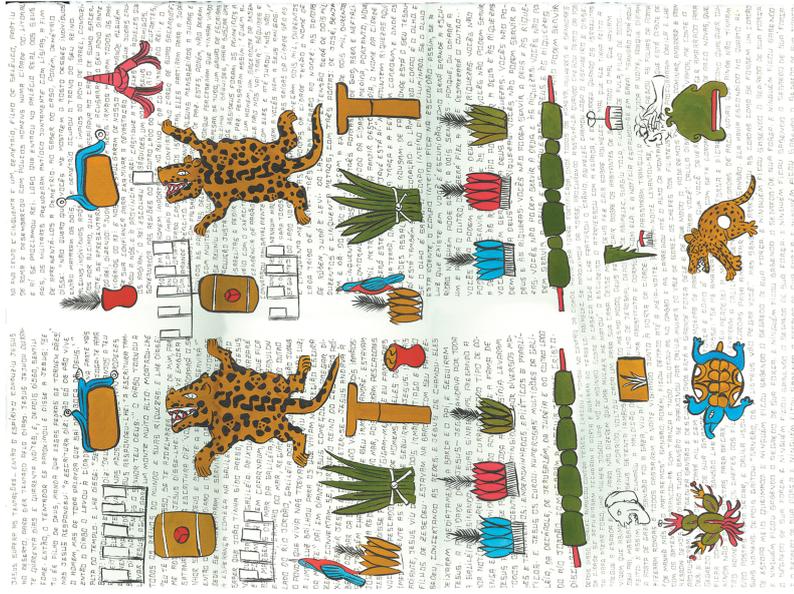
b) ÁRVORE

- 1-Coração
- /
- 0-Apoio
- 2-Intensificador

Uma estrutura de árvore pode ser definida como 1 - A estrutura vazia e 2 - Subárvores. No exemplo b) ÁRVORE, a enumeração segue o método de varredura e-r-d (= in-order traversal) tomando evidente um início básico de diferenciação computacional entre uma Sequência de a) LISTA (0-Coração, 1-Apoio, 2-Intensificador) de uma Estrutura Arbórea (0-Apoio, 1-Coração, 2-Intensificador). Em termos metalinguísticos, a própria existência destas 2 (duas) alternativas de leitura é suficiente para comprovar a naturalidade da árvore como estrutura de classificação. Em uma árvore podem ser feitas operações de busca, inserção e remoção. Extrato de perspectiva.



Escrita



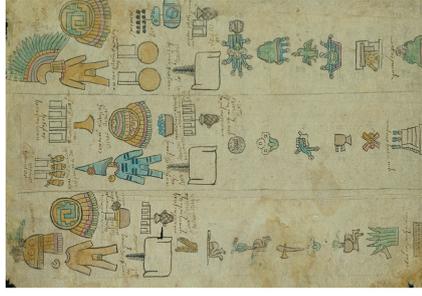
A Matricula de Tributos é uma antiga forma de registro praticada na América pré-colombiana que contabilizava os impostos que as cidades súditas (altepetl) pagavam ao México-Tenochtitlan. A leitura dessa matricula é feita da seguinte forma: "quando a margem esquerda está cheia, a margem direita segue e finalmente, a fileira superior". Esse tipo de leitura demonstra a coalescência entre o sistema de escrita e aritmética Nahuatl e a configuração visual de páginas encontradas na Internet. Perfaz um movimento da "coisa" concreta para a "coisa" enumerada, ou seja, a matricula racionaliza o mundo no papel, denomina e estrutura dados da realidade.



Estrutura para leitura da Matricula Ameríndia

a	b	1	0
1	0	2	3
2	3	4	6
4	6	x	9
7	9	0	0
8	3	7	3
1	6		

Completa Principal Segue

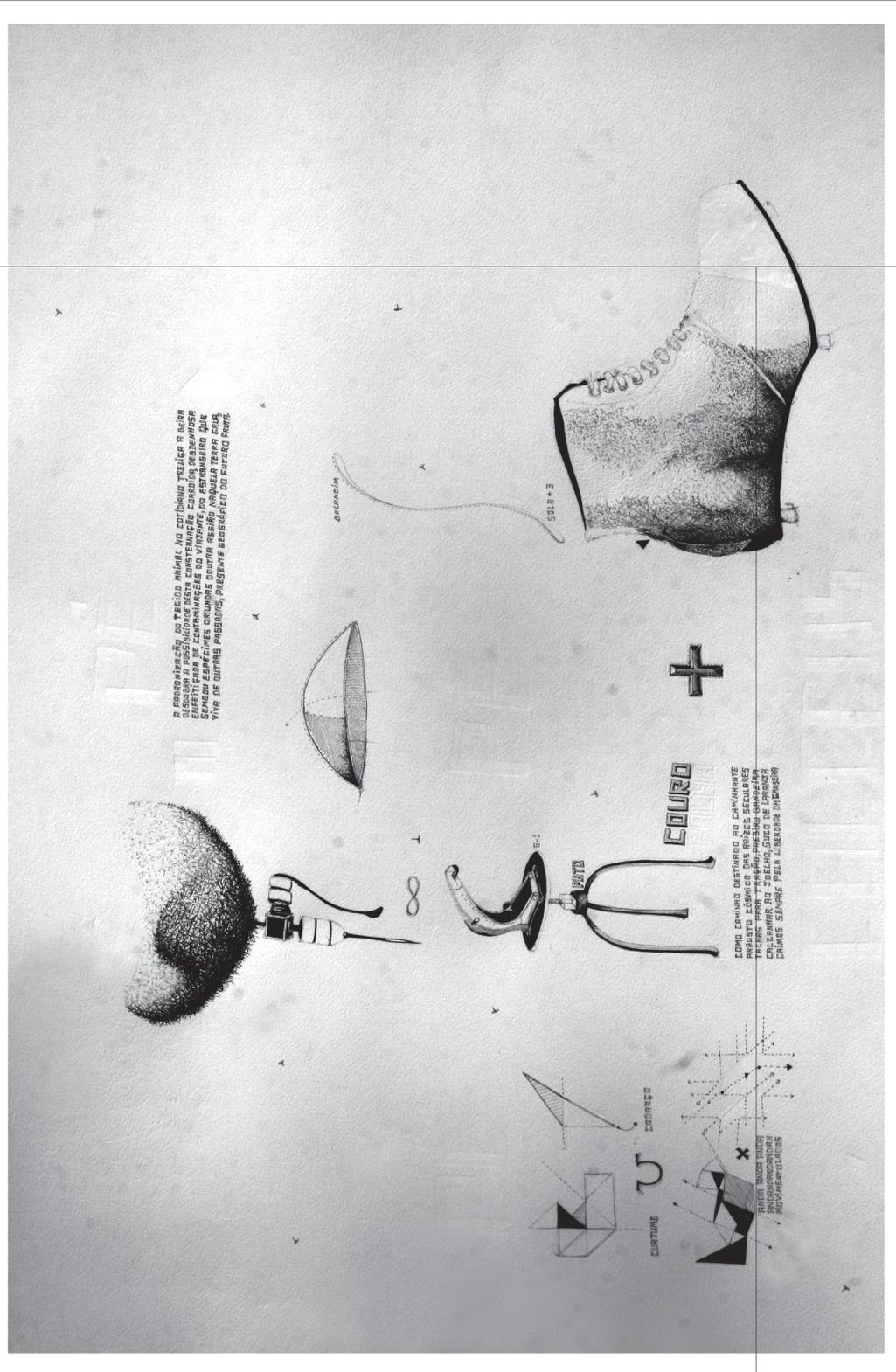


Matricula de Tributos (rolo de tributos).
México-Tenochtitlan, 1522-30 d.C.

Descrição: A Matricula de Tributos (Rolo de tributos) registra, em escrita pictográfica, os impostos que as cidades súditas (altepetl) pagavam para o México-Tenochtitlan. Cada página da Matricula representa uma das cidade de cada província aparece primeiro, na parte inferior à esquerda da página, seguindo em direção à parte superior pelas altepetl. Cada altepetl é representada por seu glifo topônimoico. Quando a margem esquerda está cheia, a margem direita segue e, finalmente, a fileira superior. No interior desta moldura de glifos topônimoicos, os itens pagos pela região são representados visualmente, em fileiras, e a quantidade esperada de cada item é registrada, usando o sistema Nahuatl de aritmética.

(Fonte: Biblioteca Virtual Mundial)

Calceus

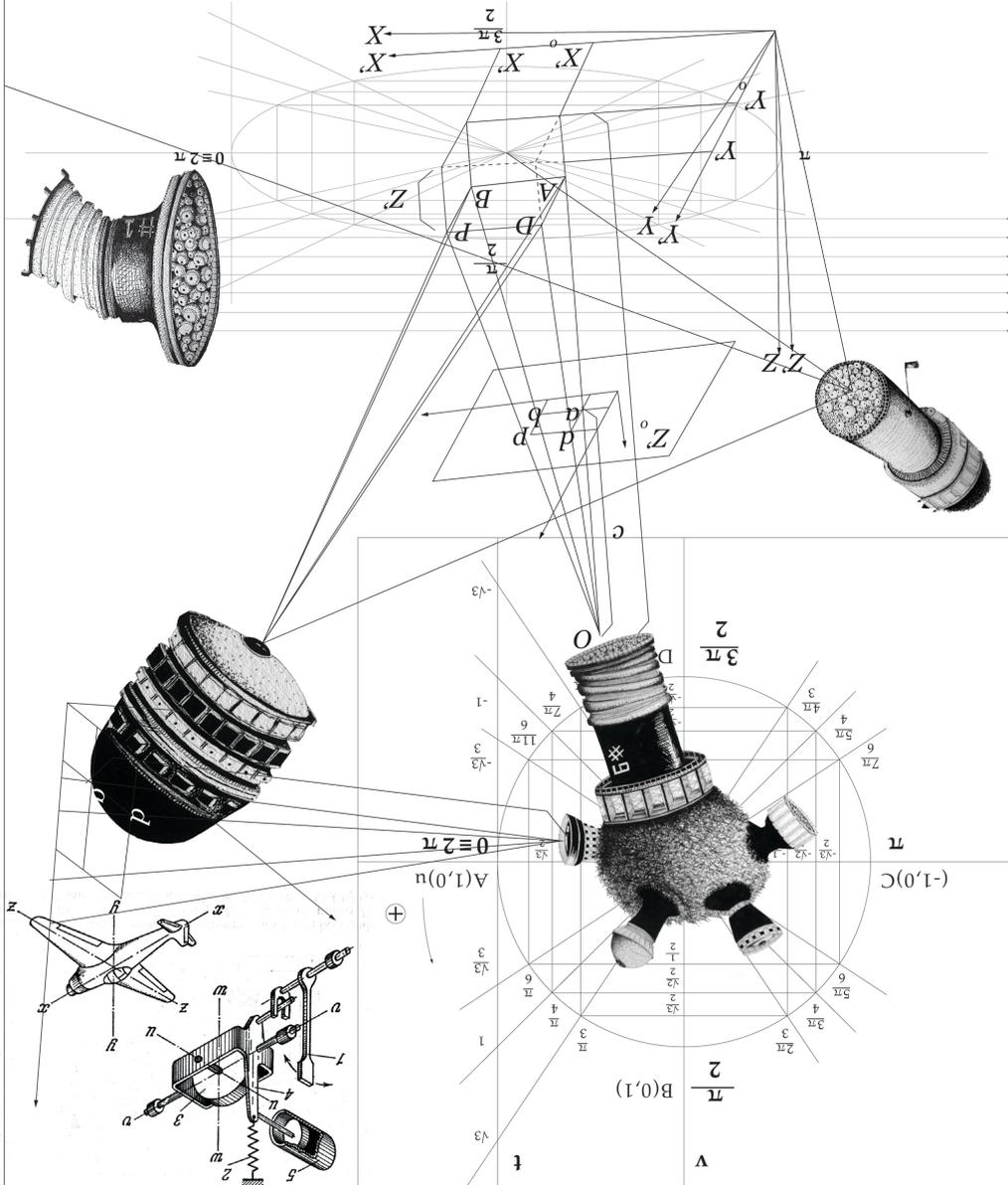


Óptica

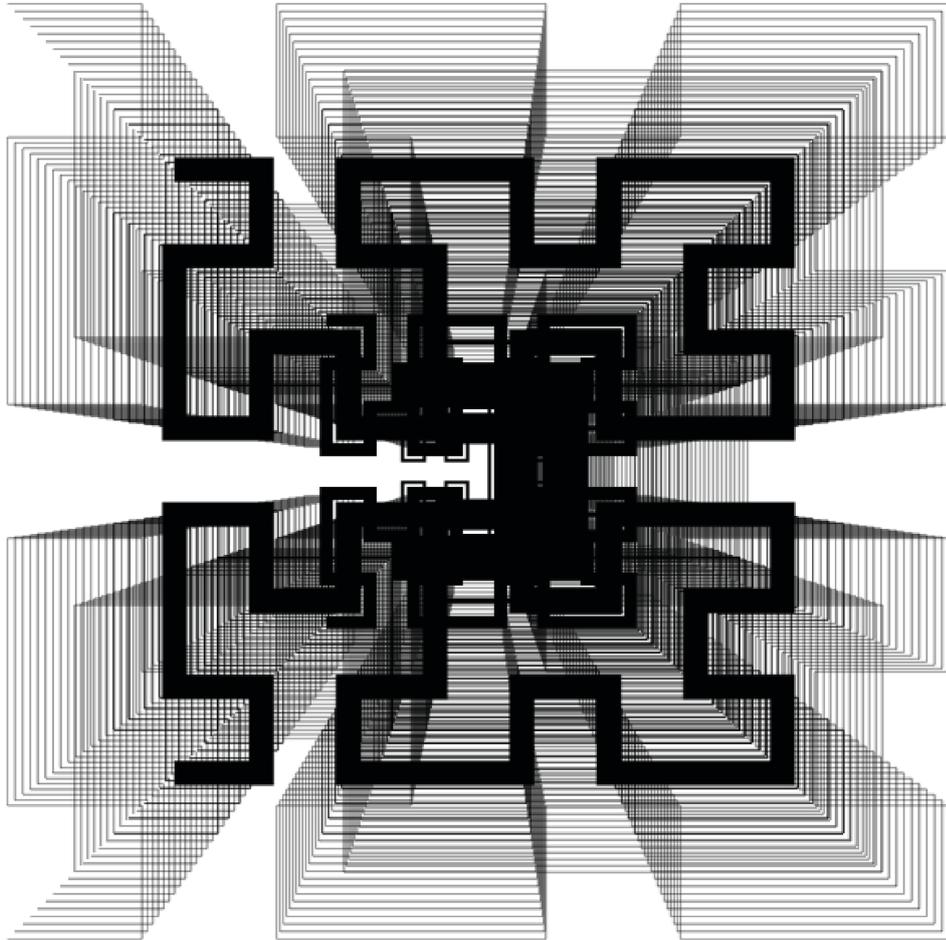


Quando desta necessidade de entendimento e apreensão do que se move, desta circulação hipnó, configurada na hipotética do princípio fogo e sua investida no plásmar ou, o desejo de fazer estático o impossível; eis a imagem. A imagem que produz a ideia, longe de ser organizativa, é repetição e mimese, criação feita com base em uma imagem também criada. Eidolon: do grego *εἶδωλον*: 'imagem, ídolo, duplo, aparição, fantasma'. O termo exato para designar este fenômeno, muito mais que o conhecido 'virtual', seria o conceito de abdução, ou seja, o processo de formação de ideias explicativas. O que o nosso universo Pindorama conheceu há 500 anos atrás, quando os invasores europeus colonizaram estas terras e abduziram o indígena pela razão, voltou a reproduzir-se, nesta novíssima maneira de virtualizar e abduzir, ou seja, a lógica materializada na internet, a transformação da óptica em lógica, do elemento fático em dispositivo explícito.

Alguns elementos baseados em BRITO & COELHO e em ARTOBOLLESKI.



Babel



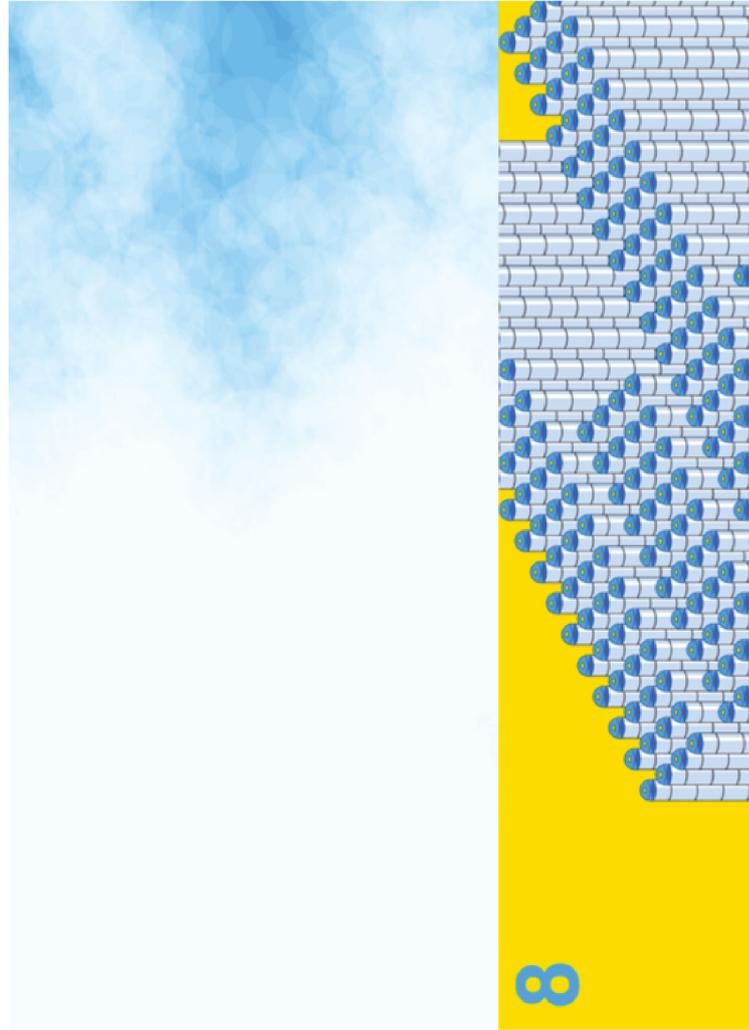
1 Toda a terra tinha uma só língua, e servia-se das mesmas palavras. 2 Alguns homens, partindo para o oriente, encontraram na terra de Senaar uma planície onde se estabeleceram. 3 E disseram uns aos outros: "Vamos, façamos tijolos e cozamo-los no fogo." Serviram-se de tijolos em vez de pedras, e de betume em lugar de argamassa. 4 Depois disseram: "Vamos, façamos para nós uma cidade e uma torre cujo cimo atinja os céus. Tornemos assim célebre o nosso nome, para que não sejamos dispersos pela face de toda a terra." 5 Mas o senhor desceu para ver a cidade e a torre que construíram os filhos dos homens. 6 "Eis que são um só povo, disse ele, e falam uma só língua: se começarem assim, nada futuramente os impedirá de executarem todos os seus empreendimentos. 7 Vamos: desçamos para lhes confundir a linguagem, de sorte que já não se compreendam um ao outro." 8 Foi dali que o Senhor os dispersou daquele lugar pela face de toda a terra, e cessaram a construção da cidade. 9 Por isso deram-lhe o nome de Babel, porque ali o Senhor confundiu a linguagem de todos os habitantes da terra, e dali os dispersou sobre a face de toda a terra.

(BÍBLIA AVE MARIA, GÊNESE, capítulo 11).

11 Torre de Babel — 1 Todo o mundo se servia de uma mesma língua e das mesmas palavras. 2 Como os homens emigrassem para o oriente, encontraram um vale na terra de Senaar e aí se estabeleceram. 3 Disseram um ao outro: "Vinde! Façamos tijolos e cozamo-los ao fogo!" O tijolo lhes serviu de pedra e o betume de argamassa. 4 Disseram: "Vinde! Construamos uma cidade e uma torre cujo ápice penetre nos céus! Façamo-nos um nome e não sejamos dispersos sobre toda a terra!" 5 Ora, Iahweh desceu para ver a cidade e a torre que os homens tinham construído. 6 E Iahweh disse: "Eis que todos constituem um só povo e falam uma só língua. Isso é o começo de suas iniciativas! Agora, nenhum desígnio será irrealizável para eles. 7 Vinde! Desçamos! Confundamos a sua linguagem para que não mais se entendam uns aos outros." 8 Iahweh os dispersou dali por toda a face da terra, e eles cessaram de construir a cidade. 9 Deu-se-lhe por isso o nome de Babel, pois foi lá que Iahweh confundiu a linguagem de todos os habitantes da terra e foi lá que ele os dispersou sobre toda a face da terra.

(BÍBLIA JERUSALÉM, GÊNESE, capítulo 11).

Nuvem

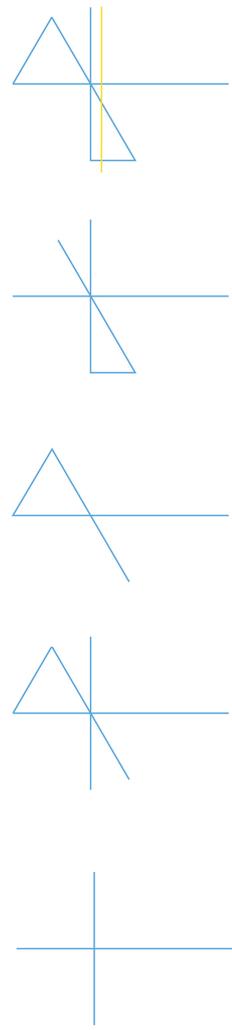


O mecanismo que cria a imagem ao lado foi estruturado a partir do processo de formação de uma nuvem natural. Trata-se de um mecanismo que concatena uma lógica de transformação em regras trigonométricas e desenho.

Duarte 2012, descreve minuciosamente sobre a formação de uma nuvem no céu, destacando que, primeiramente, é necessário que o ar esteja supersaturado, ou seja, "é preciso que a umidade relativa ultrapasse os 100% (>=) e a pressão de vapor e es é a pressão de vapor saturado)". Destaca ainda que, uma gotícula de água é formada "quando uma parcela de ar ascende na atmosfera, se expande e se resfria adiabaticamente, esta atinge a saturação e o ar não comportará mais água em estado vapor" (ídem). Esse processo é chamado de nucleação.

Caso a gotícula se constitua apenas por água pura, o processo é chamado de nucleação homogênea. Na prática, a condensação homogênea é impossível de acontecer na atmosfera terrestre, uma vez que as gotículas de água pura não se aproximam do raio crítico necessário de 1 µm e evaporam.

A formação de uma Nuvem somente se torna possível pelo processo de nucleação heterogênea. Neste processo é necessário um outro elemento, o aerossol atmosférico, um termo criado por Schummauss para designar gotículas ou sólidos suspensos na atmosfera, que possuem dimensões inferiores a 100 µm por partícula. As nuvens dependem das partículas de aerossol para se formar, de maneira que, Nuvem e aerossol são entidades inseparáveis. "Se tais partículas são hidrosscópicas", ou seja, que absorvem água, "elas podem ser referidas como núcleos de condensação, e as moléculas de vapor d'água se condensarão sobre estes núcleos, aproveitando-se de seu tamanho natural." (DUARTE 2012, p.7). Dessa forma, as gotículas de água podem ultrapassar o raio crítico de 1 µm e não evaporar. Neste momento, a gotícula se torna ativa.



Pietà

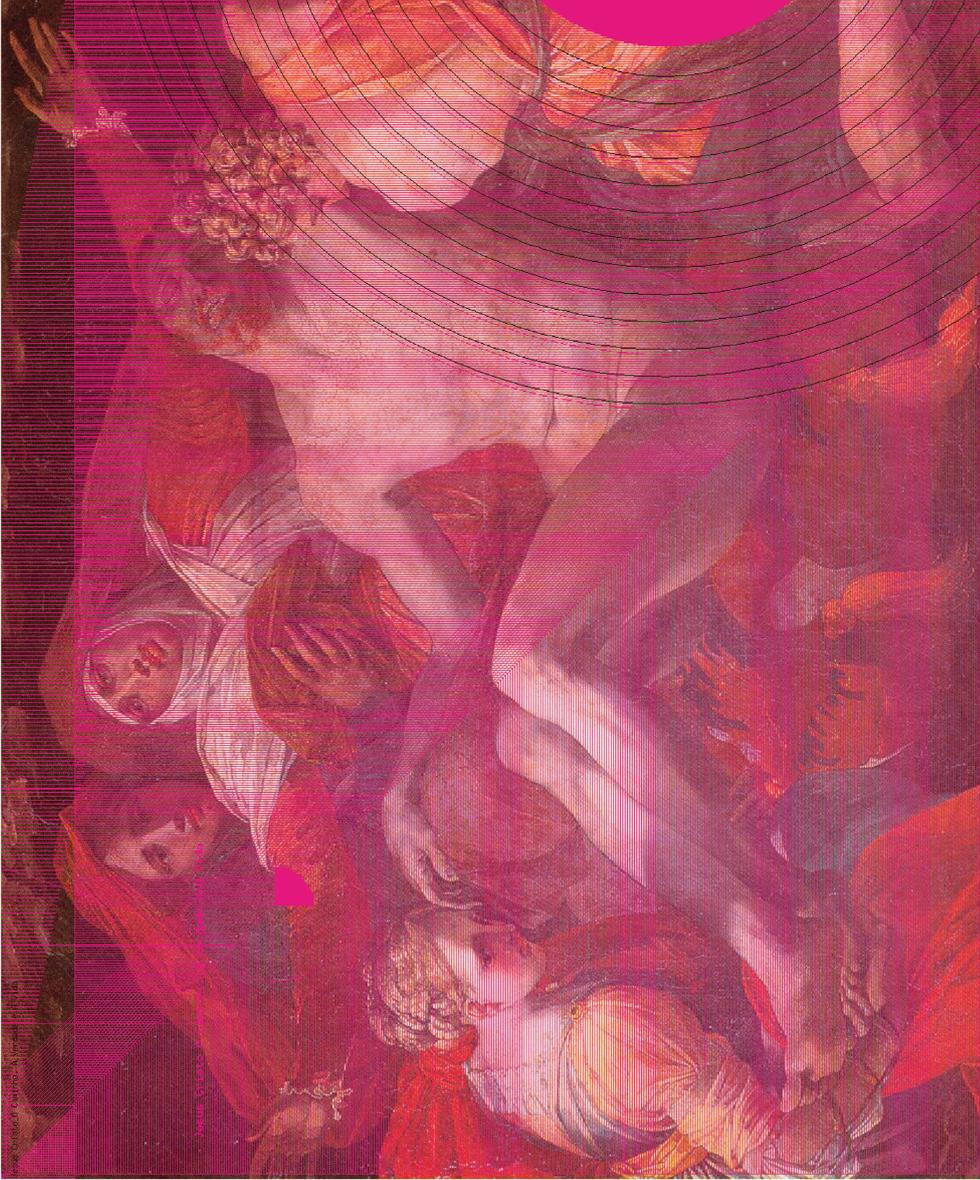


Imagem (otimizada em 54% do seu tamanho original) gerada no servidor minuspadrãoedotópicos.com.br a partir da obra *Pietà*, 1530-1540, de Rosso Fiorentino.

O primeiro, que tome cuidado com as palavras dos filósofos antigos e modernos que falam sobre essa ciência, porque a arte da alquimia tem seu assento e fundamento na capacidade do entendimento e da demonstração da experiência. Os filósofos, querendo encobrir a verdade da ciência, falaram quase todas as coisas em linguagem figurativa.

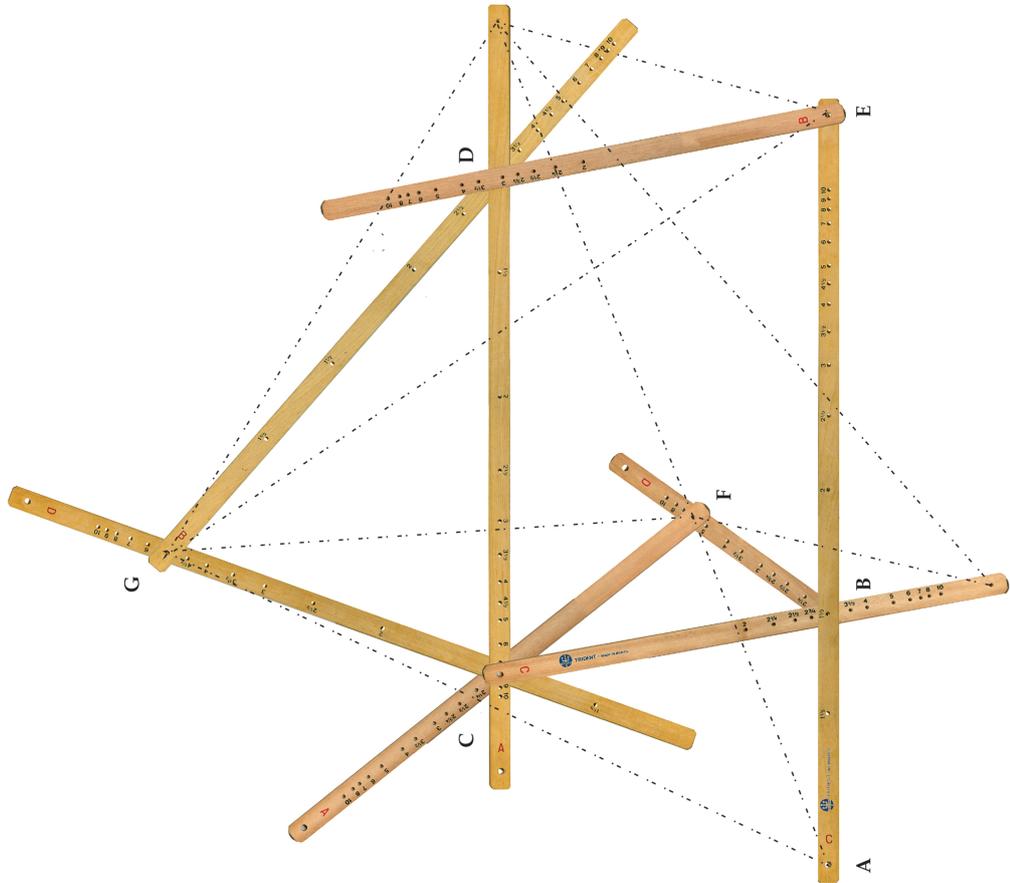
O segundo: que não aprecie muito as coisas, nem as composições de variadas espécies, porque a natureza nunca produz senão sua semelhança: porque assim como o barroto é uma produção imperfeita a partir do cruzamento do cavalo com a jumenta, alguns imitadores da ciência produzem uma multiplicidade de coisas estranhas à ela.

O terceiro, que não seja falador, nem acadêmico, mas antes ponha um guarda em sua boca assim como fizeram os sábios. Não jogue pérolas aos porcos. Esteja em paz com Deus e tenha nele o fim e a ordem do seu trabalho, sem nunca recuar perante o cansaço e as dificuldades.

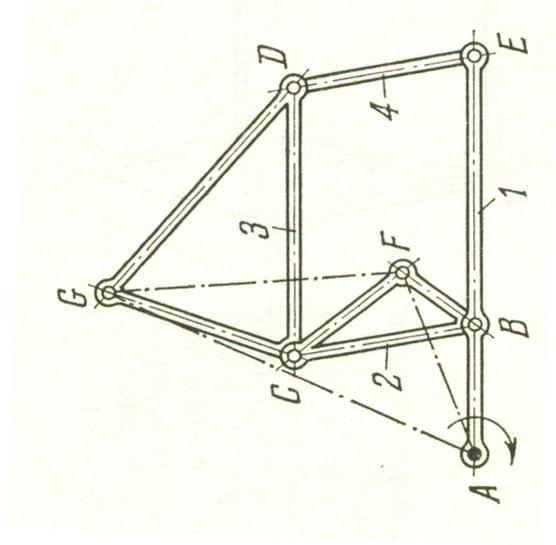
Assim, se você tiver diante dos olhos essas regras, que me foi passada por Alberto Magno, não terá a necessidade de buscar favor de reis ou poderosos, mas antes eles te darão toda honra. Porque todo aquele que é reconhecido nessa arte servirá aos reis e aos poderosos, assim como a todos os necessitados. E quem receber essa graça jamais deve oferecê-la com algum interesse.

(AQUINO, 1268)

Pantógrafo



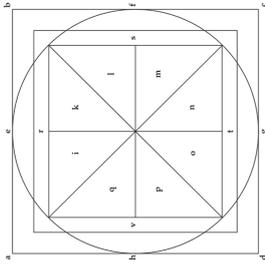
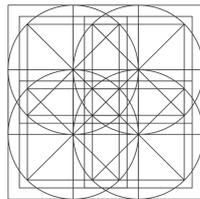
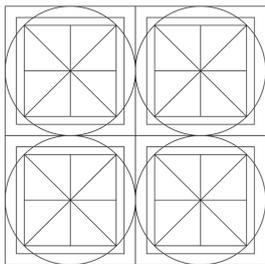
MECANISMO DE ALAVANCAS ARTICULADAS DE UM PANTÓGRAFO.



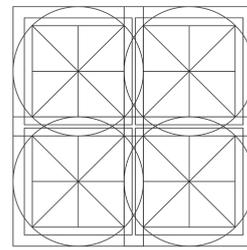
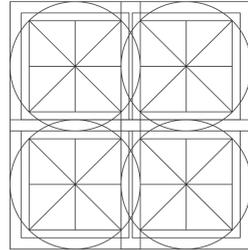
Os comprimentos dos elementos do mecanismo satisfazem as seguintes condições: $BC = DE$ e $CD = BE$, ou seja, a figura $BCDE$ é um paralelogramo. O triângulo rígido FBC é semelhante ao triângulo FAG . Independentemente da configuração do paralelogramo $BCDE$ os ângulos nos vértices FAG são constantes. Quando o elemento 1 gira ao redor do ponto fixo A , selecionado como centro de semelhança, e um dos pontos F ou G segue uma trajetória arbitrária, o outro ponto descreverá uma trajetória similar virada a um ângulo constante. O mecanismo possui a propriedade da reversibilidade, ou seja, qualquer um dos pontos A , G ou F pode ser selecionado como centro de semelhança.

(ARTOBOLSKY, mecanismo 837, 1976).

Igreja



The Master Figure, Raymond Lull.



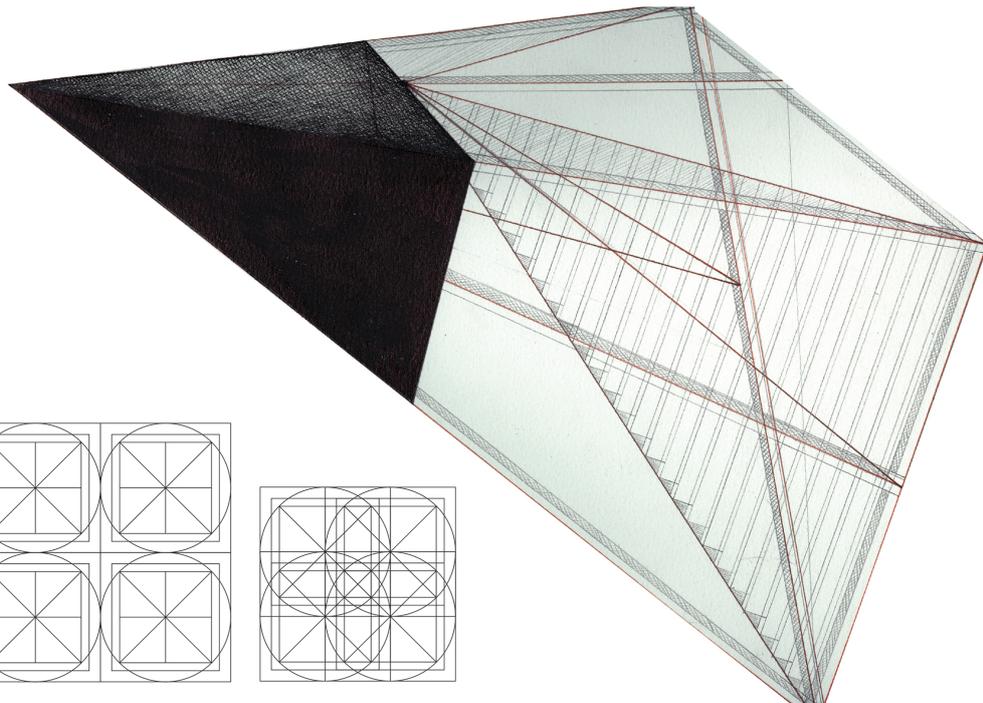
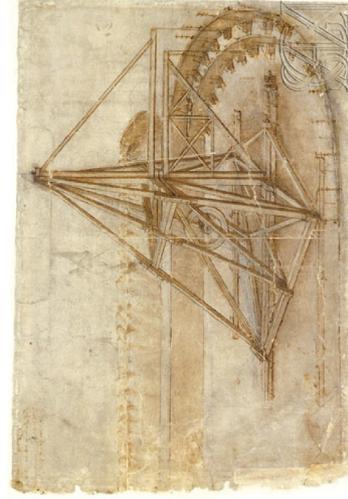
O desenho apresentado é baseado em um estudo de DA VINCI (1478-1519) e representa uma estrutura de trabalho no espaço físico. Trata-se de uma ferramenta de movimentação e armazenagem de materiais. A forma geométrica é a figura mestre descrita por LULL (1305).

Em um paralelo, a abdução trazida pela internet é tão violenta como uma ocupação territorial clássica, onde a imposição de algo estranho vai aos poucos enraizando no cotidiano do país, nas práticas setoriais e nas estruturas do pensamento. A própria ocupação pressupõe um movimento de cargas e hoje um movimento semântico da informação.

Plano: espaço dimensional estático dependente de um suporte. 3D: espaço dimensional imersivo também dependente de um suporte. Espaço físico: o espaço habitado, exterior, natural, o aqui. Qualquer objeto que venha habitar esse espaço pode ou não provir de um suporte abstrato.

Para gerar qualquer informação sobre esse sistema, primeiro é necessário a enumeração dos seus elementos: quantidade, identificação, vias, mapas, terreno, etc.

A quantidade de dados brutos desse sistema somente poderá ser lida com o auxílio de operações algorítmicas e automatizações modulares. Do contrário, esses dados seriam irrelevantes.



Tal como a construção com gravetos, que são depurados no plano e retornam para o espaço físico como ferramentas, estes exemplos poderão também retornar para o plano como objetos físicos ou mesmo outras ferramentas. A evolução do desenho *scriptivo* e dessas novas ferramentas são aspectos a serem tratados em outras ocasiões, em um contínuo esforço de compreender e quantificar o espaço pelo plano. De fato, consideramos o desenho como base para essas destinações.

Assim, o processo de trabalho observou a orientação feita pelo professor Sílvio Zamboni (2001), na obra *A pesquisa em Arte, um paralelo entre arte e ciência*, em que “processo de trabalho é uma fase da pesquisa na qual, por meio de ações sistemáticas, procura-se chegar à materialização de uma obra embasada pelas ideias e interpretações da observação.”

Texto e imagem, são como castelos de areia construídos pela linguagem. Assim colocados, em essência ou simplicidade, fazem pouco movimento e quase nada podem dizer. Relacioná-los então, trazê-los para o papel, soa como mero exercício acadêmico e puro malabarismo teórico. O texto se transforma na imagem, a imagem transmuta-se para o texto. A imagem, seja ela proposição ou visualidade, torna-se recurso para a memória, algo que fixa um significado, um significante e resvala na realidade. Quando distinguimos algo deste emaranhado natural, no pensamento ou no exterior, imediatamente operamos a plástica, do grego *plastiké*, arte de plasmar.

A imagem pode ser operacionalizada. Parece óbvio mas, estranhamente, está sendo mantido fora do entendimento comum. A imagem digital, juntamente com os meios de comunicação atuais, são mitificados e colocados como que surgidos pela magia. Não há interesse em operar uma pedagogia da imagem ou para a imagem, preferindo a massa de manobra em detrimento dos construtores e do povo livre. Entretanto, não há qualquer novidade nesse tipo de discurso.

O desenho é uma ferramenta de trabalho e planejamento capaz de expansão ilimitada. Com as tecnologias digitais e os ambientes remotos, essa forma de estruturar o pensamento ganhou fôlego e uma condição privilegiada. Porém é fato o processo de omissão nas redes atuais, não somente com o desenho, mas em toda a estrutura de comunicação. O desenho está presente nessas interfaces, o código foi escrito e está sendo executado, mas a interação com o sistema está focada no instantâneo e ainda há pouco espaço para operar a plástica.

Mas, o que acontece com um suporte de alto nível — como é o caso do ecrã — quando utilizado repetidamente e em larga escala? — Ele se desenvolve. O entretenimento não desaparecerá da internet, mas espera-se cada vez mais uma operação metalinguística nesse suporte. Com esse sentido, um desenho voltado para o código e sua escritura são aqui vistos como uma boa prática para o desenvolvimento da disciplina nos meios tecnológicos. Assim como a geometria, o desenho deve passar para as coisas reais, operar no mundo físico e ser aplicado em ferramentas práticas para o dia-a-dia, concordar ou não com a realidade: riscar o suporte.

Em pouco mais de 20 anos de acesso massivo à internet, o brasileiro presenciou o surgimento e o desaparecimento de dezenas de aplicativos e plataformas de larga escala. Os processos de transformação operados por essas experiências são difíceis de medir, mas restou algum aprendizado? Houve avanços na estrutura do fenômeno? Retrocedemos? A prática política, o nível técnico, as relações sociais, a geografia, os objetos são melhor compreendidos? Há espaço para uma outra internet que não esteja pautada na guerra dos discursos? É necessário uma ferramenta ou uma mídia?

A teoria permite a ampliação do campo de investigação, inclusive com a retomada de elementos fundamentais para o risco sobre o suporte digital, como é o caso das compreensões sobre mecânica, trigonometria, ponto, linha, pixel e curvas. Aliado a esses fundamentos, considera-se um avanço a inserção de objetos que não fazem parte do escopo da arte e nem do desenho, como é o caso das curvas de Hilbert, curvas “W”, mecanismos de alavancas, geometria sem distância e comunicações entre origens. A consolidação desse modelo só foi possível devido as intervenções contundentes da banca de qualificação.

Em um nível educacional, as bases da tecnologia estão muito próximas das bases da arte. Entre a matemática e a programação gráfica podemos exemplificar a inversão dos quadrantes do plano coordenado. Esses objetos foram aplicados com sucesso em sala de aula, com alunos de 9º ano ao Ensino Médio, no ano de 2014. É notável o interesse dessa faixa etária por questões de ordem tecnológica. Outro fato relevante é que, atualmente, a arte é a disciplina mais apta a inserir a tecnologia na escola, retomando seu papel de vanguarda e na atual necessidade de desmistificação da imagem.

CONCLUSÃO

Este trabalho oferece três contribuições principais para o desenvolvimento e a construção do Desenho na sociedade atual e a implementação de aplicações livres de desenho em ambientes cliente/servidor:

(i) A utilização de base filosófica e geométrica para o pensamento do risco sobre o plano, com o objetivo de transformar a idéia em um objeto comunicativo que possa ser compartilhado. A formulação de algoritmos autônomos presentes em 7 padrões iniciais de desenho, sendo mesmo uma forma de pensar as predefinições de uma arte tecnológica fundamentada no desenho.

(ii) Um desenho distribuído escrito em linguagem de programação *script* e em *software* livre, trabalhado em um ambiente de rede que pode ser acessado de qualquer terminal com acesso à internet. Trata-se de um simples texto que, quando codificado e inserido em um ambiente adequado, se transforma em um desenho dinâmico provindo de diversas fontes.

(iii) 10 exemplos de trabalhos que concatenam escrita e geometria. Visa as possibilidades de transformações entre o desenho analógico e o desenho digital, ou mesmo o objeto físico em concatenação com o objeto virtual.

A abordagem enuncia o desenho como uma técnica importante para o pensamento humano, uma ferramenta que compreende a transformação do espaço pelo plano conforme SANTOS (2007) e DHOMBRES (1996). Na realidade, como já foi dito, a arte e a tecnologia compartilham de um mesmo pressuposto, que é a conciliação entre a teoria e a prática. A implementação de um aplicativo pressupõe a complementação entre essas duas realidades, de forma a comunicar o humano com a máquina de forma estruturada e eficiente. Esses elementos, quando sistematizados, produzem a materialização de um objeto ou mesmo um método, e estes acabam por atender aos mais diversos escopos de produtos e tecnologias. Podem mesmo serem trabalhados em uma educação tecnológica eficiente.

Tanto a arte como a tecnologia, enquanto processos de implementação e ferramentas do conhecimento humano, guardam semelhanças estreitas. Tanto o desenvolvimento de uma quanto da outra necessita de um estudo técnico preciso e incontestável. Não há a possibilidade de se fazer arte nem tecnologia sem uma implementação precisa da teoria. Claro, aqui mencionamos uma implementação ótima. Os estudos realizados ao longo dessa dissertação trouxeram diversos problemas quanto ao real impacto dos problemas ligados à concatenação do texto com a imagem. Sem dúvida, nesse entremeio podemos encontrar diversas situações de complexidade elevada. É um trabalho que exige engenho e arte.

Um dos aspectos que merecem estudos futuros é a correlação da geometria sem distância de WITTGENSTEIN (1968) e sua implementação em dispositivos da chamada internet das coisas. Trata-se de uma visão particular que transforma os objetos colocando-os em contato direto com a estrutura lógica da imagem. A distinção interior/exterior é específica para um pensamento sobre a internet, particularmente na tendência de acesso remoto aos objetos existentes e aqueles que estão sendo criados.

A principal tarefa empreendida por esse estudo foi a utilização de ferramentas de desenho a partir de um texto. Considera-se essas ferramentas como materiais em arte, pois houve a necessidade de “montar” o quebra-cabeças e torná-las realmente algo utilizável. Depois de pesquisar diversas ferramentas disponíveis optou-se pelas ferramentas livres e destinadas ao texto. O guia foi o pressuposto básico de qualquer desenho: o papel em branco. A partir dessa filosofia foi feita uma seleção que contemplou os ambientes de cliente e servidor, bem como as ferramentas que poderiam fazer o intercâmbio de dados entre as suas diversas aplicações. O objetivo foi alcançado e está detalhado no capítulo 3.

Outro ponto relevante é o aspecto transversal e didático do texto, pois sendo um assunto de relativa complexidade, foi necessário trabalhar com o conceito de objeto e a flexibilidade entre seus procedimentos. É necessário observar que esse estudo foi uma tentativa de concatenar o desenho, a tecnologia e o texto, portanto houve um elevado esforço para torná-los perceptíveis dentro do campo da arte em paralelo com o campo da técnica. Nesse sentido, observa-se

que a técnica foi sim “questionada” pela teoria, resultando na tecnologia, ou o saber sobre a técnica. Particularmente, foi estabelecido que, ao final desse trabalho, os resultados poderiam oferecer tanto um conhecimento teórico quanto um conhecimento técnico sobre o assunto, fato que aconteceu naturalmente e com objetividade. A prova disso é a possibilidade de se desenhar em qualquer terminal que possua acesso à internet, uma dádiva alcançada nestes nossos dias de mobilidade e flexibilidade espacial.

Por fim, a perspectiva agora é o desenvolvimento de aplicações mais robustas para Desenho *scriptivo*, bem como trabalhar para a divulgação dos elementos e processos ora desenvolvidos. O corpo do presente trabalho já oferece uma coletânea variada de conceitos e ferramentas que poderão ser utilizados por praticantes da linguagem desenho que estejam preocupados com as potencialidades dessa linguagem para o uso livre em nossa sociedade. Trata-se de uma arte livre, algo que historicamente pertence ao Desenho e acompanha todos os seus representantes, desde o tempo em que o ser humano resolveu a colocar suas idéias no papel.

Algo que meu pai indagou-me um dia: “é melhor ser o mais pobre dos ricos ou o mais rico dos pobres?”. É mesmo uma honra empregar e desenvolver, pelo trabalho, os dons recebidos de Deus; sendo necessário trabalhar com ordem, paz, moderação e paciência, sem nunca recuar perante o cansaço e as dificuldades; de trabalhar sobretudo com pureza de intenção e com desapego, tendo sempre diante dos olhos a morte e a conta que deverei dar do tempo perdido, dos talentos inutilizados, do bem omitido e da vã complacência nos sucessos, tão funesta à obra de Deus. Valei-me São José.

Bibliografia

ANTON, Howard. *Cálculo, um novo horizonte*. Cyro de Carvalho Patarra e Márcia Tamanaha (trad.). 6ª ed. Porto Alegre: Bookman, 2000.

AQUINO, Santo Tomás de. *Tratado de Arte de Alquimia*. Itália, 1268.

ARTOBOLEVSKI, I.I. *Mecanismos de Palancas vol.1*. Traducido del ruso por F. Petrov. Moscou. URSS: Editorial MIR, 1976.

_____. *Mecanismos de Palancas vol.2*. Traducido del ruso por F. Petrov. Moscou. URSS: Editorial MIR, 1977.

BORGES, José Antônio; JENSEN, Leo Roberto. *Cegos e Computador: Uma Interação que Explora o Potencial do Desenho*. Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro - NCE/UFRJ. Disponível em intervox.nce.ufrj.br, acesso em 09/08/2014.

BABENKO, Dmitry; MARMANIS, Haralambos. *Algorithms of the Intelligent Web*. EUA: Manning Publications, 2009.

BARBOSA dos, Marcelo Luis. *Design hipermídia na internet: uma análise semiótica dos padrões de comunicação on-line*. São Paulo: Tese de Mestrado, Comunicação e Semiótica, PUC/SP, 2005.

BERNERS-LEE, Tim; HENDLER, James; LASSILA, Ora. *The semantic web: a new form of web content that is meaningful to computers will unleash a revolution of new possibilities*. Scientific American, 2001.

BRAGA, Theodoro. *Desenho Linear geométrico: problemas de desenho linear geométrico*. 14ª ed. São Paulo: Ícone, 1997.

BRITO, Jorge Luís Nunes e Silva; COELHO, Luiz Carlos Teixeira. *Fotogrametria digital*. EdUERJ, Rio de Janeiro, 2007.

CARVALHO, P.C.P. *Introdução à Geometria Espacial*. Rio de Janeiro, 1993.

CASTELLS, Manuel. *A sociedade em rede*. São Paulo: Paz e Terra, 2001.

CHADWICK, Henry; EVANS, Gillian Rosemary. *Atlas of the Christian church*. Editora Phaidon, 1987.

- CRESPO, Nuno. *Imagem, percepção e expressão. A estética em Wittgenstein*. Lisboa, 2008.
- CHRISTIANSEN, Tom; WALL, Larry; ORWANT, Jon. *Programming Perl, Fourth Edition*. . Sebastopol: O'Reilly Media, 2012.
- COSTA, Luciano Bedin. *O ritornelo em Deleuze-Guattari e as três éticas possíveis*. UFRGS, 2014.
- DA VINCI, Leonardo. *Códice Atlântico*. Itália, 1478-1519.
- DEITEL, H.M. H.M. Deitel, P.J. Deitel, T.R Nieto e D.C. McPhie. *Perl: como programar*. Carlos Arthur Lang Lisboa (trad.). Porto Alegre: Bookman, 2002.
- DELEUZE, Gilles & GUATTARI, Félix. *Mil Platôs vol.4*. “1837 – Acerca do Ritornelo”, Suely Rolnik (trad.), São Paulo: Editora 34, 1997.
- DERRIDA, Jacques. *Gramatologia*. São Paulo: Perspectiva, Ed.Universidade de São Paulo, 1973.
- DHOMBRES, Jean. O que é uma forma? (entrevista). In: *As ciências da forma hoje*. Émile Noël (org.). Campinas, SP: Papyrus, 1996.
- DOMINGUES, Diana (Ed.). *A arte no século XXI: a humanização das tecnologias*. São Paulo: UNESP, 1997.
- DÜRER. *Instrução para medir com régua e compasso*. Alemanha, 1525.
- EGOROVA, O. & CECCARELLI, M. *Ivan Ivanovich Artobolevski as one of the Founders of IFToMM*. 13th World Congress in Mechanism and Machine Science, Guanajuato, México, 19-25 June, 2011.
- ELAM, Kimberly. *Geometria do design: estudos sobre proporção e composição*. Claudio Marcondes (trad.). São Paulo: Cosac Naify, 2010.
- FERRETTO, Daniel. *Curvas: estudo e visualização com o software cabri-géomètre II*. Florianópolis: UFSC, 2003.
- FLEET, David; HERTZMANN, Aaron. *Computer Graphics Lecture Notes*. Canadá: Computer Science Department, University of Toronto, 2006.
- FRAGOSO, Suely. MALDONADO, Alberto Efendy (ORGs). *Internet na América Latina*. São Leopoldo: Ed. Unisinos. Porto Alegre: Sulina, 2009.
- FRANZON, Carmen Pinto; BRITO, Arlete Jesus. *Um estudo sobre Leibniz e a criação de um*

alfabeto do pensamento humano. Disponível em: <www.portalrealize.com.br>. Acesso em: 16 jul. 2012.

GIANNETTI, Claudia. *Estética Digital: sintopia da arte, a ciência e a tecnologia*. Maria Angélica Melendi (trad.). Belo Horizonte: C/Arte, 2006.

GUELICH, Scott; GUNDAVARAM, Shishir; BIRZNIEKS, Gunther. *Programação CGI com Perl*. Rio de Janeiro: Editora Ciência Moderna Ltda., 2001.

HERRINGTON, Jack D. *PHP Hacks: dicas e ferramentas para a criação de web sites dinâmicos*. João Tortello (trad.). Porto Alegre: Bookman, 2008.

HERTLEIN, Grace C. (Editor). *Computer Graphics and Art*. Universidade da Califórnia, 1976.

JACOMETTI, Welson Régis. *Geolab - Um Ambiente para Desenvolvimento de Algoritmos em Geometria Computacional*. Campinas, 1992.

LAS-CASAS, Luiz Fernando. Interferências gráficas no cinema. In: VENTURELLI, Suzete (org.) *Arte e Tecnologia: para compreender o momento atual e pensar o contexto futuro da arte*. PPG-Arte UnB: Brasília, 2008.

LEIBNIZ, Gottfried Wilhelm. *De la numération binaire*. Manuscrito disponível em: <<http://www.bibnum.education.fr/calculinformatique/calcul/de-la-numeration-binaire>>. Acesso em: 16 jul. 2012.

LIMEIRA, José L.S. *Utilização de AJAX no desenvolvimento de sistemas web*. Porto alegre: UFRGS, 2006.

LOBATO, Lúcia Maria Pinheiro. *A construção de palavras e a arquitetura da faculdade da linguagem*. Brasília: Link comunicação e design, 2010.

LULL, Raymond. *Ars Magna. The Ultimate General Art*. 1305.

MCDANIEL, Adam. *Perl and Apache, your visual blueprint[®] for developing dynamic web content*. EUA: Wiley Publishing, Inc., 2010.

MOURA, Reinaldo Aparecido. *Sistemas e técnicas de movimentação e armazenagem de materiais: materiais handling*. Reinaldo A. Moura. São Paulo, 1979.

NIXON, Robin. *PHP, MySQL, JavaScript & CSS*. EUA: O'Reilly and Associates, 2012.

NUNES, João Fernando Igansi. Estética da Interface Computacional. In: VENTURELLI, Suzete (org.) *Arte e Tecnologia: para compreender o momento atual e pensar o contexto futuro da arte*. PPG-Arte UnB:

Brasília, 2008.

ORLOV, P. *Ingeniería de diseño*. José Puig Torres (trad.). Moscou, URSS: Editorial MIR, 1974.

PRESSMAN, Roger S. LOWE, David. *Engenharia Web*. Daniel Vieira (trad.); Delano Medeiros Beder, Marcos Lordello Chaim, Paulo Cesar Masiero (revisão técnica). Rio de Janeiro: LTC, 2009.

RAY, Erik T. *Aprendendo XML*. Rio de Janeiro: Editora Campus, 2001.

ROONEY, Anne. *A História da Matemática: desde a criação das pirâmides até a exploração do infinito*. São Paulo: M.Books do Brasil Editora Ltda, 2012.

ROSA, João Guimarães. *Tutaméia (Terceiras estórias)*. 8ª. Ed. Rio de Janeiro: Nova Fronteira, 2001.

SABÓIA, Lygia Maria Maurity. *Uma poética transdisciplinar – arte, matemática, simetria e linguagem de programação Postscript. Fundamentos e instrumentos*. Tese. UNICAMP, 2001.

SANTAELLA, Lúcia. O homem e as máquinas. In: DOMINGUES, Diana (Ed.). *A arte no século XXI: a humanização das tecnologias*. São Paulo: UNESP, 1997.

SANTOS, Eduardo Toledo. Novas Tecnologias no Ensino de Desenho e Geometria. Anais do Encontro Regional do Vale do Paraíba de Profissionais do Ensino de Área de Expressão Gráfica, 2000.

SANTOS, Milton. *A Natureza do Espaço: Técnica e Tempo, Razão e Emoção*. 4.ed. São Paulo: Editora Universidade de São Paulo, 2009.

_____. *Economia Espacial: Críticas e Alternativas*. Maria Irene de Q. F. Szmrecsányi (trad.). 2ª ed., 1ª reimpr. São Paulo: Editora Universidade de São Paulo, 2007.

SILVESTRE, Ricardo Sousa. *Um curso de lógica*. Petrópolis, RJ: Vozes, 2011.

SILVA, A...[et al.]. *Desenho Técnico Moderno*. Antônio Eustáquio de Melo Pertence, Ricardo Nicolau Nassar Koury (trad.). Rio de Janeiro: LTC, 2006.

SILVA, Maurício Samy. *Fundamentos da SVG*. São Paulo : Novatec Editora, 2012.

_____. *HTML 5*. São Paulo : Novatec Editora, 2011.

_____. *JavaScript: guia do programador*. São Paulo: Novatec Editora, 2011.

SLOTTERDIJK, Peter. *Crítica da Razão Cínica*. (1981). Manuel Resende (trad.). Lisboa: Relógio D'água Editores, 2011, 671 p.

SOUZA, Carlos Henrique Medeiros de; BOLSONI, Evandro Paulo; CARDOSO, Carla. Computação Ubíqua, Cloud Computing e PLC para Continuidade Comunicacional diante de Desastres. ANAIS ELETRÔNICOS - ARTIGOS. *V Seminário Internacional de Defesa Civil - DEFENCIL*, São Paulo, 2009. (Artigo_Anais_Eletronicos_Defencil_14.pdf, 627kb) Disponível em: <www.defesacivil.uff.br>. Acesso em: 16 jul. 2012.

STARK, Jonathan; JEPSON, Brian. *Construindo aplicativos Android com HTML, CSS e JavaScript*. São Paulo: Novatec Editora; Sebastopol, CS: O'Reilly, 2012.

STRATHERN, Paul. *Pitágoras e seu teorema em 90 minutos*; Marcus Penchel (trad.); Carla Fonseca-Barbatti (consultoria). Rio de Janeiro: Jorge Zahar Ed, 1998.

TISDALL, James D. *Mastering Perl for Bioinformatics*. EUA: O'Reilly and Associates, 2003.

VARGAS, Milton. *Para uma filosofia da tecnologia*. Editora Alfa Omega: São Paulo, 1994.

VENTURELLI, Suzete (org.) *Arte e Tecnologia: para compreender o momento atual e pensar o contexto futuro da arte*. PPG-Arte UnB: Brasília, 2008.

VENTURI, Jacir J. *Cônicas e quadráticas*. 5ª ed. Curitiba, 2003.

VERBRUGGEN, Martien. *Graphic Programming with Perl*. USA: Manning Publications Co, 2002.

VIANA, Juliana Alencar; MELO, Victor Andrade de. Arte, novas tecnologias e novas redes – desafios para pensar a Animação Cultural a partir do Software Livre. In: VENTURELLI, Suzete (org.) *Arte e Tecnologia: para compreender o momento atual e pensar o contexto futuro da arte*. PPG-Arte UnB: Brasília, 2008.

VIGNOLA, Giacomo. *Regras das cinco ordens da arquitetura (Regole delle cinque ordini d'architettura)*. Roma, 1562.

VINEGAR, Ben. & KOVALYOV, Anton. *JavaScript Remoto*. São Paulo: NOVATEC editora, 2013.

VITRÚVIO. *Tratado de Arquitetura*. Martins Fontes, 2007.

WALLACE, Shawn. *Perl Graphics Programming*. EUA: O'Reilly and Associates, 2003.

WIRTH, Niklaus. *Algoritmos e Estruturas de Dados*. Cheng Mei Lee (trad.). João José Neto (revisão técnica). Rio de Janeiro: LTC, 1999.

_____. *Algorithms and Data Structures* © N. Wirth 1985. Oberon version: August 2004.

WIRTH, Uwe. As we may surf. The Relevance of Abductive Inference for surfing through the Internet. in *Semiotica*, 141-1/4, 2002, S.159-168.

WITTGENSTEIN, Ludwig. *Gramática Filosófica*. São Paulo: Edições Loyola, 2003.

_____. *Tratado Lógico-Filosófico*. São Paulo: Companhia Editora Nacional, 1968.

ZAMBONI, Sílvio. *A pesquisa em arte: um paralelo entre arte e ciência*. Campinas: Autores Associados, 1998.

INTERNET

<http://www.cpan.org/>.

<http://www.google.com>

<http://helpx.adobe.com/br/photoshop/using/file-formats.html>.

<http://www.ime.usp.br/~pf/algoritmos/aulas/bint.html>

<http://www.linhadefensiva.org/dicionario/>

<http://blogs.msdn.com/b/ie/archive/2011/04/22/thoughts-on-when-to-use-canvas-and-svg.aspx>

<http://paginas.fe.up.pt/>

FIM.