autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

e-mail: arianestolfi@gmail.com

world wide web: forma aparente e forma oculta

Ariane Stolfi

Dissertação apresentada a Faculdade de Arquitetura e Urbanismo da Universidade de São Paulo para obtenção do título de Mestre em Arquitetura e Urbanismo

Área de Concentração
Design e Arquitetura

Orientador
Carlos Zíbel Costa
São Paulo 2010.

Stolfi, Ariane

S875w World wide web: forma aparente e forma oculta. Webdesign da interface ao código / Ariane Stolfi. --São Paulo, 2010. 378 p. : il.

Dissertação (Mestrado - Área de Concentração: Design e Arquitetura) - FAUUSP.

Orientador: Carlos Zíbel Costa

1.Design 2.Linguagens 3.HTML
I.Título

CDU 7.05

*Pro meu pai, meu filho
e o espírito do tempo*

Resumo

Este trabalho pretende estudar a world wide web sob o ponto de vista da sua forma, considerando sua forma tanto a forma aparente, que é a interface, quanto sua forma oculta, dada pelos códigos que a conformam. A intenção é reunir referências tanto para designers que queiram ir mais além no domínio das linguagens de programação e marcação, quanto para programadores que queiram ter mais domínio sobre a linguagem do design gráfico. Para isso, desmembramos o trabalho em três partes.

Uma primeira, trata da formação histórica da rede, com ênfase na transformação do papel do computador, de uma ferramenta de uso militar, para o suporte para um novo meio de comunicação de massa. Em seguida, partimos para uma análise mais objetiva da web sob o ponto de vista de sua forma aparente, e para isso tentamos primeiro compreender o design gráfico como linguagem, e como linguagem, quais são seus elementos constitutivos, além de questões novas que se colocam para o design de interfaces, como arquitetura de informação e usabilidade. Na terceira parte, elaboramos um manual técnico dos protocolos, linguagens e recursos utilizados para a construção de websites. Explicamos como utilizá-los e reunimos exemplos ilustrativos para auxiliar no aprendizado.

Por fim, fazemos considerações sobre os agentes envolvidos na produção de websites — designers e programadores —, sobre a relação entre eles no processo de trabalho e introduzimos questões que se colocam no modo de construção da web de modo mais geral, a partir da relação com um novo tipo de usuário, que é também produtor. Com isso, produzimos um material que serve como introdução ao webdesign, abarcando conceitos e linguagens relacionadas ao processo, que serve como referência para aqueles que quiserem se envolverem com a disciplina.

Abstract

This work studies the world wide web under the terms of its form, considering their form as the apparent form, which is the interface, and its hidden form, given by the codes that builds it. The intention is to gather as much reference for designers who want to go further in the field of programming languages and markup, and for developers who want to have more control over the language of graphic design. For this, we have unbundled the work in three parts.

The first one deals with the historical formation of the network, with an emphasis on the transformation on the role of the computer, from a tool for military use into the support for a new medium of mass communication. Then we went to a more objective analysis of the web from the point of view of its apparent form, and we try to first understand the graphic design as a language and beeing language, what are its constituent elements, and also new issues that arise for the design of interfaces, such as information architecture and usability. In the third part, we prepared a technical manual of protocols, languages and resources used to build websites. We explain how to use them and gather illustrative examples to assist in learning.

Finally, we make considerations for those involved in the production of websites - designers and developers -, on the relationship between them in the work process and introduce issues that stand more generally in the way of building the web, related to a new type of user, which is also producer. As a result, produce a material that serves as an introduction to web design, covering concepts and languages related to the process, which serves as reference for those wishing to become involved in discipline.

Agradecimentos

Agradeço a todos que ajudaram na elaboração deste trabalho, principalmente amigos que ajudaram na revisão crítica como Diogo Gomes, que criticou e ajudou a tornar o texto menos pretensioso e redundante, Diogo Damásio, que me ajudou com uma bibliografia essencial, e também na discussão dos capítulos finais e o professor Etienne Delacroix, que inspirou este trabalho e me fez uma leitura final. Amigos que me ajudaram cedendo materiais pontuais, como Fernando Bizarri, Bruno Schiavo e Nicolas Montgermont. E ao Dr. Thiago de Lucca, que me acudiu quando essa empreitada começou a me deixar doente.

Agradeço a meu professor orientador Carlos Zibel Costa, que acreditou na potencialidade deste trabalho e me orientou ao longo dos últimos três anos, aos colegas do grupo de pesquisa DEVIR, que me deram ânimo nesta reta final de continuar a pesquisa nessa área, e ao professor Marcelo Bicudo que me ofereceu o trabalho no Laboratório de Identidade Visual, sem o qual eu talvez não tivesse conseguido terminar este mestrado.

Aos professores da minha banca de qualificação Giorgio Giorgi e Caio Vassão, que me ajudaram a dar as direções deste trabalho e a torná-lo mais viável e mais útil e Marcelo Modesto, que me ajudou com as revisões durante o período de pré-qualificação.

E principalmente a meu pai, Guido Stolfi, que revisou tudo cuidadosamente e me deu suporte e estrutura para que este trabalho pudesse ser executado, e ao Tales, meu filho, que compreendeu o esforço e a dedicação necessária para a que este trabalho pudesse ser concluído.

Índice

0. Resumo Abstract	5
1. Introdução	13
2. Uma breve história da web	35
2.1 Do computador como ferramenta militar, ao computador como amplificador do intelecto humano	37
2.2 O computador pessoal	65
2.3 As origens do hipertexto	85
2.4 A internet e a world wide web	93
2.5 Forma e conteúdo da rede: história das linguagens	117
2.6 Um novo paradigma: o Software livre	131
2.7 Web 2.0: a rede dinâmica	139
3. A forma aparente	153
3.1 A linguagem do design	155
3.5 A Interface	197
4. A forma Oculta	219
4.1 Impulsos elétricos, bits e bytes	221
4.2 Redes e Protocolos de transmissão	227
4.3 A estrutura-base: O HTML	239
4.4 Forma na estrutura: CSS	259
4.5 Forma em movimento linguagens para interatividade	297
4.6 Forma em construção: sistemas de gerenciamento de conteúdo	317
5. Forma Aparente e Forma Oculta	345
5.1 Designers e Programadores	347
5.2 Produção / consumo e produssumo	355
6. Conclusão	355
6.1 Uma estética em escala global	356
6.2 A formação de um novo campo disciplinar	357
7. Bibliografia	361

Introdução

A virada do século XX para o século XXI foi imaginada e descrita por uma série de escritores de ficção científica e filmada de diversas maneiras na história do cinema. Em 2001, *uma odisséia no espaço*, de Stanley Kubrick, um dos filmes mais significativos de ficção científica já filmados, a virada para o século XXI aparece com o homem conquistando o espaço, e com um computador que começava a adquirir sentimentos humanos e subjugar a humanidade. O domínio dos computadores sobre os homens também foi proposto em vários outros filmes e livros, vários futuros foram imaginados: ciborgues, andróides, viagens aeroespaciais, exaustão dos recursos naturais, guerras nucleares, carros voadores, superpopulação.

O caminho trilhado, no entanto, foi um tanto distinto: uma rede mundial de computadores, onde uma parte considerável da cultura humana criada até então está reunida, ao alcance de um clique; redes sociais, na qual milhões podem entrar em contato uns com os outros de forma direta; um espaço onde os homens podem comprar mercadorias de qualquer parte do mundo, disponibilizar sua mão-de-obra globalmente, encontrar parceiros e amores; um lugar aonde as pessoas podem publicar suas produções artísticas e acadêmicas livremente. Robôs existem aos montes, mas principalmente para percorrer esse imenso mar de dados em busca de informação.

A internet, fruto da guerra fria, surge com a idéia da criação de uma rede de computadores descentralizada, que pudesse sobreviver a possíveis ataques nucleares — uma resposta do militar americano ao lançamento do foguete russo Sputnik. Até o começo da década de 90, essa rede teve uso predominantemente militar e acadêmico, e foi a invenção de uma nova tecnologia embasada nela que pode fazer-lhe dar o seu grande

salto social: a World Wide Web (WWW).¹ O desenvolvimento e alastramento dessa rede se deu numa velocidade e escala nunca antes vista na história do homem. Em vinte anos, um simples documento — escrito pelo pesquisador Tim Berners-Lee, quando era consultor do Centro Europeu de Pesquisas Nucleares (CERN) — com uma proposta de um sistema de organização e compartilhamento de informações, deu origem a um dos mais importantes meios de comunicação mundiais.²

1 É muito comum haver uma certa confusão entre internet e World Wide Web. A internet é a rede física que interliga os computadores mundialmente. A World Wide Web é uma tecnologia que existe na internet. A internet é mais ampla do que a world wide web, e inclui redes de e-mail, redes de troca de arquivos, como as p2p (torrents, soulseek, e-mule, etc) e a usenet, por exemplo. Num capítulo mais adiante falaremos melhor sobre essas outras redes. A World Wide Web, é a rede de páginas, cujos endereços começam em geral com www e inclui alguns protocolos específicos, como http, ftp, stp, que também explicaremos na terceira parte desse trabalho.

2 Além da internet, o telefone também teve um grande salto evolutivo nesses últimos 15 anos, com a invenção e a popularização do telefone celular; mas o telefone sozinho é um meio restrito apenas à fala, enquanto a internet é um suporte para uma multiplicidade de formas de linguagens: texto, imagem, som e vídeo e permite, inclusive, a circulação de capital e de mão de obra. A tendência atual inclusive, é a fusão das tecnologias, com a rede sendo incorporada pelos parelhos

A internet não é somente o desenvolvimento de meios de comunicação que vieram antes dela, mas segundo Michael Dertouzos, no prefácio do livro de Tim-Berners Lee “Waving the Web”, inaugura uma forma de comunicação nunca antes experimentada pela sociedade: a comunicação muitos a muitos — numa escala global:

Ela [a world wide web] já nos deu uma gigantesca feira- livre de informação, onde indivíduos e organizações compram, vendem e trocam informações e serviços de informação livremente uns com outros. A imprensa, o rádio e a televisão nunca chegaram perto disso; tudo que eles conseguem fazer é pulverizar a mesma informação a partir de uma fonte para muitos destinos. Nem as cartas nem o telefone se aproximam do poder da Rede, porque mesmo esses meios só permitem trocas uma-um, são lentos e carecem da capacidade do computador de busca, automação e mediação. Significativamente — comparada com a prensa de Gutemberg, o telefone de Bell e o rádio de Marconi — e muito antes de estabelecer sua forma final, a Rede de Berners-Lee já estabeleceu sua singularidade.

Milhares de cientistas da computação estiveram debruçados por duas décadas sobre mesmas duas coisas — hipertexto e redes de computador. Mas só Tim concebeu como juntar esses dois elementos e criar a Rede. ³

Um novo campo de pesquisa

A relevância dessa ferramenta no cotidiano e no desenvolvimento de todas as ciências já se mostra extremamente significativa para a comunidade acadêmica. Em todas as áreas do conhecimento, uma série de pesquisadores já se volta para esse novo meio, estudando as influências e as potencialidades da internet em relação às suas respectivas disciplinas. Em uma busca no Dédalus,

celulares, televisões e videogames.

³ Dertouzos in Berners-Lee 1999 pg X

Welcome! Sign in or register.

Categories ▾MotorsStoresDaily Deal

Home > Buy > Search results for "star wars"

Find star wars in All Categories
☐ Include title and description

Related Searches: star trek, lightsaber, transformers, lego, gi joe, star wars action figures, star wars lot, star wars vintage

Refine search

Categories

- Toys & Hobbies (46,502)
 - Action Figures (30,148)
 - Building Toys (11,966)
 - Games (1,363)
 - Models & Kits (581)
 - Trading Card Games (938)
- Collectibles (15,189)
 - Science Fiction & Horror (7,291)
 - Trading Cards (2,150)
 - Decorative Collectibles (2,189)
 - Comics (1,188)
- Clothing, Shoes & Accessories (8,176)
 - Men's Clothing (3,299)
 - Costumes & Reenactment Attire (1,574)
 - Unisex Clothing, Shoes & Accs (2,055)
 - Boys (956)
- Video Games (6,872)
 - Games (5,763)
- Art (8,371)
 - Art from Dealers & Resellers (6,216)

See all categories

Price

\$ to \$

Condition

☐ New
☐ Used
☐ Not specified
Choose more...

Seller

☐ eBay Top-rated sellers
Specify sellers...

Preferences

Buying formats

☐ Auction
☐ Buy It Now
☐ Include Store inventory
Choose more...

Show only

☐ Completed listings
☐ Free shipping
Choose more...

Location

☐ US Only
☐ North America
☐ Worldwide
Choose more...

Distance

Customize preferences

See all matching eBay Stores

Recently viewed items

Rock Girl Vintage
\$13.75
19c 21n

ADVERTISEMENT (about)

worldofgood.com

ebay

All ItemsAuctions onlyBuy It Now only

96,203 results found for **star wars** [Save this search]

View as [List] [Customize view]

EBAY'S HOLIDAY TOY LIST GO!

Featured Items

STAR WARS MILLENNIUM FALCON GALACTIC HEROES NEW LOT
WE HAVE 1000'S OF STAR WARS ITEMS POWER SELLER 100%
Enlarge

Star Wars Clone Turbo Tank Vehicle+Action Figure NEW
See VIDEO Here! HUGE! Speeder Bike! MORE. 100% Feedback
Enlarge

Droids - Misc Droids Auth. Star Wars Poster
Enlarge

Star Wars-Collectable TCG Pack Vadar Front-Mixed Cards

YODA Star Wars Holiday Christmas Light Set NEW XMAS
Enlarge

star wars millenium falcon 3d puzzle 857 pieces

Star Wars Clone Wars CW21 ANAKIN SKYWALKER MOC!

Clone Wars Auth. Star Wars Poster
Enlarge

MIB Star Wars Royal Elegance Queen Amidala Episode I

STAR WARS UNLEASHED DARTH VADER SECOND CARD

Star Wars MR 45 Scaled Lightsaber Qui-Gon Jinn
Enlarge

Star Wars Clone Wars CW37 ZIRO's ASSASSIN DROID MOC!

Darth Vader Auth. Star Wars Poster
Enlarge

Yoda Auth. Star Wars Poster
Enlarge

2 Pack SWORD LIGHT SABER For Nintendo Wii STAR WARS NEW

1983 STAR WARS RETURN OF JEDI "JABBA THE HUT" GLASS
Enlarge

Star Wars Legacy BD38 CONCEPT KI-ADI-MUNDI MOCI!

Obi-Wan Kenobi Auth. Star Wars Poster
Enlarge

Star Wars Lego Set 8009 Technic R2-D2

STAR WARS UNLEASHED LUKE SKYWALKER PILOT

Star Wars Legacy BD37 CLONE COMMANDER DEVISS MOCI!

Hallmark 1999 Star Wars QUEEN AMIDALA ORNAMENT - MIB
Enlarge

Bounty Hunters - Boba Fett Auth. Star Wars Poster
Enlarge

6 LBS Mixed LEGO LEGOS STAR WARS AND OTHER SETS

Star Wars Speeder Bike Model Kit "NIB"

o banco de dados das bibliotecas da USP, podemos encontrar hoje 576 teses feitas somente na USP, em campos tão diversos como psicologia, computação, economia, direito, educação, jornalismo, comunicação, letras e até astronomia, que tem a palavra internet no seu título ou em sua descrição.

Entendendo a importância que a internet tem como meio de comunicação, e o crescimento da atividade de webdesign dentro do campo da comunicação visual, este trabalho busca ampliar a pesquisa nesta área dentro da FAU, onde ela ainda começa a se articular. Ele é fruto do desenvolvimento de uma pesquisa que já dura mais de quatro anos, e começou ainda na graduação, no curso de uma disciplina intitulada "Oficina de Arte e Programação", ministrada pelo professor Etienne Delacroix, que buscava apresentar e estimular o conhecimento de conceitos de linguagens de programação e a produção cultural e artística experimentais a partir desses recursos.

Desse caminho aberto, seguii se o desenvolvimento de um dicionário de código HTML, CSS e *JavaScript* — linguagens que são a base da World Wide Web — apresentado como TFG na FAU, que como o próprio professor Etienne afirmou, necessitava de um "livro companheiro" que fosse além no estudo e na compreensão desse meio. Seguindo essa orientação, esta pesquisa é feita nesse sentido, de produzir um entendimento sobre a internet, do ponto de partida da comunicação visual, que é a nossa área de atuação, no sentido de procurar elucidar o modo (ou modos) como se estrutura a rede.

Este trabalho, está organizado em quatro partes: primeiro, uma pesquisa histórica sobre a tecnologia da comunicação; depois, uma reflexão sobre a forma aparente desse novo meio, e conceitos que estão ligados a ela; em terceiro, um

aprofundamento nas tecnologias e protocolos que estruturam a web: a forma oculta; e, por fim, considerações sobre a relação entre forma aparente e forma oculta, no modo de produção dos websites e na forma com que a rede se configura atualmente.

Uma fonte de volume inesgotável

A internet é o coroamento de uma nova economia, que se desenvolveu a partir da última revolução tecnológica, como aponta Manuel Castells:

Uma nova economia surgiu em escala global no último quartel do século XX. Chamo-a de informacional, global e em rede [...]. É informacional porque a produtividade e a competitividade de unidades ou agentes nessa economia [...] dependem basicamente de sua capacidade de gerar, processar e aplicar de forma eficiente a informação baseada em conhecimentos. É global porque as principais atividades produtivas, o consumo e a circulação, assim como seus componentes (capital, trabalho, matéria prima, administração, informação, tecnologia e mercados) estão organizados globalmente. É rede porque, nas novas condições históricas, a produtividade é gerada, e a concorrência é feita numa rede global de interações entre redes empresariais.

A internet participa no processo produtivo de várias maneiras. Ela é um meio de circulação de informação e tecnologia, facilita a distribuição de trabalho numa escala global e aumenta o ritmo de circulação de mercadorias. Uma estimativa de volume financeiro movido apenas pelo comércio eletrônico nos Estados Unidos, chega a somas de mais de 200 bilhões de dólares em 2008⁴.

http://news.netcraft.com/archives/2009/06/site_count_history.png

Segundo a Internet Telecommunication Union, atinge cerca de 23% da população mundial, sendo superada somente (e de longe) pelos telefones celulares⁵. Em 2005, a rede já tinha atingido um bilhão de usuários⁶. No Brasil, a penetração da rede já chega a 34% da população, o que corresponde a mais de 67 milhões de usuários⁷. O gráfico acima fornecido pela pesquisa da Netcraft mostra a evolução histórica do número de domínios; e a mesma pesquisa afirma que existem mais de 200 milhões de sites no mundo.

O volume de informação disponível é um dado impossível de ser medido, uma vez que a rede é uma estrutura descentralizada, e que mesmo mecanismos de busca super potentes, como os robôs da google, não conseguem atingir o que se chama de web profunda, que é uma imensa quantidade de dados⁸ localizados em intranets,

5 Em 2008, o número de assinaturas de números de telefone celular chega a 61% do número de habitantes, segundo o documento Measuring the information Society, da International Telecommunication Union.

6 "One Billion Internet Users", do alertbox de Jakob Nielsen, dezembro de 2005. http://www.useit.com/alertbox/internet_growth.html

7 fonte: <http://internetworldstats.com/stats10.htm>

8 Em 2000, estimava-se que a web profunda tinha um volume de dados mais de cinco vezes maior do que a web superficial. fonte: How Much Information? - Internet - Summary, 2000. Disponível em: <<http://www2.sims.berkeley.edu/research/projects/how-much-info/internet.html>>

4 June 2008: Us Retail e-commerce. Pesquisa da e-marketer sobre comércio eletrônico. http://www.iab.net/insights_research/530422/1675/334589

em diretórios sem links públicos e bancos de dados particulares.

História do Presente

A história da comunicação visual vem se desenvolvendo desde os primórdios junto à história do homem, e já foi registrada e estudada por uma série de historiadores como Philip Meggs, Richard Hollins, Abrams, entre outros, de uma certa forma. Art Nouveau, Art Deco, Futurismo, Construtivismo Russo, Escola Suíça e até o Pós Modernismo, como várias outras escolas e linguagens surgidas ao longo do tempo, são partes bem conhecidas dessa história, registrada em belos livros ilustrados. É certo também que os avanços tecnológicos de cada tempo tiveram um papel central no desenvolvimento de cada uma dessas escolas e de muitas outras que existiram, mas os historiadores sempre procuram estudar aqueles avanços que tiveram ligações mais diretas com o desenvolvimento dessas linguagens, e, principalmente, aqueles que tinham mais relação com a comunicação impressa, que até pouco tempo atrás era a forma dominante de comunicação visual. Eles não tinham como supor qual seria o futuro da comunicação visual, e mesmo quando estudaram o computador, estudaram-no somente como ferramenta para a comunicação visual impressa, e não como suporte.⁹

No instante em que as atenções dos designers se voltam para esse novo meio que é a rede, a história do design gráfico se funde com a história dos meios de comunicação.

Ao longo do século passado, os avanços tecnológicos no campo da informática eram somente ferramentas para o design gráfico¹⁰, e foi só no final do século passado que o computador deixou de ser uma ferramenta e passou a ser também o suporte para uma nova forma de comunicação visual. A interface, até então, era somente objeto do design industrial, e as tecnologias de comunicação, incluindo as ciências que dão base a elas, ficaram de uma certa forma fora da historiografia do design gráfico.

E compete a nós que estamos vivendo essas transformações, vivendo esse novo meio, buscar reconstruir a história dele a partir desses novos elementos, resgatando uma série de agentes e momentos históricos responsáveis por fazer-nos chegar a nosso estado atual: matemáticos, inventores, cientistas e pesquisadores que criaram os primeiros meios de comunicação elétricos, que desenvolveram os primeiros computadores e que deram as bases para a construção dessa realidade da rede mundial de computadores entram agora na história do design gráfico — assim como os inventores do papel, da tinta e da imprensa já faziam parte dela há muito tempo.

Para essa tarefa, foram fundamentais as leituras de Marshall McLuhan, que buscou compreender o desenvolvimento dos meios de comunicação e suas consequências para a costura do tecido social. McLuhan enxerga os meios de comunicação como extensões do homem, e principalmente a tecnologia elétrica como uma extensão do nosso sistema nervoso, nas próprias palavras dele:

9 Tanto para a comunicação visual impressa quanto a audiovisual, como o desenvolvimento de aberturas de filmes e vídeos. No final do século XX, alguma atenção foi voltada para o que se chamou de *multimídia*, que incluía a criação de materiais interativos, como *CD-Roms*, jogos eletrônicos e instalações artísticas.

10 Aqui consideramos as interfaces de equipamentos eletrônicos ou mecânicos. Algum campo de trabalho para designers gráficos também já existe há um certo tempo (começo dos anos 80) na indústria do software, mas esse era um campo relativamente restrito se compararmos com o campo de trabalho que a *world wide web* abriu para os designers. Falaremos mais adiante sobre o desenvolvimento das interfaces gráficas na informática.

Ao colocar o nosso corpo físico dentro do sistema nervoso prolongado, mediante meios elétricos, nós deflagramos uma dinâmica pela qual todas as tecnologias anteriores — meras extensões das mãos, dos pés, dos dentes e dos controles do calor do corpo, e incluindo as cidades como extensões do corpo — serão traduzidas em sistemas de informação. (...) O homem deve servir à tecnologia elétrica com a mesma fidelidade servomecânica com que serviu seu barco de couro, sua piroga, sua tipografia e todas as demais extensões de seus órgãos físicos. Com uma diferença, porém: as tecnologias anteriores eram parciais e fragmentárias, a elétrica é total e inclusiva. Um consenso ou uma consciência externa se faz agora tão necessário quanto a consciência particular. Com os novos meios também é possível armazenar e traduzir tudo; e quanto à velocidade, não há problema. Nenhuma aceleração maior é possível a quem da barreira da luz.¹¹

Ele vivia no momento da euforia da televisão, e ainda não tinha a dimensão de algo com a internet, mas em alguns momentos parece já prever o desenvolvimento da tecnologia para algo maior do que os meios de seu tempo lhe indicavam. E nesse sentido, vimos ao longo dessa pesquisa que ele não foi somente um visionário, mas também foi um fundador, uma vez que suas idéias foram lidas e estudadas por uma série de pesquisadores que colaboraram no processo de desenvolvimento dos meios de comunicação eletrônicos¹². Mc Luhan acredita que uma mudança no meio tem consequências em todo o tecido social, nesse sentido, "o meio é a mensagem", como explica no seguinte parágrafo:

Numa cultura como a nossa, há muito acostumada a dividir e estilhaçar todas as coisas como meio de controlá-las, não deixa, às vezes, de ser um tanto chocante lembrar que, para efeitos práticos e operacionais, o meio é a mensagem. Isto apenas significa que as consequências sociais e pessoais de qualquer meio — ou seja, de qualquer uma das extensões de nós mesmos — constituem o resultado do novo estalão introduzido em nossas vidas por uma nova tecnologia ou extensão de nós mesmos.¹³

ou ainda:

O meio é a mensagem" significa, em termos da era eletrônica, que já se criou um ambiente totalmente novo. O "conteúdo" deste novo ambiente é o velho ambiente mecanizado da era industrial. O novo ambiente reprocessa o velho tão radicalmente como a TV está reprocessando o cinema. Pois o conteúdo da TV é o cinema.¹⁴

O momento vivido por McLuhan: a formação da cultura de massas e o domínio da televisão, era somente um passo do que ainda estava por vir. Agora, da mesma forma que a TV reprocessou o cinema, a internet reprocessa a TV, e vai muito além disso, reprocessando também todos os outros meios de comunicação anteriores num só. A interatividade que ela proporciona é um passo muito além da TV, do rádio e do cinema, é a possibilidade de comunicação instantânea através de longas distâncias, já anunciada pelo telefone, ampliada para abrigar todas as vantagens e potencialidades da cultura escrita, como a possibilidade de durar para além de seu instante.

O conteúdo de um meio, diz McLuhan (1964), "é sempre outro meio ou veículo" assim, como "o conteúdo da escrita é a fala", e "a palavra escrita

11 McLuhan, 1964 pg 78

12 O próprio Alan Kay, pesquisador do pioneiro laboratório da Xerox Park, em Palo Alto, e inventor o primeiro computador portátil afirma no livro de Bill Moggridge "Design Interactions", que se sentiu inspirado por McLuhan a pensar no computador como um meio de comunicação verdadeiramente novo. (Moggridge, 2007 p 159)

13 McLuhan, 1964 pg 21

14 Op. Cit. pg 11

é o conteúdo da imprensa". O conteúdo da internet, é, por sua vez, o conteúdo de todas as formas de comunicação anteriores: a fala, a escrita, a imprensa, a arte, a televisão e a música. A "mensagem" de um meio ou tecnologia, por sua vez, é nas palavras dele "a mudança de escala, cadência ou padrão que esse meio ou tecnologia introduz nas coisas humanas" e no caso da internet, a mudança de escala foi radical, unindo, não o mundo todo, mas todas as partes dele onde tenha eletricidade e/ou rede de telefone, numa única rede¹⁵.

A televisão já tinha consagrado a cultura de massas, mas com a internet — principalmente após a web 2.0, fenômeno que trataremos mais adiante — as massas já não são mais somente contemplativas, e passam a se tornar agentes: elas tem o poder de fazer sua própria programação, tem o poder de responder e intervir sobre as notícias, e de escolher entre infinitos "canais". A indústria cultural deixa de ser dividida entre produtores e consumidores, inaugurando uma era onde todos indivíduos, produtores em potencial, passam a ter acesso aos meios de produção culturais. Com a internet, a eletricidade já não é mais uma extensão somente do nosso sistema nervoso, mas também a extensão de vários sentidos — tato, audição, visão, fala — de várias pessoas interconectadas. É um sistema nervoso coletivo, um processo nunca vivido na história da cultura humana.

A rede permite o acesso a um volume de informações colossal: a coisas que só existiam em acervos pessoais, a vídeos que já passaram há muito tempo na televisão, a documentos que

antes eram praticamente inacessíveis, a um sem número de gravações musicais. E esse volume de informação só tende a aumentar mais e mais, de acordo com uma tendência que tem se mantido de aumento pregressivo da capacidade e velocidade dos computadores¹⁶. É certo que o amplo acesso a esse oceano de informações, de possibilidades e de indivíduos terá efeitos sobre toda organização social, sobre a política e a sobre a civilização como um todo. Fatos objetivos como esses criam condições históricas sem precedentes, nunca imaginados por teóricos, revolucionários e políticos antes de seu advento. Quais efeitos serão, é impossível de se determinar, mas alguns fatos objetivos já começam a ser observados, entre eles, a formação de um mercado global de mão de obra, a decadência da televisão como meio de comunicação predominante e a transformação da própria linguagem. O próprio McLuhan constatou:

Os novos meios e tecnologias pelos quais nos ampliamos e prolongamos constituem vastas cirurgias coletivas levadas a efeito no corpo social com o mais completo desdém pelos anestésicos. Se as intervenções se impõem, a inevitabilidade de contaminar todo o sistema tem de ser levada em conta. Ao se operar uma sociedade com uma nova tecnologia, a área que sofre a incisão não é a mais afetada. **O sistema inteiro é que muda** [grifo nosso].

É um momento aonde os produtos culturais estão se desmaterializando, e com ele, conceitos solidamente edificados ao longo de séculos de desenvolvimento da mídia impressa começam a ser questionados, como o conceito da propriedade intelectual, dos direitos autorais.

15 Na realidade, a China tem um controle de acesso à internet controlado por um bloqueio central, conhecido como "a grande muralha digital". Alguns países muito pequenos e subdesenvolvidos também não têm acesso à rede, assim como a Coreia do Norte, que como a China, também é controlada por um regime autoritário (fonte <http://www.internetworldstats.com>)

16 A lei de Moore diz que a capacidade de processamento dos computadores cresce exponencialmente, dobrando a aproximadamente cada dois anos. Kurzweil, 1999

Qualquer suposição sobre as influências que a internet pode ter nas demais partes do sistema pode ser arriscada, mas podemos estudar as transformações ocorridas na história da humanidade com o surgimento de cada novo meio, desde os primórdios, para entender como eles afetaram a sociedade no seu tempo. Assim, começaremos este trabalho com uma breve história da transformação da tecnologia informática até o momento presente, para entender os agentes que trouxeram a tecnologia até o seu estágio atual.

A proposta inicial deste trabalho não era de ser uma análise histórica, e sim o de ser uma ferramenta para auxiliar designers e programadores a lidarem com os novos desafios impostos pela internet. Contudo, vivemos hoje um momento no qual as tecnologias mudam muito rápido — nada é estável e o futuro é incerto — e, diante disso, é certo que qualquer trabalho que seja feito somente com base na tecnologia do presente poderia se tornar obsoleto em pouco tempo. De qualquer maneira, o destino desse trabalho, mais cedo ou mais tarde, seria de fazer sentido mais para a história do que para a prática cotidiana dos designers e programadores. Diante disso, a saída foi assumir desde já esse destino, e entender desde já esse trabalho como um trabalho de história, da história do presente.

Para fazer essa pesquisa, procuramos estudar o desenvolvimento da informática, procurando entender principalmente a trajetória do desenvolvimento da computação, de uma ferramenta militar, para a sua transformação em um meio de comunicação em escala global. Para isso, contamos com o trabalho de autores como Howard Rheingold, Bill Moggridge — que traçou um bom panorama do desenvolvimento do design de interações, através de entrevistas com vários dos agentes envolvidos nesse processo — e

Manuel Castells, um teórico que se também se dedicou ao estudo desse novo meio.

A forma da Rede: forma aparente e forma oculta

Poderíamos estudar vários aspectos da rede: aspectos econômicos, sociológicos, influências psicológicas da rede, centenas ou milhares de aspectos que a rede possui, mas por estarmos inseridos dentro do campo disciplinar da comunicação visual, e dentro ainda da Faculdade de Arquitetura e Urbanismo, é interessante que o tema central desse trabalho seja a internet sob um ponto de vista específico: o da sua forma. Apesar de sabermos que forma e conteúdo se relacionam, a forma, mais do que o conteúdo, será nosso objeto de pesquisa. Assim, as partes seguintes desse trabalho serão divididas no sentido de estudar dois aspectos dessa forma: a forma aparente, e o que chamamos aqui de "forma oculta".

A forma aparente é, a grosso modo, a interface: é a forma com que os sites de internet se mostram para o mundo; então procuraremos estudar a linguagem do design gráfico e questões que se colocam com relação à interface, como a usabilidade, acessibilidade e arquitetura da informação.

O que estamos chamando aqui de "forma oculta", é na verdade a estrutura dos sites de internet. Sabemos que as páginas da web se estruturam a partir de códigos, e iremos nos dedicar com mais atenção a essa forma oculta, que é construída a partir de uma série de linguagens e recursos de informática, tratando também das possibilidades novas surgidas nos últimos anos com o desenvolvimento da rede, baseadas na capacidade de multiplicação e alteração dos códigos.

A Forma Aparente

Para estudar a "forma aparente", primeiramente, estudaremos a própria linguagem do design

gráfico e seus elementos básicos e constitutivos: cor, forma, tipo, disposição, entre outros. Além disso, procuraremos entender o design gráfico como linguagem, que será importante mais tarde para a compreensão da linguagem que configura os sites da web.

Mesmo antes da web, o computador já tinha se tornado a principal ferramenta para a produção gráfica impressa, e tinha revolucionado todas as etapas da produção, desde a escrita, com a invenção dos processadores de texto, a produção gráfica, imensamente facilitada através de programas de edição de fotografias como Photoshop e de desenho vetorial como o Corel Draw, a diagramação, feita com programas específicos para composição de textos como o PageMaker e até a impressão de fotolitos, feita diretamente a partir de arquivos digitais.

O uso do computador como ferramenta nas artes gráficas foi uma nova revolução estética nas páginas impressas: multiplicou a disponibilidade de fontes tipográficas e ampliou o repertório da linguagem gráfica, facilitando a realização de efeitos especiais, as sobreposições de elementos, criando novas possibilidades de disposição e orientação, facilitando o recorte e a fotomontagem, ampliando as possibilidades de simulação 3D, facilitando recursos de tratamento de imagens, reduzindo o custo da realização de testes e simulações e sobretudo aliviando o peso do erro, com a possibilidade de se desfazer o que foi feito, só para enumerar alguns entre vários recursos proporcionados pela inclusão dessa nova ferramenta.

As possibilidades oferecidas pela linguagem de configuração das páginas web, o CSS, no entanto, são diferentes das oferecidas para o design de mídia impressa. Entender os elementos da linguagem do design de uma forma abstrata, é

importante para pensarmos como é possível se projetar as interfaces.

Como a forma aparente, na internet, é a interface, estudaremos algumas disciplinas que se voltaram para o estudo e construção das interfaces gráficas. A primeira é a arquitetura de informação, e também a usabilidade: uma palavra que é repetida à exaustão por vários experts e agentes envolvidos na construção e planejamento de websites. Usabilidade, ou design centrado no usuário, é um campo disciplinar que determinou toda uma metodologia para o desenvolvimento de websites, e nesse campo, é importante a leitura de Jakob Nielsen — chamado de "guru da usabilidade" — e Don Norman, dois dos principais agentes na formação e difusão desse conceito, que publicam constantemente pesquisas sobre aspectos relacionados a relação entre a tecnologia e os usuários.

A Forma Oculta

Tendo passado por alguns elementos que constituem a forma aparente, é o momento de nos enveredarmos pelo que estamos chamando de "forma oculta": as tecnologias, protocolos e linguagens que servem de estrutura para a Rede.

Nesse sentido, começaremos por conceitos básicos de comunicação entre máquinas: impulsos elétricos, a noção de bit e byte, o sistema binário e suas vantagens em relação ao sistema analógico. A partir disso, iremos avançar sobre a comunicação mais específica da internet: a definição de rede de computadores, os protocolos de comunicação entre computadores, a noção de endereçamento, de domínio, e o sistema de transferência de arquivos — os fundamentos da rede.

A partir disso, nos dedicaremos a estudar mais profundamente o ambiente de desenvolvimento das páginas da rede. Para começar, o HTML, ou

Hypertext Markup Language, que é uma linguagem que estrutura os elementos de uma página a partir de definições de semânticas, diferenciando, por exemplo, o que é um parágrafo do que é uma imagem, do que é uma tabela, e assim por diante. É uma linguagem que tem algumas características peculiares, definidas pelo projeto da rede, e que é fundamental para a configuração de algumas características da rede, como a abertura, a permeabilidade e a navegabilidade.

Estudaremos também a linguagem chamada de CSS ou Cascading Style Sheets, que é a responsável por dar forma ao HTML, através de definições de aparência, como cor, tamanho, posição, fundo, entre várias outras. O CSS é a transformação dos elementos da linguagem do design gráfico em linguagem funcional, textual, com algumas restrições impostas pelo meio e pelo seu estágio de desenvolvimento.

Depois avançaremos para o estudo do *JavaScript*, que é uma linguagem que permite a manipulação direta dos elementos de uma página, permitindo animações, personalizações, e interações mais avançadas. E junto ao *JavaScript* estudaremos o *Ajax*, que é uma nova forma de se usar HTML, CSS e JavaScript, que permite atualizações dinâmicas sem o recarregamento das páginas.

No capítulo seguinte, passaremos a estudar também os chamados CMSs (*Content Management Systems*, Sistemas de Gerenciamento de Conteúdo), que são aplicativos que podem ser projetados ou encontrados prontos para utilização, e em muitos casos com distribuição livre, que já contém uma solução pronta de código que pode ser instalada, adaptada e utilizada pelos designers, programadores ou produtores de

conteúdo para criação de um site. Os CMSs e bibliotecas de código abrem também uma nova perspectiva para o mercado de trabalho e a prática profissional, que é a customização e a criação de aplicativos baseados em dados de terceiros.

Nessa parte do trabalho, a intenção não é se aprofundar exaustivamente sobre cada um desses pontos, já que existe uma abundância de livros técnicos sobre cada um deles; mas apresentar considerações gerais e familiarizar o leitor com a forma-código. A intenção não é dar o domínio total sobre tudo isso que foi introduzido, mesmo porque não seria possível com o trabalho de uma pessoa nesse período de tempo, mas sim dar um panorama geral que poderá auxiliar quem quiser um domínio maior sobre o código, ou sobre a estrutura, em particular os designers. A ideia é ajudar aqueles que desejarem escolher um caminho de aprendizado por esse imenso mar de possibilidades.

Forma Aparente e Forma Oculta

Tendo estudado a interface tanto sob sua forma aparente quanto através da estrutura que a conforma, estudaremos algumas questões relacionadas à divisão do trabalho, que ocorre na produção dela.

Nesse ponto, daremos atenção especial para o papel do designer neste processo produtivo. O designer não precisa ter domínio total de programação, mas é interessante que ele tenha, pelo menos, alguma noção da organização geral do sistema; do contrário, ele é como um arquiteto antes dos modernos: um mero desenhista de fachadas. Nesse capítulo trataremos também das possibilidades de abordagem com relação ao código: escrevê-lo a mão ou utilizar um editor visual.

Numa segunda parte, estudaremos a relação entre produção, consumo e produssumo na rede, que permite vários níveis de uso. Começaremos tratando dos artistas que usam o código como meio, passando então para questões surgidas em decorrência da popularização e difusão do que se chama de web 2.0, a web dinâmica criada a partir de conteúdo gerado por usuários, que cria novas relações de produção e consumo na rede. São ferramentas que permitem a organização de comunidades virtuais, de novas formas de comunicação, de propagação de material produzido, e também de larga difusão de produtos culturais oriundos de outros meios de comunicação e de outros tempos históricos.

Por fim, concluiremos a dissertação com considerações sobre esse novo campo disciplinar, relacionando autores e disciplinas que ela engloba, e dando uma perspectiva de como o design gráfico pode ser inserido dentro dele. A rede é um espaço muito amplo, e com múltiplos aspectos e consequências sociais. Um estudo aprofundado sobre ela não pode ser fruto de uma pesquisa individual, mas deve ser resultado de esforços de uma série de pesquisadores, em uma série de campos diferentes do conhecimento humano.

O desejo que moveu essa pesquisa era o de conhecer mais a fundo o universo desse meio, e por isso optamos por não manter um foco muito definido, e sim pairar por um amplo espectro de possibilidades. A consequência dessa abordagem é que nenhum tema tratado se esgota nessa pesquisa, mas ao mesmo tempo, ela dá indícios de vários temas que poderão ser tratados mais profundamente por qualquer pesquisador que se sinta estimulado em explorar este tema tão amplo.

2

Uma breve história da web

2.1

Do computador como ferramenta militar, ao computador como amplificador do intelecto humano

No começo, era a balística

É um fato amplamente reconhecido que a necessidade massiva de destruição de meios de produção, causada pela guerra, é um processo contraditório que acaba por ocasionar um desenvolvimento dos meios de produção. Pode se dizer que o computador, peça chave no atual momento do nosso desenvolvimento histórico, foi fruto desse desenvolvimento contraditório.

A Primeira Guerra Mundial propiciou o surgimento da balística, que segundo Howard Rheingold¹ é "a ciência matemática da predição do caminho de um projétil do momento de seu lançamento até o momento de atingir o seu alvo". Ela leva em conta uma série de cálculos complexos, para calcular as trajetórias levando em consideração fatores ambientais como vento, velocidade do alvo e velocidade do veículo lançador. Quanto mais precisos esses cálculos, maior a precisão e maior a distância que pode causar destruição.

Naquele momento, os computadores ainda eram seres humanos. Assim eram chamados os funcionários especializados em realizar esses complexos cálculos durante a primeira Guerra. Com o aumento da complexidade, e da demanda por esses cálculos, houve um direcionamento da ciência no sentido de desenvolver métodos de mecanização dessa atividade.

Toda a Revolução industrial foi baseada neste mesmo processo: o processo de transformação do trabalho humano em máquina, ou, para utilizar um conceito de Marx², cristalização do trabalho vivo em trabalho morto. O computador, a ferramenta do século XX é, de uma certa

1 RHEINGOLD, Howard. Tools for thought. Howard Rheingold, 1985. Disponível em: <http://www.rheingold.com/texts/tft/>

2 MARX, Karl - O capital. Crítica da economia política. São Paulo, Abril Cultural.

Abaixo, computadoras humanas:

Some NACA Muroc personnel with snowman

In National Advisory Committee for Aeronautics (NACA) terminology of 1946, computers were employees who performed laborious and time-consuming mathematical calculations and data reduction from long strips of records generated by on-board aircraft instrumentation. Virtually without exception, computers were female; at least part of the rationale seems to have been the notion that the work was long and tedious, and men were not thought to have the patience to do it. Though equipment changed over the years and most computers eventually found themselves programming and operating electronic computers, as well as doing other data processing tasks, being a computer initially meant long hours with a slide rule, hunched over illuminated light boxes measuring line traces from grainy and obscure strips of oscillograph film. Computers suffered terrible eyestrain, and those who didn't begin by wearing glasses did so after a few years. But they were initially essential employees at the Muroc Flight Test Unit and NACA High-Speed Flight Research Station, taking the oscillograph flight records and "reducing" the data on them to make them useful to research engineers, who analyzed the data.

<http://www.dfrc.nasa.gov/Gallery/Photo/People/Small/E49-0212.jpg>

maneira, o grande coroamento deste processo. A diferença essencial, neste caso, em relação a todos os processos semelhantes ocorridos anteriormente, de automatização de trabalhos humanos, é que o trabalho cristalizado na forma computador-máquina, não é um trabalho mecânico, mas sim um trabalho de raciocínio, um trabalho essencialmente intelectual, que inicialmente era somente o de calcular.

Aparatos voltados para o auxílio do cálculo existem desde 3000 aC, com a criação dos ábacos pelos Babilônicos. Os gregos também inventaram dispositivos mecânicos para o auxílio do cálculo do calendário lunar, mas as primeiras calculadoras mecânicas da civilização moderna foram desenvolvidas no século XVII, pelos filósofos Blaise Pascal, e posteriormente Gottfried Leibnitz³, que é considerado o inventor do cálculo e também foi quem introduziu a matemática binária. Esses dispositivos podem ser considerados de uma certa maneira como os primeiros passos dados pelo homem moderno no sentido de mecanizar atividades intelectuais, mesmo antes da eletricidade.⁴

No século XIX, um excêntrico visionário Chamado Charles Babbage começou o desenvolvimento de aparatos engenhosos, que seriam capazes de calcular diversas funções matemáticas⁵, e dispendeu fortunas no desenvolvimento desses equipamentos, que apesar de ainda mecânicos, eram capazes de calcular funções matemáticas mais complexas. Babbage, nas suas necessidades, ainda estava à frente das capacidades técnicas

3 Blaise Pascal criou em 1642 uma calculadora chamada de pascaline, que era capaz de realizar adições e subtrações, em 1674, Leibnitz, que também é considerado o inventor do cálculo, criou uma calculadora capaz também de adicionar e multiplicar

4 fonte: http://www.warbaby.com/FG_test/Timeline.html

5 Um primeiro dispositivo chamado de Difference Engine começa a ser desenvolvido em 1822, e em 1834, Babbage começa a trabalhar em um segundo equipamento, mais desenvolvido chamado Analytical Engine

De cima para baixo, Pascal e Pascaline, Leibnitz e sua calculadora, Charles Babbage e o Analytical Engine, e Ada Lovelace

da indústria da época, mas seus experimentos foram fundamentais no desenvolvimento posterior da computação. Babbage dotou sua máquina de um sistema de alimentação através de cartões perfurados⁶, que foi retomado posteriormente na computação moderna.

Um algoritmo é um procedimento que envolve um conjunto de operações para a resolução de um determinado problema, e é a raiz da computação. Ada Lovelace, companheira de Babbage, e que compreendia em essência o funcionamento do Analytical Engine, dedicou boa parte da sua vida a escrever séries de algoritmos para ele, em cartões perfurados, e reaproveitando determinados conjuntos de cartões, é considerada a primeira programadora da história.

Em 1854, George Boole, um jovem professor de matemática de fora do circuito acadêmico tradicional da época, publica um trabalho intitulado "*An investigation of the laws of thought, on which are founded the Mathematical Theories of Logic and Probabilities*". Sua idéia era entender como se dava o funcionamento do pensamento humano, mas o que ele fez foi criar uma lógica aritmética, que era também um método de colocar a lógica em um sistema formal, baseada em apenas dois estados, que ele chamava de "*o nada*" e "*o universo*", como explica Rheingold⁷:

6 Babbage tomou emprestada a idéia de alimentação por cartões perfurados do francês Jacquard que, segundo Rheingold, "who had triggered a revolution on the textile industry by inventing a mechanical method of weaving patterns in cloth" <http://www.rheingold.com/texts/tft/2.html>)

7 The concept Boole used to connect the two theretofore different thinking tools of logic and calculation was the idea of a mathematical system in which there were only two quantities, which he called "the Universe" and "Nothing" and denoted by the signs 1 and 0. Although he didn't know it at the time, Boole had invented a two-state system for quantifying logic that also happened to be a perfect method for analyzing the logic of two-state physical devices like electrical relays or vacuum tubes. Rheingold, Howard. Tools for thought. Howard Rheingold, 1985. disponível em: <http://www.rheingold.com/texts/tft/>

AN INVESTIGATION
OF
THE LAWS OF THOUGHT,
ON WHICH ARE FOUNDED
THE MATHEMATICAL THEORIES OF LOGIC AND
PROBABILITIES.

BY

GEORGE BOOLE, LL. D.
PROFESSOR OF MATHEMATICS IN QUEEN'S COLLEGE, CORK.

CONTENTS.

CHAPTER I.

NATURE AND DESIGN OF THIS WORK,.....1

CHAPTER II.

SIGNS AND THEIR LAWS, 17

CHAPTER III.

DERIVATION OF THE LAWS, 28

CHAPTER IV.

DIVISION OF PROPOSITIONS,.....37

CHAPTER V.

PRINCIPLES OF SYMBOLIC REASONING, 48

CHAPTER VI.

OF INTERPRETATION, 59

CHAPTER VII.

OF ELIMINATION, 74

OF REDUCTION, 87

O conceito que Boole usou para ligar as duas até então diferentes ferramentas de pensamento, a lógica e o cálculo foi a idéia de um sistema matemático no qual havia apenas duas quantidades, o que ele chamou de “O Universo” e “Nada” e denotado por sinais de 1 e 0 . Embora ele não sabia disso na época, Boole inventou um sistema de dois estados para a quantificação lógica que também passou a ser um método ideal para analisar a lógica de dispositivos físicos de dois estados como relés eléctricos ou tubos de vácuo.

Apesar de ter uma importância histórica enorme, por conseguir formalizar a ciência da lógica — que até então era baseada em silogismos — Boole não foi reconhecido no seu tempo, e demorou quase um século até que Claude Shannon, que é também conhecido como o fundador da “Teoria da Comunicação”, descobrisse que essa lógica simbólica desenvolvida por Boole era perfeita também para construir sistemas lógicos a partir de circuitos eléctricos — abrindo campo para a computação digital.

Alan Turing

A próxima contribuição essencial no desenvolvimento da computação foi dada por outro jovem, Alan Turing, que, como aponta Howard Rheingold, “resolveu um dos problemas mais cruciais da era moderna”, e neste processo “criou as bases teóricas para a computação:

Em 1936, com 24 anos, Alan M. Turing estabeleceu-se como uma das maiores prodígios matemáticos de todos os tempos, quando ele apontou para os colegas de que era possível executar cálculos na teoria dos números por meio de uma máquina — uma máquina que incorporou as regras de um sistema formal. Apesar de a máquina em si não existir como um modelo de trabalho, Turing enfatizou desde o início que essas máquinas poderiam realmente construídas. Sua descoberta foi um marco no esforço de formalizar a matemática

e, ao mesmo tempo, um divisor de águas na história da computação.⁸

A máquina de Turing, como ficou conhecida sua proposta, junto à redescoberta da lógica Booleana por Shannon em 1937, abriu o caminho para a criação de uma máquina que pudesse simular alguns dos processos do pensamento humano, ainda segundo Rheingold:

Se a lógica formal era o sistema que mais se aproximava das operações da razão humana, e se as tabelas de Boole da verdade poderia simular um sistema formal de raciocínio, então o uso de tabelas de verdade como tabelas de “instrução” que Turing discutia, e com dispositivos de comutação como relés para representar os estados “das máquinas (ou de células da fita), seria possível construir circuitos elétricos que poderiam simular algumas das operações lógicas do pensamento humano.⁹

Mas foi a eclosão da Segunda Guerra Mundial o fato que estimulou o primeiro grande salto rumo ao computador digital:

O surgimento da computação digital, baseada nos princípios da máquina de Turing, foi estimulada pela Segunda Guerra Mundial, que ainda estaria a quatro anos no futuro. Em

8 In 1936, at the age of twenty-four, Alan M. Turing established himself as one of the greatest mathematical prodigies of all time when he pointed out to his colleagues that it was possible to perform computations in number theory by means of a machine — a machine that embodied the rules of a formal system. Although the machine itself didn't exist as a working model, Turing emphasized from the beginning that such machines could actually be built. His finding was a milestone in the effort to formalize mathematics and, at the same time, a watershed in the history of computation. Op cit pg: <http://www.rheingold.com/texts/tft/3.html>

9 If logic was the formal system that most closely matched the operations of human reason, and if Boole's truth tables could embody such a formal system of simulated reasoning, then the use of truth tables as the “instruction tables” Turing discussed, and with switching devices like relays to represent the “states” of the machines (or the cells of the tape), it would be possible to build electrical circuits that could simulate some of the logical operations of human thought. Op. cit.

O Colossus, considerado o Primeiro computador

1936, Claude Shannon tinha ainda que descobrir que a álgebra inventada por George Boole para formalizar operações lógicas era idêntica com a matemática utilizada para descrever os circuitos de computação. John von Neumann e seus colegas tinham ainda que elaborar o conceito de programação armazenada. Norbert Wiener não havia formalizado a descrição de circuitos de feedback em sistemas de controle. Vários desenvolvimentos cruciais da eletrônica estavam ainda por vir." ¹⁰

A segunda Guerra Mundial

Na Inglaterra, em 1940, o governo chamou Alan Turing para participar de um projeto secreto, chamado de *Ultra*, que visava decodificar o código gerado por uma máquina chamada "*Enigma*", que criptografava toda a comunicação via rádio do exército alemão. O trabalho ultra secreto desse laboratório, chamado *Bletchley Park*, que a partir de 1943 já começou a decodificar o tido como inviolável código de criptografia alemão, foi essencial para a vitória dos aliados.

E enquanto lá Turing e seus companheiros desenvolviam o *Colossus*, computador que quebrou o código nazista, nos Estados Unidos também se fazia um movimento no sentido de contruir

máquinas para cálculos de balística.¹¹ A IBM começou vendendo máquinas para tabulação de dados para a indústria e o comércio, e com a segunda guerra, também passou a colaborar com a marinha americana no desenvolvimento de computadores voltados para a execução de cálculos de balística. Em 1944, um computador para uso geral chamado *Mark 1* baseado em válvulas, estava já em funcionamento, e em 1943, o exército começou o projeto de construção do primeiro computador totalmente eletrônico, o ENIAC, guiado por John von Neumann.

ENIAC era monstruoso — 100 metros de comprimento, 10 metros de altura, 3 metros de profundidade, pesando 30 toneladas — e quente o suficiente para manter a temperatura da sala em cerca 120 graus F, enquanto calculava equações diferenciais multivariáveis por meio de suas mais de 17.000 válvulas, 70.000 resistores, 10.000 capacitores e 6.000 interruptores de mão. Utilizava-se uma quantidade enorme de energia — a história apócrifa é que as luzes de Filadélfia se esmaeceram quando quando ele foi ligado.

¹²

O fim da guerra, com todo o desenvolvimento obtido neste processo, tornou possível o início da

¹⁰ The emergence of the digital computer, based on the principles of Turing's machine, was stimulated by World War II, which was still four years in the future. In 1936, Claude Shannon had yet to discover that the algebra invented by George Boole to formalize logical operations was identical with the mathematics used to describe switching circuits. John von Neumann and his colleagues had yet to devise the concept of stored programming. Norbert Wiener hadn't formalized the description of feedback circuits in control systems. Several crucial electronic developments were yet to come. Op cit.

¹¹ Mesmo antes da guerra, já haviam sido significativos avanços em direção à computação também nos Estados Unidos. Hollerith, por exemplo, um funcionário do escritório do censo americano, havia desenvolvido em 1890 uma máquina capaz de automatizar cálculos a partir de uma alimentação feita por um sistema de cartões perfurados. A partir de seu invento, que tornou possível a contagem da crescente população americana em dois anos, ele fundou o que viria a ser uma das maiores companhias de tecnologia do século XX, a Tabulating Machine Company, que posteriormente se transformou na International Business Machines, hoje conhecida pela familiar sigla IBM.

¹² ENIAC was monstrous — 100 feet long, 10 feet high, 3 feet deep, weighing 30 tons — and hot enough to keep the room temperature up toward 120 degrees F while it shunted multivariable differential equations through its more than 17,000 tubes, 70,000 resistors, 10,000 capacitors, and 6,000 hand-set switches. It used an enormous amount of power — the apocryphal story is that the lights of Philadelphia dimmed when it was plugged in. Op. cit, pg <http://www.rheingold.com/texts/tft/4.html>

Uma das salas de controle
do ENIAC

construção das primeiras máquinas de Turing. A partir desse momento, essa história passa a ter muito mais agentes e detalhes do que os que estão sendo expostos até aqui, mas não nos interessa entrar nesses meandros¹³. Mas é importante notar que essa primeira geração de computadores eram máquinas muito grandes, formadas por dezenas de milhares de válvulas, e que dispendiam muita energia. O tamanho e o custo elevado desses primeiros computadores eletrônicos limitava extremamente seu uso, como aponta Rheingold:

Poucos dos pioneiros da computação pessoal que ainda trabalham na indústria de computadores podem se lembrar do nascimento e do sonho, quando a noção de computação pessoal era uma heresia obscura nas fileiras do sacerdócio da computação. Trinta anos atrás, a esmagadora maioria das pessoas que concebiam, fabricavam, programavam, e usavam computadores compartilhavam de uma única idéia sobre o lugar adequado (e possível) dos computadores na sociedade: “computadores são dispositivos misteriosos concebidos para ser utilizados em cálculos matemáticos”. Ponto. Acreditava-se que tecnologia informática era muito frágil, preciosa e complicada para os não-especialistas.

Em 1950, você poderia contar as pessoas que pensavam uma excessão para esse dogma dogma nos dedos de uma mão. O ponto discordante da opinião partilhada por aquelas poucas pessoas envolvidas em uma maneira diferente de pensar sobre como os computadores podem ser usados. Os ponto de vista compartilhado entre essas pessoas envolvidas em um pensamento diferente sobre como os computadores podiam ser usados. Os dissidentes compartilhavam uma visão de computação pessoal na qual os computadores seriam usados para melhorar os aspectos mais criati-

13 para isso, fica a sugestão do livro de Roward Reinghol, *tools for thought*, disponível em <http://www.rheingold.com/texts/tft/>

vos da inteligência humana — para todos, não apenas os *technocognoscenti*.¹⁴

O caminho rumo à miniaturização só foi possível de ser trilhado com a invenção do transistor, nos anos 50, e a invenção posterior dos circuitos integrados, que culminou três décadas mais tarde no computador pessoal. Mas esse caminho não foi somente um desenvolvimento técnico em direção à uma portabilidade, mas envolveu também uma mudança de visão do papel que o computador poderia cumprir na sociedade, e, nesse sentido, alguns agentes foram fundamentais para isso.

As we may think

Vannevar Bush, cientista militar de alta patente do exército americano é um desses agentes. Bush escreveu um texto chamado "*As we may think*", em julho de 1945, quando a Segunda Guerra estava por acabar. Nele, Bush ressalta os grandes desenvolvimentos ocorridos em vários campos da ciência por ocasião da guerra, nas suas próprias palavras: O mundo chegou a uma era de dispositivos complexos baratos de de grande confiabilidade; e algo deverá surgir disto¹⁵.

14 A few of the pioneers of personal computing who still work in the computer industry can remember the birth and the dream, when the notion of personal computing was an obscure heresy in the ranks of the computing priesthood. Thirty years ago, the overwhelming majority of the people who designed, manufactured, programmed, and used computers subscribed to a single idea about the proper (and possible) place of computers in society: "computers are mysterious devices meant to be used in mathematical calculations." Period. Computer technology was believed to be too fragile, valuable, and complicated for nonspecialists. In 1950 you could count the people who took exception to this dogma on the fingers of one hand. The dissenting point of view shared by those few people involved in a different way of thinking about how computers might be used. The dissenters shared a vision of personal computing in which computers would be used to enhance the most creative aspects of human intelligence — for everybody, not just the technocognoscenti. Op. cit.

15 The world has arrived at an age of cheap complex devices of great reliability; and something is bound to come of it. Bush, 1945 pgXXX

Diante de tantos avanços, ele então propõe caminhos para a pesquisa científica no pós guerra no sentido de direcionar os grandes desenvolvimentos alcançados para busca fomentar a ampliação do conhecimento humano de modo geral.

Um desses caminhos, seria um dispositivo hipotético que ele chamou de "*memex*", um equipamento do tamanho de uma mesa, que seria capaz de armazenar o conteúdo de uma biblioteca em rolos de microfilme, e faria o trabalho lógico de pesquisar conteúdo nessa gigante biblioteca miniaturizada. Por essa proposta, que em conceito se assemelha muito à web que nós temos hoje, Bush é considerado por muitos como o pai do hipertexto.

Engelbart, e o Augmentation Research Center (ARC)

Douglas Engelbart era ainda técnico de radar quando leu o artigo de Vannevar Bush em 1945, ao voltar da guerra. E "infectado pela idéia da criação de uma ferramenta para a extensão da mente"¹⁶, começou a desenvolver "suas idéias de como utilizar máquinas para ampliar a inteligência humana", procurando, na verdade, um objetivo maior para a sua própria vida, como aponta Rheingold:

Engelbart percebeu, assim como Vannevar Bush, que a humanidade estava entrando em uma era em que a complexidade e a urgência dos problemas globais estavam superando as ferramentas dispendiosas de tempo para lidar com os problemas. (...) A tarefa mais importante não residia na concepção de novas formas de expandir nossa acumulação de conhecimento, mas em saber onde procurar as respostas que já estavam armazenados em algum lugar. "Se você pode melhorar a nossa capacidade de lidar com problemas complicados, você fez um impacto significativo na

16 Rheingold <http://www.rheingold.com/texts/tft/9.html>

ajuda à humanidade. Esse era o tipo de retorno que eu queria, e foi isso que me propus a fazer”.¹⁷

Por sua experiência com radares, ele sabia que se um computador poderia imprimir informações em cartões perfurados ou em papel, ele poderia também desenhar informações numa tela. E começou a reascunhar um sistema como o que Bush havia sugerido. Ele queria utilizar os computadores para ajudar as pessoas a aprender, queria que as pessoas pudessem interagir com eles, utiliza-lo como uma ferramenta para a ampliação do intelecto. Suas idéias não foram muito bem recebidas num meio onde o computador era considerado uma ferramenta complexa e altamente especializada, mas depois de mais de uma década de pesquisa, ele conseguiu publicar um artigo — “*A Conceptual Framework for the Augmentation of Man's Intellect*”— que como Rheighold¹⁸ aponta, era um “manifesto para o lançamento de um novo campo do saber”, como fica aparente neste trecho de sua introdução:

Por “Ampliação do intelecto humano” nós queremos dizer: aumentar a capacidade de um homem se aproximar de uma situação problemática complexa, e ganhar compreensão para atender suas necessidades específicas, e obter soluções para seus problemas. Aumento da capacidade nesse respeito, entende-se por uma mistura do seguinte: que a compreensão pode ser obtida mais rapidamente; que uma melhor compreensão pode ser adquirida; que um grau útil de compreensão pode ser adquirida

em áreas onde anteriormente a situação era muito complexa, que as soluções podem ser produzidas mais rapidamente, melhor que as soluções podem ser produzidos; que as soluções podem ser encontradas anteriormente um ser humano não poderia encontrar nenhuma. E por “situações complexas” incluímos os problemas profissionais dos diplomatas, executivos, cientistas sociais, cientistas biológicos, advogados, designers — quer a situação problemática exista por vinte minutos ou vinte anos. Nós não falamos de truques inteligentes isolados que ajudariam em situações particulares. Nós nos referimos a um modo de vida em um domínio integrado onde palpites, recortar e tentar, problemas intangíveis, e “sensação de uma situação” humana útil coexistem com conceitos poderosos, a tecnologia e notação simplificada, métodos sofisticados e de ajuda eletrônicos de alta potência.¹⁹”

Entre essas ferramentas de ajuda à ampliação do conhecimento humano estava uma proposta relativamente simples, mas que teve consequências avassaladoras quando implantada em larga escala: um processador de texto. Ele sugeria que o computador poderia ser utilizado para a automatização de tarefas de relativamente baixo nível, como a datilografia, por exemplo, e que através disso todo o sistema poderia ser alavancado.

19 By “augmenting man’s intellect” we mean increasing the capability of a man to approach a complex problem situation, gain comprehension to suit his particular needs, and to derive solutions to problems. Increased capability in this respect is taken to mean a mixture of the following: that comprehension can be gained more quickly; that better comprehension can be gained; that a useful degree of comprehension can be gained where previously the situation was too complex; that solutions can be produced more quickly; that better solutions can be produced; that solutions can be found where previously the human could find none. And by “complex situations” we include the professional problems of diplomats, executives, social scientists, life scientists, attorneys, designers — whether the problem situation exists for twenty minutes or twenty years. We do not speak of isolated clever tricks that help in particular situations. We refer to a way of life in an integrated domain where hunches, cut-and-try, intangibles, and the human “feel for a situation” usefully coexist with powerful concepts, streamlined technology and notation, sophisticated methods, and high-powered electronic aids. Op cit.

17 Engelbart realized, as had Vannevar Bush, that humankind was moving into an era in which the complexity and urgency of global problems were surpassing time-honored tools for dealing with problems. (...) The most important task no longer lay in devising new ways to expand our accumulation of knowledge, but in knowing where to look for the answers that were already stored somewhere. “If you can improve our capacity to deal with complicated problems, you’ve made a significant impact on helping humankind. That was the kind of payoff I wanted, so that’s what I set out to do”. Op. Cit.

18 Op. Cit

O primeiro mouse

Douglas defendia a idéia de mobilidade, de arranjo e de manipulação de símbolos pelos usuários e propôs a criação de um laboratório dedicado à pesquisa para a ampliação do intelecto humano. Inesperadamente, recebeu uma ajuda da Advanced Research Project Agency (ARPA)²⁰ para o financiamento do projeto. Em 1968, o Augmentation Research Center (ARC), havia produzido o suficiente para uma demonstração pública, e seu time juntou o melhor do equipamento multimídia disponível na época para pro-

20 A ARPA era uma agência do governo americano criada em 1957 em resposta ao lançamento do primeiro satélite russo, o sputnik. Mais adiante falaremos mais da ARPA e de seu papel na fundação da internet

duzir aquilo que até hoje é conhecida como "a mãe de todas demonstrações".

The mother of all demos

Diante de uma "platéia heterogênea de matemáticos diletantes e hippies no San Francisco Civic Auditorium"²¹, Douglas Engelbart fez uma apresentação que até hoje impressiona. Utilizando recursos de multimídia, e uma tela de projeção aonde alternavam e se sobrepunham uma câmera que filmava suas ações e a tela de seu computador, ele fez a demonstração de de um sistema que transformou toda a trajetória da computação. Ele tinha uma tela, que era conectada ao seu computador central através de uma rede de micro-ondas, um teclado convencional, um teclado de cinco teclas e um "mouse". Com essa ferramenta totalmente inovadora, Engelbarth percorria a tela do computador, que podia ser dividida em várias "janelas", e cada uma delas podia apresentar um texto ou uma imagem. Com isso, ele criava o espaço informacional — a interface contemporânea, nas palavras de Bill Moggridge²²:

Esta foi a demonstração que mudou o mundo. A comunidade científica da computação passou do ceticismo à ovação de pé em uma hora e meia, e a idéia de manipulação direta de uma interface gráfica com o usuário ficou alojada na consciência comum. (...)

A demo foi verdadeiramente surpreendente, provando que a computação interativa poderia ser usada para manipulações em tempo real de informações de formas que muito poucas pessoas tinham imaginado antes.²³

21 Johnson, Steve, Cultura da Interface

22 Moggridge, Bill. Designing Interaction. Massachusetts: MIT Press, 2007

23 This was the demo that changed the world. The computer science community moved from skepticism to standing ovation in an hour and a half, and the ideas of direct manipulation of a graphical user interface became lodged in the communal consciousness. (...) The demo was truly amazing, proving that interactive computing could be used for real-time manipulations of information in ways that very few people had imagined before. Moggridge, 2007 pg 34

O que ele criou foi um mapeamento da tela em bits, e transformou a tela, que tradicionalmente era um espaço linear, com informações textuais, em um espaço bidimensional que poderia, através do mouse, ser manipulado e conquistado pelo usuário, como aponta Steve Johnson em seu livro "Cultura da Interface":

A grande investida de Engelbart envolveu o princípio de manipulação direta. Representar um documento de texto como uma janela ou um ícone era uma coisa, mas, a menos que o usuário tivesse algum controle sobre essas imagens, a ilusão seria remota, pouco convincente, como um filme projetado a poucos fotogramas por segundo. Para que a ilusão de espaço-informação funcionasse, devíamos poder sujar as mãos, mexer as coisas de um lado para outro, fazer coisas acontecerem. Foi aí que entrou a manipulação direta. Em vez de teclarem comandos obscuros, o usuário podia simplesmente apontar alguma coisa e expandir seus conteúdos, ou arrastá-las através da tela. Em vez de dizer ao computador para executar uma tarefa específica — "abra este arquivo" —, os usuários pareciam fazê-lo eles próprios. A manipulação direta tinha uma qualidade estranhamente paradoxal: na realidade, a interface gráfica havia acrescentado uma outra camada entre o usuário e sua informação. Mas a imediatez tátil da ilusão dava a impressão de que agora a informação estava mais próxima, mais à mão, em vez de mais afastada. Sentíamos que estávamos fazendo alguma coisa diretamente com nossos dados, em vez de dizer ao computador que a fizesse por nós. (pgs 21-22)

Com esses dois novos paradigmas: o espaço informacional, e a manipulação direta, é aberto o caminho para a transformação do computador, de uma ferramenta para um meio de comunicação.

ARPAnet

Trechos da "mother of all demos, de Douglas Engelbarth

J. R. C. Licklider era um pesquisador cuja principal área de interesse era a psicoacústica, e, com o fim da guerra, foi atraído para o MIT para um grupo de pesquisas que procurava criar mecanismos eletrônicos de representação do cérebro humano²⁴. Nesse processo, ele percebeu que a maior parte do seu tempo de trabalho era dispendido em procurar informação ao invés de em digerí-la. Ele chegou à conclusão de que essas tarefas, que tomavam a maior parte do tempo dos pesquisadores, poderiam ser melhor executadas por máquinas, como aponta Rheingold²⁵.

A preocupação dele não era com processamento de dados, que poderiam ser feitos pelos programas e computadores que a IBM já oferecia na época; ele acreditava que os computadores poderiam ajudar os cientistas a formular e a desenhar "modelos", como os necessários para o desenvolvimento de aeronaves. Trabalhando em uma empresa de consultoria,²⁶ ele passou a pesquisar novas formas de interação com a máquina, e percebeu que ao invés de alimentar o computador através de cartões perfurados, era possível alimentá-lo através de uma fita magnética, que poderia ser alterada durante a execução de um determinado programa. Com isso, ele cria a possibilidade de computação interativa, e, com ela, ele começou a acreditar que o papel do computador na sociedade poderia ser transformado, como aponta essa passagem do texto de Rheingold²⁷:

Embora de estar convencido por sua “conversão religiosa para computação interativa” — uma frase que tem sido usado vez ou outra

O Projeto SAGE, coordenado por Licklider, precursor da computação interativa

24 Num movimento inspirado pelo conceito de cibernética, proposto por Norbert Wiener.

25 Rheingold, 1985 pg <http://www.rheingold.com/texts/tft/7.html>

26 Com um modelo de computador de aproximadamente 250.000 dólares, que era considerado um mini-computador na época.

27 Ibid.

por aqueles que participaram nos acontecimentos que se seguiram — Licklider ainda sabia muito pouco sobre a economia da tecnologia da computação para ver como ela poderia tornar realmente possível a construção de um assistente de laboratório inteligente. Embora ele não sabia como ou quando os computadores se tornariam suficientemente poderosos e baratos o suficiente para servir como “ferramentas para o pensamento”, ele começou a perceber que o computador de uso geral, se se pudesse ser criado de forma com que os seres humanos podusessem interagir diretamente, poderia evoluir para algo completamente diferente dos processadores de dados e números dos anos 1950. Embora a possibilidade de criar uma ferramenta pessoal parecia ainda economicamente inviável, a idéia de modernizar recursos comunais, como uma biblioteca, começou a ser interessante para ele. Ele foi incendiado pela a idéia que Vannevar Bush havia mencionado em 1945, o conceito de um novo tipo de biblioteca para englobar o novo sistema mundial de conhecimento.²⁸

Essas pesquisas culminaram em uma nova forma de computação, que envolvia uma “simbiose homem-computador”, os computadores compartilhados, com acesso remotos por meio de terminais, e o processamento tipo “batch”, ou via linha de comando, que permitia uma intera-

28 Although he was convinced by his “religious conversion to interactive computing” — a phrase that has been used over and over again by those who participated in the events that followed — Licklider still knew too little about the economics of computer technology to see how it might become possible to actually construct an intelligent laboratory assistant. Although he didn’t know how or when computers would become powerful enough and cheap enough to serve as “thinking tools,” he began to realize that the general-purpose computer, if it was set up in such a way that humans could interact with it directly, could evolve into something entirely different from the data processors and number crunchers of the 1950s. Although the possibility of creating a personal tool still seemed economically infeasible, the idea of modernizing a community-based resource, like a library, began to appeal to him. He got fired up about the idea Vannevar Bush had mentioned in 1945, the concept of a new kind of library to fit the world’s new knowledge system. Rheingold, op. cit.

ção em tempo real entre os programadores e o computador.

Em 1958, ao mesmo tempo em que o projeto SAGE²⁹ foi completado, os americanos receberam uma notícia que tornava obsoleto todo seu sistema de defesa anti-bombas: a Rússia havia lançado o “Sputnik”, seu primeiro satélite, e essa notícia fez desencadear a guerra-fria, uma disputa de décadas pela supremacia tecnológica militar, mesmo em tempos de paz. Diante disso, o Departamento de Defesa americano cria a ARPA, que como aponta Rheingold, tinha o objetivo de financiar projetos robustos, entre eles o

29 O Projeto SAGE era um projeto de interligação dos sistemas de radares americanos. Seu predecessor, o Whirlwind foi a primeira experiência de computação em tempo real.

ARC, de Engelbarth, que pudessem alavancar as tecnologias defesa americanas.

Os EUA queriam se concentrar na produção de micro-componentes eletrônicos, para produção de mísseis, e essa decisão ocorreu em um momento extremamente propício, uma vez que a eletrônica finalmente havia atingido o estágio de fabricação de circuitos integrados. Vários engenheiros de computação e pesquisadores migraram para essa nova agência, e entre eles, Licklider e seu projeto de "computação interativa":

Time-sharing was to be the first, most important step in the transition from batch processing to the threshold of personal computing (i.e., one person to one machine). The idea was to create computer systems capable of interacting with many programmers at the same time, instead of forcing them to wait in line with their cards or tapes.³⁰

EM 1962, ele escreveu uma série de memorandos a respeito de um conceito de "Galactic network", uma rede globalmente interconectada onde qualquer um poderia rapidamente acessar dados e programas de qualquer lugar. Esse conceito deu origem à ARPAnet, criada em 1966, e com ela, foi possível a invenção do correio eletrônico, o e-mail.

A ARPAnet foi desenhada para sobreviver a quedas da rede, e interligava redes de pesquisa nos Estados Unidos. Posteriormente essa rede pioneira se converteu na internet que utilizamos até hoje.

30 Op. cit.

2.2

O computador pessoal e a interface gráfica

Dez anos após a apresentação do primeiro artigo sobre a ampliação do conhecimento humano, Engelbart, junto com outros dois companheiros de laboratório, Watson e Norton, apresentaram um novo artigo intitulado "*The Augmented Knowledge Workshop*". Nele, enfatizavam que o trabalho intelectual e o conhecimento ganhavam cada vez mais importância entre os demais trabalhos sociais existentes. Eles propunham o que ficou mais tarde conhecido como "o escritório do futuro", que seria um ambiente de trabalho totalmente desenhado para o "setor do conhecimento". Chegava a hora em que os desenvolvimentos conquistados no ARC deveriam ser economicamente viáveis para ser vendidos como equipamento de escritório.

O ARC foi, de 1963 até 1975, um ambiente fértil de novas idéias e tecnologias, mas em um determinado ponto, a ARPA decidiu reduzir drasticamente seu financiamento. No começo dos anos 70, alguns de seus membros deixaram o laboratório para trabalhar no PARC, o novo centro de pesquisa que a Xerox abriu, interessada em desenvolver uma nova idéia, que se tornara possível graças ao desenvolvimento das tecnologias de fabricação de circuitos impressos em larga escala: um computador poderoso que pudesse caber em uma mesa de escritórios. O Xerox PARC, era, nas palavras de Rheingold "a nova Meca para aqueles que viam o computador como uma ferramenta para a ampliação do intelecto humano". Entre eles, estava Bill English, parceiro de Engelbart desde os primórdios do ARC.

A oposição à guerra do Vietnã, e a maior militarização da ARPA, acabaram atraindo um certo número de pesquisadores ao Xerox PARC, como aponta Rheingold:

Em 1970, a combinação de uma crescente oposição à guerra do Vietnã, e a militarização de todas as pesquisas da ARPA, ocasionou que

uma extraordinária coleção de talentos nas novas áreas de redes de computadores e computação interativa passou a procurar pastos mais verdes, num momento em que uma corporação decidiu para fornecer os mais verdes pastos imagináveis.

Em 1969, Peter McColough, CEO da Xerox Corporation, anunciou sua intenção de fazer da Xerox “a arquiteta de informação” para o futuro. Para esse fim, uma organização de pesquisa foi montada em Palo Alto, na década de 1970. McColough pos um homem chamado George Pake como responsável. Uma das primeiras coisas Pake fez foi contratar o melhor visionário de longo-prazo, organizador de pesquisa e coletor de pessoas que ele poderia encontrar — Bob Taylor.¹

Bob Taylor havia sido diretor do Escritório de Técnicas de Processamento de Informação da ARPA, onde foi responsável pela criação da ARPAnet, a rede que foi precursora da internet.

Se alguma vez houve um ambiente modelo para uma vanguarda tecnológica do “setor de ponta do conhecimento”, o PARC era ele. De físicos nos laboratórios de laser e do engenheiro-artesãos nas fábricas de microchips personalizados aos projetistas de linguagem de computador, programadores de inteligência artificial, fisiologistas cognitivos, VJs, engenheiros de som, maquinistas, bibliotecários, secretários, cozinheiros, faxineiros e guardas

¹ In 1970, a combination of growing opposition to the Vietnam war, and the militarization of all ARPA research, meant that an extraordinary collection of talent in the new fields of computer networks and interactive computing were looking for greener pastures at a time when one corporation decided to provide the greenest pastures imaginable.

In 1969, Peter McColough, CEO of Xerox Corporation, announced his intention to make Xerox “the architect of information” for the future. To that end, a research organization was assembled in Palo Alto, in the early 1970s. McColough put a man named George Pake in charge. One of the first things Pake did was hire the best long-term computer visionary, research organizer, and people-collector he could find — Bob Taylor. Op. cit, pg <http://www.rheingold.com/texts/tft/10.html>

de segurança, você tem um agradável, modelo quase utópico de todo mundo que encontrou.²

Até então, o modelo de computador utilizado nesses laboratórios de pesquisa era o de "time sharing", um computador central de uso compartilhado, acessível por terminais individuais; mas havia uma pequena quantidade de usuários que estava de certa maneira guiada pelo desejo de ter uma espécie de computador de uso pessoal. O uso pessoal não era só um desejo individual, mas uma forma de possibilitar também a execução de softwares mais complexos, uma vez que no modelo anterior a capacidade de processamento do computador era dividida entre os vários usuários que o utilizavam.

Até o começo dos anos 60, os computadores eram somente processadores de dados. Um programador escrevia o programa e o convertia em uma série de cartões perfurados, entregava o programa para o administrador do computador, e buscava o resultado um certo tempo depois. Os computadores de tempo compartilhado foram um extremo avanço nesse sentido, pois permitiam o acesso direto aos programadores, por meio de um terminal aonde era possível se executar comandos. Os editores de texto disponíveis eram voltados para a manipulação via teclado. O mouse, criado pelo ARC, foi a primeira ferramenta que permitiu criar uma extensão da mão, e as janelas, uma extensão do espaço físico para dentro do computador.

2 If there was ever a model environment for the technological cutting edge of the "knowledge sector," PARC was it. From the physicists in the laser laboratories and the engineer-artisans in the custom microchip shops to the computer language designers, artificial intelligence programmers, cognitive physiologists, video jockeys, sound engineers, machinists, librarians, secretaries, cooks, janitors, and security guards, you got a nice, model-utopian feeling from everybody you encountered. Op. Cit.

Em 1972, engenheiros e pesquisadores do laboratório de ciência dos computadores da Xerox em Palo Alto, o Xerox PARC, discutiam ainda o legado das janelas de Engelbart, do mapeamento de bits e do mouse, e entre eles, havia um pesquisador específico chamado Alan Kay, que estava dedicado a implementar um sistema de janelas no SRI³. Mas as janelas do SRI eram precárias e não se sobrepunham, e acabavam competindo pelo espaço pequeno da tela. Já na Xerox, com mais recursos disponíveis, Kay conseguiu desenvolver um modelo de janelas que se sobrepunham, que até hoje é o modelo conceitual da maioria das interfaces utilizadas nos computadores pessoais, quer nos sistema operacional Window, no OS da Apple, como nas interfaces gráficas de Linux e Unix, como aponta Johnson:

Engelbart e Sutherland haviam dotado o computador digital de espaço; as janelas sobreponíveis de Kay lhe deram profundidade. Podíamos entrar e sair da paisagem da tela, puxar coisas na nossa direção ou afastá-las. A revolução do mapeamento de bits nos dera uma linguagem visual para a informação, mas as pilhas de papel de Kay sugeriam uma abordagem mais tridimensional, um espaço-tela em que era possível entrar. Toda a idéia do computador como um ambiente, um mundo virtual, tem origem nessa inovação aparentemente modesta, embora fossem ser necessários muitos anos para que esse legado se tornasse visível

Alan Kay, nesse momento, já havia percebido o potencial do computador não só como uma ferramenta, mas como um meio de comunicação. Ele, como muitos outros pioneiros, haviam aprendido com McLuhan que os meios de comunicação eram tecnologias capazes de transformar todos os padrões de pensamento de uma

3 Vários dos pesquisadores do Xerox PARC haviam migrado do SRI

Demonstração do Sketchpad, de Ivan Sutherland, precursos dos programas de edição de gráficos

geração. Johnson cita uma frase de Alan Kay que expõe essa visão:

O computador é um meio de comunicação! Eu sempre tinha pensado nele como uma ferramenta, talvez como um veículo — uma concepção muito fraca ... Se o computador pessoal [era] um veículo totalmente novo, o próprio uso dele iria realmente mudar os padrões de pensamento de uma geração inteira.

O paradigma de interface gráfica até então, era um programa de computação gráfica que havia sido criado por Ivan Sutherland, chamado "*Sketchpad*":

Pessoas como Alan Kay ainda ficam animadas quando falam sobre Sketchpad: “o Sketchpad tinha aspectos maravilhosos, além do fato de que ele ter sido primeiro verdadeiro programa de computação gráfica. Não era apenas uma ferramenta para desenhar as coisas. Era um programa que obedecidas as leis que você gostaria que fossem verdadeiras. Então, para desenhar um quadrado no Sketchpad, você desenha uma linha com uma caneta óptica e dizia: ‘Copiar-copiar-copiar, anexar-anexar-anexar anexar. Esse ângulo é de 90 graus, e essas quatro coisas devem iguais.’ Sketchpad faria Zap! E você teria um quadrado. 4

Antes mesmo de entrar para o SRI, Kay havia se baseado na possibilidade que o *sketchpad* tinha de manipulação de objetos para criar a linguagem de programação LOGO, voltada para a educação infantil. Ele "queria criar um meio que fosse tanto um aplicador da fantasia como um amplificador do intelecto", nas palavras de Rheingold.

4 People like Alan Kay still get excited when they talk about Sketchpad: “Sketchpad had wonderful aspects, besides the fact that it was the first real computer graphics program. It was not just a tool to draw things. It was a program that obeyed laws that you wanted to be held true. So to draw a square in Sketchpad, you drew a line with a lightpen and said: ‘Copy-copy-copy, attach-attach-attach. That angle is 90 degrees, and these four things are to be equal.’ Sketchpad would go zap! and you’d have a square. Rheingold, pg (<http://www.rheingold.com/texts/tft/11.html>)

Com um tipo certo de linguagem de programação e um tipo certo de hardware, ele pensava que "um novo tipo de computador — um computador pessoal — poderia ser possível."⁵

E esse computador pessoal era o que vários pesquisadores do Xerox PARC estavam buscando. Para isso, desenvolveram uma linguagem de programação específica, chamada *smalltalk*, que incluía uma interface gráfica completa, com ícones,

5 Op. Cit.

O Xerox Alto, precursor dos computadores pessoais

menus, janelas e um dispositivo de apontamento. Para a criação da imagem na tela, eles desenvolveram uma técnica chamada mapeamento de bits:

um termo que significa que cada elemento da imagem, cada ponto de luz na tela, foi conectado a um bit de informação em um lugar específico na memória do computador, criando assim uma espécie de mapa informacional de sentido duplo na tela. ⁶

Foi lá também que surgiu a metáfora da escrivaninha — ou desktop — para o desenvolvimento dos ícones da interface gráfica. A simplicidade de uma representação 2D em ícones de elementos comuns de um escritório, como a lata de lixo, o arquivo de pastas, a impressora, foi essencial na conquista dos usuários, que só foi se dar mais tarde, sob o comando de outra companhia. A junção desses elementos, criou um paradigma que até hoje não foi superado, a interface gráfica.

Através da interface gráfica, o computador — uma máquina extremamente complexa, que funciona de forma totalmente misteriosa para os "homens normais" — podia ser usado pelas pessoas comuns, que se apoiam em metáforas do mundo real, do mundo dos escritórios, como o desktop, as pastas de arquivo, latas de lixo e coisas desse tipo. Através disso, a interface dá o controle sobre uma estrutura que no princípio não tem nenhuma relação com esses elementos. A interface gráfica nesse sentido é somente uma mediação que facilitou uma geração inteira de trabalhadores a adotar uma ferramenta totalmente nova.

6 a term that meant that each picture element, each dot of light on the display screen, was connected to one bit of information in a specific place in the computer's memory, thus creating a kind of two-way informational map of the screen. Op. Cit.

Apesar do Xerox PARC ter conseguido a criação de protótipos de computador pessoal com interfaces gráficas bastante desenvolvidas, eles nunca conseguiram dar saída comercial para seu produto. A popularização do computador pessoal aconteceu por um outro caminho

A Apple Company

Em 1975, o único computador pessoal disponível no mercado era o Altair 8800. Steve Wozniak era um jovem hacker que participava de um clube de computação no Vale do Silício. Era um ambiente fervilhante de idéias e interesses em torno da informática. Nesse ambiente, Wozniak desenhou o Apple I, que era um computador computador pessoal feito artesanalmente, "desenhado estritamente na base do hobby e da diversão"⁷, segundo suas próprias palavras, não para se tornar um produto. Ele fazia demonstrações no clube e recebia retorno de seus colegas, durante seu desenvolvimento. Utilizava um televisor comum e um teclado adaptado. Seu amigo de escola, Steve Jobs, propôs-lhe uma parceria para transformar o Apple I em um produto, e, no primeiro ano, eles construíam seus computadores numa garagem adaptada.

Wozniak gostava de enfatizar a idéia de que seu computador era construído com poucos chips, com poucos recursos. Ao longo do ano, eles conseguiram uma encomenda significativa, e Jobs conseguiu também pegar um empréstimo para fabricar essa primeira leva de computadores. Em pouco tempo, seus nomes já estavam estabelecidos, e conseguiram com um colega dinheiro para montar uma companhia e lançar o Apple II, que tinha gráficos coloridos e uma grande capacidade de expansão. Na primeira feira de computação da

O primeiro Apple, fabricado artesanalmente, e um anúncio do lançamento do Apple II

⁷ Wozniak, Stephen. Homebrew and How the Apple Came to Be, Digital Deli. Retrieved on 2007-03-02. http://www.atariarchives.org/deli/homebrew_and_how_the_apple.php

Costa Oeste, eles apresentaram comercialmente o Apple II e obtiveram um imenso sucesso comercial. Como ele mesmo cita, eles conseguiram trazer a revolução na computação para a casa das pessoas:

Naquela época, a pequena cena da computação era baseada na crença de que estávamos todos no topo de uma revolução. Todos os participantes do clube em 1975-76 sabiam que havia uma grande revolução da computação ocorrendo e o resto do mundo não estava ciente disso ainda. É por isso que havia tanta excitação e espírito. Estávamos indo finalmente ganhar controle sobre nossos próprios computadores. E não era uma coisa de milhões de dólares, que pertencia à empresa onde se trabalhava. Essa coisa gigante e tão valiosa, e que nós queríamos usar e controlar, estava finalmente chegando perto. Olhe como muitas empresas surgiram do nosso Homebrew Computer Club. Na última contagem, era algo como vinte e um! Nós conseguimos trazer a revolução do computador para casa.⁸

E ainda como aponta Bill Moggridge:

O Apple II foi um de uma enxurrada de máquinas amadoras que eram baratas o suficiente para tornar a computação acessível para o povo entusiasmado que queria escrever seus próprios programas e montar as peças de hardware eles próprios. A chave para que ele fosse uma máquina pessoal foi o preço, tornando-se disponível para pessoas que não pensam nele como uma ferramenta de trabalho, paga pela empresa ou universidade, mas

⁸ Back then, the small computer scene was based on the belief that we were all on top of a revolution. Everyone attending the club in 1975-76 knew there was a big computer revolution occurring and the rest of the world wasn't aware of it yet. That's why there was so much excitement and spirit. We were finally going to get control of our own computers. It wasn't a million-dollar thing that belonged to the company you worked for. This big thing that had so much value, and that we wanted to use and control, we finally were getting close to. Look at how many companies have sprung out of our Homebrew Computer Club. At last count, it was something like twenty-one! We managed to bring the computer revolution home. Wozniack, op. Cit.

Precursoras da Interface gráfica do Usuário, acima, o Xerox Star, abaixo, o OS da Apple

sim como um brinquedo para a noite e retoques finais de semana.⁹

A Xerox havia feito um grande investimento na criação de um computador pessoal, mas seu computador não conseguia lugar no mercado pois era muito caro comparado a esses novos competidores que haviam surgidos, como o Apple II e o Commodore¹⁰. Como eles não conseguiriam baratear seu custo de fabricação, a Xerox decidiu investir na Apple. O intuito era começar como investidor e então comprar a empresa, e no processo descobrir meios de baratear a produção de seus computadores, como aponta Bill Moggridge. Nesse processo, Steve Jobs conseguiu negociar também uma transferência de tecnologia do PARC para a Apple, levando consigo o mouse.

O próximo computador que a Apple desenvolveria, o Lisa, já iria ser dotado também de uma interface gráfica para o usuário, que também foi inspirada no *Smalltalk* e no *Star*, desenvolvidos pelo PARC. Entre as inovações da interface gráfica do Lisa estavam os menus no topo que permanecem até hoje nas interfaces dos sistemas operacionais da Apple. O Lisa, contudo, apesar de extremamente inovador, era caro e desengonçado, e apesar de ter ajudado a Apple crescer, seu sucesso não se compara ao sucesso de seu sucessor, o Macintosh.

Bill Atkinson

⁹ The Apple II was one of a flurry of hobbyist machines that were inexpensive enough to make computing accessible for the enthusiastic people who wanted to write their own programs and plug together the hardware themselves. The key to it being a personal machine was the price, making it available to people who did not think about it as a tool for work, paid for by the company or the university, but rather as a plaything for evening and weekend tinkering. Moggridge, Bill. *Designing Interaction*. Massachusetts: MIT Press, 2007

¹⁰ Enquanto um Apple II custava cerca de \$800 dólares, o Alto, fabricado pela Xerox não saía por menos de \$10.000.

O primeiro Macintosh

Parte das inovações feitas para o Lisa, como o gerenciador de janelas, o gerenciador de menus e o gerenciador de eventos, assim como os gráficos, foram escritos por um programador Chamado Bill Atkinson. Todas foram incorporadas no novo produto da empresa, que foi lançado apenas 9 meses depois. O Mac, como ficou conhecido, integrava essas inovações de design de interface a um design de hardware também inovador. Nesse momento, a IBM já havia lançado o PC, que funcionava com o sistema operacional Ms-DOS, licenciado pela Microsoft. O Mac prontamente se colocava em oposição ao PC, que rodava uma interface baseada em linha de comando. Johnson coloca que havia uma idéia de que o PC era um computador voltado para os homens de terno, o "Homem da Empresa", enquanto o Mac transmitia uma idéia de jovialidade, de expressão, de identidade visual, de criatividade. Algumas primeiras críticas foram feitas de que o Mac, com seus ícones, se tornava algo mais próximo a um brinquedo, que seria engraçadinho demais. Um editorial da revista Forbes de 1984 faz o seguinte comentário sobre o novo Macintosh:

As melhores características [do Macintosh] são para os novatos em computador: MacPaint, um programa que cria padrões gráficos de fantástica complexidade, e MacWrite, um programa de processador de palavras que chega ao engenhoso requinte de dispor a tela para se parecer com uma máquina de escrever. Ambos são controlados pelo "mouse" da máquina, que move o cursor sem que o usuário toque no teclado. Tamanha simplicidade não é o que almejam as grandes empresas. O executivo médio tem pouca necessidade da habilidade gráfica do MacPaint. A maioria deles já tem de penar o suficiente para escrever relatórios, sem ter de se preocupar também com a apresentação gráfica deles. (Steve Johnson)

Naquele tempo, tarefas como publicar um arquivo e fazer um cartaz tinham pouco de informatização. Já haviam alguns computadores que

faziam composição eletrônica, mas a grande maioria das publicações eram montadas manualmente, a partir de textos fotocompostos e letra-set, e exigiam um grande uso das habilidades manuais dos designers e ilustradores. O MacPaint, com sua facilidade de uso, foi um precursor do movimento que aconteceu com as artes gráficas nessas últimas décadas, com o avanço das ferramentas de computação nas artes gráficas, e sua consequente popularização, de um circuito muito reduzido de profissionais altamente especializados, para uma ampla utilização por diversos leigos e amadores numa escala massificada.

Outro aplicativo pioneiro desenvolvido pela companhia, foi o Hypercard, desenvolvido pelo mesmo criador do Mac Paint e da interface gráfica do Mac, Bill Atkinson. O Hypercard é especialmente importante aqui nessa pesquisa pois é, de uma certa forma, um dos precursores do Hipertexto. Ele consistia em um banco de dados em forma de cartões que poderia ser alimentado pelo o usuário. Esses cartões poderiam conter texto e imagem, bem como links para outros cartões, o que de certa forma lembra o funcionamento de um browser atual de internet, como fala o seu próprio criador, em entrevista para o livro *Designing Interactions*:

Sob muitos aspectos, o HyperCard foi o precursor do primeiro navegador da Web. Era como um navegador que foi acorrentado a um disco rígido, em vez de ser ligado à World Wide Web. Páginas na Web são como os cartões de uma pilha — todos eles tem elementos gráficos, texto e uma linguagem de script.¹¹

11 In many ways, HyperCard was the precursor to the first Web browser. It was like a Web browser that was chained to a hard disk, instead of connecting to the World Wide Web. Pages on the Web are like cards on a stack—they all have graphics, text, and a scripting language. Moggridge, pg 104

windows 83 pré-version

windows 1.x

windows 2.x

windows 3.11 fonte: <http://toastytech.com/guis/>

A própria Wikipédia, segundo entrevista dada pelo seu criador, foi baseada em um aplicativo feito para Hypercard. Neste aplicativo, cada cartão trazia informação sobre uma palavra-chave. Caso não houvesse ainda um cartão com aquela determinada palavra-chave, o hypercard criava um automaticamente.¹²

Nesse momento, os PCs da IBM já eram uma realidade, mas, rodando um sistema operacional baseado em linha de comando, estavam bem aquém em termos de desenvolvimento do pacote completo oferecido pela Apple.

A Microsoft, que licenciava o sistema operacional MS-DOS, correu atrás de produzir uma paródia do sistema operacional do mac e desenvolveu o Windows, um sistema operacional com interface gráfica para o usuário, para licenciar para as companhias que produziam PCs. Nas primeiras versões, eles não obtiveram sucesso comercial, mas com a versão 3.1, o Windows passou a ser adotado em larga escala pelo mercado, o que acabou por transformar a Microsoft não somente na maior empresa de software do mundo, mas na maior empresa do mundo de uma maneira geral.

O PC com Windows foi, desde então — e tem sido até hoje — o conjunto que levou o computador não só às pequenas empresas e ao pequeno comércio, mas também à casa das pessoas. A lei de Moore, que dizia que o poder de processamento dos computadores de um determinado custo dobra a cada 18 meses¹³, provou-se correta, e no começo dos anos 90 o custo de um computador já era baixo o suficiente para que pudesse se tornar um equipamento doméstico.

12 An Evening With Wiki Inventor Ward Cunningham in Conversation with John Gage <http://video.google.com/videoplay?docid=-7739076742312910146>

13 Nielsen, Jakob. *Projetando Websites*. trad. Ana Gibson. Rio de Janeiro: Campus, 2000

2.3

As origens do hipertexto

COMPUTERLIB

"Computers are simply a necessary and enjoyable part of life, like food and books. Computers are not everything, they are just an aspect of everything, and not to know this is computer illiteracy, a silly and dangerous ignorance.... I have an axe to grind: I want to see computers useful to individuals, and the sooner the better, without necessary complication or human servility being required. Anyone who agrees with these principles is on my side, and anyone who does not, is not.... Simply as a matter of citizenship, it is essential to understand the impact and uses of computers in the world of the future. Computers belong to all mankind."

(Ted Nelson, Computer lib)¹

"As we may think", o texto de Vannevar Bush que inspirou vários visionários na transformação do papel do computador na história inspirou também a criação do conceito de *hipertexto*. Ted Nelson, o homem que cunhou o termo foi também um deles. Enquanto Douglas Engelbart se concentrou no desenvolvimento de aparatos que permitiriam a manipulação de símbolos e a automatização de tarefas simples, e Licklider concebeu o conceito de rede de computadores, Ted se inspirou no memex para conceber aquele que é considerado o mais longo projeto de software da história da computação: o projeto Xanadu, como aponta o jornalista Gary Wolf:

Xanadu, um sistema de publicação de global de hipertexto, é a mais longa história

1 Wolf, Gary. "the curse of Xanadu" Wired Archive <http://www.wired.com/wired/archive/3.06/Xanadu/5.1.html>

vaporware² na história da indústria de computadores. Ele está em desenvolvimento há mais de 30 anos.³

Nos anos 60, Nelson era um estudante universitário preocupado em como lidar com seu amontoado de notas sobre seus sonhos e esquemas. Encontrou o texto de Vannevar Bush e "abraçou a idéia de que ele poderia usar um computador para "manter a trilha de seus próprios prodigiosos fluxos de pensamentos e esboços", como escreve Rheingold:

O curso de Harvard em programação de computadores que Ted cursou em 1960 usava o único computador então disponível em Harvard, o IBM 7090 no Observatório Smithsonian. Como um projeto de longo prazo, Ted decidiu escrever um programa em linguagem de máquina que lhe permita armazenar suas anotações e manuscritos no computador, para alterar e editar projetos de várias maneiras, e produzir versões finais impressas. Em algum momento por volta de quarenta mil linhas de código, ocorreu-lhe que a sua primeira estimativa da magnitude da tarefa — e a quantidade de tempo que seria necessário terminá-la — tinha sido demasiado otimistas.

(...)

Durante os próximos dois anos, quando lecionou sociologia pensava sobre a complexidade de armazenamento e cruzamento que o tinha impedido de terminar a seu programa de anotações, Nelson percebeu que ele estava tentando criar um novo tipo de coisa. Era uma ferramenta, mas também era uma biblioteca e um novo meio, e uma legião de escravos-bibliotecários. Em meados da década de 1960,

quando ele estava trabalhando em uma empresa de livros, ele começou a chamar a todo projeto de Xanadu. Ele diz que é "um nome tradicional para um lugar mágico na memória literária", mas é importante notar que o poema de Coleridge assim intitulado, como o projeto de Nelson, ficou inacabado.⁴

No curso desse projeto, surgiu a idéia dos links, a partir de um desejo de se ligar partes de um texto a partes de outro, ou outras partes do mesmo texto. Ainda segundo Rheingold:

A facilidade de ligação, Nelson insistiu desde o princípio, ofereceria algo muito mais poderoso do que um meio de unir probabilidades e extremidades. Um sistema com backtrack, controle de versoes e links criaria a possibilidade de uma nova maneira de organizar os pensamentos em palavras, uma forma não seqüencial de escrita que nunca foi possível antes dos computadores, uma forma literária que ele chamou de hipertexto.⁵

4 The Harvard course in computer programming that Ted took in 1960 used the only computer then available at Harvard, the IBM 7090 at the Smithsonian Observatory. As a term project, Ted decided to write a machine-language program that would enable him to store his notes and manuscripts in the computer, to change and edit drafts in various ways, and produce final printed versions. Somewhere around the forty-thousandth line of his program, it dawned on him that his first estimates of the magnitude of the task — and the amount of time it would take to establish it — had been over-optimistic.(...)

Over the next two years, while he taught sociology and thought about the complexities of storing and cross-referencing that had prevented him from finishing his note-keeping program, Nelson realized that he was trying to create a new kind of thing. It was a tool, but it was also a library, and a medium, and a legion of slave-librarians. In the mid-1960s, when he was working at a book firm, he started to call the whole scheme Xanadu. He says it is "a traditional name for a magic place of literary memory," but it is worth noting that Coleridge's poem of that name, like Nelson's term project, was unfinished. Rheingold Op. cit. pg <http://www.rheingold.com/texts/tft/14.html>

5 The link facility, Nelson insisted from the first, provides something far more powerful than a means of attaching odds and ends. A system with backtrack, versioning, and links would create the possibility of a new way of organizing thoughts into words, a nonsequential form of writing that was never possible before computers, a literary form he called hypertext. Op. cit. Rheingold, pg <http://www.rheingold.com/texts/tft/14.html>

2 Vaporware é um termo que diz respeito a projetos de software muito anunciado antes de sua conclusão, mas que nunca chega a ser concluído, segundo a definição do dictionary.com

3 Xanadu, a global hypertext publishing system, is the longest-running vaporware story in the history of the computer industry. It has been in development for more than 30 years. Gary Wolf, The Curse of Xanadu, wired Issue 3.06 | Jun 1995 <http://www.wired.com/wired/archive/3.06/xanadu.html> (nota vaporware)

A idéia do Xanadu possibilitaria que qualquer pessoa pudesse incluir um texto no sistema, que todo mundo pudesse citar qualquer outro texto, num amontoado de conteúdo, o que é, de uma certa forma, muito parecido com o que a web é hoje. O que Rheingold aponta, é que ele estava programando não uma tecnologia, mas uma comunidade, uma "*comunidade eletrônica*". Mas sua idéia de uma biblioteca global que contivesse toda a cultura humana acessível era totalmente megalomaniaca diante da capacidade de processamento dos computadores da época.

O projeto seguiu com ajuda de alguns colaboradores por décadas, com altos e baixos financeiros, reunindo um certo número de hackers ao seu redor, mas nunca conseguiu ser concretizado.

Os hackers do projeto Xanadu podem nunca ter produzido uma versão funcional de seu projeto, mas eles tinham um profundo conhecimento prévio da crise de informações que em breve seria produzida pela tecnologia digital. Eles foram precisamente mortais quando esboçaram um futuro da comunicação de muitos para muitos, a publicação digital universal, as ligações entre documentos, e da capacidade de armazenamento infinita. Quando começou, eles estavam à frente de seu tempo. Mas em meados dos anos 1980, eles mal estavam à sua frente. 6

6 The Xanadu hackers may never have produced a working implementation of their design, but they possessed a profound foreknowledge of the information crisis soon to be produced by digital technology. They were dead-accurate when they sketched a future of many-to-many communication, universal digital publishing, links between documents, and the capacity for infinite storage. When they started, they had been ahead of their time. But by the mid-1980s, they were barely ahead of it. Wolf, Gary, op Cit pg <http://www.wired.com/wired/archive/3.06/xanadu.html?pg=13&topic=>

No final dos anos 80, John Walker, um ex-hacker que ficou conhecido por fundar a Autodesk⁷, se interessou em investir no projeto, e então o projeto migrou para sua empresa, ainda sob o comando de Nelson. Walker acreditava que com investimento corporativo, o projeto deslancharia, como fica evidente nesta sua declaração:

“Em 1964”, anunciou Walker, confiantemente, “O Xanadu foi um sonho de uma única mente. Em 1980, foi o objetivo comum de um pequeno grupo de tecnólogos brilhantes. Em 1989, será um produto. E em 1995, ele vai começar a mudar o mundo. 8

O projeto Xanadu, previa um complexo sistema de recolhimento de direitos autorais dos textos, a cada leitura, seria cobrada uma taxa dos usuários. Na grande livreria Xanadu, cada texto só existiria como original, e as citações sempre se refeririam a eles. O projeto era muito conectado à idéia de direito autoral, e manter a propriedade das idéias a seu dono era uma preocupação central do projeto.

Uma biblioteca universal, democrática, eles decidiram, seria só o começo. O Xanadu também poderia fornecer uma ferramenta para a discussão racional e tomada de decisão entre os grupos muito grandes. No docuverse Xanadu, uma afirmação pode sempre ser se-

7 A Autodesk é uma gigante companhia de software que produz entre outros, softwares 2D e 3D para projetos de engenharia e arquitetura, bem como para produção de realidade virtual, como o Autocad, o Maya e o 3D Studio. Fonte autodesk.com

8 In 1964,” announced Walker, confidently, “Xanadu was a dream in a single mind. In 1980, it was the shared goal of a small group of brilliant technologists. By 1989, it will be a product. And by 1995, it will begin to change the world. Ibid.

guida de sua fonte original. Uma idéia nunca seria destacada de seu autor.⁹

O projeto era um tanto pretensioso, e seus autores acreditavam que com ele, estavam "ajudando a vida na terra a evoluir para uma forma inteiramente nova". (Wolff, 1995) Mas embora eles estivessem resolvendo questões gerais relacionadas ao conceito de *hipertexto*, eles "não tinham um consumidor em mente", e sendo assim, o Xanadu não se configurava como um produto, que pudesse ser vendido por uma companhia como a Autodesk. Além disso, várias disputas aconteciam dentro do time de programadores, e mesmo com anos se passando, o programa não ficava pronto.

Em 1992, A autodesk anuncia um ano de fracasso comercial, o valor de suas ações cai rapidamente, e a empresa procura fazer uma reorganização interna. A nova CEO, Carol Bartz, decide, depois de quatro meses no comando, encerrar o investimento da companhia no projeto. Programadores envolvidos no projeto, assim como Ted Nelson ainda tentaram de diversas maneiras a continuidade do projeto, mas, por volta de 1989, o conceito de hipertexto passa finalmente a ter uma implementação bem-sucedida, proposta por outro pesquisador: a World Wide Web, e, no começo dos anos 90, o Xanadu já se tornava obsoleto pela emergência dessa nova rede.

⁹ A universal democratic library, they decided, was only the beginning. Xanadu could also provide a tool for rational discussion and decision making among very large groups. In the Xanadu docuverse, an assertion could always be followed back to its original source. An idea would never become detached from its author. Ibid.

2.4

A internet e a world wide web

Quando alguém fala em usar a internet, ou navegar na internet, é muito possível que esteja falando em navegar na *World Wide Web*. Embora haja uma certa confusão entre os dois termos, é importante salientarmos antes de tudo, a diferença entre os dois conceitos.

A internet é a rede das redes de computador, uma infra estrutura de cabos que conecta os computadores numa escala global, de modo que cada computador possa se conectar com outro que também esteja conectado nessa rede. Para que aconteça essa comunicação, eles compartilham o uso de um determinado protocolo chamado TCP/IP. Dentro dessa rede, circulam informações de sistemas distintos, como por exemplo, e-mails, comunicadores instantâneos, VoIP, serviços tipo p2p¹, e também o sistema de hipertexto conhecido como World Wide Web. Assim, é importante reforçar que essas duas coisas — internet e web — são coisas distintas, e que uma contém a outra.

Como já dissemos anteriormente, as origens da internet se dão na ARPA, num projeto que procurava unir os sistemas de radares americanos, o SAGE. Posteriormente, Licklider, que trabalhava no projeto, passa a projetar uma rede entre computadores, que começa a funcionar em 1969, ligando a UCLA ao SRI. Em 72, pesquisadores decidem fazer a primeira demonstração pública da ARPAnet, na International Conference on Computer Communications, e instalam um terminal no hotel Hilton, em Washington, que dava acesso ao público a uma série de aplicativos e arquivos hospedados por toda a América.²

¹ p2p ou peer-to peer são serviços de compartilhamento de arquivos diretamente entre usuários, como bit torrent, e-mule ou o finado napster.

² Gromov, Gregory. The Roads and Crossroads of Internet History. disponível em <http://www.netvalley.com/cgi-bin/intval/net_history.pl?chapter=1>

A partir de 73, eles direcionam a pesquisa para determinar protocolos de transmissão que permitissem que os computadores pudessem se comunicar através de várias redes. Eles chamaram esse projeto de "Internetting project" e estabelecem um sistema de protocolos de comunicação que ficou conhecido como TCP/IP — *Transmission Control Protocol e Internet Protocol*. Essa primeira rede era basicamente constituída de universidades e centros de pesquisa, e controlada também a partir de uma colaboração entre essas instituições. A idéia do protocolo IP, que havia sido proposta por Paul Baran nos anos 50 era, como aponta Rheingold:

que as mensagens poderiam ser divididas em unidades de igual tamanho e que rede direciona essas unidades de mensagem por um caminho até para o seu destino, onde elas seriam reagrupadas em conjuntos coerentes.³

Com isso, a rede não teria um ponto vulnerável, como um centro de controle, que pudesse ser atingido, interrompendo as comunicações.

Aos poucos, começam a aparecer versões comerciais dessa rede, como a *Telnet*, em 74. Em 79, surge a *Usenet*, um sistema voltado para a comunicação por e-mail e *newsgroups*. Outras redes e

1969

1970

1971

1972

1973

³ That messages be broken into units of equal size and that the network route these message units along a functioning path to their destination where they would be reassembled into coherent wholes. Rheingold, virtual community, cap 3 <http://www.rheingold.com/vc/book/3.html>

ARPANET GEOGRAPHIC MAP, OCTOBER 1980

Desenvolvimento da internet ao longo do tempo

BBSs⁴ também surgem. Na década de 80, computadores pessoais já eram realidade, e com essa nova realidade, o computador começa finalmente a funcionar como meio de comunicação:

Quando pessoas bastante passaram a ter computadores suficientemente potentes em suas casas, era inevitável que alguém algum ida fosse descobrir uma maneira de ligar os PCs aos telefones. Todos os que você pode plugar em conjunto para fazer esse trabalho, as tecnologias de “habilitação” para as telecomunicações pessoais, estavam disponíveis e o preço estava caindo. Com os poderosos computadores à disposição dos cidadãos de hoje e modems acessíveis, você não precisa de condutor caro de alta velocidade como a internet usa. Você só precisa ligar direito à sua linha telefônica, perfeitamente legal — até agora — e publicar seus manifestos ou organizar as suas reuniões.⁵

Nos anos 80, A National Science Foundation (NSF) passa a financiar a criação da Internet, baseada na infra-estrutura criada pela ARPAnet. Diversos campus universitários americanos já

4 Conhecidos como BBSs, ou bulletin-board systems, eram redes que surgiram na década de 80 e interconectavam computadores individuais via rede telefônica. Segundo Rheingold: “sixty thousand BBSs operated in the United States alone in 1993, fourteen years after the first BBSs opened in Chicago and California. Each BBS supports a population of a dozen to several hundred, or even thousands, of individual participants. There are religious BBSs of every denomination, sex BBSs of every proclivity, political BBSs from all parts of the spectrum, outlaw BBSs, law enforcement BBSs, BBSs for the disabled, for educators, for kids, for cults, for nonprofit organizations—a list of the different flavors of special-interest BBSs is dozens of pages long. The BBS culture has spread from the United States to Japan, Europe, Central and South America.

5 When enough people brought sufficiently powerful computers into their homes, it was inevitable that somebody would figure out a way to plug PCs into telephones. All the off-the-shelf devices that you could plug together to make that work, the “enabling technologies” for personal telecommunications, were available and the price was dropping. With the powerful computers available to citizens today and affordable modems, you don’t need an expensive, high-speed conduit like Internet uses. You just plug right into your telephone line, perfectly legally—so far—and publish your manifestos or organize your meetings. Rheingold, op cit.

tinham muitos computadores disponíveis, e o computador passa fazer parte da realidade intelectual dos estudantes, mesmo fora das tradicionais disciplinas de computação, ciência e engenharia:

Não só muitos centros de computação das universidades aderiram ao backbone de alta velocidade da internet, milhões de estudantes ganharam acesso discado à Internet. Milhares se engancharam na *Usenet*, *MUDs*, Internet Relay Chats, e listas de e-mails.

Em termos de crescimento da população, a comunidade ARPANET original que tinha cerca de mil pessoas em 1969. Um pouco mais de vinte anos depois, a população de internautas é estimada em entre cinco e dez milhões. A taxa de crescimento é muito rápida para medições precisas neste momento, com internetworking em todo o mundo conectando o conjunto das pequenas e médias redes, que vem crescendo ao longo da última década. O número total de redes conectadas cresceu de poucas centenas no início dos anos 1980 para sobre 7.500 no início dos anos 1990, atingindo pessoas em mais de setenta e cinco países. Em 1991 na edição de setembro da revista *Scientific American*, o então senador Albert Gore notou uma estimativa que tem sido apoiado por outros: nos últimos cinco anos, a Internet foi

crescendo em número de usuários em cerca de 10 por cento ao mês. 6

Em 1983 foi criado o *Internet Activities Board* (IAB), para guiar o crescimento da internet e de seu protocolos.⁷

A World Wide Web

Em 1989, a internet tinha ainda pouca significância na Europa. A International Standards Organization (ISO) procurava criar protocolos específicos para redese européias. Mas alguns cientistas do CERN, entre eles Tim Berners-Lee, estavam percebendo que a internet estava realmente conectando as universidades e centros de pesquisa americanos. O CERN, que construiu o LHC⁸, é um centro de pesquisas internacional, que recebe visitas de milhares de cientistas de diferentes partes do mundo todos os anos, e era, segundo Tim Berners-Lee, um ambiente caótico. Para Tim, a grande questão que se colocava era como eles conseguiriam acompanhar e manter

6 Not only did many university computer centers join the high-speed backbone of the Net, millions of students gained dial-up access to the Net. Thousands got hooked on Usenet, MUDs, Internet Relay Chat, and electronic mailing lists.

In terms of population growth, the original ARPANET community numbered around a thousand in 1969. A little over twenty years later, the Internet population is estimated at between five and ten million people. The rate of growth is too rapid for accurate measurements at this point, with worldwide internetworking plugging together all the little and medium-size networks that have been growing over the past decade or so. The total number of connected networks grew from a couple hundred in the early 1980s to over seventy-five hundred by the early 1990s, reaching people in more than seventy-five countries. In the September 1991 issue of *Scientific American*, then-senator Albert Gore noted an estimate that has been supported by others: for the past five years, Internet alone has been growing in numbers of users at around 10 percent per month. Op. cit.

7 Howe, Walt. *Histories of the Internet - A Brief History of the Internet and Related Networks*, 1998, disponível em <<http://www.isoc.org/internet/history/cerf.shtml>>

8 O LHC ou Large Hadron Collider, é o maior acelerador de partículas do mundo, e é construído por um convênio entre vários países europeus. fonte <http://lhc.web.cern.ch/lhc/>

racionais, ele precisava criar "um sistema com regras comuns que pudessem ser aceitáveis para todo mundo", e isso, segundo ele, significava que deveria ser praticamente "sem regra nenhuma"¹³.

Esta rede "sem regras" exigiu uma arquitetura específica que pudesse facilitar a circulação de informação, e certos pressupostos assumidos foram essenciais para a configuração da rede como nós a conhecemos hoje: aberta e descentralizada. Tim lista então uma série de requerimentos iniciais, dentro dos quais podemos destacar:

- Acesso remoto através das redes, que daria a garantia de que os arquivos estariam acessíveis a qualquer momento, em qualquer lugar.
- Heterogeneidade, ou seja, possibilidade de acesso a partir de diferentes sistemas operacionais, para poder incluir os mais diferentes sistemas utilizados no momento e futuros novos sistemas que surgissem.
- Não centralização: o novo sistema deveria permitir a interconexão de sistemas existentes sem a exigência de nenhum controle central. Isto foi crucial para a propagação da rede nas mais diferentes culturas, e nas mais diferentes condições econômicas.
- A possibilidade de geração de análises de dados, como listas de e-mail, criações de modelos topológicos de uma organização ou projeto, permitindo que se consiga enxergar a estrutura real do projeto.
- Links vivos: os dados das pesquisas feitas no CERN mudavam a todo tempo. Esse sistema deveria ser vivo, ou seja, os documentos poderiam mudar conforme as pesquisas fossem se

desenvolvendo. Isso modelou a rede da forma mutante como a conhecemos.

- Ausência de discussão de conceitos de direitos autorais. Ele aponta que essas questões são de importância secundária no projeto do CERN, aonde a troca de informações é ainda mais importante que a confidencialidade.

Esquema de relacionamento da rede. Tim Berners-Lee. Web History and Web Science. IBM, London, UK, March, 2008

Podemos notar que nessas linhas guia para a criação da web estava implícito um certo ponto de vista ideológico, que tinha a liberdade de comunicação e expressão colocadas à frente do direito à propriedade na informação. Se fosse necessário ter controle de acessos de direitos autorais, é possível que a complexidade do projeto o tornasse inviável, ou então que a rede não tivesse se propagado da maneira como ocorreu. Um ponto de vista progressista, que foi essencial para a disseminação da rede numa escala mundial. O projeto *Xanadu*, por exemplo, era muito centrado na questão de direitos autorais e propriedade intelectual, e mesmo com milhões de dólares dispendidos na sua execução, nunca chegou a ser completado. A web, pelo contrário, se baseou em um modelo "sem regras" e, com um orçamento inicial de apenas \$60.000 dólares, conseguiu ser

13 Ibid.

desenvolvida em cerca de um ano. A web foi estruturada desde o início de uma maneira em que a comunicação das informações era mais importante do que as estruturas hierárquicas.

Para propor o sistema de organização de informações do hipertexto, Berners-Lee analisou os sistemas de organização de informações existentes anteriormente. A maior parte deles era baseado em uma estrutura tipo árvore, aonde arquivos são organizados a partir de uma estrutura hierárquica (figura acima). Um sistema desse tipo é o sistema de organização de arquivos do *Windows*, por exemplo, aonde os arquivos são distribuídos em pastas numa estrutura hierárquica, a partir de um diretório de raiz. O problema desse tipo de sistema de informação é que é bastante inflexível, e, às vezes, informações relacionadas se encon-

travam em troncos diferentes, o que prejudica o acompanhamento de discussões, por exemplo.

Outro sistema de organização de informação disponível funcionava por meio de palavras-chave, que permitiam o acesso a arquivos localizados em pontos desconhecidos da estrutura. Um modelo semelhante ao utilizado hoje em vários sites, conhecido como sistema de tags, ou

folksonomias. O maior problema desse tipo de sistema é que dificilmente as pessoas escolhiam a mesma palavra-chave, o que dificultava a localização das informações por aqueles que desconhecem a estrutura geral.

A inovação do sistema proposto por Tim Berners-Lee é que ele funcionava como um diagrama de informação com círculos interligados uns aos outros, aonde os arquivos poderiam ter uma relação unidimensional uns com os outros. Assim, um arquivo poderia se relacionar com outro, que por sua vez se relacionaria com outro e outro, e assim por diante, independente do lugar que cada arquivo estivesse situado na estrutura hierárquica de informação de cada computador, como ele mesmo explica abaixo:

Os textos estão ligados entre si de uma forma que um pode ir de um conceito a outro para encontrar a informação que deseja obter. A rede de *links* é chamada de web. A web não precisa ser hierárquica e, portanto, não é necessário “subir uma árvore” todo o caminho novamente antes de ir para baixo para um assunto diferente, mas relacionado. A web também não é completa, pois é difícil imaginar que todas as ligações possíveis seria colocado pelos autores. No entanto, um pequeno número de ligações é geralmente suficiente para que se vá de qualquer lugar para qualquer outro lugar em um pequeno número de saltos.¹⁴

Em maio de 1990, Tim redigiu essa proposta na forma de um projeto para o CERN. Seriam

¹⁴ The texts are linked together in a way that one can go from one concept to another to find the information one wants. The network of links is called a web. The web need not be hierarchical, and therefore it is not necessary to “climb up a tree” all the way again before you can go down to a different but related subject. The web is also not complete, since it is hard to imagine that all the possible links would be put in by authors. Yet a small number of links is usually sufficient for getting from anywhere to anywhere else in a small number of hops. Berners-Lee, Tim. World Wide Web: Proposal for a Hypertext Project. 1990. <http://www.w3.org/Proposal>

necessários 6 meses de dedicação de uma equipe de apenas 4 engenheiros de software e um programador. A idéia era construir um programa que fosse capaz de localizar e interpretar arquivos de hipertexto localizados em provedores já disponíveis no CERN. Em setembro do mesmo ano, o CERN comprou aquele que viria a ser o primeiro servidor da web, um *NeXt Cube*, e, em outubro, Tim começou a construção de um programa que incluía um browser e um editor de Hipertexto com interface gráfica. Já prevendo a magnitude que sua proposta poderia alcançar, Tim deu a este programa Tim nome de *World Wide Web* (WWW).

Embora desde o início já estivesse previsto a possibilidade de uso de outras formas de mídia, como gráficos e som, o desenvolvimento inicial do projeto se concentrou na universalidade do texto.¹⁵ O *browser* seria um programa que localizasse e decodificasse as informações do hipertexto usando a interface nativa do sistema utilizado no computador do usuário, além fazer a conversão necessária de formatos entre o servidor e a máquina do usuário.

Além do *browser* era necessária a criação de um programa servidor, que rodasse em um determinado computador e que administraria os nós da rede naquela máquina e negociasse a apresentação dos dados para a máquina dos usuários e também era necessário criar um sistema de formatos para a organização dos documentos e dados, assim como um sistema de endereçamento dos arquivos. Em novembro de 1990, esse conjunto inicial já havia sido desenvolvido, e a primeira página da web já estava no ar.

Em maio de 1991, o WWW foi disponibilizado nas máquinas centrais do CERN, e no final do

mesmo ano, foi instalado o primeiro servidor web fora da Europa. Nesse momento histórico, como falamos no início do capítulo, a internet já era utilizada em escala mundial, apesar de restrita ainda a um certo grupo de estudantes e pesquisadores acadêmicos, que estavam mais intimamente ligados à tecnologia¹⁶. Mas naquele momento, estava restrita a alguns serviços existentes, como e-mail e grupos de discussão, como BBC's e a *Usenet*. Em fevereiro de 1992 um browser à base de linha de comando foi anunciado em grupos de discussão na internet.

Mas foi em fevereiro de 1993 que um grande passo foi dado na popularização da web em termos mundiais: a criação do *Mosaic*, o primeiro browser com interface gráfica. Nesse momento, o protocolo WWW ocupava 0,1% do tráfego da internet no servidores de backbone da NFSnet¹⁷. Apesar de ainda pequena, a web começava a atrair cada vez mais interesse dessa comunidade de usuários da internet.

No dia 30 de abril desse mesmo ano, diante desse crescente interesse geral pela web, os diretores do CERN cederam os direitos de uso do sistema para domínio público. Nesse documento, eles descrevem a *World Wide Web*, ou W3, como um projeto que provia um sistema colaborativo de informação, independente de plataforma de hardware e software e localização física. Através dessa atitude, eles colocavam o browser desenvolvido por eles, um servidor básico e uma biblioteca de códigos à disposição para qualquer um duplicar, modificar e redistribuir os seus códigos,

16 Uma pesquisa de uso da internet de 1994 revelava por exemplo, que apenas 5% dos usuários da rede eram mulheres, e que 92% dos usuários utilizavam Unix, um sistema operacional de uso mais restrito à pesquisadores do que a usuários domésticos.

17 A National Science Foundation Network, era uma rede que conectava departamentos de ciência da computação de várias universidades americanas. Era o principal backbone da internet no começo da década de 90. Fonte: w3.org/history.

15 Ibid.

buscando com isso atingir "compatibilidade para além, práticas comuns e padrões de networking e colaboração na computação"¹⁸.

Até este momento, o uso comercial da internet era restrito pelas normas da NSFNET, e estavam em discussão no congresso americano projetos de privatização da internet, pois o desejo da indústria e do comércio de se apropriar dessa tecnologia estava começando a entrar em contradição com as políticas de não comercialização da rede. Assim, a NSF, que era responsável pela maior parte do entroncamento dos servidores de internet, depois de várias reuniões e debates no congresso, chegou a um termo que procurasse satisfazer as necessidades tanto dos pesquisadores acadêmicos quanto aos interesses dos negócios. A solução encontrada foi a privatização da NSFnet, e sua consequente liberação para uso comercial, e criação de um novo *backbone* de alta velocidade, de uso educacional e científico, a *National Research and Education Network* (NREN).¹⁹

A consequência imediata desses dois acontecimentos, um que dava a WWW ao domínio público, e outro que permitia o uso comercial da internet, fizeram com que o uso da WWW explodisse. Em setembro desse mesmo ano, a web já ocupava 1% do tráfego da rede NSF. Nesse mesmo mês, versões do *Mosaic* já estavam disponíveis para todas os sistemas operacionais comuns: X, *Windows* e *Macintosh*.²⁰

A partir desse momento, a rede já começava a tomar forma, e seu uso começa a ser amplamente disseminado. Surgem os primeiros provedores de acesso comerciais. Diante dessa explosão inicial, começaram a ocorrer algumas conferências internacionais sobre a WWW, e em outubro de 1994, Tim-Berners Lee, que havia escrito as primeiras especificações para URI, HTTP e HTML, funda, no MIT, em colaboração com o CERN, e com apoio da DARPA e da Comissão Européia, o *World Wide Web Consortium* (W3C), cuja missão, segundo declaração própria é:

Guiar a World Wide Web a seu potencial pleno, desenvolvendo protocolos e linhas guias que garantam o crescimento de longo prazo para a Rede.²¹

Essa organização busca construir especificações comuns para a rede, padrões e linhas-guia, e serve como um fórum aberto para a discussão da rede, como eles mesmos declaram em seu website. Para que a rede funcione em seu potencial pleno, é essencial que as tecnologias mais fundamentais sejam compatíveis umas com as outras. Para isso é necessário que exista algum consenso entre as especificações das tecnologias utilizadas, como por exemplo, as especificações do HTML, linguagem de marcação utilizada para construir a base do hipertexto. Essa organização já publicou mais de uma centena de especificações, que eles chamam de recomendações da W3C.

Essas recomendações são escritas de forma pública, para que possam ser seguidas pelos programadores que trabalham no desenvolvimento dos vários *browsers* e servidores, sem o pagamento de nenhum *royalty*. Apesar da fundação do W3C datar de 1994, os browsers da década de 90 eram

18 Berners-Lee, 2003. Ten Years Public Domain for the Original Web Software in: <http://tenyears-www.web.cern.ch/tenyears-www/Welcome.html>

19 Thomson, John. Privatization of the New Communication Channel - Timeline, 2000 disponível em <https://my-webspace.wisc.edu/jcthomsonjr/web/j561/timeline.html>

20 Cailliau, Robert. A little history of the World Wide Web. 1995 Disponível em: <http://www.w3.org/History.html>

21 To lead the World Wide Web to its full potential by developing protocols and guidelines that ensure long-term growth for the Web". Fonte: <http://www.w3.org/Consortium/Points/>

focalizados em tecnologias proprietárias, isto quer dizer que várias tecnologias implementadas por eles funcionavam apenas neles próprios; eles seguiam as recomendações feitas pelo W3C apenas parcialmente. Somente os *browsers* mais modernos passaram a adotar essas especificações de forma mais ampla, em grande parte devido à pressão de web designers e programadores que reivindicavam uma rede mais padronizada.²²

Estradas para informação.

McLuhan, em o "Os meios de comunicação como Extensão do homem" já apontava, a construção da estradas e das ferrovias esteve desde sempre muito relacionada à circulação de informação. Meios de transporte são também meios de comunicação:

Foi só com o advento do telégrafo que a mensagem começou a circular mais rápido que o mensageiro. Antes dele, as estradas e a palavra escrita eram estreitamente interligadas. Com o telégrafo, a informação se destacou de certos bens sólidos, como a pedra e o papiro, tal como o dinheiro se desligara antes do couro, do lingote de ouro e dos metais, para terminar em papel. O termo "comunicação" tem sido empregado exaustivamente em conexão com estradas e ponte, rotas marítimas, rios e canais, antes mesmo de se ver transformado em "movimento da informação".²³

As tecnologias de transmissão de informação elétricas, transformaram a sociedade mundial numa aldeia global, nas palavras dele "uma reunificação instantânea de todas suas partes mecanizadas num todo orgânico". Mesmo antes da internet, ele já constatava que o espaço deixa de ser o "fator principal das disposições sociais"²⁴

A web leva esse processo a seu estado maior, é possível se ter qualquer tipo de relação, seja de trabalho, de amizade ou íntima com indivíduos e organizações por todo o mundo, basta só que esses indivíduos estejam conectados por essa imensa rede de cabos e computadores.

Na medida em que surgem as tecnologias de comunicação elétricas, o homem passa a construir novas rotas dedicadas à ela. Em 1866, construiu-se o primeiro cabo atlântico bem sucedido entre a Europa e os Estados Unidos e a partir de então, vários outros cabos foram construídos, para possibilitar a comunicação telefônica através de impulsos elétricos. Aos poucos, foi se desenvolvendo a tecnologia para a produção de cabos capazes de transmitir informação por meio de sinais luminosos através de fibra ótica. Nos anos 70, os laboratórios da *Bell* desenvolveram uma espécie de cabo que apresentava baixíssima perda, que ainda é o padrão de produção de cabos até hoje.²⁵

Information Superhighway

Na década de 90, com o grande crescimento da demanda pela internet, o governo americano começa a discutir a necessidade de criação do que eles vieram a chamar de "*Electronic superhighway*", que seria uma infra-estrutura física de rede de cabos de fibra-ótica de alta velocidade, para suprir essa demanda. Em 1993, já haviam 10 milhões de usuários conectados nessa rede global, e o número dobrava a cada ano.²⁶

Em 1994, gigantes conglomerados de mídia e rovedores de informação já estavam fazendo projetos-pilotos para a web, e as potencialidades eram imensas, como aponta James Gleick:

25 Timbercon, Inc. History of the Fiber Optics, 2009. disponível em <<http://www.timbercon.com/History-of-Fiber-Optics/index.html>>

26 Markoff, John, Building the Electronic Superhighway. in: <http://www.nytimes.com/1993/01/24/business/building-the-electronic-superhighway.html>

22 Zeldman, Jeffrey. Designing with web standards. Berkeley: New Riders Publishing, 2003

23 McLuhan, op. cit. pg 108

24 McLuhan, op. cit. pg 114

Cabos atlânticos submarinos. fonte: <http://image.guardian.co.uk/sys-images/Technology/Pix/pictures/2008/02/01/SeaCableHi.jpg>

Nunca nos anais dos negócios tantas grandes empresas corram com seus seus motores de forma tão violenta por causa de uma visão tão sedutora... Vídeos transmitidos sobre fios de telefone... Chamadas telefônicas digitalizadas e compartimentadas sobre cabos de televisão ... Dados e interatividade em todos os lugares. Se você quiser estudar a terceira base, enquanto outros espectadores estão assistindo a massa, você pode — pelo menos nos laboratórios. Quer ler o jornal segundos depois que ele é transmitido ao seu computador de bolso? Ou navegar por 500 canais? *Replay* daquela cena de sexo? Responder a Connie Chung? Fácil. Venha e tenha! Pelo menos essa é a idéia.²⁷

Mas a infra-estrutura necessária para isso ainda era incipiente. Nesse sentido, a construção dessa “*superhighway*” se fazia necessário, e a discussão que havia era quem é que haveria de construí-la; governo ou iniciativa privada. Al Gore, que foi vice-presidente na era Clinton, foi maior defensor da construção dessa rede pelo governo acreditava que sua construção traria uma nova revolução para a história da informação, nas suas próprias palavras:

Neste momento, a fibra ótica está começando uma revolução tecnológica imprevisível e dramática, tanto de computação quanto de comunicações. A Academia Nacional de Engenharia de lista a comunicação por fibra óptica como uma das mais importantes realizações da

27 Never in the annals of business have so many great enterprises raced their engines so violently in the cause of so seductive a vision.... Videos beamed over telephone wires.... Telephone calls digitized and packetized over television cable.... Data and interactivity everywhere. If you want to study the third baseman while other viewers are watching the batter, you can — at least in the laboratories. Want to read the newspaper seconds after it's transmitted to your pocket computer? Or surf 500 channels? Replay that sex scene? Talk back to Connie Chung? Easy. Come and get it! At least that's the idea. Gleick, James. The information Future: Out of Controll (And its a good thing too.) . New York times, 1 Maio 1994. disponível em <<http://www.nytimes.com/1994/05/01/magazine/the-information-future-out-of-controll-and-is-a-good-thing-too.html>>

engenharia científica dos últimos 25 anos. Claramente, é tão revolucionária como foi o transistor. Um único cabo pode transportar centenas de milhares de vezes o número de canais transmitidos pelo maior cabo de cobre de televisão. Esse simples fato significa que a fibra ótica traz oportunidades inimagináveis com cobre e torna possível a digitalização da América, incluindo a televisão digital, o que tornará a HDTV tão desatualizada como discos de 78 rpm. Irá tornar mais fácil a transferência de bilhões de *bits* de dados de uma costa à outra em segundos. Em muitos aspectos fundamentais, fibra ótica vai mudar a nossa forma de ver o mundo, assim como a revolução de Copérnico fez.²⁸

Hoje, todos os continentes são interligados por redes de fibra ótica que somam mais de 150.000 quilômetros, explorados por diversas empresas. A capacidade de transmissão de informação chega a 7.1 *tera bytes* por segundo, segundo o *the guardian*. Esse esforço em montar essa rede global de infra-estrutura fez fomentar a explosão inicial da web, e também fez levá-la a um novo patamar: imagem, som e vídeo na velocidade da luz.

28 Right now, fiber optics is starting an unpredictable and dramatic technological revolution in both computing and communications. The National Academy of Engineering lists fiber-optic communication as one of the most important scientific engineering achievements of the last 25 years. Clearly, it is every bit as revolutionary as was the transistor. A single cable can carry hundreds of thousands of times the number of channels carried by the largest copper television cable. That simple fact means that fiber will provide opportunities unimaginable with copper and make possible the digitization of America, including digital television, which will make HDTV as outdated as 78 rpm records. It will make it easy to shift billions of bits of data from one coast to the other in seconds. In many fundamental ways, fiber optics will change the way we view our world, just as the Copernican Revolution did. Gore, Albert Jr. *Information Superhighways: The next information revolution*. The futurist, 1991. Disponível em <<http://www.allbusiness.com/technology/internet-technology/150025-1.html>>

2.5

Forma e conteúdo da rede: história das linguagens

Como dissemos no capítulo anterior, a linguagem base para a estruturação do Hypertexto na web é o HTML, ou *Hypertext Markup Language*. Apesar de ser uma linguagem codificada, o HTML não é uma linguagem de programação, mas somente de marcação. Antes de entendermos o porque dessa decisão, cabe entender o que são linguagens de programação, e como elas podem ser utilizadas.

Como vimos na seção 2.1 deste trabalho, a origem das linguagens de programação remontam à Ada Lovelace e ao Analytical engine, de Charles Babage. Computadores são ferramentas capazes de executar algoritmos, ou seja, de executar certos procedimentos de cálculo para executar determinadas tarefas. Para executar esses cálculos, o computador executa uma série de procedimentos lógicos que ocorrem no nível dos circuitos elétricos. Nos primeiros computadores, era necessário se determinar cada um desses processos:

O movimento exato de dados entre memória interna do processador (registros) e memória interna tinha de ser especificado, juntamente com as operações aritméticas apropriadas. Este nível mais baixo, forma menos abstrata de linguagens de programação é o mais difícil para o ser humano entender e usar.¹

O primeiro passo para facilitar essa interação foi o de dar nomes mais fáceis para certos procedimentos mais repetitivos, e criar um programa chamado *assembler* que é o que faz passar essas instruções de um nível mais simbólico para a memória do computador, como ADD para adicionar, por exemplo. Esse tipo de linguagem, que ainda opera num nível de *hardware* é chamada

¹ The exact movement of data between the processor's internal storage (registers) and internal memory had to be specified, along with the appropriate arithmetic operations. This lowest level, least abstract form of programming languages is hardest for humans to understand and use. Encyclopedia of computer Science pg 388.

de linguagem de programação de baixo nível. A maior parte dos programadores, no entanto, trabalha utilizando o que são chamadas de linguagens de alto nível, que são linguagens que em geral se assemelham mais à linguagem comum. Uma linguagem de programação tem um certo número de *tokens*² e uma estrutura sintética funcional pré-determinados. Uma vírgula ou uma quebra de linha, por exemplo, podem significar o fim de uma determinada instrução, parênteses podem ser utilizados para realizar tarefas em bloco e assim por diante.

As linguagens de programação de alto nível em geral podem executar conjuntos de instruções em sequência, até que uma determinada condição seja atingida, o que chama-se de loop, ou podem executar um procedimento pré-determinado em outra parte do programa, ou funções. Além disso, existem variáveis de diversos tipos que podem ser definidas e declaradas pelos programadores. As tarefas são escritas na forma de conjunto de instruções lógicas, para determinar quais tarefas são executadas de acordo com quais interações sofridas. Por exemplo, um programa pode receber um determinado valor, compará-lo com outro e determinar uma resposta dependendo do resultado dessa comparação.

Cada linguagem tem sua sintaxe definida pelos programadores, e é compreensível por aqueles que tenham domínio dessa linguagem, mas para que possam funcionar no nível de *hardware*, tem de ser compiladas, isto é, transformadas em linguagem de baixo nível, para que o computador consiga executar as operações determinadas pelo programa. No momento em que o código é compilado, ele deixa de ser acessível. É assim que se dá a proteção aos direitos intelectuais do fabricante de *software*: uma vez que é vendido em um

pacote fechado, compilado para um determinado sistema operacional, é muito difícil entender como o *software* funciona, e sendo assim, fica muito difícil de se apropriar daquele código.

As vezes, para se executar algumas tarefas, não é necessário se escrever um programa compilado, uma grande gama de comandos pode ser acessado a partir dos terminais dos sistemas operacionais, através do que se chama de *shell script*, e, diante da grande quantidade de tarefas exigidas para administradores de rede, surgiram novas linguagens baseadas nessa agilidade dos *scripts* de *shell*, como aponta Henderson, na Enciclopédia de Ciência e Tecnologia da Computação:

a complexidade das redes modernas e, em particular, da Internet, tem levado os administradores e programadores a buscar linguagens que combinam a natureza rápida e interativa de *shells*, as características estruturais de extração de dados e outras tarefas. O resultado foi o desenvolvimento de uma série de linguagens populares de *script*.³

Uma linguagem de *script* é também um conjunto de instruções em texto, mas não precisa ser compilada. Quando se executa um arquivo, o processador da linguagem (ou *parser*) processa cada declaração e a executa instantaneamente. Linguagens desse tipo foram feitas para ser simples de usar e para construir aplicativos para a web, boa parte das linguagens utilizadas são linguagens desse tipo, como *Python*, *Php* e *JavaScript*.

² Tokens são palavras que tem significados semânticos específicos em uma determinada linguagem

³ the complexity of modern networks and in particular, the Internet, has driven administrators and programmers to seek languages that would combine the quick, interactive nature of shells, the structural features of data extraction, and other tasks. The result has been the development of a number of popular scripting languages. Encyclopedia of Computer Science.

HTML, no entanto, como dissemos, não é uma linguagem de programação. Isto significa que um código HTML não executa nenhum comando, não realiza nenhuma ação. Ele apenas marca partes de um determinado texto e indica, através de suas marcações, relações de estrutura semântica, como por exemplo, se um determinado trecho de texto é um título ou um parágrafo; ou relações com outros arquivos, como por exemplo a inserção de imagens e links. Nesse sentido, como linguagem, o HTML é bastante limitado, o que na verdade não é um problema, mas uma decisão muito acertada de projeto, como justifica seu criador, Tim Berners-Lee:

HTML é uma linguagem limitante: você pode usá-la apenas para expressar documentos de hipertexto. Java, pelo contrário, não é: você pode escrever um pouco de Java para fazer quase tudo. Linguagens limitantes são úteis porque você pode, por exemplo, analisar uma página HTML elemento por elemento, convertê-la para outros formatos, indexar, ou o quer que seja. É claro para que cada bit serve. As pessoas fazem todo tipo de coisas com as páginas HTML diferentes do que as páginas foram originalmente destinadas. Um *applet* Java é diferente. Porque Java é uma linguagem de programação completa, você pode usá-lo para fazer qualquer coisa, incluindo a criação de um pinguim que faz cambalhotas. No entanto, porque Java é tão poderoso, a única maneira de descobrir o que um *applet* Java vai fazer é executá-lo e prestar atenção. Quando eu projetei HTML para a web, eu o escolhi evitar dar-lhe mais poder do que fosse absolutamente necessário — um princípio de “menos poder”, ao qual eu fiquei preso para sempre. (...)

A razão para o sucesso da web é que o hipertexto é um meio tão flexível que a web não

constrange o conhecimento que tenta representar.⁴

Linguagens de marcação não eram novidade quando Tim escreveu o HTML. As origens desse tipo de linguagem remontam à linguagem simbólica utilizada pelos tipógrafos para marcar uma manuscrito, antes da composição em metal; e mesmo na informática, elas já eram utilizadas. Existia um conjunto de linguagens de marcação, denominadas *Standard Generalized Markup Languages*, ou SGML, que já era utilizado por várias comunidades internacionais para documentação, e inclusive era utilizado nas máquinas IBM no CERN para construção de bancos de dados. Ele desenhou o HTML inspirado no SGML, de modo que essa nova linguagem fosse familiar àqueles que já estivessem acostumados ao SGML.

No SGML, certas palavras, chamadas de *tags*, faziam o papel de marcar determinado texto, mas era uma linguagem altamente expansível, uma vez que qualquer palavra poderia se transformar em *tag* uma vez que estivesse cercada por chaves angulares (<>). Assim, cada pessoa ou empresa construía seu banco de dados próprio, a partir de palavras que fossem significativas em cada caso.

4 HTML is a limiting language: You can use it only to express hypertext documents. Java, by contrast, isn't: You can write a bit of Java to do almost anything. Limiting languages are useful because you can, for example, analyze an HTML page element by element, convert it into other formats, index it, and whatever. Its clear what every bit is for. People do all kinds of things with HTML pages that the pages were never originally intended for. A Java applet is different. Because Java is a complete programming language, you can use it to do anything, including creating a penguin that does somersaults. However, because Java is so powerful, the only way to figure out what a Java applet will do is to run it and watch. When I designed HTML for the web, I choose to avoid giving it more power than it absolutely needed — a “principle of least power,” which I have stuck to ever since. (...)

A reason for the success of the web is that hypertext is so flexible a medium that the web does not constrain the knowledge it tries to represent”. Berners-Lee 1999, pg 182-183

O que Tim fez para contruir o sistema de hipertexto foi reduzir o SGML a um certo número de *tags* pré-determinadas, baseadas em funções necessárias ao hipertexto, como <p> para parágrafo para imagem e <a> para link, por exemplo.

Eu nunca tive a intenção de que o código fonte HTML (o material com os colchetes) fosse visto pelos usuários. Um navegador/editor permitiria que a um usuário simplesmente ver ou editar a linguagem de uma página de hipertexto, como se ele estava usando um processador de texto. A idéia de pedir às pessoas para escrever os colchetes à mão foi para mim, e eu assumi que para muitos, tão inaceitável como pedir para alguém preparar um documento do Microsoft Word, escrevendo o seu formato binário codificado. Mas a legibilidade humana do HTML foi uma bênção inesperada. Para minha surpresa, as pessoas rapidamente se familiarizaram com as tags e começaram a escrever seus próprios documentos HTML diretamente.⁵

Forma e Conteúdo - A Linguagem CSS

Em 1994, o CERN já havia doado a tecnologia necessária para o funcionamento da *World Wide Web* para domínio público, já existiam *browsers* gráficos, como o *Mosaic* e a rede já estava começando a ser utilizada como plataforma para inúmeras publicações eletrônicas, mas havia uma parte crucial faltando: não haviam formas de dar estilo aos documentos (Håkon Wium Lie and Bert Bos, 1999); de diagramá-los de maneiras

diferente do que estava previsto na visualização padrão da linguagem HTML.

Como o HTML é uma linguagem apenas de marcação, nada que esteja no arquivo determina sua aparência no computador do usuário. Quem interpreta o código e determina a forma com que os arquivos são exibidos é o *browser*. Para isso, os *browsers* utilizavam um recurso chamado de *folha de estilos*, que determina a aparência de cada elemento estrutural do HTML. Mas essa folha de estilo não estava publicada como uma sintaxe de linguagem, porque seu criador, Tim Berners-Lee acreditava que cada navegador deveria determinar a forma que o código deveria ser mostrado para seus usuários. E sendo assim, cada programa tinha uma linguagem própria para determinar a apresentação do código.

Com a ampliação do uso comercial da rede, principalmente por grandes empresas que estavam partindo para empreitadas no mundo virtual, essa falta de possibilidade de escolha começou a se tornar um desconforto para os novos profissionais que estavam se aventurando nesse novo campo de trabalho, a criação de websites, que tinham pouco controle sobre como seus sites iriam aparecer para seus usuários.

Nesse momento, a linguagem HTML era apenas uma recomendação e o W3C estava no início da sua existência. Em resposta a essa tensão inicial, os *browsers* começaram dar suporte para tags inventadas por eles mesmos, especialmente para lidar com questões de aparência, como foi o caso da primeira versão do *Mozilla*, que lançou o suporte para a tag <center>, que centralizava o texto na página. Como o HTML não estava totalmente consolidado como linguagem, vários *browsers* das primeiras gerações tinham suporte para tags próprias.

⁵ I never intended HTML source code (the stuff with the angle brackets) to be seen by users. A browser/editor would let a user simply view or edit language of a page of hypertext, as if he was using a word processor. the idea of asking people to write the angle-brackets by hand was to me, and I assumed to many, as unacceptable as asking one to prepare a Microsoft word document by writing out its binary coded format. But the human readability of HTML was an unexpected boon. To my surprise, people quickly became familiar with the tags and started writing their own HTML documents directly. Berners-Lee, 1999 pg 42.

Era um princípio de projeto da rede, manter separadas a apresentação (forma), da informação (conteúdo); como justifica Berners-Lee:

O principal primário por trás da independência de dispositivo, e acessibilidade, é a separação da forma do conteúdo. Quando o significado de um documento é armazenado separadamente do jeito que ele dever ser exibido, a independência de dispositivos e acessibilidade tornam-se muito mais fácil de manter. Grande parte disso é conseguido com uma folha de estilo - um conjunto de instruções sobre como apresentar a página ou transformá-la numa página impressa.⁶

Nesse sentido, começaram a ser desenvolvidos os primeiros estudos para a criação de uma linguagem própria para isso — o CSS, ou *Cascading Style Sheets* (folhas de estilo em cascata). Apesar de ter sido desenvolvido principalmente para aplicação conjunta com o HTML, o CSS é uma linguagem independente, que pode ser aplicada a quaisquer outras linguagens de marcação.

Na primeira fase do web design, formas alternativas eram atingidas mediante o uso de *image maps*, que era uma forma de colocar hiperlinks em imagens, ou através do uso de tabelas, que permitiam a divisão modulada da janela. O problema maior do uso de *image maps*, é que imagens, mesmo que sejam contínuas de texto, não podem ser pesquisadas por mecanismos de busca e nem acessadas por *browsers* que transformam sites em áudio, para cegos, por exemplo.

A falta de alternativa dos designers para lidar com apresentação de imagens e também com interação, acabou fazendo com que milhares aderissem ao *Flash*, na época produzido pela *Macromedia*.

O CSS, segundo o rascunho inicial de sua proposta, foi criado para dar aos leitores um ambiente visualmente (ou sonoramente, ou tátil) mais rico, ao mesmo tempo que os permitisse manter um certo controle quando necessário, e dar aos autores uma influência estética e estilística sobre as páginas sem que eles precisassem recorrer a linguagens de descrição de página. O CSS permite uma maior portabilidade, pois o designer tem a opção de criar versões específicas do site para uso impresso, ou para dispositivos móveis, por exemplo.

A linguagem CSS é uma linguagem muito simples de ser utilizada. Ela funciona apenas de uma forma declarativa, e haviam várias críticas quanto à sua capacidade de fazer o que se propunha: formatar as páginas de uma maneira em que satisfizesse as necessidades dos usuários, dos autores e do meio utilizado. Em 1995 o W3C começou a funcionar de fato como pólo gerador de referências para a web, e vários programadores passaram a se dedicar à construção dessa linguagem de apresentação. Nessa época, havia uma disputa entre a *Microsoft* e a *Netscape* pelo mercado de navegadores, e cada um adotava tecnologias diferentes para apresentação. A *Microsoft* sinalizou que adotaria suporte ao CSS, mas a *Netscape* relutava. Finalmente, em 1996 surgiu a primeira versão do CSS como uma recomendação do W3C.

Em fevereiro de 1997 o W3C criou um grupo específico somente para o CSS, que começou a trabalhar em questões problemáticas e ausentes na versão inicial. Em maio de 1998 a versão 2 do

6 The primary principle behind device independence, and accessibility, is the separation of form from content. When the significance of a document is stored separately from the way it should be displayed, device independence and accessibility become much easier to maintain. Much of this is achieved with a style sheet — a set of instructions on how to present or transform a printed page. Berners-Lee, 1999.

CSS foi lançada como recomendação. É importante notar que o CSS é uma linguagem meramente descritiva, ou seja, ela descreve atributos de determinados elementos na página (mais adiante entraremos em maiores detalhes de como essa linguagem funciona). Como linguagem não passa nenhum tipo de instrução para o *browser*, apenas indica como ele deve fazer o processamento da página.

Um problema acontece porque cada programa tem maneiras diferentes de processar a mesma mensagem, e tem compreensão diferente de como essas recomendações devem ser lidas. Principalmente na década de 90, quando os *browsers* implementavam de maneira bem precária as recomendações feitas pelo W3C, muitos designers acabavam tendo que fazer um trabalho múltiplo, de programar um site diferente para cada tipo de navegador. Através de um reconhecimento da plataforma através de script, o código aplicava um outro estilo determinado a uma mesma página. (Zeldman, 2003)

Hoje, os *browsers* já avançaram muito no sentido de adoção dos padrões das linguagens, mas mesmo assim ainda persistem algumas diferenças de interpretação, que ainda causa dificuldades no design e construção de sites.

Flash

As potencialidades que o HTML oferecia na década passada para os desingers eram mito pequenas, existiam apenas poucos elementos de formatação e o CSS ainda não estava desenvolvido como linguagem a ponto de permitir a criação de peças elaboradas. Em 1996 surgiu no mercado o *software Macromedia Flash*, que permitia a criação de gráficos e animações interativas, baseadas em objetos vetoriais, que poderia ser inserido numa página HTML, como um *applet* em java.

O *Flash* permitia — e permite até hoje — que os designers exercessem sua liberdade criativa de uma maneira nunca antes alcançada nos meios anteriores. Além de possibilitar a execução de arranjos tradicionais de arranjo que eram possíveis de se conseguir na mídia impressa, como uso de fontes especiais, textos fora do ângulo reto, ele permitia também a execução de interações animadas e pequenos filmes, assim como a criação de jogos. Como são baseadas em imagens vetoriais, os websites realizados em *Flash* também são relativamente leves, e para os padrões de velocidade da internet de 1996, eram mais viáveis do que sites baseados em imagens *bitmap*, como *gif* e *jpg*.

Diante dessa potencialidade, o *Flash* foi largamente adotado pela comunidade de designers gráficos, que através desse recurso, puderam exercer sua liberdade criativa plena e realizar uma série de experimentos artísticos e interativos. Até hoje o *Flash* ainda é a alternativa mais fácil para se realizar aplicativos leves com altos graus de interação. A tecnologia, no entanto, de propriedade da *Adobe Corporation*, e funciona como um adendo à web. Embora o *Flash* player seja distribuído gratuitamente e seja "um dos softwares mais distribuídos da história da web"⁷, ele ainda é um sistema proprietário, e para se desenvolver aplicativos baseados nele é preciso comprar um *software*. Além disso, o conteúdo de uma animação em *Flash* é, em geral, inacessível por mecanismos de busca e por indexadores.

Há um tempo atrás, era muito comum sites inteiros desenvolvidos em *Flash*. Hoje, com o domínio total dos mecanismos de busca bem como questões de acessibilidade que se impõe, as empresas tendem a valorizar a permeabilidade do

⁷LENTZ, Jon Warren; TURNER, Bill e CHIA, Ian. *Flash: The Future*. San Fransicis: No Starch Press, 2002 pg XXVII

site e, com o desenvolvimento da linguagem CSS, bem como de bibliotecas de código *JavaScript*, que permitem a execução fácil de interações ricas para o usuário, bem como o desenvolvimento de tecnologias livres para execução de *server side scripting* e banco de dados, é, em muitos casos, mais interessante a realização de páginas em HTML do que a execução de um site em *Flash*. Mas essa tecnologia continua sendo essencial em certos aspectos, como na execução de players de áudio e vídeo — o *player* do *youtube.com*, por exemplo, é desenvolvido nela — e na execução de games online, por exemplo. A falta de permeabilidade que um arquivo *Flash* tem pode ser inclusive vantagem em muitos casos, como por exemplo para impedir o acesso aos usuários a arquivos de áudio, que tenham *copyright* de terceiros envolvidos.

A versão 5 do HTML, que ainda está começando a ser implementada pelos *browsers*, vai trazer suporte para novas *tags*, como `<video>` e `<audio>`, e, com esse suporte, programadores poderão criar dispositivos de controle multimídia apenas com HTML e *JavaScript*. Há quem diga que quando o HTML 5 for totalmente implementado vai tornar o *Flash* obsoleto⁸. Se isso realmente irá ocorrer, não podemos dizer ainda, mas a capacidade de se trabalhar com formatos abertos de compressão de áudio e vídeo gera também novas possibilidades de interação, que vão além do que o *Flash* apresentou de recursos até agora.

8 Rogério, Pedro, 2009 in <http://www.pinceladasdawe.com.br/blog/2009/06/17/html5-e-o-fim-do-Flash-player/> e Schonfeld, Erick, 2009 <http://www.techcrunch.com/2009/06/09/demo-firefox-35-treats-videos-like-web-pages-why-cant-Flash-do-that/>

2.6

Um novo paradigma: O Software livre

Quando comecei a trabalhar no Laboratório de Inteligência Artificial do MIT em 1971, me tornei parte de uma comunidade de compartilhamento de software que já existia há muitos anos. O compartilhamento de software não estava limitado a nossa comunidade em particular, é tão antigo quanto os computadores, assim como o compartilhamento de receitas é tão antigo como cozinhar. Mas fazíamos mais do que a maioria.¹

Nos anos 70, entre as comunidades de *hackers* e programadores, a prática de compartilhar códigos era muito difundida; o objeto central da indústria da computação era o desenvolvimento de *hardware*, e o software ainda não era muito valorizado como produto. A *Microsoft*, primeiro com o licenciamento do MS-DOS para a IBM e posteriormente fabricando e vendendo o *Windows* é uma das grandes responsáveis por essa mudança de paradigma. Logo, programadores diversos e *hackers* perceberam que era no *software* que residia o grande valor dos computadores, tanto para os indivíduos como para as empresas; e nos anos 80, a grande maioria dos programas eram patenteados, fechados e vendidos.

O *copyright*² tinha servido para indústria do livro impresso — bem como para a indústria da música — durante séculos, para garantir aos autores e às editoras a viabilidade econômica da atividade intelectual, mas até então, esses produtos da indústria cultural eram também objetos materiais, que requeriam certo dispêndio de material e forças produtivas para sua produção e distribuição.

¹ When I started working at the MIT Artificial Intelligence Lab in 1971, I became part of a software-sharing community that had existed for many years. Sharing of software was not limited to our particular community; it is as old as computers, just as sharing of recipes is as old as cooking. But we did it more than most. Stallman, 2008

² Por copyright, entende-se aqui a legislação regulatória de direitos autorais, que tem origem no século XVII, para proteger a indústria do livro. Cada país tem sua legislação específica, mas aqui estamos tratando à como um todo.

O termo “software livre” é, por vezes incompreendido — não tem nada a ver com o preço. É sobre liberdade. Aqui, portanto, está a definição de software livre: um programa de software é livre, para você, um usuário em particular, se:

*** Você tem a liberdade de executar o programa, para qualquer finalidade.**

*** Você tem a liberdade de modificar o programa para atender às suas necessidades. (Para tornar esta liberdade efetiva, na prática, você precisa ter acesso ao código-fonte, uma vez que fazer alterações em um programa sem ter o código-fonte é extremamente difícil.)**

*** Você tem a liberdade de redistribuir cópias, gratuitamente ou por uma taxa.**

*** Você tem a liberdade de distribuir versões modificadas do programa, de modo que a comunidade possa beneficiar das suas melhorias.**

Richard Stallman -

The GNU Project

<http://www.gnu.org/gnu/thegnuproject.html>

The term “free software” is sometimes misunderstood—it has nothing to do with price. It is about freedom. Here, therefore, is the definition of free software: a program is free software, for you, a particular user, if:

* You have the freedom to run the program, for any purpose.

* You have the freedom to modify the program to suit your needs. (To make this freedom effective in practice, you must have access to the source code, since making changes in a program without having the source code is exceedingly difficult.)

* You have the freedom to redistribute copies, either gratis or for a fee.

* You have the freedom to distribute modified versions of the program, so that the community can benefit from your improvements.

A digitalização vinda com a informática transforma essa realidade, como cita Stallman:

Mais recentemente, a humanidade desenvolveu uma nova maneira de distribuir informação: computadores e redes. Eles facilitaram a cópia e a manipulação de informações, incluindo software, gravações musicais, livros e filmes, e ofereceu a possibilidade de acesso ilimitado a todos os tipos de dados de uma utopia de informação.³

Com a informatização, dados e informações passam a poder trafegar e circular independente de um gasto com suporte. Um programa pode ser copiado e distribuído quantas vezes for necessário, sem que tenha seu valor reduzido, sem que perca suas funcionalidades. O mesmo acontece com qualquer informação digital: áudio, vídeo, literatura, documentação. Essa reproduzibilidade ilimitada da informação digital abre caminho para que certos grupos de indivíduos discutam um novo tipo de cultura: a cultura da livre circulação das ideias.

Em 1983, o *software* já havia se tornado uma mercadoria valiosa e vários códigos que haviam sido desenvolvidos sob essa perspectiva de compartilhamento, como o próprio sistema operacional UNIX, haviam sido fechados. Para se executar qualquer software, mesmo que livre, era necessário comprar um sistema operacional proprietário, como cita Stallman. O sistema operacional é o suporte para todos os *softwares*; sem ele, o computador não tem nenhum uso. Diante desse gargalo, Stallman, junto a outros camaradas que compartilhavam essa ideologia de cultura livre e livre circulação do conhecimento, começam

³ More recently, humanity developed a new way of distributing information: computers and networks. They facilitated copying and manipulating information, including software, musical recordings, books, and movies, and offered the possibility of unlimited access to all sorts of data—an information utopia. Stallman, 2009.

um projeto de criação de um sistema operacional livre: o GNU⁴. Logo, diante de um trabalho tão grande, decidiram criar a *Free Software Foundation*, inicialmente para ajudar a angariar recursos para o projeto.

A ideia era de se criar um sistema que fosse compatível com o *Unix*, mas que fosse 100% livre. Centenas de programadores do mundo todo se envolveram no projeto, mas por muitos anos ele ficou incompleto, até que em 1991, um estudante da Universidade de Helsinki, Linus Torvalds, começou a trabalhar no desenvolvimento do *kernel*⁵ de um sistema operacional, e decide lançá-lo sob licença GPL⁶. Suas ambições não eram muito grandes, ele estava interessado apenas em estudar mas acabou colocando seu projeto em um *newsgroup* na *Usenet*, e diante de respostas, logo começou a se configurar uma comunidade de ativistas interessados no desenvolvimento desse sistema.

Em 1993, programadores já haviam juntado o *kernel* produzido por Linus, em colaboração com centenas de ativistas, com o OS desenvolvido pela *Free Software Foundation*, criando a base de um sistema operacional completo, chamado de GNU/Linux. Por serem softwares livres, eles permitem que a comunidade se aproprie de seu códigos e os transformem de acordo com seus interesses, de acordo com as quatro liberdades listadas na página ao lado. Por causa disso, diversas versões desse sistema são distribuídas no

mercado, cada uma de acordo com interesses específicos de comunidades de desenvolvedores.

Um novo modelo de negócios

O que de início pareceu ser um ataque à indústria capitalista, logo foi apropriado pela indústria como um novo modelo de negócios⁷. Várias empresas passaram a enxergar no *software* livre não uma ameaça ao sistema, mas uma potencialidade para diminuir gastos. Várias empresas passaram a adotar *software* livre como base de suas operações, para evitar gastos imensos com licenças e para conseguirem livremente implementar os *softwares* de acordo com suas necessidades.

A *Google*, por exemplo, é uma das maiores financiadoras de programadores de código aberto. Eles usam *Linux* como sistema operacional na maioria das suas máquinas, e todo ano investem massivamente em desenvolvimento de *Free/Libre/Open Source Software* (FLOSS)⁸, no *Google Summer of Code*, por exemplo, que eles descrevem abaixo:

O *Google Summer of Code* é um programa global que oferece a estudantes de escrever código para vários projetos de *software open source*. Temos trabalhado com vários *open-source*, *softwares* livres e grupos de tecnologias relacionadas para identificar e financiar vários projetos ao longo de um período de três meses. Desde sua criação em 2005, o programa reuniu cerca de 2500 alunos participantes bem sucedidos e 2500 mentores de 98 países em todo o mundo, tudo por amor ao código. Através do *Google Summer of Code*, os estudantes candidatos aceitos são emparelhados com um mentor ou mentores dos projetos participantes, tendo assim uma exposição do mundo real de cenários de desenvolvimento de *software* e oportunidades de emprego em

4 GNU é um acrônimo para GNU is not Unix, uma vez que Unix era um dos principais sistemas operacionais utilizados pela comunidade de programadores na época.

5 O Kernel é a parte do sistema operacional responsável por fazer a relação entre o hardware e os aplicativos

6 A GNU Public License, ou GPL é uma licença criada pela Free Software foundation para determinar que um determinado trabalho pode ser copiado e distribuído bem como alterado livremente, desde que distribuído sob a mesma licença.

7 kleiner e Wyrick, 2007

8 FLOSS é uma denominação que inclui *softwares* livres publicados em licença GPL ou *open source* em geral publicados por outras licenças de uso

áreas relacionadas às suas atividades acadêmicas. Em troca, os projetos participantes são capazes de identificar mais facilmente e ganhar novos desenvolvedores. Melhor de tudo, mais código-fonte é criado e liberado para uso e benefício de todos.⁹

A IBM também é outra empresa, que como aponta Pamela Samuelson (2006) passou a utilizar *software* livre e conseguiu com isso aumentar suas margens de lucro e atingir crescimento¹⁰. A mudança de paradigma nesse caso é que ao invés de venderem *software*, eles passam a vender suporte e adaptação dos sistemas, uma vez que programas de código aberto em geral, por oferecem total possibilidade de alteração do código, têm uma possibilidade muito maior de serem adaptados para fins específicos do que softwares proprietários de uma maneira geral.

O desenvolvimento massivo de tecnologia voltada para web, principalmente FLOSS no começo deste século, como linguagens de programação utilizadas na web (*php*, *python*), tecnologias para criação de bancos de dados, servidores e browsers, por exemplo, ajudou a transformar a web de uma maneira profunda, como veremos a seguir.

⁹ *Google Summer of Code* is a global program that offers student developers stipends to write code for various open source software projects. We have worked with several open source, free software, and technology-related groups to identify and fund several projects over a three month period. Since its inception in 2005, the program has brought together nearly 2500 successful student participants and 2500 mentors from 98 countries worldwide, all for the love of code. Through *Google Summer of Code*, accepted student applicants are paired with a mentor or mentors from the participating projects, thus gaining exposure to real-world software development scenarios and the opportunity for employment in areas related to their academic pursuits. In turn, the participating projects are able to more easily identify and bring in new developers. Best of all, more source code is created and released for the use and benefit of all. (<http://code.google.com/intl/pt-BR/soc/>)

¹⁰ Em 2006, a IBM já havia investido mais de 100 milhões de dólares no desenvolvimento do Linux e em outros projetos de software de código aberto”. Samuelson, 2006

2.7

Web 2.0: a rede dinâmica

Com a grande explosão inicial da web, houve um investimento massivo na rede, e várias companhias que começaram pequenas tiveram abertura de capital e suas ações extremamente valorizadas no final da década de 90. De janeiro de 1994 até fevereiro do ano 2000, o índice de composição da Nasdaq¹, subiu de 776.80 para 4696.69, um aumento de 605%, segundo artigo de James Galbraith e Travis Hale². No entanto, o grande volume de investimento nessas empresas acabou não resultando em um volume de operações lucrativas na mesma proporção, e de 2000 a 2001, a dita "bolha ponto com" explodiu, levando consigo os ganhos astronômicos acumulados nesse período.

Com essa súbita desvalorização, várias companhias, endividadas pelos seus próprios planos expansionistas, entraram em falência, gerando perdas de cerca de cinco trilhões de dólares em valor de mercado em apenas dois anos. No entanto, algumas grandes companhias que operavam online sobreviveram à bolha, e demonstraram que estavam amparadas em um modelo de negócios sustentável; entre elas a *Amazon*, ea *Ebay*

Blogs

A primeira fase da web foi marcada por um processo produtivo determinado. Para produzir conteúdo e colocar no ar, era necessário ter algum tipo de domínio sobre as tecnologias de publicação, ter acesso a um servidor e a um domínio, e transferir os arquivos que se desejasse para a web. O conteúdo era em geral estático e só mudava com uma nova atualização. As companhias

1 A Nasdaq é uma bolsa de valores americana especializada principalmente em empresas de alta tecnologia, informática e telecomunicações.

2 James K. Galbraith and Travis Hale (2004). Income Distribution and the Information Technology Bubble. University of Texas Inequality Project Working Paper.

que investiram em produção de conteúdo para web nessa primeira fase eram geralmente grandes e contratavam mão-de-obra especializada pra a publicação de conteúdo online.

Uma ferramenta que ajudou a mudar a realidade da web foram os primeiros blogs, que segundo Brandon Schauer³ (2005), surgiram ainda por volta de 1994. Através de uma programação específica que é executada no servidor, essas ferramentas são pioneiras em permitir aos usuários a publicação de conteúdo online sem a necessidade de se utilizar FTP⁴, ou seja, através de uma interface online, sem a necessidade de se editar arquivos HTML e fazer *upload* dos mesmos no servidor. Ao longo dos anos, serviços como esses foram se desenvolvendo, e em 1999 já existiam websites como o *blogger.com*, aonde qualquer usuário poderia criar um *blog* pessoal e colocar seu próprio conteúdo online.

Os blogueiros, tem algo em comum com os *hackers* apaixonados que desenvolvem software livre. Inicialmente, trabalham por paixão e não por dinheiro, o que de certo modo pode colocá-los em vantagem com relação aos seus competidores comerciais, como aponta Paul Graham em seu artigo "*What Bussiness Can Learn From Open Source*":

Como *open-source*, o *blogging* é algo que as pessoas fazem por si mesmas, de graça, porque gostam. Como *hackers* de código aberto, blogueiros competem com pessoas que trabalham por dinheiro, e muitas vezes ganham. O método para se garantir a qualidade também é o mesmo: darwiniano. Empresas garantem a qualidade através de regras para prevenir

os empregados de fazer besteiras. Mas você não precisa disso quando o público pode se comunicar um com o outro. As pessoas apenas produzem o que querem, as coisas boas se espalham as ruins ficam ignoradas. E em ambos os casos, o *feedback* do público ajuda o trabalho a melhorar.⁵

Com os *blogs*, as idéias podem emergir de baixo para cima, sem a intermediação dos proprietários dos meios de produção. Isto de certa maneira já ocorria em círculos como a *Usenet*, mas com os *blogs*, estes conteúdos produzidos pelos usuários podem atingir toda a rede de usuários da web. Os blogs inauguram de uma certa maneira o acesso universal a um tipo de comunicação "*um a muitos*", que desde o princípio da história dos meios de comunicação, esteve restrito aos donos dos meios de comunicação e a seus empregados. A forma de comunicação dominante, era, como aponta Graham (2005), do topo para baixo:

Há duas forças que, juntas, orientam o projeto: idéias sobre o que fazer a seguir, e o reforço da qualidade. Na era dos canais, todas escorriam do topo. Por exemplo, editores de jornais atribuíam histórias aos repórteres, e depois editavam o que eles escreveram.

Softwares open-source e blogs nos mostrar que as coisas não precisam funcionar dessa forma. Idéias e mesmo o reforço da qualidade, podem fluir de baixo para cima. E em ambos os casos, os resultados não são apenas

3 SCHAUER, Brandon. Experience Attributes: Crucial DNA of Web 2.0. Adaptive Path, dezembro de 2005. Disponível em: <<http://www.adaptivepath.com/ideas/essays/archives/000547.php>>

4 File Transfer Protocol, ou protocolo de transmissão de arquivos. Explicaremos melhor esse protocolo no quarto capítulo: A forma Oculta

5 Like open source, blogging is something people do themselves, for free, because they enjoy it. Like open source hackers, bloggers compete with people working for money, and often win. The method of ensuring quality is also the same: Darwinian. Companies ensure quality through rules to prevent employees from screwing up. But you don't need that when the audience can communicate with one another. People just produce whatever they want; the good stuff spreads, and the bad gets ignored. And in both cases, feedback from the audience improves the best work. Schauer, Brandon. Experience Attributes: Crucial DNA of Web 2.0. Adaptive Path, dezembro de 2005in: <<http://www.adaptivepath.com/ideas/essays/archives/000547.php>> (Graham, Paul. What Bussiness Can Learn From Open Source. in <<http://www.paulgraham.com/opensource.html>>

aceitáveis, mas melhores. Por exemplo, softwares de código aberto são mais confiáveis, precisamente porque é fonte aberta, e qualquer um pode encontrar erros.⁶

Uma linguagem *open source* que contribui imensamente tanto para o desenvolvimento das tecnologias para criações de blogs quanto como plataforma para vários tipos de aplicativos da web que povoam a web dinâmica dos dias de hoje é o PHP. Uma sigla para "*Personal Home Page Tools*", essa linguagem que começou a ser desenvolvida em 1995, permite que se criem aplicativos que ao invés de serem instalados no computador do usuário, como a maioria dos *softwares* comerciais de até então, são instalados e realizam todo seu processamento no computador que serve como servidor de internet. Com este tipo de recurso, conhecido como *server-side-scripting*⁷, é possível a construção de sites complexos, multi-páginas, através de um relacionamento com um banco de dados. O PHP não é a primeira tecnologia desse tipo a existir mas, por ser *open-source*, ampliou o acesso dos programadores a esse tipo de tecnologia e permitiu o surgimento de uma grande quantidade de experimentos em aplicativos web.

6 There are two forces that together steer design: ideas about what to do next, and the enforcement of quality. In the channel era, both flowed down from the top. For example, newspaper editors assigned stories to reporters, then edited what they wrote.

Open source and blogging show us things don't have to work that way. Ideas and even the enforcement of quality can flow bottom-up. And in both cases the results are not merely acceptable, but better. For example, open source software is more reliable precisely because it's open source; anyone can find mistakes. Graham, Paul. What Bussiness Can Learn From Open Source. Paul Graham. paulgraham.com: 2005. disponível em: <<http://www.paulgraham.com/opensource.html>>

7 Nas web atual, existem tanto aplicativos rodando no modelo *server-side-scripting*, aonde os aplicativos são executados no servidor, como *client-side-scripting*, baseados principalmente em JavaScript, aonde os aplicativos são transferidos para o computador do usuário.

Utilizando tecnologias desse tipo, pode-se criar apenas um modelo de cada página, e por meio do processamento no servidor, o código cria automaticamente as páginas HTML exibidas no *browser* do usuário, buscando o "*conteúdo*" em um banco de dados. O banco de dados pode ser alimentado diretamente pelos usuários, através de um processamento no servidor. O *Wordpress*, por exemplo, uma das mais utilizadas ferramentas para a construção de *blogs*, é construída a partir de PHP.

Web 2.0

Em 2005, Tim O'Reilly, que é o presidente da maior editora de livros técnicos dos EUA, a O'Reilly Media, lança um artigo intitulado "*What is Web 2.0 - Design Patterns and Bussiness models for the Next Generation of Software*". Nesse artigo, ele analisa a retomada da web pós-crash, e aponta uma série de fatores comuns a essa nova geração de websites, construídos a partir dessas tecnologias de *server-side-scripting*, e os batiza de "Web 2.0", um termo que já estava se difundindo pela rede. Este termo é polêmico; há quem diga — como o próprio fundador da web, Tim-Berners Lee — que é só uma questão de marketing, uma vez que a web continua, em sua estrutura, a mesma de sempre, como apontam Dimitri Kleiner e Brian Wyrick, no artigo *InfoEnclure 2.0*:

A internet (que é mais do que na web, na verdade) foi sempre sobre o compartilhamento entre os usuários. Na verdade, a Usenet, um sistema de mensagens distribuídas, está em funcionamento desde 1979! Desde muito antes mesmo da Web 1.0, tem acolhido debates, jornalismo 'amador', e permitindo o compartilhamento de fotos e arquivos. Como a Internet, é um sistema distribuído não detido ou controlado por ninguém. É esta qualidade, a falta de controle central e propriedade, que

a diferenciam os serviços como a Usenet de outros surgidos a partir da Web 2.0.⁸

O que a dita Web 2.0 faz é trazer, de certa maneira, essa experiência de comunidades de usuários para a web também de uma maneira geral. A Amazon, por exemplo, conseguiu se manter na liderança e atingir crescimento ao permitir aos usuários fazerem comentários sobre os livros que compravam, criarem listas temáticas e ao indicar livros semelhantes baseada no perfil de compras dos usuários. Assim, ao invés de pagar pessoal especializado para fazer esse tipo de serviço, ela economiza, utilizando seus próprios consumidores para agregar valor aos serviços oferecidos:

A empresa Web 2.0 muda fundamentalmente o modo de produção de conteúdos na Internet. Aplicações Web e serviços tornaram-se mais baratos e fáceis de implementar, e permitindo o acesso dos usuários finais a esses serviços, a empresa pode efetivamente terceirizar a criação e a organização dos seus conteúdos para os usuários finais. Ao invés do modelo tradicional de um provedor de conteúdo publicar seu próprio conteúdo e o usuário final consumindo-o, o novo modelo permite que o site da empresa atue como o portal centralizador entre os usuários que são os criadores e consumidores.⁹

8 The internet (which is more than the web, actually) has always been about sharing between users. In fact, Usenet, a distributed messaging system, has been operating since 1979! Since long before even Web 1.0, Usenet has been hosting discussions, 'amateur' journalism, and enabling photo and file sharing. Like the internet, it is a distributed system not owned or controlled by anyone. It is this quality, a lack of central ownership and control, that differentiate services such as Usenet from Web 2.0. Kleiner, Wyrick, 2007

9 A Web 2.0 company fundamentally changes the mode of production of internet content. Web applications and services have become cheaper and easier to implement, and by allowing the end users access to these applications, a company can effectively outsource the creation and the organisation of their content to the end users themselves. Instead of the traditional model of a content provider publishing their own content and the end user consuming it, the new model allows the company's site to act as the centralised portal between the users who are both creators and consumers. Kleiner e Wyrick, 2007.

1. Google
google.com	26. RapidShare
rapidshare.com
2. Facebook
facebook.com	27. 网易Yeah邮箱
163.com
3. YouTube
youtube.com	28. Я н д е к с
yandex.ru
4. Yahoo!
yahoo.com	29. Google 日本
google.co.jp
5. Windows Live
live.com	30. Google
google.com.br
6. Wikipedia
wikipedia.org	31. П о ч т а @Mail.ru
mail.ru
7. Blogger.com
blogger.com	32. FC2
fc2.com
8. Baidu.com
baidu.com	33. Flickr
flickr.com
9. Microsoft Network (MSN)
msn.com	34. LiveJasmin.com
livejasmin.com
10. Yahoo!
yahoo.co.jp	35. V Kontakte
vkontakte.ru
11. QQ.COM
qq.com	36. Google
google.it
12. Google India
google.co.in	37. The Internet Movie
Database
imdb.com
13. Twitter
twitter.com	38. Craigslist.org
craigslist.org
14. Myspace
myspace.com	39. Google España
google.es
15. Google
google.cn	40. LinkedIn
linkedin.com
16. 新浪新闻中心
sina.com.cn	41. AOL
aol.com
17. Google
google.de	42. Go
go.com
18. Amazon.com
amazon.com	43. Double Click
doubleclick.com
19. WordPress.com
wordpress.com	44. BBC Newsline Ticker
bbc.co.uk
20. Microsoft Corporation
microsoft.com	45. 搜狐
sohu.com
21. eBay
ebay.com	46. 1e100.net
1e100.net
22. Bing
bing.com	47. Photobucket image
hosting and photo sharing
photobucket.com
23. 淘宝网
taobao.com	48. Orkut.com.br
orkut.com.br
24. Google UK
google.co.uk	49. hi5
hi5.com
25. Google.fr
google.fr	50. Full Free Porn Videos
& DVDs - Sex, Porno, Porn
Tube, Free XXX Porn
pornhub.com

Lista dos 50 websites mais visitados do mundo, segundo o site Alexa.com
<http://www.alex.com/topsites/global>

Esse princípio é um princípio comum compartilhado entre vários dos maiores sites em número de visitação atualmente, como podemos ver na lista dos 50 sites mais visitados na página ao lado, segundo o site *Alexa*, que agrega estatísticas de visitação de sites por todo o mundo; a grande maioria deles são sites de forte conteúdo agregado pelos usuários.

O próprio *Google*, que em suas várias versões locais de todo o mundo ocupa várias das primeiras posições do ranking, é também um deles. Seu mecanismo de busca conta com um algoritmo que leva em conta as escolhas dos usuários para a posição de um determinado site no seu mecanismo de busca.

Google

O'Reilly considera a *Google* a grande "estandarte" da Web 2.0. Sua abertura de capital, diz ele, foi o evento definidor dessa nova era, como a abertura da Netscape, — empresa líder do mercado de navegadores durante parte da década de 90 — havia sido para a Web 1.0 — como ele chama a rede antes dessa nova era de aplicativos online.

A *Google* começa sua trajetória rumo ao domínio do mercado de mecanismos de buscas oferecendo um aplicativo funcionando totalmente online, seu poderoso mecanismo de busca, que funciona hoje como o grande agregador de páginas, e, para muitos usuários, é a porta de entrada para a web. A *Google*, ao invés de oferecer um produto, como um *software* instalado no computador do usuário, oferece apenas um serviço, e sustenta sua competência através de um imenso

banco de dados, aonde classifica e organiza centenas de bilhões de páginas ao redor do mundo.¹⁰

A empresa criou um sistema inovador de publicação de anúncios, o *AdSense*, que também é o motor de muitas das empresas da web 2.0. Através desse serviço, uma empresa pode embutir anúncios de terceiros em seu site, que lhe gera receita líquida a cada clique que um dos seus leitores faça. As empresas anunciantes têm a possibilidade de escolher certas palavras mais relevantes para o posicionamento de seus anúncios, e o conteúdo dos anúncios é de uma certa maneira relacionado ao conteúdo do website que se está visitando. Qualquer pessoa física, de qualquer lugar do mundo pode facilmente fazer um contrato de venda de espaço para a *Google* através desse sistema, e assim, centenas de milhares¹¹ de websites se sustentam.

Posteriormente, a companhia ampliou seu leque de serviços, oferecendo serviços de e-mail, aplicativos como editores de texto online, um sistema cartográfico mundial, redes sociais, um sistema operacional e browser e hoje amplia seus horizontes para aparelhos de celular. A *Google* começou também um processo de escaneamento de todos livros já escritos, e fez um acordo com a *Authors Guild*, entidade de arrecadação de direitos autorais nos Estados Unidos, para distribuir

10 The first Google index in 1998 already had 26 million pages, and by 2000 the Google index reached the one billion mark. Over the last eight years, we've seen a lot of big numbers about how much content is really out there. Recently, even our search engineers stopped in awe about just how big the web is these days — when our systems that process links on the web to find new content hit a milestone: 1 trillion (as in 1,000,000,000,000) unique URLs on the web at once! <http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html>

11 Segundo o Google AdSense Usage Statistics do website build with, "We know of an estimated 594,773+ websites that are using Google AdSense", mas isto é certamente menos do que o número real de websites, uma vez que a base de pesquisa desse site não é toda a web <http://trends.builtwith.com/ads/Google-AdSense>

online, de maneira paga o conteúdo de livros que estejam fora de catálogo¹².

Com aplicativos online como o *Google Docs*, é possível se acessar arquivos remotamente a partir de qualquer computador com internet, e compartilhar a edição de arquivos com outras pessoas. Em troca, de todos esses serviços, a companhia monitora os *e-mails* de todos usuários, afim de, segundo seus termos de serviços, direcionar seu sistema de anúncios de acordo com os hábitos individuais de cada um.

O conteúdo de um novo meio é o conteúdo de todos os meios anteriores

A web 2.0 inaugura não só um novo modelo de negócios para as companhias, mas toda uma nova forma de apropriação da web pelos usuários. Com esses aplicativos online, foi possível contruir uma nova geração de websites, aonde o conteúdo gerado pelos usuários é a força motriz dessa nova economia. Se antes a web era construída por pessoas que tinham que ter um certo domínio de linguagens de marcação ou, no mínimo, de protocolos de transferência de arquivos com essas novas ferramentas qualquer usuário com noções básicas de informática pode se transformar em produtor de conteúdo online.

Essa é a grande base do *Youtube*, por exemplo, ferramenta onde os usuários podem fazer o *upload* de vídeos caseiros, pessoais, ou então de programas gravados de qualquer origem. O *Youtube* é um dos grandes fenômenos de comunicação dessa nova era. Com esse gigante banco de dados construído pelos usuários do site ao longo dos anos, boa parte de toda produção televisiva e musical de todas as décadas precedentes está

disponível gratuitamente, a um simples clique para qualquer um. Inaugura-se um novo momento na cultura de massas; novas celebridades são criadas; notícias são espalhadas; momentos históricos são resgatados. E tudo isso independente de um controle central, como havia na televisão até então.

Quando o livro surgiu, a maioria dos impressos eram obras medievais que foram publicadas e ganharam nova dimensão com esse novo meio¹³. Com o *Youtube* (assim como aplicativos semelhantes), o processo é semelhante. O conteúdo é a princípio, o conteúdo dos meios anteriores — da televisão, do rádio, do cinema —, mas, com a velocidade das transformações da nossa era, o processo de construção do novo já começa a surgir. Em todo o mundo, milhares de indivíduos se colocando como agentes de sua própria história, se expondo e contruindo suas relações nesse novo ambiente social.

Criam-se as bases para uma nova forma de jornalismo; como foi o caso do acompanhamento das últimas eleições presidenciais no Irã, aonde uma série de indivíduos contribuíram para a divulgação dos protestos através de vídeos no *Youtube* e comentários no *Twitter*. A televisão já sente o impacto dessa concorrência, assim como os jornais impressos e as revistas. Mas se todas essas possibilidades podem apontar para uma perspectiva de liberdade dos meios, também tem aspectos contraditórios, como apontam Kleiner e Wyrick no artigo *infoenclosure 2.0*:

Web 2.0 é o boom de investimento na internet 2.0. Web 2.0 é um modelo de negócio, que significa a captura privada de valor criado pela comunidade. Ninguém nega que a tecnologia de sites como o YouTube, por exemplo, é trivial. Isso é mais do que comprovado pelo

12 O Acordo, segundo Doctorow, dá à companhia um virtual monopólio sobre toda a literatura americana. Doctorow, Cori. Google Book search gives Google a virtual monopoly over literature. BoingBoing.net <<http://boingboing.net/2009>

13 McLuhan, 1967

grande número de serviços idênticos, como DailyMotion. O valor real do YouTube não é criado pelos desenvolvedores do site, mas sim pelas pessoas que fazem upload de vídeos no site. No entanto, quando o YouTube foi comprado por mais de um bilhão de dólares em ações pelo Google, quanto deste valor foi repassado àqueles que fizeram todos esses vídeos? Zero. Zilch. Nada. Um bom negócio se você é um proprietário de uma empresa de Web 2.0.

O valor produzido por usuários da Web 2.0 em serviços como o YouTube é capturado por investidores capitalistas. Em alguns casos, o conteúdo concreto do material é de propriedades de terceiros. Apropriação privada do valor criado pela comunidade é uma traição à promessa de tecnologias de compartilhamento e cooperação livre.¹⁴

De fato, a construção do capitalismo, como aponta Marx em *O Capital*, é baseada na idéia de expropriação do trabalho de terceiros. A web 2.0 nesse sentido, leva tudo a um novo patamar. Agora, ao invés de pagarem baixos salários a seus trabalhadores, as companhias contam com um exército de voluntários, que trabalham pelo puro prazer de compartilhar seus interesses com outros e se tornarem também, de certa maneira, produtores.

Em troca da infra-estrutura de rede, e do livre acesso, esses voluntários abastecem redes privadas com valor, *for free*.

14 Web 2.0 is Internet Investment Boom 2.0. Web 2.0 is a business model, it means private capture of community-created value. No one denies that the technology of sites like YouTube, for instance, is trivial. This is more than evidenced by the large number of identical services such as DailyMotion. The real value of YouTube is not created by the developers of the site, but rather it is created by the people who upload videos to the site. Yet, when YouTube was bought for over a billion dollars worth of Google stock, how much of this stock was acquired by those that made all these videos? Zero. Zilch. Nada. Great deal if you are an owner of a Web 2.0 company.

The value produced by users of Web 2.0 services such as YouTube is captured by capitalist investors. In some cases, the actual content they contribute winds up the property of site owners. Private appropriation of community created value is a betrayal of the promise of sharing technology and free cooperation. Kleiner e Wyrick, op. cit.

3

A forma aparente

3.1

A linguagem do design

Vimos brevemente como se deu parte do desenvolvimento das tecnologias de comunicação ao longo da história, e como o computador se transformou de uma ferramenta militar para um meta-meio, que cumpre papel de ferramenta em todos os campos da economia, da cultura e do lazer e como vem se tornando, nos últimos tempos, o mais importante meio de comunicação e difusão das informações numa escala global. Partiremos agora para uma análise mais objetiva da forma da web, partindo, como já foi dito, da forma aparente.

Embora a web seja portadora de uma série de formas de comunicação, como podcasts e vídeos, a forma dominante é a comunicação visual, e mesmo elementos de áudio e vídeo, normalmente acabam inseridos dentro de uma interface que é sobretudo gráfica. Antes de entrarmos nas especificidades da interface web, que leva em conta a interação do usuário, é importante compreendermos o design gráfico como linguagem e, como linguagem, quais são seus elementos constitutivos.

Comunicação Visual

Em *Design e Comunicação Visual*, Bruno Munari declara que "praticamente tudo o que nossos olhos vêem é comunicação visual"¹ mas que no entanto, entre todas as mensagens que nossos olhos recebem como estímulo, é possível se distinguir entre a comunicação casual e a comunicação intencional. Num processo intencional "a comunicação deveria ser recebida na totalidade do significado pretendido pelo emissor."²

Conceitos como emissor, receptor, mensagem e informação, importantes para a compreensão

1 MUNARI, Bruno. *Design e comunicação visual* Lisboa. Edições 1970, pg 65.

2 Ibid.

do processo de comunicação como um todo — comunicação visual aí incluída — foram inicialmente desenvolvidos por Claude Shannon, nos laboratórios da companhia telefônica norte-americana Bell. Shannon, junto a Warren Weaver, criou um dos primeiros modelos de comunicação que se utiliza para explicar o funcionamento de sistemas de comunicação (figura 1). A comunicação acontece aí entre um emissor, que emite uma determinada "mensagem"; essa mensagem sofre influência do meio, o ruído, e é recebida pelo receptor. Este modelo é um tanto mecanicista, pois foi desenvolvido para aplicação em sistemas de comunicação elétricos, como a telefonia. A questão é que este modelo não permite a múltipla interpretação de uma determinada mensagem, o que ocorre na maioria das formas de comunicação entre humanos, e quase nunca (a não ser em caso de erro), na comunicação entre máquinas. Alguns teóricos, no entanto, procuraram desenvolver modelos semelhantes ao de Shannon, mais adequados à aplicação à comunicação humana, inclusive a visual.

Figura 1. Modelo esquemático de um sistema geral de comunicação. fonte: Shannon e Weaver. *The Mathematical Theory of Communication*.

Figura 2. Modelo de comunicação desenvolvido por Umberto Eco, inclui conceitos como sobrecodificação e subcodificação, bem como um esforço do leitor em reconstruir o código do Autor. fonte: Wallerr, Robert. *The Typographic Contribution to Language*

Um deles foi Humberto Eco, que propõe um modelo (figura 2) que leva em conta o contexto e o leitor, com um "esforço filológico de reconstruir o código do remetente". Eco propõe um conceito de subcodificação, que é quando o leitor deve fazer uma interpretação provisória quando se depara com o incerto, e sobrecodificação que descreve o uso de frases feitas e clichês e padrões que podem diminuir a possibilidade de interpretação correta pelo destinatário. Eco remete a instâncias contextuais (onde pode se encaixar a tipografia) como instâncias de sobrecodificação que permi-

tem ao destinatário a escolha do subcódigo apropriado.³

Pensando mais especificamente no papel do design gráfico no processo comunicativo, Robert Waller, em sua tese "*The Typographic Contribution to Language*", desenha um modelo diferente (figura 3), onde uma determinada informação passa por três estágios diferentes de existência: escrita, publicação e leitura. Nele, a comunicação não ocorre diretamente entre o leitor e o autor. Há um processo de comunicação entre o autor e um leitor imaginário, seguido de uma quebra no tempo e espaço — que é onde o trabalho do designer se insere — e uma comunicação entre o leitor e um autor imaginário, cuja imagem os aspectos formais do texto ajudam a construir.

Figura 3: Modelo de Comunicação proposto por Robert Waller, leva em conta uma mudança de tempo e espaço — assim como de forma — que separa o texto do escritor do texto do leitor. fonte: Waller, op cit.

3 WALLER, Robert - The typographic contribution to Language. Towards a model of typographical genres and their underlying structures. unpublished PhD thesis, Department of Typography and Graphic Communication, University of Reading, August 1987. pg 169. <http://www.fb10.uni-bremen.de/anglistik/langpro/projects/gem/robwaller.html>

Principalmente no momento em que essa comunicação ocorre entre um produtor e um consumidor de uma determinada mercadoria — como entre uma editora de um livro e um leitor, ou entre uma indústria que fabrica determinado produto e seu consumidor — essa construção da imagem do autor passa a ter um papel na constituição do valor do objeto. Neste momento, surge a figura do profissional especializado em fazer esse papel: o designer.

No Brasil, a profissão "designer", seja gráfico ou industrial, não é regulamentada, ou seja, é uma profissão que não existe juridicamente perante as leis brasileiras. Sendo assim, não existe formalmente um conjunto de disciplinas e conhecimentos que o designer deve ter para o exercício da profissão, e qualquer um pode se considerar designer por auto-declaração, apesar de existir uma grande quantidade de cursos de nível superior ou técnico voltados para a formação profissional. É uma disciplina de caráter misto, tendo aspectos de arte e aspectos de ciência, e por essa falta de regulamentação acaba também sofrendo com a competição de uma série de outros profissionais, como aponta Alexandre Wollner, no texto "O depoimento de um pioneiro":

"A princípio, todo mundo pode ser um designer. Isto só depende de treinamento. Quando, por exemplo, um empresário precisa de design, mas não sabe exatamente o que vem a ser isso, manda chamar alguém que tenha talento e contrata um artista — que é intuitivo, mas não está preparado para aplicar a tecnologia. Ele também pode chamar um diretor de arte, um publicitário. Este, em geral, tem formação um pouco mais experimentada, porque trabalha com jornais, impressos, televisão. Em parte domina a tecnologia. Mas a usa de uma maneira aleatória e lida com o efêmero. Pode ser bom para conceber anúncios, mas não para fazer design. Depois temos o arquiteto, que recebe alguns preceitos de design. Mas ele só adquire informações que o auxiliam a desenvolver projetos um pouco mais elaborados que o artista. (...)

O designer propriamente dito produz para um público amplo. tem um talento de ordem intuitiva. Como é um artista que se interessa por formas, tem que complementar suas habilidades natas com tecnologia e ciência, pois deve alcançar o equilíbrio entre a intuição e a

técnica. Se for só técnico, vira engenheiro; não pode ser só intuitivo, senão vira arista. Vai precisar de um ferramental técnico, de uma linguagem, de uma tecnologia, e terá que saber falar, comunicar-se, explicar o conceito para o cliente, justificar determinadas escolhas e caminhos. "⁴

Tradicionalmente, como aponta Amy Arnston, o design gráfico é definido como "a resolução de problemas em uma superfície lisa, bi-dimensional"⁵. Mais recentemente, com a introdução do mercado de webdesign, incluíram-se questões relacionadas ao tempo e à interação, que trataremos no próximo item deste capítulo. O designer, ainda segundo ela "concebe, planeja e executa" peças que "comuniquem uma mensagem específica a uma audiência específica com certas limitações", que podem ser "financeiras, físicas ou psicológicas". Mas a essência do design é a comunicação.

Resolução de problemas, no caso do designer, significa em geral planejar, organizar informações de maneira criativa, econômica, e que satisfaça na medida do possível, as necessidades do cliente — que no modelo de comunicação de Waller é representado pelo autor — e dos usuários, ou leitores. Para tanto, é necessário, como apresentou Wollner, o domínio de um ferramental técnico, e o domínio de uma linguagem, que inclui certos fundamentos teóricos.

Linguagem

Antes de nos dedicarmos a compreender o que pode ser essa linguagem do design, cabe entender o que é que define uma linguagem. Phillip Meggs em *Type and Image: The language of Graphic Design*, afirma que linguagem:

"é um meio sistemático de comunicação através de signos (sons, gestos visuais ou marcas) que tenham significados compreendidos"⁶

Um signo, de acordo com a definição de Pierce, "é aquilo que, sob certo aspecto ou modo, representa algo para alguém"⁷ e ainda "O signo representa alguma coisa, seu objeto". Para realizar qualquer processo comunicativo, precisamos de representações de um objeto para podermos nos referir a ele. Isto, o signo, pode ser uma palavra, uma imagem ou um símbolo.

Signos são classificados, segundo a semiótica de Pierce, em três categorias distintas, como descreve Décio Pignatari em *Informação Linguagem Comunicação*:

"Ícone, quando possui alguma semelhança ou analogia com o seu referente. exemplos: uma fotografia, uma estátua, um esquema, um pictograma.

Índice ou índice, quando mantém uma relação direta com o seu referente, ou a coisa que produz o signo. Exemplos: chão molhado, indício de que choveu; pegadas, indício de passagem de animal ou pessoa; uma perfuração de bala, uma impressão digital etc.

Símbolo, quando a relação com o referente é arbitrária, convencional. As palavras, faladas ou escritas, em sua maioria, são símbolos. Quando eu pronuncio os fonemas

4 Wollner, em ADG. O valor do Design. São Paulo: SENAC, 2004

5 Arnston, Amy. *Graphic Design Basics*. Belmont: Thomson Wadsworth, 2007 pg 4.

6 Meggs, Philip. *Type and Image: The Language of Graphic Design*. Devers: John Wiley & Sons inc, 1992

7 PEIRCE, Charles Sanders - *Semiótica*. São Paulo, Perspectiva, 1977 pg 46.

correspondentes a mesa, por exemplo, o som complexo que eu emito designa um objeto por convenção estabelecida (...)»⁸

Toda linguagem pressupõe um código, ou como chama a semiótica da cultura, sistemas modelizantes, que, nas palavras de Irene Machado:

podem ser entendidos como sistemas de signos, como conjunto de regras (códigos, instruções, programas) para a produção e textos no sentido semiótico amplo e como totalidade de textos e suas funções correlatas.⁹

Iuri Lotman, importante teórico desta disciplina, afirma que:

todo o sistema que serve os fins da comunicação entre dois ou vários indivíduos pode ser definido como uma linguagem.¹⁰

E ainda que:

por linguagem, entendemos todo sistema de comunicação que utiliza signos ordenados de modo particular.¹¹

Assim sendo, a arte pode ser compreendida como linguagem, e como a arte, também as artes gráficas e o design.

A linguagem gráfica, no entanto, diferencia-se claramente da linguagem verbal, aonde é possível, dentro de um certo limite, obter-se uma interpretação determinada de uma dada mensagem. Na linguagem verbal, não existe uma relação direta entre estímulo e significado, salvo no caso de poucas exceções, como no caso de

diagramas e sinais de trânsito, imagens e composições visuais são pansêmicas, ou seja, permitem múltiplas interpretações¹². Sendo assim, na comunicação visual não existem regras, como aponta Donis Dondis:

o que existe, é um alto grau de compreensão do que vai acontecer em termos de significado, se fizermos determinadas ordenações das partes que nos permitam organizar e orquestrar os meios visuais. Muitos dos critérios para o entendimento do significado na forma visual, decorrem da investigação do processo da percepção humana.¹³

A investigação da percepção humana

O estudo da percepção humana até agora foi melhor desenvolvido como ciência por um campo disciplinar conhecido como psicologia da gestalt. *Gestalt*, nas palavras de Rudolph Arheim :

substantivo comum alemão, usada para configuração ou forma tem sido aplicada desde o início do nosso século [século XX] a um conjunto de princípios científicos extraídos principalmente de experimentos de percepção sensorial.¹⁴

12 Segundo Bertin, como cita Robert Waller, imagens podem ser monossêmicas, polissêmicas ou pansêmicas, de acordo com a quantidade de interpretações que se pode ter delas:

Not all media are equally explicit, and Bertin restricts his analysis to rulebound diagrammatic and cartographic images on the grounds that they are monosemic, as distinct from polysemic or pansemic. Monosemic, polysemic and pansemic images are those which are, respectively, capable of only one interpretation, capable of several interpretations, and capable of an infinite number of interpretations. Pansemic images are comparatively rare (abstract painting might be an example). In practice most but not all pictures are polysemic, although as we have noted they are usually anchored to one interpretation by a caption. Diagrams are (in Bertin's opinion, at least) said to be monosemic, so long as they make use of explicitly coded graphic conventions.

13 Dondis, Donis A. *Sintaxe da Linguagem visual*. São Paulo: Martins Fontes, 2007 pg 29.

14 ARNEHEIM, Rudolf - *Arte e percepção visual: psicologia da visão criadora*. São Paulo, EDUSP. 1078 introdução.

8 PIGNATARI, Décio - *Informação, linguagem, comunicação*. São Paulo, Perspectiva, 1968 pgs 30-31

9 Machado, Irene. *Semiosphera — Espaços semióticos compartilhados*. <http://www.usp.br/semiosphera/>

10 Lotman, Iuri. *A estrutura do texto artístico*, Lisboa, Editorial Estampa, 1978

11 Ibid.

Psicólogos gestaltistas, como Wolfgang Köhler, Kurt Kofka e Max Wertheimer, desenvolveram uma série de experimentos em laboratório para tentar compreender os princípios da percepção humana. Segundo Kofka, "gestalt é o produto de uma organização; a organização é o processo que leva a uma gestalt"¹⁵ e ainda "a organização, como uma categoria, é diametralmente oposta à mera justaposição ou distribuição ao acaso".

A *gestalt* conseguiu listar alguns fenômenos — que alguns como Kofka, chamam de "leis" e outros, como Arnston preferem chamar de "princípios" — que determinam como a organização das formas em um determinado campo é percebida pelos homens. A máxima gestaltista, divulgada amplamente em manuais de design¹⁶ prega que "o todo é maior que a soma das partes", e nos permite constatar que "a aparência de qualquer elemento depende de seu lugar e da sua função num padrão total", segundo Arheim. Assim, uma palavra, por exemplo, é mais do que a soma de suas letras.

Essas partes, interagem umas com as outras por meio do que os gestaltistas chamam de tensões, como expõe Arheim:

A experiência visual é dinâmica. [...] O que uma pessoa ou animal percebe não é apenas um arranjo de objetos, cores e formas, movimentos e tamanhos. É talvez, antes de tudo, uma interação de tensões dirigidas. Essas tensões não constituem algo que o observador acrescenta, por razões próprias, a imagens estáticas. Antes, estas tensões são inerentes a qualquer percepção como tamanho, configuração,

localização ou cor. Uma vez que as tensões possuem magnitude e direção, pode-se descrevê-las como "forças" psicológicas¹⁷

Os objetos possuem, segundo Arheim, linhas de tensão, representadas na terceira figura. Assim, um objeto que esteja deslocado em relação ao centro, como na segunda figura, parece mais tenso do que se estivesse em algum dos pontos centrais. fonte: Arheim, op. cit. pgs 1 - 6

15 Koffka, Kurt. Princípios da Psicologia da Gestalt. São Paulo: Cultrix, 1975 pg 691

16 The gestalt school of psychology, which began in Germany around 1912, investigated how we see and organize visual information into a meaningful whole. The conviction developed that the whole is more than the sum of its parts. This whole cannot be perceived by a simply addition of isolated parts. Each part is influenced by those around it". Arnston, 2007, pg 78

17 Arheim, Op Cit. pg 4

A idéia é de que o plano visual contém forças similares às forças da gravidade, como a força de atração entre objetos, o equilíbrio de elementos no campo, o peso dos elementos, mas não exatamente como no mundo físico. No ambiente da psicologia da percepção, a cor pode influenciar no peso do objeto, por exemplo. O equilíbrio, é segundo Dondis, "a referência visual mais forte e firme do homem"¹⁸. Os processos de organização, e reconhecimento de tensões e equilíbrio são, ainda segundo ela, processos intuitivos e inconscientes. As composições estruturadas de forma mais tensa, ou seja, fora dos eixos predominantes de equilíbrio (eixo vertical central) são consideradas por ela como mais atrativas para os espectadores, uma vez que possuem mais tensão.

A *gestalt* tem como principal "lei" a *prägnanz*, que pode ser formulada, segundo Koffka, da seguinte maneira:

a organização psicológica será sempre tão 'boa' quanto as condições reinantes permitirem. Nesta definição, o termo 'boa' é indefinido. Abrange propriedades tais como a regularidade, a simetria, a simplicidade, e outras¹⁹

Essas propriedades, ele descreve ao longo de seu livro, *Princípios de psicologia da Gestalt*. Todos os tais princípios da *gestalt*, como continuidade, proximidade, fechamento e similaridade derivam, de uma certa maneira, dessa lei da *prägnanz* (prägnanz). Através de uma série de exemplos, a teoria da *gestalt* explica então como se dá sua máxima, que revela que:

o todo é alguma coisa outra que a soma de suas partes, porque somar é um procedimento vazio de significado
(...)

A significação, para a percepção, se encontra na relação entre o todo e as partes.²⁰

Nesse sentido, a organização e a consequente 'boa forma', não pode ser resolvida separadamente a partir de cada objeto, mas somente pode ser conseguida através de soluções para o todo.

A seguir, apresentamos graficamente alguns desses princípios de conformação da "boa forma".

Na figura acima, tendemos a enxergar um círculo sobre um quadrado cinza. Na nossa percepção, o quadrado está completo. A mesma imagem poderia ser formada por um quadrado furado, mas tendemos a conceber formas mais simples

18 Dondis, Op Cit pg 32

19 Koffka, op cit pg 121

20 Koffka, op cit. pg 186

O Princípio da chamada Ascii art, arte feita a partir de caracteres ASCII base-se no princípio da gestalt de que o todo é maior do que a soma das partes. Nesta página, nossa percepção constrói a imagem das pernas de uma mulher, aonde em essência, existem só caracteres alinhados

<http://www.textfiles.com/art/>

Nesta capa de Dick Bruna, apesar de não haver nenhuma linha diferenciando as árvores, podemos enxergar claramente na imagem quatro árvores, e não uma figura irregular com oito pontas. Isto porque nossa percepção tende a enxergar formas mais simples.

Aqui, o designer Nicolas Feltron utilizou o princípio da continuidade para gerar uma forma coesa a partir de uma série de formas triangulares.

A artista Tauba Auerbach utiliza o conceito de figura e fundo para criar a imagem acima, aonde é impossível de se distinguir o que é figura e o que é fundo

A percepção tende a agrupar elementos que estejam próximos uns dos outros. Na figura acima, enxerga-se claramente dois grupos de figuras.

A figura acima é um exemplo clássico da teoria da gestalt. Apesar de não existir formalmente, nossa percepção constrói a imagem do cubo a partir da continuidade de suas arestas.

Assim como a proximidade, a similaridade entre as figuras também pode ser um fator de agrupamento. Na figura acima, distinguem-se claramente dois grupos de objetos diferentes.

Cor

Cor, assim como forma é um dos elementos essenciais da linguagem da comunicação visual. Toda diferença de forma perceptível é resultado de uma diferença percebida de cor, na verdade variações de frequência luminosa, percebidas pelo nosso aparato ótico como cor. Tais frequências, podem ser geradas de duas maneiras: ou pela reflexão de faixas de frequências luminosas, ou pela irradiação das mesmas.

No meio impresso, cores são formadas pela soma de pigmentos; quanto menos pigmento, mais clara, e a cor só pode ser percebida se houver luz ambiente; num sistema cromático chamado de aditivo. No processo de impressão tradicional, as cores são geradas pela soma de quatro pigmentos: ciano, magenta, amarelo e preto, ou sistema CMYK.²¹

No meio digital, a cor geradas pelos televisores e monitores de computador, por outro lado, é gerada pela soma de pequenas luzes; quanto mais acesas, mais claro, num sistema conhecido como aditivo. As cores são geradas por uma soma de

luz vermelha (red), verde (green) e azul (blue), ou sistema RGB.

As cores complementares de cada sistema formam as cores básicas do outro. Justapondo os dois esquemas de cor, construímos o círculo cromático, que serve como “mapa básico de relação entre as cores”²². As cores do amarelo ao violeta, passando pelo vermelho, são chamadas de cores quentes; do amarelo até o violeta, passando pelo azul, são consideradas cores frias. Dependendo do tom e das demais cores da composição, amarelo e violeta podem ser considerados tanto quentes como frias.

21 Lupton, Ellen. Novos Fundamentos do Design pg 76

22 Ibid, pg 77

As cores podem variar quanto ao tom, saturação e brilho. Tom, é o aspecto que os leigos denominam como cor, na verdade corresponde a uma diferença de comprimento de onda das frequências luminosas. Saturação varia a vividez da cor, de uma cor mais pura e sólida, para o cinza. Brilho é a quantidade aparente de luz que a cor emite, variando de preto a branco. Na imagem a seguir, a saturação corresponde à variação horizontal, enquanto a de brilho corresponde à variação vertical.²³

²³ Color Usage research lab. Nasa Ames research center. Using color on Information Display graphics. http://colorusage.arc.nasa.gov/color_concepts_1.php

Dentro do espectro de ondas eletromagnéticas existente, a faixa que corresponde à luz visível é uma faixa muito pequena, que começa no vermelho e termina no violeta.

Das frequências eletromagnéticas existentes, nosso aparato de visão só percebe uma faixa pequena

Quanto mais curto o comprimento da onda que produz a cor, melhor ela se mistura com um cinza de mesma luminância.²⁴ O vermelho, é a cor que mais segrega, e o azul a que menos segrega. Koffka chama as cores que mais segregam, como o amarelo, o laranja e o vermelho de cores duras, e as cores que mais se misturam ao cinza, como azul, verde e violeta de cores brandas. O branco é considerado mais duro do que o preto. As cores

²⁴ Koffka, op. cit. pg 137

duras, são sobre o cinza, como que dois líquidos de densidades diferentes que não se misturam²⁵, e a zona delimitadora entre esses dois campos gera uma tensão na percepção visual, ou contraste.

Existem dois tipos de contraste, um é o contraste de brilho, que aumenta conforme a diferença de luminosidade entre as cores, outro, é o contraste de cor, que aumenta conforme a diferença de posição das cores na roda cromática. O contraste é gerado pois as cores tendem a formar uma área de ação oposta ao seu redor. Observando um círculo vermelho, por exemplo, pode-se perceber pequenas manchas esverdeadas ao seu redor.²⁶ Por essas tensões, uma cor pode variar de aparência dependendo da cor que estiver ao seu redor.

Um conjunto de cores escolhidas para uma determinada aplicação é chamado de palheta. De acordo com a posição das cores em relação ao círculo cromático, a palheta pode ser:

Monocromática, se for formada por variações de valor do mesmo tom;

Complementar, se for formada por cores contrastantes;

Ttriádica, se for formada por uma cor oposta à intermediária entre as duas outras

Tetrádica, se formada por 4 cores base, opostas duas a duas;

Análoga, se as cores estiverem próximas umas das outras;

25 Ibid.

26 Ibid. pg 144

Análoga acentuada, se formada por uma série de cores análogas mais uma complementar a elas;

Ou aleatória, se não houver relação entre elas;

O Color Scheme Designer é um aplicativo online que produz palhetas cromáticas a partir de uma roda de cores interativa, foi utilizado para produzir os esquemas de cor ilustrativos acima, pode ser encontrado no endereço <http://colorscheme-designer.com/>

Há quem diga, como Eric Danger, que cores podem sugerir certas sensações psicológicas como:

Vermelho: calor, paixão, excitação, fogo, inverno

Azul: frescor, água, mar, verão, homens.

Verde: campo, frescor, descanso, primavera.

Amarelo: sol, calor energia

Cinza: dignidade, calma

Branco: asseio

Rosa: gentileza, mulheres ²⁷

Essas definições, no entanto, são mais geradas por convenções sociais do que por características inerentes das cores como fenômeno físico, como ressalta Lupton:

Da mesma forma [que pode variar em função das outras ao redor dela], a cor muda de sentido de uma cultura para outra. As cores carregam diferentes conotações em diferentes sociedades. O branco representa virgindade e pureza no Ocidente, porém, é a cor da morte nas culturas orientais. O vermelho, usado pelas noivas no Japão, é considerado extravagante e erótico na Europa e nas Américas. As cores entram e saem de moda, e toda uma indústria foi criada para guiar e prever seu curso.²⁸

Imagens

As primeiras formas de comunicação visual que se têm registro, eram representações de aspectos da vida pré-histórica em forma de imagens. Desde os primórdios da comunicação, a imagem cumpre um papel essencial na transmissão de certos tipos de conteúdo.

Imagens são superfícies; em essência, signos contínuos, onde as unidades de conteúdo não podem ser separadas umas das outras; mas todas informação digital é composta de signos discretos, caracteres alfanuméricos. Para que se possa utilizar imagens em ambientes digitais, foi preciso criar formas de tradução — sistemas de codificação — capazes de transformar as imagens em signos discretos.

Existem duas formas básicas de se fazer isso, que geram as duas famílias principais de imagens existentes no ambiente digital:

A primeira forma, é decompor (ou compor) as imagens em termos de figuras abstratas, ou objetos, que podem ser descritos em termos de suas características geométricas. São assim construídas as chamadas imagens vetoriais, construídas a partir de descrições matemáticas de suas características físicas. Um exemplo de linguagem

²⁷ Danger, Eric, A cor na comunicação. Rio de Janeiro: Forum, 1973 pg 27

²⁸ Lupton, op. cit. pg 71

vetorial é o SVG. Um arquivo SVG, por ser composto a partir de XML, chega a ser compreensível por humanos, quando inspecionado seu código:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://
www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" id="Layer_1" xmlns="http://www.
w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/
xlink" x="0px" y="0px" width="212.6px" height="212.6px"
viewBox="0 0 212.6 212.6" enable-background="new 0 0
212.6 212.6" xml:space="preserve">
<circle fill="#FF0000" cx="100" cy="70" r="50"/>
</svg>
```

Num arquivo vetorial, o tamanho da imagem não dita o tamanho do arquivo, mas sim sua complexidade. Quanto mais objetos descritos, maior é o arquivo, e o tamanho de exibição é independente.

Outra forma de compor imagens é tratando-as como um conjunto de pontos de cor variante. Imagens construídas assim são conhecidas como *bitmap*, e são formadas a partir de um mapeamento de cada ponto, cada um com um valor determinado de cor. Diferentes formatos têm diferentes níveis de compactação e precisão. Imagens construídas a partir de um mapa de *bits*, estão sempre restritas a um determinado tamanho, e a uma determinada resolução, que equivale à quantidade de pontos existentes em uma polegada. Cada suporte tem uma resolução limite diferente; a tela, por exemplo, tem em geral 72 ou 92 dpi, ou *dots per inch* (pontos por polegadas), enquanto uma folha impressa pode ter 300 ou 600 pontos por polegada.

Quanto mais pontos tiver uma imagem, maior o tamanho do arquivo resultante de sua codificação. Existem uma série de tipos de compactação que podem ser empregadas para diminuir o tamanho de uma imagem. Na web, os formatos aceitos são: gif, jpg e png. Gif e png são formatos com suporte para transparência, png pode suportar também graus de transparência diferentes.

Abaixo, a mesma imagem convertida em uma série de formatos diferentes, com graus de compactação diferentes (com 72dpi):

Tiff - sem compactação 136 kb

Png 24 bits - 64 kb

Png 8 bits - 8kb

Gif 256 cores - 12 kb

Gif 64 cores - 8 kb

Gif 8 cores - 4 kb

jpg qualidade alta - 4 kb

jpg qualidade média 4kb

jpg qualidade baixa 4kb

Tipografia

Boa parte do trabalho do designer gráfico é compor informação escrita, às vezes relacionando-a com imagens, às vezes somente texto. Para tanto, deve saber escolher as famílias de fonte mais adequadas, e dominar certos parâmetros de uso de fontes, uma vez que dentro de certos limites, uma má escolha pode tornar uma peça completamente ilegível ou facilitar sua leitura, como aponta Muller Brockmann:

O conhecimento da qualidade de um tipo de letra é da maior importância para os efeitos funcionais, estéticos e psicológicos do material impresso. Novamente, o design tipográfico, i. e corrigir os espaços entre as letras e palavras, a duração eo espaçamento das linhas condutoras para uma leitura fácil, fazem muito para reforçar a impressão criada.²⁹

29 Knowledge of the quality of a typeface is of the greatest importance of the functional, aesthetic and psychological effect of printed matter. Again, the typographic design, i. e the correct spaces between letters and word and the length and spacing of lines conducive to easy reading, does much to enhance the impression created. Brockman, grid systems in graphic Design, 2001. pg 19

Ao longo da história, vários pesquisadores se dedicaram a investigar como a variação dos elementos tipográficos pode influenciar na legibilidade dos textos. Miles Tinker, por exemplo, conduziu uma série de testes para chegar em zonas ótimas de formatação baseadas na velocidade de leitura.

Mas velocidade de leitura, por si mesma não garante uma comunicação efetiva. Um pesquisador que conseguiu chegar em uma metodologia científica que fosse capaz de medir a capacidade de comunicação de uma peça gráfica foi Colin Wheildon³⁰. Partindo de acadêmicos ingleses como Herbert Spencer, que disse que “a verdadeira economia da impressão deve ser medida pelo quanto é lido e compreendido, e não pelo quanto é produzido”³¹, ele desenvolveu uma metodologia que consistia em dar a dois grupos de 112 pessoas vários artigos para ler, mas para cada grupo os artigos eram apresentados em formas diferentes, com uma variação tipográfica. Depois de lerem os artigos, de interesse geral ou específico, como informativos, eram feitas perguntas para se determinar a compreensão do texto, separando em nível bom, razoável e pobre. Com essa metodologia ele descobriu que variações de tipográficas conseguiam ter grandes influências na compreensão do texto.

Famílias Tipográficas

Escolher uma família tipográfica adequada é um dos primeiros passos que o designer deve tomar na composição de um texto. Existem diversos tipos de classificação das famílias tipográficas³²,

30 WHEILDON, Colin; WARWICK, Mal, ED.— Type and layout: How typography and design can get in your message across – or get in the way; Berkeley: Strathmoor, 1996

31 SPENCER, Herbert—The Visible Word. London, Lund Huphries in association with the Royal College of art, 1969. pg 9

32 Cláudio Rocha, no Livro Tipografia Comparada” adota um sistema completo de classificação, para quem tiver interesse.

que podem incluir as mais várias nuances de desenho tipográfico. A linguagem CSS, no entanto, que é utilizada para diagramar páginas web, reconhece somente 4 famílias genéricas de fonte: *serif* (serifadas), *sans-serif* (sem-serifas), *monospace* (monoespaçadas) e *fantasy* (fantasia).

Existem milhares de famílias disponíveis para uso, mas nem todas são adequadas para o uso em texto. Além disso, existem ótimas famílias de fonte que funcionam bem no meio impresso, mas têm uma aparência pobre quando usadas na tela. Por isso, em se tratando de design para o meio digital, recomenda-se o uso de um pequeno número de famílias tipográficas, cujo desenho foi desenvolvido para uso em tela, entre elas: Verdana, Lucida sans, Arial, Georgia e Serifa. Mais adiante, na quarta parte, trataremos mais especificamente de fontes para a web.³³

Fontes serifadas, obviamente, são aquelas que possuem serifas, pequenas terminações prolongadas nos pés das letras. A pesquisa desenvolvida por Collin Weildon, feita com impressos, constatou que a simples variação do texto, de uma fonte serifada para uma fonte sem-serifa, por exemplo, fazia com que mais de 40% dos leitores deixassem de ter boa compreensão do texto.

Uma clássica fonte serifada: Palatino

Se para texto impresso, letras serifadas foram demonstradas como mais legíveis, o mesmo não ocorre para texto computador. Na tela do computador, as serifas costumam a criar pequenos borrões nas pontas da letra, por causa da resolução dos monitores, que é menor que a do papel. É verdade que desde o surgimento da internet a tecnologia de produção dos *displays* de computador melhoraram muito, mas não dá pra prever a qualidade do monitor que seus usuários devem ter, sendo assim, recomenda-se a utilização de fontes sem-serifa na produção de peças para web.

Uma clássica fonte sem serifa: Univers

³³ Harris, Daniel, The Best faces for the screen. Typofile Magazine. <http://www.will-harris.com/typoscrn.htm>

Fontes monoespaçadas são como as fontes das antigas máquinas de escrever; cada fonte ocupa a mesma largura da tela; em geral são utilizadas para diagramação de trechos de código, pois é interessante manter o alinhamento vertical do texto.

```
$cmd = 'index.php';
$_SERVER['HTTP_HOST']      = 'default';
$_SERVER['PHP_SELF']       = '/index.php';
$_SERVER['REMOTE_ADDR']    = '127.0.0.1';
$_SERVER['SERVER_SOFTWARE'] = 'PHP CLI';
$_SERVER['REQUEST_METHOD'] = 'GET';
$_SERVER['QUERY_STRING']   = '';
$_SERVER['PHP_SELF']       = $_SERVER['REQUEST_URI'] = '/';|
```

Trecho de código do CMS Drupal, com fonte Monaco, monoespaçada.

Fontes fantasia são fontes decorativas em geral. Existem inúmeras famílias disponíveis, com os temas mais variados possível. Fontes gratuitas são, em sua maioria, dessa família, pois o fato de serem temáticas e decorativas restringe seu potencial comercial. Muitas delas são mal feitas, ou feitas sem o cuidado que uma família de fonte voltada para utilização em texto, como *kernin* adequado e bom balaceamento entre letras.

De cima para baixo, em sentido horário: Rosewood, Cracked, Aristocrat, Marcelle Script, Blockhead e Birch, fontes fantasia

Pela baixa legibilidade, recomenda-se um uso cuidadoso e limitado desse tipo de fontes, restringindo seu uso a determinados títulos e evitando seu uso em texto corrido.

Tamanho de texto

Existe uma zona de conforto para lermos texto corrido. Se o texto é muito pequeno, não conseguimos enxergar os caracteres. Se é muito grande, dificulta a compreensão das palavras como um todo. Para texto impresso, essa zona de conforto vai de 8 a 12 pontos, em média; para texto em tela, de 10 a 14, sendo que o tamanho 12 é o mais confortável³⁴.

Designers que estejam produzindo texto para tela devem levar em conta a capacidade que o usuário tem de determinar o tamanho mais confortável para a leitura, visto que usuários podem ter diferentes níveis de acuidade visual, bem como usar diferentes resoluções ou tamanhos de tela.

Tamanho da linha

Linhas de texto muito curtas fazem com que o ritmo de leitura seja interrompido muito constantemente, linhas muito compridas, por outro lado, por não oferecer muitas pausas, trazem fadiga e fazem com que, muitas vezes, o leitor se perca no meio da linha. O tamanho ideal de linha é relacionado com o tamanho dos caracteres. Amy Arnston afirma que o tamanho ideal de linha deve conter entre 50 e 70 caracteres³⁵. Na web, existe ainda a possibilidade de deixar que as caixas de texto sejam redimensionáveis pelos usuários, que podem ajustar o tamanho da linha ao que acharem mais confortável.

Cor

Para manter uma boa legibilidade, a cor do texto e a cor do fundo devem ter um alto grau de contraste de brilho. Collin Weildon, em seu livro *Communication or Just Making Pretty Shapes* afirma que deve existir uma diferença de no mínimo

70% entre o valor de luminosidade das cores para que haja uma boa compreensão do texto. Além disso, obtém-se uma compreensão maior quando o texto é escuro e o fundo claro. Suas pesquisas, no entanto, foram feitas somente em mídia impressa; o branco, na tela, irradia luminosidade, e há quem prefira, por esse motivo, ler em branco sobre fundo preto. Outras pesquisas³⁶, no entanto, sugerem que como na mídia impressa, texto em preto sobre branco é mais legível do que ao contrário.

A imagem abaixo mostra as variações possíveis entre cor da fonte e fundo. fonte: Color Usage Research Lab
http://colorusage.arc.nasa.gov/legib_1.php#1

Peso e Variantes

Apesar de texto em negrito ser mais visível do que texto normal, é considerado menos legível. Estudos, tanto de Tinker (1969) quanto de Wheildon (1996), encontraram diferenças significativas de legibilidade no caso de texto em negrito³⁷. O peso das hastes diminui os espaços em branco no interior das letras, e torna, assim, os caracteres menos legíveis. É indicado utilizar negrito somente em títulos, subtítulos e palavras em destaque. Fonte itálica, no entanto, pelo menos para texto impresso, não torna o texto menos legível³⁸, mas na web pode gerar problemas de renderização, pelo grande número de hastes inclinadas.

36 Strand, Iennart, Legibility on a Podcast: Color and Typefaces. Advances in Computer and Information Science and engineering Springer Netherlands, 2008 e Timpany, Claire. The affect of type and colour on readability in print and screen based environments. World Conference on educational Multimedia, Hypermedia and Telecommunications. Honolulu, 2009

37 Stolfi, op. cit.

38 Wheildon, op cit.

34 Stolfi, Ariane. Legibilidade e Evolução das Mídias. CnPQ, 2001. disponível em: <http://finetanks.com/reference.html>

35 Arnston, op.cit. pg 106

Visibilidade

Existe uma relação entre o tamanho da fonte e a distância em que ela pode ser lida. A relação varia de acordo com a proporção dos caracteres. Para pôsters, que são lidos normalmente a cerca de 75 cm, o tamanho adequado de letra é 14pt. Para fontes com a espessura do traço de cerca de 1/8 da altura, foi encontrada a relação de 1/2 cm por metro de distância. Para razão entre a espessura do traço e a altura de 1/6, a proporção é de 1/3 de cm por metro de distância.³⁹

Composição

O trabalho do designer gráfico, assim como do webdesigner, é essencialmente, num primeiro momento, organizar elementos visuais em um plano. Os elementos visuais são os aqui apresentados: cor, texto, imagens e elementos gráficos.

Para organizar os elementos em uma página, o designer trabalha com unidades de conteúdo, ou seja, texto em blocos, títulos, figuras e listas. Na web dinâmica, onde os usuários adicionam conteúdo às páginas, nem sempre se pode prever o tamanho exato do conteúdo, e portanto, nem sempre se pode prever o tamanho final da página. Para organizar unidades de conteúdo, um recurso valioso pode ser o uso de grades construtivas:

Alguém disposto a encarar o trabalho necessário, e preenchê-lo, com o auxílio do sistema de rede, estará melhor equipado para encontrar uma solução para seus problemas de design que é funcional, lógico e também mais agradável esteticamente.⁴⁰

³⁹ Sanders e Mc Cormick, 1993 em Wijnholds, 1996

⁴⁰ Anyone willing to take the necessary trouble, will find that, with the aid of the grid system, he is better fitted to find a solution to his design problems which is functional, logical and also more aesthetically pleasing. Buch in Brockmann, op. cit. pg 9

Grades, se usadas com disciplina, podem transmitir uma série de valores e conceitos, além de facilitar a tarefa de lidar com tipos diferentes de conteúdo, como cita Brockmann no livro “Grid systems in graphic design”:

A utilização do sistema de grade implica a vontade de sistematizar, para clarificar a vontade de penetrar no essencial, de concentrar a vontade de cultivar a objetividade ao invés da subjetividade a vontade de racionalizar os processos de produção criativa e técnica a vontade de integrar elementos de cor, forma e material a vontade de alcançar o domínio arquitetônico sobre a superfície e o espaço a vontade de adotar uma atitude positiva, olhando pra frente (...)

Todo o trabalho criativo visual é uma manifestação do caráter de seu criador, o reflexo de seu conhecimento, sua habilidade e sua mentalidade⁴¹

A construção do grid tipográfico é a subdivisão do plano bi-dimensional em unidades menores.⁴² Tradicionalmente, o grid é construído a partir de medidas tipográficas, como o entrelinhamento, afim de gerar uma mancha gráfica homogênea, e atingir a “boa forma” pelos princípios de similaridade e continuidade.

⁴¹ The use of the grid system implies the will to systematize, to clarify the will to penetrate to the essentials, to concentrate the will to cultivate objectivity instead of subjectivity the will to rationalize the creative and technical production processes the will to integrate elements of colour, form and material the will to achieve architectural dominion over surface and space the will to adopt a positive, forward looking attitude (...)

Every Visual creative work is a manifestation of the character of the designer, it's a reflection of his knowledge, his ability and his mentality. Brockmann, 2001 - pg 10

⁴² Ibid, pg 11

A grade pode ser usada pelo profissional de comunicação visual para estabelecer um padrão de design, e facilitar a resolução de problemas, Como aponta Muller Brockmann, “reduzindo os elementos pictóricos a poucos formatos do mesmo tamanho” ⁴³(pg 12). Segundo o autor, essa padronização aumenta a credibilidade da informação e induz a um senso de confiança.

No meio impresso, as grades são construídas de forma fixa a partir da determinação do formato da página. Podem ser construídas grades simétricas ou assimétricas, de acordo com as necessidades de design e o desejo do designer.

Na web, onde o espaço de trabalho é fluido, o designer pode tanto impor um tamanho fixo máximo e construir a grade a partir dele, deixando eventuais bordas de um lado ou de outro, ou centralizando o elemento, como construir grids fluidos, a partir de porcentagens do espaço da tela.

Na mídia impressa, quando trabalhamos com textos de grandes dimensões, é natural separar o texto em colunas. Quando o texto chega no final da página, programas de *desktop publishing* automaticamente mandam o texto sobressaliente para a coluna seguinte. Na web, ainda não existe uma maneira simples de se diagramar texto multi-colunas. É mais fácil estabelecer colunas para unidades de conteúdo diferentes. Por isso, faz mais sentido, na web, definir uma grade horizontal, para posicionamento dos elementos, do que uma grade vertical, uma vez que as unidades de conteúdo podem ter tamanhos variáveis.

Pode-se também construir grids tipográficos a partir do emprego da unidade de medida *em*, que corresponde a um quadratim, unidade de medida tipográfica que equivale ao tamanho

horizontal da letra ‘M’. Um quadratim é uma unidade de medida relativa, e cada corpo de fonte, possui um tamanho diferente dela. Neste caso, o tamanho dos elementos na página muda caso o usuário decida, por sua conta, aumentar ou diminuir o tamanho base da fonte.

The Grid System

The ultimate resource in grid systems.

* The grid system is an aid, not a guarantee. It provides a number of possible uses and each designer can look for a solution appropriate to his personal style. But one must learn how to use the grid; it is an art that requires practice. * Josef Müller-Brockmann

Articles	Tools	Books	Templates	Blog	Inspiration
The Golden Proportion The golden section – a precise way of dividing a line, music or anything else is showed up early in nature. 06.Jan.2010	Fluid 960 Grid System A jQuery based fluid grid system template based on Nathan Smith's 960 Grid System. Allows multiple layouts including 12 or 16 fluid columns and 12 or 16 fixed columns. 06.Jan.2010	Universal Principles of Design Universal Principles of Design is the first comprehensive, cross-disciplinary encyclopedia of design. 04.Nov.2009	Photoshop 4 Column Grid A new 4 column Photoshop grid template for a 1024x768 screen resolution by Ray Guzik. 08.Jan.2009	Buy Slammer = Help Haiti The grid-generating app Slammer is part of the individualized initiative. Buy it and all proceeds from today (01/02) will be donated to Haiti. 20.Jan.2010	Ace Jet 170 AldoOne AldoOne BBOK Blanka Build Corporate Risk Watch Counter Print David Alary Design Assembly Dirty Moose Experimental Jetset Form Fifty Five Grain Edit Grain Edit Graphic Hug I Love Typography Landscape Mark Boulton Minimal Sites Neubau NewWork OC-INK Original Linkage Robbin Haman Sampsonlay Schneiders September Industry Souffle Subtraction Swiss Legacy The International Office Thinking for a Living This Studio Toko Typographic Posters Visualize Xavier Erichson Year of the Sheep
The Golden Ratio in Web Design You would be surprised to find out that most aesthetically pleasing designs, works of art, objects and even people have math in common. Specifically the Golden Ratio... 06.Jan.2010	Masonry Masonry is a layout plugin for jQuery that arranges elements vertically then horizontally according to a grid. 06.Jan.2010	Designing for the Web A Practical Guide to Designing for the Web has written explanations of the core principles of graphic design in relation to the web. 08.Oct.2009	InDesign 568x792 Grid System (12) By David Galvagno. Adobe InDesign file with a grid system for a 568x792 page that is divided into 12 columns and rows using the Golden Ratio. Includes a 12pt baseline grid. 18.Apr.2009	Eternal Network Lovely minimal grid based site design that's reminiscent of old computer paper. 06.Jan.2010	
Myths & Misconceptions About Grid Systems A few myths and misconceptions about grids exist in the design community that can be detrimental. 28.Nov.2009	Slammer 1.0 A new, robust version of Slammer that superimposes a customizable grid over any window. 22.Nov.2009	The Way of Typography The book also goes into depth on how to create grid systems by hand with only pencil, straightedge and compass. 18.Sep.2009	Illustrator 874px Grid System (12) By Neil Bradley. Adobe Illustrator file with a grid system for a 874px wide page that is divided into 12 columns and rows using the Rule of Thirds (Golden Ratio). Includes a 18pt baseline grid. 18.Feb.2009	Limited Edition 2010 Letterpress Calendar Grid-based letterpress calendar set in Helvetica. 21.Dec.2009	
About Made popular by the International Typographic Style movement and pioneered by legends like Josef Müller-Brockmann and Wim Crowell, the grid is the foundation of any solid design. The Grid System is an ever-growing resource where graphic designers can learn about grid systems, the golden ratio and how to use them. Created by Antonio Carosone, graphic designer and author of the design and typography blog AldoOne . Special thanks to Duane King for his help and wisdom. If you want to say hi or for general inquiries, send an email to: hello@thegridsystem.org Have an article, tool, template or news you want to submit? Send an email to: submit@thegridsystem.org	View All Articles --	View All Tools --	View All Books --	View All Templates --	View All Blog Posts --
Subscribe Want to stay updated with the most recent content? Subscribe below via email for daily updates or to the RSS feed. Subscribe by Email <input type="text"/> <input type="button" value="Subscribe"/>	Archives January 2010 December 2009 November 2009 October 2009 September 2009 August 2009 July 2009 June 2009 May 2009 April 2009 March 2009 February 2009 January 2009 December 2008 November 2008	Goodies LegiStyles AldoOne Store Wallpapers The Grid System Group Inter-Type-Style Group Wim Crowell Group	Colophon Made on a Mac Set in Helvetica Themed in Futurity Hosted by Media Temple Powered by Wordpress		
Designers Bookshop Grid Calculator by Designers Bookshop. A tool for creating grid systems. 	Thinking For A Living Thinking for a Living™ An ever-growing platform dedicated to open source design education. 	Basic Maths Basic Maths A grid-rick WordPress theme like no other. Get it here >			
The Grid System Flickr Group 	TypeNeu TypeNeu AN ODYSSEY IN TYPOGRAPHY. 	Blanka Blanka 			

O *The Grid System* é um site de inspiração claramente moderna, que se faz notar desde a citação de Josef Muller-Brockmann no topo da página, até o próprio tema do site, e o conteúdo, todo voltado para o estudo e a prática de construção de sistemas de grid. O layout do site, é baseado em uma grelha tipográfica tradicional, mas as larguras são constuídas em *pixels*; por isso, quando o leitor amplia o tamanho da fonte, os elementos saem da grade. O site é uma customização do

43 Ibid, pg 12

sistema *Wordpress*, que estudaremos mais adiante. <http://www.thegridsystem.org/>

The Grid System

The ultimate resource in grid systems.

** The grid system is an ad, not a guarantee.
It permits a number of possible uses and most designers can look for a suitable approach to the personal style. But one must know how to use the grid if it is so or the require pattern."
— Josef Müller-Brockmann*

HomeGrid

Join The Forum

Search

Templates

Photoshop 4 Column Grid

A free 4 Column Photoshop grid template for a 1004x788 screen resolution by Ray Cudico
June 8, 2009 /
Tags: columns, grid system, photoshop, template

InDesign 568x792 Grid System (12)

by Darío Galvagno. Adobe InDesign file with a grid system for a 568pts x 792pts page that is divided into 12 columns and rows using the Golden Ratio. Includes a 12pt baseline grid.
April 16, 2009 /
Tags: grid system, indesign, magazine, print, template

Illustrator 974px Grid System (12)

by Neil Greyling. Adobe Illustrator file with a grid system for a 974px wide page that is divided into 12 columns and rows using the Rule of Thirds (Golden Ratio). Includes a film timeline grid.
February 18, 2009 /
Tags: grid system, illustrator, print, template, website

InDesign A4 Complex Grid System (12)

Adobe InDesign file with a grid system based on Karl Gerstner's work for Capital Magazine. The grid features a 4, 6 or 12 columns and 4, 6 or 8 row size.
February 9, 2009 /
Tags: grid system, indesign, ISO 216, Karl Gerstner, print, template

InDesign A3 Grid System (12)

Adobe InDesign file with a grid system for an A3 (297x400) page that is divided into 12 columns and rows using the Rule of Thirds (Golden Ratio). Includes a 12pt baseline grid.
January 6, 2009 /
Tags: grid system, indesign, ISO 216, print, template

InDesign A3 Grid System (16)

Adobe InDesign file with a grid system for an A3 (297x400) page that is divided into 6 columns and rows using the Rule of Thirds (Golden Ratio). Includes a 12pt baseline grid.
January 6, 2009 /
Tags: grid system, indesign, ISO 216, print, template

InDesign A4 Grid System (12)

Adobe InDesign file with a grid system for an A4 (210x297) page that is divided into 12 columns and rows using the Rule of Thirds (Golden Ratio). Includes a 12pt baseline grid.
December 15, 2009 /
Tags: grid system, indesign, ISO 216, print, template

InDesign A4 Grid System (6)

Adobe InDesign file with a grid system for an A4 (210x297) page that is divided into 6 columns and rows using the Rule of Thirds (Golden Ratio). Includes a 12pt baseline grid.
December 15, 2009 /
Tags: grid system, indesign, ISO 216, print, template

Categories

Articles

Blog

Books

Templates

Toys

Archives

January 2010

December 2009

November 2009

October 2009

September 2009

August 2009

July 2009

June 2009

May 2009

April 2009

March 2009

February 2009

January 2009

December 2008

November 2008

Tags

align

alignment

Application

Arno Hoffman

article

balance

baseline

Blanka

Blog

blueprint

book

Books

brockmann

browser

build

business

calculator

columns

community

component

composition

conversion

construct

construction

content

cover

css

design

development

dimension

drive proportion

drive proportions

Articles	Tools	Books	Templates	Blog
The Golden Proportion The golden section—a precise way of dividing a line, music, or anything else—is showed up early in mathematics. 06.Jan.2010	Fluid 960 Grid System A jQuery based fluid grid system template based on Nathan Smith's 960 Grid System. Allows multiple columns including 12 or 16 fluid columns and 12 or 16 fixed columns. 06.Jan.2010	Universal Principles of Design Universal Principles of Design is the first comprehensive, cross-disciplinary encyclopedia of design. 04.Nov.2009	Photoshop 4 Column Grid A free 4 Column Photoshop grid template for a 1024x768 screen resolution by Ray Gullick. 08.Jun.2009	Buy Help The g Stirr indie it and today donat 20.Ja

A grid tradicional, ou seja, simétrica e bem definida, pode ser um dos princípios a se utilizar, mas não é o único possível. Nem sempre o designer quer transmitir os valores que Brockmann salienta; nem sempre a intenção é de clarificar, sistematizar ou cultivar a objetividade.

Neste caso, o designer pode abolir as linhas guia e agir de modo intuitivo. Para web, infelizmente, as possibilidades ficam restritas à ortogonalidade, a não ser que se use alguma tecnologia complementar ao HTML e o CSS.

O website do baterista Mauricio Takara é um exemplo de site que rompe totalmente com a idéia de grid, construído com recursos relativamente simples de HTML e CSS e JavaScript. As caixas de conteúdo ficam desordenadas, mas quando se passa o mouse sobre uma delas, ela sobe para cima dos demais e ganha opacidade. Além disso, as caixas podem ser arrastáveis pelos usuários. (<http://mtakara.com>)

[illegible][illegible]

3.2

A interface: O domínio da Usabilidade

Até a invenção da interface gráfica para o usuário, criada junto com os primeiros sistemas operacionais com ambientes gráficos, o design de interfaces era trabalho de designers industriais, principalmente o projeto de interfaces de eletrodomésticos, painéis de controle e equipamentos eletrônicos em geral.¹

No design gráfico, reinavam as composições estéticas. O software, e mais tarde as publicações "multimídia" e principalmente a web, foram responsáveis por inserir a interação no campo disciplinar do design gráfico. Com isso, ocorre, antes de tudo uma mudança de paradigma do público: ele deixa de ser somente leitor, e passa a ser também usuário. Se para a mídia impressa, o designer deveria ter plena capacidade de composição, para a web, é necessário mais do que isso, é necessário o domínio sobre o que se chama de "arquitetura de informação".

Arquitetura de informação

A inserção da interação como elemento de projeto, faz com que se tenha necessidade de um planejamento maior do objeto que se desenha sob seus aspectos estruturais. É preciso planejar os caminhos que os usuários percorrem para chegar a uma determinada informação, para que este não se perca ou deixe de encontrar a informação desejada.

A web é um espaço dinâmico, e o tempo de um clique é o que separa um website de outro. Se os usuários não encontram a informação desejada em um minuto, afirma Nielsen², "concluem que não vale a pena gastar seu tempo. E saem".

1 Thimbleby, Harold. User Interface design. reading: ACM Press, 1990

2 Nielsen, Jakob. Projetando Websites, 2000

O campo disciplinar conhecido como "arquitetura da informação" é, segundo Rosenfeld e Morville (2002):

1. A combinação de organização, rotulação e esquemas de navegação dentro de um sistema de informação.
2. O design estrutural de um espaço de informação para facilitar a realização de tarefas e acesso intuitivo ao conteúdo.
3. A arte e a ciência de estruturar e classificar websites e *intranets* para ajudar as pessoas a encontrar e gerenciar informações.
4. Uma disciplina emergente e prática comunitária focada em trazer princípios do design e da arquitetura à paisagem digital.³

A questão central, no entanto, é organizar a informação de modo a facilitar os fluxos de usuários pelo sistema projetado. Para tanto, é necessário organizar as informações em "átomos", que podem ser artigos, posts e comentários, por exemplo, e decidir quais são as relações que esses "átomos" tem entre si; criar categorias e organizar esses átomos dentro delas e criar meios para que estas informações sejam facilmente encontráveis pelos usuários.

Um arquiteto de informação é, ainda segundo os autores, algo diferente de um designer gráfico, de um programador ou de um engenheiro de usabilidade. Apesar disso, eles ressaltam que não existem fronteiras muito determinadas entre todos esses campos, e que no trabalho de um designer gráfico, por exemplo, boa parte é também

arquitetura de informação⁴. Nem todos websites necessitam de um profissional dedicado somente à arquitetura de informação, mas em grandes ambientes de informação, isto quase sempre é necessário, como apontam os autores:

Grandes organizações como IBM, Microsoft e Vanguard já têm equipes de arquitetos de informação dedicadas à estratégia e design de longo prazo de seus web sites e *intranets*. As organizações menores tendem a envolver arquitetos de informação a título de consultoria durante um *redesign* do site. Isso permite que o arquiteto de informação possa fazer uma contribuição maior sem quebrar a banca.⁵

O trabalho do arquiteto de informação abrange construir diagramas de relacionamento entre páginas, esquemas ou wireframes das páginas, esquemas de metadados e definição do vocabulário aceitável. Para isso, é preciso levar em conta três aspectos essenciais: usuários, conteúdo e contexto.

Um diagrama de arquitetura de informação representa a estrutura do site, e determina a relação hierárquica entre as páginas. fonte: Kemper, 2006

3 1. The combination of organization, labeling, and navigation schemes within an information system.

2. The structural design of an information space to facilitate task completion and intuitive access to content.

3. The art and science of structuring and classifying web sites and intranets to help people find and manage information.

4. An emerging discipline and community of practice focused on bringing principles of design and architecture to the digital landscape. Rosenfeld, Louis e Morville, Peter. Information Architecture for the World Wide Web. Sebastopol: O'Reilly Media, 2002

4 Ibid.

5 Large organizations like IBM, Microsoft, and Vanguard already have teams of information architects dedicated to the long-term strategy and design of their web sites and intranets. Smaller organizations tend to involve information architects in a consulting capacity during a site redesign. This allows the information architect to make a major contribution without breaking the bank. Ibid. pg 33.

12

Um wireframe é o esqueleto de um layout da página, com todos elementos essenciais, mas sem as imagens de fundo e detalhes gráficos.

Contexto é fundamental de se compreender pois é relacionado com a intenção do projeto; conteúdo diz respeito aos "documentos, aplicativos, serviços e metadados que as pessoas precisam usar ou encontrar" no website projetado. Várias questões se colocam em relação ao conteúdo, como: propriedade dos dados, formatos utilizados, estrutura, metainformação, volume e dinamismo, por exemplo. Além disso, é também necessária compreensão sobre os usuários; quem é o público-alvo, que tipo de comportamento ele tem, que equipamento possui, etc.

Como um arquiteto de informação, seu objetivo é fazer o melhor para aprender sobre as necessidades de informação e prováveis comportamentos de busca de seus usuários. Então, determinar quais componentes da arquitetura que melhor suprem essas necessidades e comportamentos, equilibrando estes critérios de seleção centrados no usuário com fatores contextuais, tais como quanto os componentes arquitetônicos ideais vão custar, e fatores de conteúdo, como se uma taxonomia nave-

gável poderá ser ampliada para cobrir sua rapidamente crescente coleção de documentos.⁶

Taxonomias são formas de categorizar o conteúdo de um determinado site como: sistemas de navegação globais, sistemas de navegação locais, mapas de site (*sitemaps*) ou tábua de conteúdos, índices, guias, *wizards*⁷ ou sistemas de links contextuais. Hoje, além dessas formas tradicionais de organização de informação, muitos websites se apóiam nas chamadas *folksonomias*, que são formas de organização de conteúdo criadas pelos próprios usuários, como as *tag clouds*.

Taxonomias organizam a informação no Google diretório. <http://directory.google.com/>

6 As an information architect, your goal is to do your best to learn about your users' major information needs and likely information seeking behaviors. Then determine which architectural components will best address these needs and behaviors, balancing these user-centric selection criteria with contextual factors, such as how much the ideal architectural components will cost, and content factors, such as whether a browsable taxonomy will scale to cover your rapidly growing collection of documents. Morville, Ibid.

7 Wizards são, segundo Rosenfeld e Morville (2002), "Supplementary navigation systems that lead users through a sequential set of steps; may also link to a related subset of the site's content." pg 62

Folksonomias, como a *tag cloud* do Flickr, acima, são formas muito utilizadas hoje em dia para organização de conteúdo na web. <http://www.flickr.com/explore/>

Além dessas formas de organização de conteúdo, muitas vezes a informação é encontrada por via de um mecanismo de busca, externo ou interno ao site. O campo de busca é um dos campos mais utilizados pelos usuários de um determinado site⁸, por isso deve ser bem posicionado e acessível. Para se implantar um sistema de busca dentro de um site pode se construir um sistema específico, ou então utilizar um serviço desenvolvido por terceiros, como o sistema de busca que a *Google* fornece para realização de buscas dentro de sites, por exemplo.

A arquitetura de informação engloba também uma série de recursos para facilitar a busca dentro de um website como a construção títulos, rótulos indicando o tema do conteúdo de determinado artigo, segmentação da informação em partes, organização de listas e avisos sobre o andamento de determinadas tarefas, tudo o que pode ajudar o usuário a encontrar a informação desejada em um website.

Pesquisa que medem a intensidade do olhar dos usuários em uma página revelam que o campo de busca é um dos mais vistos. Acima: Facebook, Twitter e Youtube. fonte: <http://blog.searchenginewatch.com/090713-134945>

A questão fundamental é que para se transformar dados em informação, é necessária uma organização, como aponta Nathan Shedroff⁹. Ele sugere maneiras de se organizar informação: alfabeticamente, por localização, cronológica, gradativamente, numericamente, por categorias ou randômicamente. De acordo com o tipo de informação que se procura organizar, cada maneira pode ser

8 Nielsen, Jakob. *Projetando Websites*. trad. Ana Gibson. Rio de Janeiro: Campus, 2000

9 Shedroff, Nathan. *Organizing Things*. Information Interaction Design: A Unified Theory of Design. 1994 disponível em <<http://www.nathan.com/thoughts/unified/6.html>>

mais ou menos eficiente, e dependendo do caso, mais de uma maneira pode ser necessária.

E assim como é importante se organizar os dados da forma que se ache mais conveniente, é necessário também organizar a relação entre eles e o fluxo entre as determinadas seções. Para isso, uma ferramenta valiosa são os diagramas de arquitetura de informação, aonde são definidas as seções e as relações hierárquicas entre elas.

Pra se construir uma arquitetura de informação adequada, é importante, segundo Rosenfeld e Morville, aquilo que se chama de design centrado no usuário, que inclui algumas metodologia bem desenvolvidas de projeto.

Usabilidade

Na medida em que a web foi se inserindo cada vez mais no setor comercial e da indústria, a necessidade de se teorizar, compreender e estudar essa relação do objeto com o usuário vai se ampliando, e com isso surgem teorias e metodologias para lidar com essa relação. Um desses conceitos é a usabilidade, definida a seguir pelo U.S. Department of Health & Human Services, no site *Usability.gov*:

Usabilidade mede a qualidade da experiência do usuário ao interagir com um produto ou sistema — seja um site, uma aplicação de software, tecnologia móvel, ou qualquer dispositivo operado pelo usuário.

Em geral, a usabilidade se refere a quão bem os usuários podem aprender e usar um produto para alcançar seus objetivos e como eles estão satisfeitos com esse processo. Usabilidade, como definido por Joseph Dumas e Janice (Ginny) Redish, significa que as pessoas que usam o produto podem usá-lo com rapidez e facilidade para realizar suas tarefas.

Usabilidade pode também considerar fatores como de eficácia de custo e utilidade.¹⁰

Um método essencial para atingir uma usabilidade satisfatória é o Design Centrado no Usuário. Autores como Jakob Nielsen, considerado o "guru da Usabilidade"¹¹ insistem que planejar, desde o início, pensando no usuário pode aumentar a chance de sucesso de um website e inclusive evitar gastos futuros potenciais de um eventual redesenho que se tenha que fazer para consertar problemas de interface. Maria Laura Martinez descreve um processo de design centrado no usuário:

No ciclo de desenvolvimento de software, o projeto centrado no usuário é diferente ao ciclo de vida comum, centrado na tecnologia. Ao centrar o processo de desenvolvimento no usuário, é necessário conhecer suas características, necessidades, interesses, habilidades, conhecimentos atitudes e adequar a interface a eles.¹²

Ainda segundo a autora, projetistas e usuários têm visões diferentes sobre as necessidades do produto, por isso é necessário uma metodologia específica de desenvolvimento de projetos. "Esse

10 Usability measures the quality of a user's experience when interacting with a product or system—whether a Web site, a software application, mobile technology, or any user-operated device.

In general, usability refers to how well users can learn and use a product to achieve their goals and how satisfied they are with that process. Usability, as defined by Joseph Dumas and Janice (Ginny) Redish, means that people who use the product can do so quickly and easily to accomplish their tasks. Usability may also consider such factors as cost-effectiveness and usefulness. U.S. Department of Health & Human Services, no site *Usability.gov*

11 Meet the guru of web page "usability," a man whom Web design is not a matter of taste or aesthetics — it's matter of science. Ritchtel, Matt. Making Web Sites More 'Usable' Is Former Sun Engineer's Goal. NyTimes.com, Julho de 1998 disponível em: <<http://www.nytimes.com/library/tech/98/07/cyber/articles/13usability.html>>

12 Martinez, Maria Laura. Um método de webdesign baseado em usabilidade. São Paulo, 2002 pg 20

é o campo da usabilidade". Os princípios fundamentais de projeto são, segundo ela, três:

- Focalizar desde o começo o usuário e as tarefas realizadas por ele, em um determinado ambiente (não adianta saber quais funções o software tem de executar sem saber quem é o usuário e porque precisaria dessas funções)
- Medidas empíricas da usabilidade do produto observando a interação do usuário com ele (testes com usuários realizando tarefas representativas)
- Utilizar um processo de design iterativo, com refinamentos sucessivos (onde o design pode ser modificado após as fases de prototipação ou testes), e análise de custo-benefício.¹³

Para tanto, ela sugere uma metodologia dividida em três etapas fundamentais. A primeira seria de planejamento, e inclui análise de requisitos; identificação do perfil do usuário, com aplicação de questionários; definição dos princípios de design, que inclui análise de sites similares; análise de necessidades; especificação do conteúdo; planejamento de tarefas; definição de padrões de layout; prototipação; avaliação iterativa, e especificação da implementação. A segunda seria de Implementação, que é o desenvolvimento e programação do website, e a terceira seria de distribuição, que envolve a publicação e a divulgação do site depois de pronto.

No método de design centrado no usuário, parte essencial do trabalho está na fase de planejamento, uma vez que concentrados esforços nesta etapa, problemas futuro de implementação podem ser evitados. Se o planejamento de tarefas for feito de maneira eficiente, pode-se identificar as atividades essenciais, e programar a inclusão futura de novas funcionalidades, que podem ocorrer ao longo do tempo.¹⁴

Seu método, no entanto, foi desenvolvido em um momento onde o CSS ainda não era plenamente suportado pelos browser mais comuns, por isso a definição do layout gráfico precisava ser feito antes da prototipação em HTML. Antes do CSS, era muito mais difícil de se fazer mudanças globais de layout em um site, uma vez que o layout da interface era parte da linguagem de marcação. Com a disseminação dessa linguagem, a prototipação em HTML pode ser feita independente da definição do layout final; e mudanças de layout podem ser conseguidas, em muitos casos, sem que haja desperdício de trabalho dos programadores HTML.

O design centrado no usuário em suas etapas, se ampara em uma série de metodologias, entre elas:

Card Sorting

Um método que utiliza grupos de usuário para ajudar a criar a arquitetura de informação do website, mediante o agrupamento de cartões que simbolizam unidades de conteúdo do site.¹⁵

Card sorting. Fonte: Ideo Method Cards

¹³ Ibid.

¹⁴ Ibid. pg 57

¹⁵ http://www.usability.gov/methods/design_site/card-sort.html

Testes de usabilidade

Inclui testes com protótipos que devem ser feitos com usuários reais, e devem medir tanto a satisfação subjetiva quanto a performance do sistema projetado. Podem ser feitos na forma de entrevistas contextuais, quando o teste é feito no local de trabalho dos usuário, ou em laboratórios próprios.¹⁶

Grupos focais

Discussões com grupos de entre 8 e 12 participantes, onde se discute o conteúdo e se ouvem as necessidades de recursos de potenciais usuários do site.¹⁷

Avaliação heurística

Feita por especialistas afim de identificar potenciais problemas na usabilidade da interface¹⁸

Design Paralelo

É um processo de design que inclui o desenvolvimento de uma série de layouts iniciais diferentes por diferentes pessoas, para que no fim se reúna e se sintetize as melhores propostas de cada design.¹⁹

Personas

Criação de personagens fictícios que representem potenciais usuários do site, afim de entender melhor as necessidades dos usuários potenciais.²⁰

Prototipagem

Criação de uma versão preliminar do site, seja em papel ou em HTML, para se conduzir testes de usabilidade antes do desenvolvimento final do site. A prototipagem pode evitar erros futuros

que impliquem em revisões mais drásticas de projeto²¹.

Prototipagem rápida. fonte: Ideo Method Cards

Pesquisas online

Entrevistas online, feitas com potenciais usuários, afim de identificar necessidades, perfis, índices de satisfação, requisitos do sistema e receber sugestões.²²

Análise de tarefas

Consiste em criar determinadas tarefas, como encontrar uma determinada informação ou enviar um arquivo, e analisar os passos necessários para que aquela tarefa se concretize, afim de compreender os requisitos da interface.²³

Casos de uso

Inclui a descrição da realização de uma determinada tarefa, por meio de um ator que representa um usuário determinado.²⁴

16 http://www.usability.gov/methods/test_refine/learnusa/index.html e http://www.usability.gov/methods/analyze_current/learn/contextual.html

17 http://www.usability.gov/methods/analyze_current/learn/focus.html

18 http://www.usability.gov/methods/test_refine/heuristic.html

19 http://www.usability.gov/methods/design_site/parallel.html

20 http://www.usability.gov/methods/analyze_current/personas.html

21 http://www.usability.gov/methods/design_site/prototyping.html

22 http://www.usability.gov/methods/analyze_current/learn/surveys.html

23 http://www.usability.gov/methods/analyze_current/analysis.html

24 http://www.usability.gov/methods/design_site/use-cases.html

Escrever para a web

Compreender que as condições de leitura na web são diferentes das de leitura impressa, e escrever de forma mais sucinta, utilizando parágrafos menores, listas tabuladas, imagens, diagramas e ilustrações, títulos e subtítulos, adicionando índices de conteúdo e espaço em branco para separar unidades de conteúdo. O conteúdo principal deve estar no começo do texto e deve ser utilizada linguagem familiar aos usuários.²⁵

Linhas Guia para Usabilidade

Além desses métodos apresentados pelo governo americano, Nielsen²⁶ enumera uma série de linhas guia para o desenho da interface, afim de atingir uma boa usabilidade. Não listaremos todas aqui, pois são mais de uma centena, mas alguns eixos centrais são importantes de mencionar:

Informando o conteúdo do site

Nielsen aponta que quando um usuário entra em uma página, deve saber o mais precisamente possível para que aquele site serve. Por isso, uma atenção especial deve ser dedicada à *home-page*, que deve ser claramente especificada, deve trazer o logotipo da empresa, um slogan e deve enfatizar as tarefas principais claramente em um menu de navegação. A página deve também oferecer sempre formas de contato.

Criação do conteúdo

É importante categorizar as seções do site de acordo com os interesses do usuário, e não da empresa, diz Nielsen. Além disso, deve-se evitar conteúdo redundante e escrever em linguagem familiar ao usuário.

Revelando o conteúdo por meio de exemplos

Inserir miniaturas e exemplos de cada categoria, ou então explicar o que acontece em cada link. Em caso de notícias, um pequeno resumo de cada.

Links

Diferenciar claramente os *links* do demais conteúdo do site, e torná-los fáceis de visualizar, evitar usar declarações genéricas como "Clique aqui!" e sempre indicar para onde o link aponta, diferenciar as cores de *links* visitados dos demais e utilizar de preferência as cores padrões de *link*, ou seja, sublinhado azul.

Navegação

Posicionar o menu de navegação em uma área destacada na página; agrupar por proximidade os itens semelhantes do navegação; evitar criar termos novos, e utilizar palavras familiares para os usuários nos itens do menu.

Home page

Disponibilizar caixa de pesquisa na home page, oferecer atalhos para as tarefas mais executadas e não incluir ferramentas desnecessárias.

Gráficos e animações

Evitar utilizar gráficos como decoração, somente quando tiver relação com conteúdo real e evitar animações que possam distrair os usuários.

Design gráfico

Evitar utilizar fontes decorativas e fontes em excesso, usar texto com contraste adequado em relação ao fundo e evitar rolagem horizontal da tela. Colocar os elementos essenciais acima da dobra, ou seja, na parte visível da janela e utilizar layout fluido que se adapte a diferentes tamanhos de tela.

Componentes da interface com o usuário

Certificar que áreas que pareçam clicáveis sejam funcionais e usar de modo cauteloso itens suspensos.

²⁵ http://www.usability.gov/methods/design_site/writing4web.html

²⁶ NIELSEN, Jakob. Homepage: Usabilidade - 50 web-site desconstruídos. Rio de Janeiro: Campus, 2002

Títulos de janelas

Criar títulos que expliquem o conteúdo da página. Incluir uma descrição resumida do site no título.

URLs

Utilizar um nome de domínio correspondente ao país em que o site esteja localizado, registrar, quando possível, nomes de domínio semelhantes e redirecionar os acessos que provenham desses domínios para a página principal.

Notícias e comunicados à imprensa

Utilizar títulos sucintos para matérias e artigos e escrever sinopses sucintas e informativas.

Janelas *pop-up* e páginas intermediárias

Evitar utilizar animações de abertura e páginas intermediárias, evitar janelas *pop-up*.

Publicidade

Manter anúncios nas bordas da página, e de maneira discreta; rotular anúncios para evitar que os usuários os confunda com o conteúdo da página.

Essas linhas guia são instruções que em muito lembram um tipo de instrução dada tradicionalmente em manuais de design gráfico, do tipo: Não deixe viúvas; ou use margens amplas! A diferença é que em boa parte dessas recomendações de usabilidade se amparam em testes e pesquisas realizadas com usuários reais que identificaram os problemas mencionados.

Usabilidade, como aponta o guia de usabilidade do governo federal americano, www.usability.gov, pode ser inclusive mensurada em termos de:

- Eficácia: capacidade de um usuário de usar com sucesso um site para encontrar informações e realizar tarefas.
- Eficiência: capacidade de um usuário de realizar rapidamente tarefas com facilidade e sem frustração.

- Satisfação: o quanto os usuários gostam de usar o site.
- frequência e gravidade de erro: quantas vezes os usuários cometem erros durante a utilização do sistema, quão sérios são esses erros, e como os usuários podem se recuperar deles?
- Facilidade de Memória: se um usuário tenha utilizado o sistema antes, ele ou ela pode se lembrar o suficiente para usá-lo efetivamente na próxima vez ou se o usuário tem que começar a aprender tudo de novo.²⁷

Um sistema bem utilizável, pode ainda segundo eles, gerar economia, aumentando a produtividade, a satisfação dos consumidores, ampliar vendas e rendimentos, diminuir o tempo de desenvolvimento do site, o custo de treinamento e capacitação dos funcionários e os custos com suporte e manutenção do site. Eles citam o exemplo da IBM, que calculou que um investimento de 60.000 dólares em usabilidade gerou uma economia de seis milhões de dólares somente no primeiro ano:

Usabilidade aumenta a satisfação do cliente, a produtividade, e gera confiança e fidelidade do cliente. Os consumidores ficaram mais exigentes quanto a usabilidade. Aplicando a usabilidade no projeto inicial pode-se reduzir bastante a possibilidade de extensa remodelação, manutenção e suporte técnico ao cliente.²⁸

27 • Effectiveness: A user's ability to successfully use a Web site to find information and accomplish tasks.

• Efficiency: A user's ability to quickly accomplish tasks with ease and without frustration.

• Satisfaction: How much a user enjoys using the Web site.

• Error frequency and severity - How often do users make errors while using the system, how serious are these errors, and how do users recover from these errors?

• Memorability - If a user has used the system before, can he or she remember enough to use it effectively the next time or does the user have to start over again learning everything (<http://www.usability.gov/basics/measured/index.html>)

28 Usability increases customer satisfaction, productivity, and leads to customer trust and loyalty. Consumers have become more demanding about usability. Applying usability in the initial design can greatly reduce extensive redesign, maintenance, and customer support. Usability.gov

Acessibilidade

Além das linhas-guia para usabilidade, uma outra questão que se aponta para o projeto de websites é a acessibilidade. Todos websites do governo americano, por exemplo, devem seguir certas linhas-guias para garantir esse acesso a portadores de necessidades especiais, como Apontam Koyanl, Balley e Nall:

Sites devem ser projetados para assegurar que toda pessoa e, inclusive os usuários que têm dificuldade em ver, ouvir e fazer movimentos precisos, pode usá-los. Geralmente, isso significa garantir que os sites facilitem a utilização de tecnologias de ampliação ao acesso.²⁹

Para garantir a acessibilidade, o designer deve garantir que exista texto equivalente a cada elemento gráfico, garantir que os scripts permitam acessibilidade, permitir que os usuários fujam de links repetitivos de navegação, prover títulos para frames e sincronizar os elementos multimídia.

Boa parte dos mecanismos de auxílio são browser baseados em texto, que transformam o texto em áudio ou que permitem que se comande a navegação pela voz. Por isso é importante prover sempre alternativas escritas ao conteúdo multimídia oferecido. Além disso, uma arquitetura de informação mais simples facilita também a acessibilidade.³⁰ O W3C lista alguns dos principais recursos de design que o site deve incluir afim de facilitar o acesso a esses mecanismos de auxílio:

1. Imagens e animações: Use o atributo *alt* para descrever a função de cada elemento visual.

²⁹ Websites should be designed to ensure that everyone, including users who have difficulty seeing, hearing, and making precise movements, can use them. Generally, this means ensuring that websites facilitate the use of common assistive technologies. Koyanl, Sanjal, Bailey, Robert, Nall, Janice. Research-based web design & Usability Guidelines. Maryland: Communication Technologies Branch, 2003

³⁰ Ibid. pg 22

2. Image maps. Use o mapa client side e texto para pontos-chave.
3. Multimedia. Disponibilize legendas, transcrições de áudio e descrições de vídeo.
4. Links de hipertexto. Utilize texto que faça sentido quando lido fora do contexto. Por exemplo: evite “clique aqui”.
5. Organização de página. Use cabeçalhos, listas e uma estrutura consistente. Use CSS para layout e estilo sempre que possível.
6. Gráficos e esquemas. Resumir ou utilizar o atributo longdesc.
7. *Scripts, applets e plug-ins*. Forneça conteúdo alternativo, se as características ativas não forem acessíveis ou não tiverem suporte.
8. *Frames*. Utilize a anotação *noframes* e empregue títulos significativos.
9. Tabelas. Faça a linha por linha sensível à leitura. Resumir.
10. Verifique o seu trabalho. Validação. Use as ferramentas, *checklist*, e linhas guia em <http://www.w3.org/TR/WCAG>³¹

A decisão de não seguir normas de acessibilidade implica simplesmente em jogar fora uma parte dos usuários de um site, e sem acessibilidade, um site não pode ser considerado usável, como aponta Steve Krug:

³¹ 1. Images & animations: Use the alt attribute to describe the function of each visual.

2. Image maps. Use the client-side map and text for hotspots.

3. Multimedia. Provide captioning and transcripts of audio, and descriptions of video.

4. Hypertext links. Use text that makes sense when read out of context. For example, avoid “click here.”

5. Page organization. Use headings, lists, and consistent structure. Use CSS for layout and style where possible.

6. Graphs & charts. Summarize or use the longdesc attribute.

7. Scripts, applets, & plug-ins. Provide alternative content in case active features are inaccessible or unsupported.

8. Frames. Use the noframes element and meaningful titles.

9. Tables. Make line-by-line reading sensible. Summarize.

10. Check your work. Validate. Use tools, checklist, and guidelines at <http://www.w3.org/TR/WCAG> (<http://www.w3.org/WAI/quicktips/>)

A menos que você estiver tomando uma decisão geral de que as pessoas com deficiência não são parte do seu público, você realmente não pode dizer que seu site é utilizável a menos que seja acessível.³²

O CSS, que estudaremos no próximo capítulo, é nesse sentido um grande avanço, pois permite eliminar qualquer elemento que não seja informativo do HTML, facilitando a permeabilidade por browsers baseados em texto. Agora que compreendemos o design gráfico como linguagem, e alguns dos seus elementos constitutivos, bem como algumas metodologias utilizadas para a construção da interface, a forma aparente, avançaremos para compreendê-la mais profundamente, estudando alguns dos processos e linguagens que a estruturam: A forma oculta.

32 Unless you're going to make a blanket decision that people with disabilities aren't part of your audience, you really can't say your site is usable unless it's accessible. Krug, Steve. Dont Make me think — A Common Sense Approach to Web Usability. Berkeley: New Riders, 2006. pg 169

4

A forma oculta

4.1

Impulsos elétricos, bits e Bytes

Já foi dito anteriormente que tudo que circula pela web é codificado. A forma aparente, que é a imagem que aparece na janela do *browser*, que estudamos em parte no capítulo anterior, é, na web, gerada sempre por textos, formados apenas por caracteres alfanuméricos. Assim, podemos dizer em essência que toda a comunicação feita pela rede é feita através de signos discretos.

Antes de entendermos como se dá a construção dessa forma aparente por meio dos códigos, é importante compreender como acontece essa distribuição da informação na internet de uma maneira geral, e como a web se insere dentro desse mecanismo maior.

O princípio das telecomunicações está embasado no fato de ser possível transmitir informação entre dois ou mais pontos através de algum meio. Por meio, entendemos qualquer meio físico por onde podem passar sinais elétricos, ondas sonoras, eletromagnéticas ou luminosas¹. Mas essa informação, enquanto está se propagando pelo meio, ou canal, não pode ser utilizada e interpretada por nós, seres humanos: ela está codificada de uma forma específica, determinada historicamente a partir do trabalho de uma série de engenheiros e pesquisadores, como pudemos ver em parte no trabalho anterior.

Uma informação pode ser digital ou analógica, dependendo de sua composição. Um texto, por exemplo, é essencialmente uma informação digital, na medida em que pode ser dividido em vários dígitos (caracteres). O discurso também é considerado digital, por poder ser dividido em várias unidades semânticas diferentes (sílabas), mas a onda de som gerada por um discurso, por sua vez, é uma informação analógica, um sinal contínuo que varia em função do tempo, assim

1 Lathi, 1983 pg 4

como a temperatura e a variação de pressão atmosférica.²

Existem vários modos possíveis de transmissão de dados, seja por ondas eletromagnéticas (rádio, TV, celular), por pulsos elétricos (cabos) ou sinais luminosos (cabos de fibra óptica). Esses meios de transporte podem transmitir tanto informação analógica quanto digital, mas a informação digital tem uma vantagem sobre a analógica quando levamos em conta o ruído que normalmente acomete o canal (Lathi, 1983 pg 5). Para compreendermos melhor, selecionamos algumas imagens que representam o comportamento de um sinal analógico e o de um sinal digital em um meio sujeito a um determinado nível de ruído.

Observe um exemplo de um sinal analógico:

E um exemplo de um sinal de ruído:

Quando o sinal se sobrepõe ao ruído, fica impossível para o decodificador separar o que é a informação original e o que é ruído. Mesmo com técnicas de filtragem de ruído, é possível que certas informações originais sejam perdidas.

Agora, observemos um sinal digital:

Um sinal digital, assim como o analógico, também sofre de interferência do ruído do meio, como podemos observar abaixo:

Mas é mais fácil para o decodificador reconhecer o que é informação e o que é ruído ou erro, principalmente quando essa informação digital é binária, ou seja, só existem duas escolhas possíveis a serem feitas pelo decodificador.

O sistema binário

O sistema numérico que utilizamos no nosso dia-a-dia é o sistema decimal, isto é, é um sistema numérico que tem uma base de 10 algarismos, a partir dos quais conseguimos obter a representação de todos os demais números que constituem o sistema.

Um sistema binário é baseado em apenas dois algarismos, no caso, o zero e o um. A partir desses dois algarismos conseguimos também obter toda a gama possível de números.

O sistema binário é um sistema aonde a quantidade de valores possíveis se determina exponencialmente a partir da quantidade de bits disponíveis, ou seja: para 4 bits é possível um valor de 2^4 de valores, ou seja 16, para 8 bits 256 valores diferentes, e assim por diante.

Mais com menos

Uma maneira bem didática de mostrar como é possível, através do sistema binário contar mais números com menos recursos pode ser alcançada com os dedos da mão.

² Ibid

Com os dedos de uma mão, podemos indicar valores até cinco (seis valores se contarmos o zero), através do sistema decimal, enquanto através do sistema binário podemos contar até 31 (32 valores contando com o zero) (figura abaixo)

Ainda existem nos dias de hoje vários sistemas de comunicação que funcionam por uma base analógica, como algumas linhas de telefone, ou a televisão analógica, por exemplo; mas ao longo das últimas década, a tendência é de digitalização dos sistemas, inclusive de captação de vídeo, foto e telecomunicações. Primeiro pela vantagem apresentada acima em relação à resistência ao ruído; segundo, pela facilidade que informações digitais têm de ser processadas e manipuladas posteriormente por *softwares*. Além disso, meios de armazenamento de dados digitais — como CD-ROMs, *pen drives*, *Hard Disks* e cartões de memória — tem se tornado cada vez mais baratos, o que permite uma redução crucial dos custos de produção e reprodução das informações digitais³.

Além do meio e do sinal, existe uma parte importante do processo de comunicação que não está representado nesse modelo, que é o endereço. Quando uma pessoa pega o telefone para ligar para outra, existe um processo anterior à comunicação entre os dois indivíduos, que é um processo de localização de um ponto específico na rede dentre todos os pontos possíveis existentes. Quando um número de telefone é discado, uma máquina seleciona um número específico entre milhões, mas ela não procura cegamente entre todos os números possíveis. Ela se concentra apenas no primeiro número, seleciona um grupo, depois no segundo número, e assim por diante.⁴ Este processo de localização é parte essencial da estrutura da rede, como veremos mais adiante.

3 Um bom exemplo da economia conseguida através da digitalização é no cinema. Um metro de película é muito caro, e a edição de um filme feito em película é muito dispendiosa e demorada. A possibilidade de se filmar digitalmente em alta resolução barateou muito a produção, a edição e a distribuição de filmes, dando abertura para a realização de filmagens com orçamentos muito menores, ampliando as possibilidades de cineastas independentes

4 Bush, Vanevar, 1945. *As we may think*

4.2 redes e protocolos de comunicacao

Na internet, que é a rede mundial de computadores, a comunicação ocorre pela transmissão de informação por uma série de meios: cabos telefônicos, cabos de televisão, cabos de fibra óptica, frequências de rádio ou via satélite. Hoje, já existe a possibilidade técnica de se transmitir informação por meio da rede elétrica também. Redes podem ser configuradas de diversas maneiras: centralizadas, descentralizadas e distribuídas, como podemos ver na figura ao lado. A internet, é em essência, uma rede distribuída, aonde a informação pode circular livremente entre computadores.

Para que essa circulação ocorra além das conexões físicas, é preciso um sistema de codificação que inclua um sistema comum de codificação e um sistema de endereçamento. A internet, como já dissemos no segundo capítulo, é uma rede baseada na troca distribuída de pacotes de informação, para isso, um sistema específico chamado *Internet Protocol* ou Ip foi desenvolvido. Esse sistema implementa um sistema de endereçamento e um sistema de fragmentação de informação em pacotes.

O sistema de endereço Ip consiste em um número de 32 bits, agrupados em 4 grupos de 8, separados por um ponto¹. Cada grupo de *bits* tem um valor que varia entre 0 e 255. Cada servidor e cada rede local recebe um endereço específico. Esse sistema é conhecido pelo nome de protocolo TCP/IP²:

¹ Este sistema pode gerar um número de 4,294,967,296 endereços, ou 2^{32} , e como uma série de valores estão reservados para situações específicas, é previsto que chegará a sua exaustão em breve. Devido à essa exaustão iminente, há atualmente esforços direcionados ao redesenho do protocolo de internet, buscando construir endereços de 128 bits

² <http://www.ciscosystems.ch/en/US/docs/internet-working/technology/handbook/Internet-Protocols.html#wp4155>

rede centralizada:
a quantidade de conexões exige uma grande capacidade do servidor central. Além disto, é bastante vulnerável a qualquer falha no servidor

rede descentralizada:
a quantidade de conexões exige uma menor capacidade dos servidores central. Apesar disto, ainda é vulnerável

rede distribuída:
a comunicação ocorre independente de servidores e a rede sobrevive em caso de qualquer nó ser perdido

O TCP/IP é um protocolo orientado a conexão que fornece transmissão de dados completamente duplex (transmissão de duas vias). TCP/IP é conhecido por sua confiabilidade e integridade dos dados. É o padrão para a Internet e para a maioria das redes locais. TCP/IP é uma tecnologia de troca de pacotes que oferece vantagens distintas em relação às tecnologias de troca de circuito anteriores. Primeiro, a troca de pacotes estabelece rotas lógicas em vez de caminhos físicos. Por consequência, aumenta a transmissão de dados bem-sucedida, por causa das múltiplas vias de disponíveis.³

A comunicação na rede ocorre entre dois tipos diferentes de computador. Um é chamado de cliente, que é o computador que requisita a informação, como o equipamento dos usuários comuns; outro é chamado de servidor, que é o equipamento onde as informações estão hospedadas. Pela arquitetura da rede, qualquer computador pode se tornar um servidor, bastando, para isso que tenha instalado um programa específico. O *software* mais utilizado com servidor no mundo hoje, é o *open source* Apache que roda em cerca de 47% dos servidores mundiais⁴.

Para receber os dados, por sua vez, o computador cliente precisa também de um software específico para isso, dependendo da aplicação, se estiver na world wide web, este software é o browser. Existem vários browsers disponíveis no mercado, cada um com sua particularidade. Hoje, os navegadores mais importantes são o Internet Explorer, o Mozilla Firefox e o Safari.

³ TCP/IP is a connection-oriented protocol that provides full duplex data transmission (two-way transmission). TCP/IP is noted for its reliability and data integrity. It is the standard for the Internet and most LANs. TCP/IP is a packet switching technology that provides distinct advantages over the previous circuit switching technology. First, packet switching establishes logical routes instead of physical routes. Consequently, successful data transmission increases, because of the of multiple pathways available (Shindler, 2002)

⁴ Fonte Netcraft: Market Share for Top Servers Across All Domains August 1995 - January 2010. disponível em http://news.netcraft.com/archives/web_server_survey.html

Além disso, o Google Chrome vem ganhando importância. Alguns browsers, como o Firefox, por exemplo, permitem a instalação de plugins que podem auxiliar no desenvolvimento para web. Um deles é o Firebug, que permite a visualização dos códigos HTML, CSS e JavaScript, e também do DOM, que explicaremos adiante.

O firebug, que mostra a estrutura do documento em uma janela dentro do navegador

Um computador se comunica com outro a partir de uma série de trocas de mensagens, como indica a figura abaixo.

Estabelece a conexão antes de mandar ou receber dados:

Envia dados durante a conexão:

Packet #1, Direction: Out, Time:22:41:39.470

Ethernet II

Destination MAC: 00:00:E8:7B:B5:49
Source MAC: 00:02:DD:7C:7E:7F
EtherType: 0x0800 (2048) - IP

IP

IP version: 0x04 (4)
Header length: 0x05 (5) - 20 bytes
Type of service: 0x00 (0)
Precedence: 000 - Routine
Delay: 0 - Normal delay
Throughput: 0 - Normal throughput
Reliability: 0 - Normal reliability
Total length: 0x0030 (48)
ID: 0x7076 (28790)
Flags

Don't fragment bit: 1 - Don't fragment
More fragments bit: 0 - Last fragment

Fragment offset: 0x0000 (0)
Time to live: 0x80 (128)
Protocol: 0x06 (6) - TCP
Checksum: 0xDEFB (57083) - correct
Source IP: 192.168.21.2
Destination IP: 192.168.21.3
IP Options: None

TCP

Source port: 1041
Destination port: 4000
Sequence: 0x003C2879 (3942521)
Acknowledgement: 0x00000000 (0)
Header length: 0x07 (7) - 28 bytes
Flags: SYN
URG: 0
ACK: 0
PSH: 0
RST: 0
SYN: 1
FIN: 0
Window: 0x2000 (8192)
Checksum: 0x7B63 (31587) - correct
Urgent Pointer: 0x0000 (0)
TCP Options
Maximum Segment Size: 0x05B4 (1460)
Sack-Permitted
Data length: 0x0 (0)

Raw Data:

```
0x0000 00 00 E8 7B B5 49 00 02 DD 7C 7E 7F 08 00 45 00 ..èµl..Yl~ ..E.
0x0010 00 30 70 76 40 00 80 06 DE FB C0 A8 15 02 C0 A8 .0pv@.ë.pûÀ..À"
0x0020 15 03 04 11 0F A0 00 3C 28 79 00 00 00 00 70 02 .....<(y....p.
0x0030 20 00 7B 63 00 00 02 04 05 B4 01 01 04 02 .....{c.....
```

Packet #2, Direction: In, Time:22:41:39.470

Ethernet II

Destination MAC: 00:02:DD:7C:7E:7F
Source MAC: 00:00:E8:7B:B5:49
EtherType: 0x0800 (2048) - IP

IP

IP version: 0x04 (4)
Header length: 0x05 (5) - 20 bytes
Type of service: 0x00 (0)
Precedence: 000 - Routine
Delay: 0 - Normal delay
Throughput: 0 - Normal throughput
Reliability: 0 - Normal reliability
Total length: 0x0030 (48)
ID: 0xB35F (45919)
Flags

Don't fragment bit: 1 - Don't fragment
More fragments bit: 0 - Last fragment

Fragment offset: 0x0000 (0)
Time to live: 0x80 (128)
Protocol: 0x06 (6) - TCP
Checksum: 0x9C12 (39954) - correct
Source IP: 192.168.21.3
Destination IP: 192.168.21.2
IP Options: None

TCP

Source port: 4000
Destination port: 1041
Sequence: 0x0041F062 (4321378)
Acknowledgement: 0x003C287A (3942522)
Header length: 0x07 (7) - 28 bytes
Flags: SYN ACK
URG: 0
ACK: 1
PSH: 0
RST: 0
SYN: 1
FIN: 0
Window: 0x2238 (8760)
Checksum: 0x8876 (34934) - correct
Urgent Pointer: 0x0000 (0)
TCP Options
Maximum Segment Size: 0x05B4 (1460)
Sack-Permitted
Data length: 0x0 (0)

Raw Data:

```
0x0000 00 02 DD 7C 7E 7F 00 00 E8 7B B5 49 08 00 45 00 ..Yl~ ..èµl..E.
0x0010 00 30 B3 5F 40 00 80 06 9C 12 C0 A8 15 03 C0 A8 .03@.ë.æ.À..À"
0x0020 15 02 0F A0 04 11 00 41 F0 62 00 3C 28 7A 70 12 ....Aðb.<(zp.
0x0030 22 38 88 76 00 00 02 04 05 B4 01 01 04 02 .....8v.....
```


O sistema utiliza um conjunto de domínios de “alto nível” para categorizar esses nomes. Um conjunto de domínios é baseada na natureza dos sites envolvidos, incluindo: .com (comercial, empresarial), .edu (instituições de ensino), .gov (governamental), .mil (militar), .Org (organizações sem fins lucrativos), .int (organismos internacionais), .net (prestadores de serviços de rede, e assim por diante).⁶

Além disso, pode haver um prefixo local, em caso de domínios registrados fora dos Estados Unidos. Cada país tem uma política específica para registro de domínios, e pode cobrar taxas diferenciadas para a manutenção de um determinado nome, mas a autoridade reguladora internacional é a ICANN (*Internet Corporation for Assigned Names and Numbers*). O sistema de DNS é na verdade o único sistema centralizado da rede.

Toda a atividade na Internet começa no lado do cliente, quando um usuário inicia o seu navegador. O navegador começa carregar um documento inicial, quer de armazenamento local ou a partir de um servidor por meio de alguma rede, tais como a Internet, uma intranet corporativa, ou uma extranet municipal. Ao iniciar-se na rede, o navegador do cliente consulta previamente um sistema de nomes de domínio (domain name system DNS) para traduzir o nome do servidor da página, como www.oreilly.com, em um endereço IP, antes de enviar um pedido para o servidor através da Internet. Este pedido (e da resposta do servidor) é formatado de acordo com as regras do padrão Hypertext Transfer Protocol (HTTP).⁷

6 The system uses a set of “top-level” domains to categorize these names. One set of domains is based on the nature of the sites involved, including: .com (commercial, corporate), .edu (educational institutions), .gov (government), .mil (military), .org (nonprofit organizations), .int (international organizations), .net (network service providers, and so on). (Handerson, 2009 p. 158)

7 All web activity begins on the client side, when a user starts his browser. The browser begins by loading a home page document, either from local storage or from a server over some network, such as the Internet, a corporate intranet, or a town extranet. When starting

É sabido que o nome de um website é parte importante de seu sucesso comercial. Nomes muito longos, ou de difícil pronúncia, são difíceis de se lembrar; assim, desde o começo do crescimento da web, houve uma corrida pelo registro de certos nomes que possivelmente teriam grande potencial comercial, e uma vez que se tenha garantido o direito sobre um determinado domínio, é possível negociá-lo por preços muito maiores aos praticados pelas autoridades reguladoras. Hoje, nomes de domínios muito curtos, com duas ou três letras, são praticamente indisponíveis.

URL

URL, ou *Universal Resource Identifier*, é o que chamamos de endereço web. Cada página tem um endereço diferente e único, que é composto pelo nome do domínio, seguido da localização do arquivo no servidor.

O endereço abaixo, por exemplo:

<http://pravida.org/capital.html>

Indica que a página *capital.html* está localizada no diretório de raiz do domínio *pravida.org*

Cada “/” indica um diretório de arquivos dentro do domínio. Por exemplo, o arquivo de imagem:

<http://pravida.org/mundo/pignatari3.gif>

Está localizado dentro de um diretório chamado *mundo*, dentro de *pravida.org*.

up on the network, the client browser first consults a domain name system (DNS) server to translate the home page server’s name, such as www.oreilly.com, into an IP address, before sending a request to that server over the Internet. This request (and the server’s reply) is formatted according to the dictates of the Hypertext Transfer Protocol (HTTP) standard. (Kennedy e Musciano 2006)

Em websites dinâmicos, a URL pode ser contruída a partir de variáveis, como no exemplo abaixo:

http://livi.finetanks.com/fichas_preenchidas.php?decada=1970&ficha=12

fichas de análise de impressos		Livi, 2008
Lista de Fichas	CeSAD - centro de Colet, Sistematização e Fornecimento de Dados	
1950		
1970	CeSAD - centro de Colet, Sistematização e Fornecimento de Dados	
Systematic Method For Designers	estudo da publicação:	
O Edifício	fonte:	lpg
Textos Informes	localização:	537
A Moral dos Objetos	data:	1980
A Herança Mourisca da Arquitetura no Brasil	editor responsável:	Jodete Sócrates, Maria Barreto e Maria Belda
Bibliografia Preliminar - Consórcio de Desenvolvimento Integrado do Vale do Paraíba	assunto/tópicos:	informativo do setor
Notas de um estudo sobre Objetivos do Ensino da Arquitetura e Meio para Alargá-los em Trabalho de Projeto	departamento:	FAU
Um Guia para a História da Têcnica no Brasil Colônia	projeto gráfico:	Lúcio Gomes Machado
Permanência do Fazer Artístico no Japão - Notas sobre o Japão	diagramação:	Vicente Lemos Cardoso
Metodologia de Metodologia de Planejamento Macroeconômico no Setor Habitacional Volume I	autor da capa:	lucio gomes machado
A Problematiza da Localização Industrial na Grande São Paulo	frequência:	única
A Indústria da Construção e o plano Habitacional do ESP	no. de páginas:	24
Calendário Escolar da USP/78	formato:	20,5 x 20,4 x --
Estudos de geometria descritiva	acabamento:	grampo à cavalo
Geometria da Insolação	miolo:	papel: offset 80g cores: 1 (preto)
Boletim 2	capa:	papel: offset flexa (com verniz brilhante) cores: 2 (vermelho e pink)
A invenção da Máquina à vapor	tipografia miolo:	univers
Glossário Crítico	itulos:	univers bold
O Estudo sobre Reprodução Humana no Distrito de São Paulo	comprimeto da linha:	8,4 (duas)
A Óvela Comenda	entrelinha:	4
Ou... no. 2 - O autor como produtor	altura de x:	1
Ventilação Natural dos Edifícios	margens:	1,9 - 1,0 - 2 - 1,8
Sistemas Coordenados para Projeto e Construção Universitárias	alinhameto:	à esquerda
Topografia - Estudo de Planta	intensização:	sim
Topografia	escaneado:	sim
Regimento Interno - FUPAM	observações:	Projeto gráfico simples e bem composto, com tamanho de fonte adequado. A capa apresenta bem os recursos da gráfica e as imperfeições impostas pela máquina.
Cândido Malta Campos Filho - Grande São Paulo: trabalhos e entrevistas de 1965 a 1973		
Paisagem		
Boletim FAU/SP no 1		
Faculdade De Arquitetura e Urbanismo - Relatório da Diretoria		
Arquitetura e expressionismo		
CeSAD - centro de Colet, Sistematização e Fornecimento de Dados		
Manual para Proposta, Apresentação e Execução de Projetos de Pesquisa		
Racionalização Da Construção		
Evolução e Industrialização da construção no Brasil		
Bibliografia Mínima para Escolas de Arquitetura		
FAU - informações básicas		
1980		
1990		
2000		
2010

Index of /img

Name	Last modified	Size	Description
Parent Directory		-	
200px/	29-Oct-2008 11:50	-	
FAU001_lpg_093_capa.jpg	22-Oct-2008 09:24	5.8M	
FAU001_lpg_093_ccapa.>	22-Oct-2008 09:26	4.7M	
FAU001_lpg_093_indic.>	22-Oct-2008 09:27	2.5M	
FAU001_lpg_093_rosto.>	22-Oct-2008 09:29	4.0M	
FAU001_lpg_1005_capa.>	22-Oct-2008 09:30	1.4M	
FAU001_lpg_1005_ccap.>	22-Oct-2008 09:32	6.7M	
FAU001_lpg_1005_indi.>	22-Oct-2008 09:34	2.5M	
FAU001_lpg_1005_rost.>	22-Oct-2008 09:36	5.7M	
FAU001_lpg_1011_capa.>	29-Oct-2008 11:22	2.9M	
FAU001_lpg_1011_capi.>	29-Oct-2008 11:22	461K	
FAU001_lpg_1011_imag.>	29-Oct-2008 11:23	1.0M	
FAU001_lpg_1011_indi.>	29-Oct-2008 11:23	677K	
FAU001_lpg_1011_mioi.>	29-Oct-2008 11:23	1.2M	
FAU001_lpg_1011_rost.>	29-Oct-2008 11:24	617K	
FAU001_lpg_1012_capa.>	29-Oct-2008 11:25	2.4M	
FAU001_lpg_1012_capi.>	29-Oct-2008 11:25	298K	
FAU001_lpg_1012_imag.>	29-Oct-2008 11:25	773K	
FAU001_lpg_1012_indi.>	29-Oct-2008 11:26	483K	
FAU001_lpg_1012_mioi.>	29-Oct-2008 11:26	1.2M	
FAU001_lpg_1012_rost.>	29-Oct-2008 11:27	425K	
FAU001_lpg_1014_capa.>	22-Oct-2008 09:37	2.8M	
FAU001_lpg_1014_capi.>	22-Oct-2008 09:38	1.9M	
FAU001_lpg_1014_ccap.>	22-Oct-2008 09:42	8.5M	
FAU001_lpg_1014_indi.>	29-Oct-2008 09:42	1.3M	
FAU001_lpg_1014_not.>	22-Oct-2008 09:43	2.4M	
FAU001_lpg_1014_rost.>	22-Oct-2008 09:44	577K	
FAU001_lpg_1015_capa.>	29-Oct-2008 11:27	1.8M	
FAU001_lpg_1015_ccap.>	29-Oct-2008 11:28	1.3M	
FAU001_lpg_1015_imag.>	29-Oct-2008 11:28	757K	
FAU001_lpg_1015_mioi.>	29-Oct-2008 11:29	490K	
FAU001_lpg_1015_rost.>	29-Oct-2008 11:29	940K	
FAU001_lpg_1017_capa.>	29-Oct-2008 11:29	456K	
FAU001_lpg_1017_ccap.>	29-Oct-2008 11:30	432K	
FAU001_lpg_1017_imag.>	29-Oct-2008 11:30	906K	
FAU001_lpg_1017_mioi.>	29-Oct-2008 11:31	1.3M	
FAU001_lpg_1017_suma.>	29-Oct-2008 11:31	437K	
FAU001_lpg_1020_capa.>	22-Oct-2008 09:46	6.5M	
FAU001_lpg_1020_ccap.>	22-Oct-2008 09:51	1.0M	
FAU001_lpg_1021_capa.>	22-Oct-2008 09:53	5.3M	
FAU001_lpg_1021_capi.>	22-Oct-2008 09:54	749K	
FAU001_lpg_1021_ccap.>	22-Oct-2008 09:54	1.9M	
FAU001_lpg_1021_fich.>	22-Oct-2008 12:39	1.4M	
FAU001_lpg_1021_imag.>	22-Oct-2008 12:41	3.8M	
FAU001_lpg_1021_indi.>	22-Oct-2008 12:41	1.9M	
FAU001_lpg_127_capa.jpg	22-Oct-2008 12:43	3.9M	
FAU001_lpg_268_capa.jpg	22-Oct-2008 12:45	4.0M	
FAU001_lpg_268_capi.>	22-Oct-2008 12:46	2.0M	
FAU001_lpg_268_ccapa.>	22-Oct-2008 12:48	4.3M	
FAU001_lpg_268_rosto.>	22-Oct-2008 12:49	3.7M	
FAU001_lpg_284_capa.jpg	22-Oct-2008 12:50	2.0M	
FAU001_lpg_293_capa.jpg	22-Oct-2008 12:52	5.6M	
FAU001_lpg_293_ccapa.>	22-Oct-2008 12:55	5.3M	
FAU001_lpg_297_capa.jpg	22-Oct-2008 12:57	5.6M	
FAU001_lpg_297_ccapa.>	22-Oct-2008 12:59	6.4M	
FAU001_lpg_297_image.>	22-Oct-2008 13:01	1.9M	

Já se tentarmos acessar o diretório:

<http://livi.finetanks.com>

A primeira parte “livi.” indica que este domínio é um subdomínio do domínio “finetanks.com” que o segue. Em seguida *fichas_preenchidas.php* indica o nome do arquivo PHP que está sendo executado no servidor. A seguir, aparecem as variáveis que foram selecionadas pelo usuário para a exibição da página: “decada=1970” e “ficha=12”

Diretórios

Através de FTP, podemos transferir arquivos para um computador servidor, e esses arquivos podem ser organizados dentro de subpastas, como no computador do usuário. Se soubermos o endereço desse diretório, podemos acessá-lo pelo *browser*. Por exemplo, o diretório: <http://livi.finetanks.com/img/>

É um diretório acessível pela sua URL, conseguimos acesso a uma lista de todos arquivos localizados dentro dele.

Encontramos uma página HTML. Isto acontece porque neste diretório existe um arquivo chamado *index.html*, que é um arquivo que vemos quando entramos no endereço. O *browser* sempre quando procura um determinado endereço, procura um arquivo chamado *index* localizado na pasta desejada; se não encontrá-lo, exibe então a lista de arquivos ou então um aviso de que o acesso ao diretório é negado.

4.3

A estrutura-base: O Html

Uma vez que apresentamos algumas ferramentas indispensáveis para a circulação da informação na rede, como os protocolos de transmissão e o sistema de endereçamento, iremos nos direcionar mais atentamente para a compreensão da forma da web propriamente dita. E na estruturação da rede, a linguagem mais importante e essencial é o HTML, ou *Hypertext Markup Language*.

Apesar da rede comportar uma gama de tipos de mídia, como áudio vídeo e imagens, todo tipo de arquivo digital pode ser transmitido pela internet, mas quando organizados em um sistema de página na web, são inseridos dentro de um sistema de hipertexto, cuja estrutura base é essa linguagem marcação, o HTML — ou uma variação dela o XHTML. Quando estivermos tratando de ambas linguagens, usaremos a sigla (X)HTML. Mesmo sites desenvolvidos em *Flash*, ou em qualquer outro tipo de tecnologia, são convertidos ou inseridos em um arquivo de hipertexto para circular na web.

Como já explicamos no segundo capítulo, (X)HTML é apenas uma linguagem de marcação, o que significa que não realiza qualquer tipo de processamento ou operação lógica. O princípio dessa linguagem é simples, e de fácil compreensão. A base dessas linguagens é de inserir instruções para o *browser* de como apresentar cada elemento do texto. O princípio é o de organizar os elementos em relação à sua função na estrutura do texto, identificando, por exemplo, parágrafos, imagens, *hyperlinks* e títulos.

Apesar de todos os as novas tecnologias multimídia de construção de layout de página, e as tecnologias novas que podem dar vida a documentos HTML/XHTML pela Internet, também é importante para entender as “limitações” das linguagens. Elas não são ferramentas de processamento de texto, aplicativos de editoração eletrônica, ou mesmo linguagens de programação. Seu propósito fundamental

é definir a estrutura dos documentos e das famílias de documentos para que eles possam ser transportadas de forma rápida e fácil a um usuário através de uma rede para a representação em uma variedade de dispositivos de visualização; paus pra toda obras, mas mestres de ninguém, por assim dizer.¹

As páginas da web, como já dissemos então, são arquivos de texto, recebidos e interpretados pelo *browser*, no computador do usuário. Assim, todo código (X)HTML é acessível ao usuário, que pode investigá-lo normalmente através de ferramentas fornecidas pelo *browser* de inspeção do código-fonte. Um arquivo (X)HTML não necessita de um servidor para ser visualizado. Uma vez salvo no computador, pode ser aberto pelo *browser* mesmo sem acesso à internet.

Para escrever um arquivo de hipertexto, qualquer editor simples de texto, como *Notepad* é suficiente, no entanto, existem alguns programas específicos para programação, que podem oferecer recursos extra, como colorização automática para identificar quais partes do arquivo são códigos e para identificação de eventuais erros. É possível se encontrar bons editores de (X)HTML gratuitos na internet.

Os elementos da linguagem são determinados por *tags*, certas palavras reservadas circundadas por sinais de *menor que* (<) e *maior que* (>), <p>, por exemplo, é uma *tag* que indica o começo de um parágrafo; </p>, indica o final do mesmo.

1 Despite all their new, multimedia-enabling page-layout features, and the hot technologies that give life to HTML/XHTML documents over the Internet, it is also important to understand the languages' limitations. They are not word processing tools, desktop publishing solutions, or even programming languages. Their fundamental purpose is to define the structure of documents and document families so that they may be delivered quickly and easily to a user over a network for rendering on a variety of display devices; jacks-of-all-trades but masters of none, so to speak. (Kennedy e Mucsiano, 2006)

Assim, tudo que estiver entre essas duas marcações, estará contido dentro de um parágrafo, como no exemplo abaixo:

```
<p>Este é um parágrafo, todo texto entre essas
marcas são partes de um mesmo parágrafo</p>
```

Alguns elementos especiais no HTML não necessitam dessa segunda marca que indica o fechamento, que são os elementos auto constituídos, ou vazios, por exemplo, a indicação de uma imagem , ou a indicação de uma quebra de linha
. No XHTML, que é uma versão do HTML com regras mais rígidas, esses elementos já contém em si o fechamento, por exemplo ou
.²

Os elementos podem também conter atributos específicos. Um atributo de um *link*, *href*, por exemplo, é o endereço para onde o *link* aponta:

```
<a href="http://pravida.org/palavra">palavra</a>
```

Um atributo de uma imagem, por exemplo, *src*, é o local aonde está hospedado o arquivo desta mesma imagem:

```

```

A fonte "*src*" pode ser um arquivo dentro do diretório aonde o website está hospedado, e para tanto se escreve o nome do diretório, seguindo a notação de "/" para separar cada diretório. Pode ser também um diretório que esteja acima na hierarquia do diretório onde o arquivo (X)HTML esteja hospedado, e para isso usa-se a notação ../; pode ainda, ser uma arquivo externo ao site, e para isso usa-se a URL completa do arquivo:

```

```

Alguns atributos são específicos para uma determinada *tag*, enquanto outros podem ser aplicados à maioria delas, um exemplo disso são atributos como *classe*, *id* e *style*, que explicaremos melhor no próximo subcapítulo, e são utilizados para dar forma à estrutura marcada pelas *tags*.

A linguagem não é estanque, e como quase todas as linguagens funcionais, foi se desenvolvendo com o tempo. Além disso, como é só uma linguagem de marcação, o suporte às *tags* é definido pelos navegadores. Da primeira versão até a versão corrente, muitas novas *tags* surgiram e muitas também foram suprimidas. Os *browsers* costumam a manter suporte às *tags* antigas, para que as pessoas continuem tendo acesso às páginas feitas segundo os preceitos anteriores. A versão mais nova do HTML, a 5.0, inclui uma série de novas *tags* voltadas ao suporte de conteúdo multimídia, como `<audio>` e `<video>`, por exemplo. Para distinguir a versão utilizada, para que o *browser* saiba como lidar com as *tags*, existe uma declaração que pode ser feita no começo de cada documento indicando a versão utilizada, e inclusive para indicar se está sendo utilizado HTML ou XHTML. Essa declaração é chamada de DTD, ou *Document Type definition*.

Cada versão possui uma declaração diferente, que pode ser por exemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.
w3.org/TR/html4/loose.dtd">
```

para HTML 4.01 numa versão que suporta grames, por exemplo, ou para XHTML 1.0:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
```

Para a versão 5 do HTML, uma versão mais simplificada funciona:

```
<!DOCTYPE html>
```

A estrutura de um documento HTML

Documentos HTML, XHTML ou XML, organizam-se no que se chama de estrutura em “árvore”. Numa estrutura em árvore, o elemento que contém o próximo é chamado de pai, e o elemento contido é chamado filho. Cada elemento é pai dos que estão dentro dele e filho do que está contido. A figura abaixo ajuda a entender essa estrutura:

Apesar de não ser obrigatória, é comum que documentos HTML obedeçam a uma certa estrutura padrão³, até mesmo para facilitar a organização das partes. A estrutura básica é a seguinte:

```
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

Isto quer dizer que a primeira divisão de um arquivo HTML é entre o que se chama de *head* e

³ No HTML essas *tags* não são necessárias, e o *browser* prevê a posição do corpo do documento. Contudo, elementos que são inseridos no `<head>` devem vir antes dos elementos inseridos no `<body>`. No XHTML, todos elementos são necessários, incluindo a declaração `!DOCTYPE`

Depois, o código chama um arquivo externo do tipo CSS, externos, através da *tag* `<link>` utilizado para definir a formatação do site:

```
<link href="pravida.css" type="text/css"
rel="StyleSheet">
```

Em seguida aparece o título do site, que é o que aparece na borda da janela do navegador

```
<title>PRAViDA</title>
```

A seguir temos ainda uma definição de estilo para um determinado elemento, que estudaremos mais aprofundadamente na próxima seção:

```
<style type="text/css">
    A {
        color:#ffffff;font-weight:normal;
text-decoration:none;}
</style>
```

E ainda o endereço de um determinado *script*, que no caso é um serviço de estatística oferecido pela Google, o *Google Analytics*.

```
<script src="http://www.google-analytics.com/
urchin.js" type="text/javascript">
</script>
```

E, por fim, há a indicação de que a seção `<head>` terminou:

```
</head>
```

Todas as *tags* listadas aqui, com exceção dos comentários e da *script*, que podem aparecer em qualquer parte do arquivo, são *tags* exclusivas da seção *head* do documento, e não funcionam se forem colocadas dentro da seção *body*.

O corpo do documento

Como já dissemos, o projeto da web leva em conta uma separação entre *conteúdo* e *forma*., Uma vez que a intenção é a organização de informação, está pressuposto que a informação seria o *conteúdo*, e a *forma* seria só um modo de organizá-lo (embora muitas vezes a forma pode ser o que interessa). Assim, o ideal é que os documentos (X)HTML sejam somente uma organização

estrutural da informação que se deseja compartilhar, e que a forma seja algo flexível, que possa ser definida à parte da estrutura do documento. Para isso, a linguagem reserva uma série de *tags* específicas, que podem ser utilizadas para categorizar os elementos do texto de acordo com sua função semântica.

Exemplos de *tags* utilizadas para isso, são as *tags* de cabeçalho, ou *headings*: `<h1>`, `<h2>`,...e `<h6>`. Em ordem de importância, da `<h1>` até a `<h6>`, são utilizadas para marcação de títulos. A *tag* `<p>`, indica um parágrafo. O algoritmo do mecanismo de busca a Google, por exemplo, leva em conta se determinado trecho de texto se encontra no título ou no corpo do texto, para classificação dos resultados das buscas.

Código Exemplo:

```
<html lang="">
<head>
  <meta charset="UTF-8">
</head>
<body>
  <h1>Cabeçalho1</h1><h2>Cabeçalho2</h2>
  <h3>Cabeçalho3</h3><h4>Cabeçalho4</h4>
  <h5>Cabeçalho5</h5><h6>Cabeçalho6</h6>
  <p>As Universidades Brasileiras estão em crise, porque
se recusam a compreender que chegaram à época muda, que é
preciso trocar a epiderme e as entranhas aristocráticas pelo
pêlo democrático e de massas</p>
</body>
</html>
```

Aparência:

O html não leva em conta quebras de linha do texto, nem no código, a não ser caso o texto esteja circundado por *tags* especiais, como `<pre>` ou `<xmp>`. Para que o *browser* faça uma quebra de linha em um lugar determinado, é necessário ou separar o texto em parágrafos, ou inserir uma quebra de linha, a *tag* `<br>`. Abaixo, vemos um exemplo disso no poema “A plenos pulmões”, de Maiakovski.

Código Exemplo:

```
<html lang="">
<head><meta charset="UTF-8"></head>
<body>
Camarada vida,
    vamos,
        para diante,<br>
galopemos
    pelo quinquênio afora.<br>
Os versos
    para mim
        não deram rublos,<br>
nem mobílias
    de madeiras caras.<br>
Uma camisa
    lavada e clara,<br>
e basta, —
    para mim é tudo.<br>

<pre>
Ao
    Comitê Central
        do futuro
            ofuscante,
sobre a malta
    dos vates
        velhacos e falsários
apresento
    em lugar
        do registro partidário
todos
    os cem tomos
        dos meus livros militantes.
</pre>
```

Aparência:

Camarada vida, vamos, para diante,
galopemos pelo quinquênio afora.
Os versos para mim não deram rublos,
nem mobílias de madeiras caras.
Uma camisa lavada e clara,
e basta, — para mim é tudo.

```
Ao
    Comitê Central
        do futuro
            ofuscante,
sobre a malta
    dos vates
        velhacos e falsários
apresento
    em lugar
        do registro partidário
todos
    os cem tomos
        dos meus livros militantes.
```

Além de parágrafo, o HTML tem *tags* especiais para definir listas, formulários e tabelas, como podemos ver nos exemplos a seguir.

Hiperlink

A *tag* que define um *hiperlink*, `<a>`, é possivelmente a mais importante da linguagem HTML. A partir dela que se constrói o espaço do hipertexto. O princípio é bastante simples, basta circundar um elemento pela *tag* `<a>`, que ele se torna automaticamente um link para um endereço determinado. Para isso, é necessário que esta *tag* tenha um atributo especial chamado *href*, que determina para onde aquela *link* aponta.

Código exemplo:

```
<html lang="">
<head><meta charset="UTF-8"></head>
<body>
Leia o poema <a href="aplenos.html">A Plenos Pulmões</a>
</body>
</html>
```

Aparência:

Leia o poema [A Plenos Pulmões](#)

Imagens e objetos

Imagens e objetos externos podem ser inseridos num documento Html por meio de *tags* específicas. Em comum eles compartilham um atributo *src*, o endereço da fonte (*source*) do objeto ou imagem. Imagens podem ser .jpg, .png ou gif e são inseridas pela *tag* `<img>`.

Código exemplo:

```
<body>

</body>
```

Aparência:

Através do objeto `<map>`, é possível de se inserir *hyperlinks* dentro de imagens.

Código exemplo:

```
<html lang="">
<head><meta charset="UTF-8"></head>
<body>

<map name="asterisco">
    <area shape="circle" coords="60,50,50" href="asterisco.htm">
</map>
</body>
</html>
```

Aparência:

A tag `<area>` define uma área clicável dentro da imagem. No caso acima, com o atributo `shape="circle"`, a área é definida pelas coordenadas do centro e o raio da esfera.

Objetos podem ser inseridos pela tag `<object>` ou no HTML 5.0 pela tag `<embed>`. Os objetos mais comuns são alicativos desenvolvidos em *Flash*, em *Java* ou em *Processing*, além de arquivos de áudio. De qualquer maneira, necessitam de um plugin especial instalado no *browser* do usuário para serem interpretados.

É muito comum a utilização de mini-aplicativos em *Flash* para criar players de áudio ou vídeo, por exemplo, como os utilizados pelo Youtube ou pelo *Vimeo*. Jogos on-line também costumam ser desenvolvidos em *Flash*.

Aqui está o trecho do código onde o objeto `.swf`, que corresponde a um arquivo de *Flash* está embebido:

```
<div class="vimeo_holder">
  <div id="vimeo_player_2707010" class="player" style="width:640px;height:480px;">
    <div id="vimeo_swf821ad996fb8e67a98d3ef7d5c-0be2016" style="width: 100%; height: 100%; display: none;" class="swf_holder">
      <object class="swf_holder" type="application/x-shockwave-flash" width="640" height="480" data="/mooga-loop_local.swf?clip_id=2707010&server=vimeo.com&autoplay=0&fullscreen=1&show_portrait=0&show_title=0&show_byline=0&color=00ADEF&context=user:1105593&context_id=&hd_off=0&buildnum=33157">
```

```
        <param name="quality" value="high" />
        <param name="allowfullscreen"
value="true" />
        <param name="AllowScriptAccess"
value="always" />
        <param name="wmode"
value="transparent" />
        <param name="scale" value="showAll" />
        <param name="movie" value="/mooga-loop_local.swf?clip_id=2707010&server=vimeo.com&autoplay=0&fullscreen=1&show_portrait=0&show_title=0&show_byline=0&color=00ADEF&context=user:1105593&context_id=&hd_off=0&buildnum=33157" />
      </object>
    </div>
  </div>
</div>
```

Listas

Existem dois tipos de lista em HTML: listas ordenadas `<ol>`, que recebem uma numeração automática, e listar desordenadas `<ul>`, que não recebem numeração. Dentro de uma lista, cada item é definido pela tag `<li>`. Autores como McFarlane (2006) indicam a utilização de listas não-ordenadas para a criação de menus de navegação.

Código exemplo:

```
<html>
<head><meta charset="UTF-8"></head>
<body>
<h4>Discografia Michael Jackson</h4>
<ol>
  <li>Got To Be There (1971)</li>
  <li>Ben (1972)</li>
  <li>Music and Me (1973)</li>
  <li>Forever, Michael (1975)</li>
  <li>Off the Wall (1979)</li>
  <li>One Day in your Life (1981)</li>
  <li>Thriller (1982)</li>
  <li>Farewell my Summer Love (1984)</li>
  <li>Bad (1987)</li>
  <li>Dangerous (1991)</li>
  <li>HIStory: Past, Present and Future – Book I (1995)</li>
  <li>Invincible (2001)</li>
</ol>
```

Aparência:

Discografia Michael Jackson

1. Got To Be There (1971)
2. Ben (1972)
3. Music and Me (1973)
4. Forever, Michael (1975)
5. Off the Wall (1979)
6. One Day in your Life (1981)
7. Thriller (1982)
8. Farewell my Summer Love (1984)
9. Bad (1987)
10. Dangerous (1991)
11. HIStory: Past, Present and Future – Book I (1995)
12. Invincible (2001)

A seguir, uma tabela com os elementos (X) HTML utilizados para construir listas.

Elemento	Descrição
	Define uma lista desordenada
	Define uma lista ordenada
	Define um item da lista
<dl>	Cria uma lista de definições
<dt>	Define um termo em uma lista de definições
<dd>	Define a descrição de um termo em uma lista de definições

Tabelas

O elemento <table> cria uma tabela, que pode ser subdivida em uma série de linhas, com o elemento <tr> e cujas linhas podem ser divididas em células. Por essa razão, tabelas inclusive foram utilizadas (e ainda são até hoje) para formatar o layout das páginas. Existe uma série de elementos específicos para formatação de tabelas, como cabeçalhos e rodapés, que podem ser conferidos na lista abaixo.

Elemento	Descrição
<table>	Define uma tabela
<caption>	Define o título da tabela
<th>	Define uma célula de cabeçalho
<tr>	Define uma linha em uma tabela
<td>	Define uma célula de uma tabela
<thead>	Agrupar o conteúdo do cabeçalho
<tbody>	Agrupar o conteúdo do corpo da tabela
<tfoot>	Define o rodapé de uma tabela
<col />	Define atributos de uma coluna da tabela
<colgroup>	define grupos de colunas em uma tabela

Código exemplo:

```
<meta charset="UTF-8">
<table border="1">
  <thead>
    <tr>
      <th>Mês</th><th width="200px">Atividade</th><th>Custo</th>
    </tr>
  </thead>
  <tfoot>
    <tr><td>Total</td><td></td><td>R$180</td></tr>
  </tfoot>
  <tbody>
    <tr>
      <td>Janeiro</td><td>Projeto</td><td>$100</td>
    </tr>
    <tr>
      <td>Fevereiro</td><td>Revisão</td><td>$80</td>
    </tr>
  </tbody>
</table>
```

Aparência:

Mês	Atividade	Custo
Janeiro	Projeto	\$100
Fevereiro	Revisão	\$80
Total		R\$180

Pela possibilidade que elas dão de se dividir o campo da página em colunas e células, de largura fixa ou variável, eram o melhor recurso para criação de layout na época em que os *browsers* não ofereciam suporte adequado à CSS. Hoje em dia, utilizar tabelas para criação de *layouts* é desencorajado por vários autores⁴, mas tabelas ainda devem ser usadas para criar conteúdo tabular.

Abaixo, um exemplo de layout construído com base em tabelas. Mostramos em azul a divisão das tabelas que estrutura a página:

Formulários

Formulários são a forma mais comum de receber dados dos usuários. Existem uma série de tipos de entrada diferentes, como áreas de texto, *checkboxes*, botões e *inputs* para o envio de arquivos. A seguir uma tabela com os elementos utilizado para definir formulários.

Elemento	Descrição
<form>	Define um formulário HTML
<input />	Define um elemento de entrada (<i>input</i>)
<textarea>	Uma área de <i>input</i> para texto multi-linha
<button>	Um botão clicável
<select>	Uma lista <i>drop down</i> para seleção
<optgroup>	Um grupo de opções relacionadas em uma lista <i>drop down</i>
<option>	Uma opção em uma lista de seleção
<label>	Um rótulo para um elemento <i>input</i>
<fieldset>	Um campo de informações. Põe uma borda ao redor de alguns elementos num formulário
<legend>	Define o título de um campo de informações

O elemento *input* pode ser de vários tipos diferentes. A seguir uma tabela com os tipos de *input* possíveis:

Tipo de input	Descrição
<input type="button">	Cria um botão
<input type="checkbox">	Cria uma caixa selecionável
<input type="file">	Input para envio de arquivos
<input type="hidden">	Transmite informação sobre interação cliente/servidor
<input type="image">	Uma imagem que quando clicada envia o formulário
<input type="password">	Um campo para entrada de senha
<input type="radio">	Um botão redondo que pode ser selecionável
<input type="reset">	Um botão que reseta o conteúdo do formulário
<input type="submit">	Um botão de envio do formulário
<input type="text">	Um campo para entrada de texto

Podemos formatar um formulário em (X)HTML, mas este é o trabalho de construir somente sua interface. Para enviá-lo, bem como para processar

seus dados é preciso utilizar alguma linguagem de programação, como PHP ou ASP.

Código Exemplo:

```
<meta charset="UTF-8">
<form name="ficha" action="enviaFormulario.php" method="post">
  <fieldset>
    <legend>Dados Pessoais</legend>
    <label for="nome">Nome Completo: </label>
    <input type="text" name="nome" id="nome" size="40">
    <br>
    <label for="login">Login: </label>
    <input type="text" name="login" id="login" size="20">
    <label for="login">Senha: </label>
    <input type="password" name="senha" id="senha" size="20">
    <br><br>
    <fieldset>
      <legend>Sexo</legend>
      <label for="masculino">Masculino</label>
      <input type="radio" name="sexo" id="masculino">
      <br>
      <label for="feminino">Feminino</label>
      <input type="radio" name="sexo" id="feminino">
    </fieldset>
    <fieldset>
      <legend>Interesses:</legend>
      <label for="interesses"></label>
      <input type="checkbox" name="noticias" id="nome">
      <label for="noticias">Noticias</label><br>
      <input type="checkbox" name="musicas" id="nome">
      <label for="musicas">Músicas</label><br>
      <input type="checkbox" name="promo" id="nome">
      <label for="promo">Promoções</label><br>
    </fieldset>
    <br>
    <input type="reset">
    <input type="submit">
  </fieldset>
</form>
```

Aparência:

Elementos textuais

O (x)HTML reserva uma série de tags para definir elementos textuais de apoio, tais como citações, texto em itálico, texto enfatizado, texto removido, entre outros. Abaixo, uma lista dos elementos auxiliares que ainda são suportados no HTML 5.

Elemento	Descrição
<abbr>	define uma abreviação
<adress>	para marcar um endereço
	texto em negrito
<bdo>	efine a direção do texto
<blockquote>	um bloco de citação
<cite>	uma citação

Elemento	Descrição
<code>	para marcar código de computador
	marca texto removido (aparece riscado)
<dfn>	Um termo numa lista de definições
	texto enfatizado
<i>	texto itálico
<ins>	texto inserido (aparece sublinhado)
<kbd>	texto em fonte de máquina de escrever
<pre>	texto pré-formatado
<q>	uma citação curta
<samp>	exemplo de código
<small>	texto em fonte pequena
	texto forte (negrito)
<sub>	texto subscrito
<sup>	texto superscrito
<var>	determina uma variável
*<ruby>	anotações de pronúncia
*<rt>	esplicação da pronúncia
*<rp>	explica o que mostrar quando não hover suporte para <ruby>
*<mark>	define texto marcado
* elementos novos do HTML 5	

Código exemplo:

```
<meta charset="UTF-8">
<body>
<q>Código central ou básico</q>
<blockquote>
  A comunicação é uma função do código e do repertório,
  que , por sua vez, estão na dependência da <i>informação</i>
  <i>Informação</i> não é uma "coisa", mas uma relação
  estatística entre o que se conhece e o que não se conhece,
  entre o <strong>previsível</strong> e o <b>imprevisível</b>.
  grau de imprevisibilidade dos sinais — da informação, portan
  — vincula-se à capacidade de manipulação do código.
</blockquote>
<cite>Décio Pignatari, <ins>Contracomunicação</ins></cite>
</body>
```

Aparência:

"Código central ou básico"

A comunicação é uma função do código e do repertório, que por sua vez, estão na dependência da *informação*. *Informação* não é uma "coisa", mas uma relação estatística entre o que se conhece e o que não se conhece, entre o **previsível** e o **imprevisível**. O grau de imprevisibilidade dos sinais — da informação, portanto — vincula-se à capacidade de manipulação do código.

Décio Pignatari, Contracomunicação

Elementos estruturais

O HTML reserva alguns elementos com funções meramente estruturais. Até o HTML 4.0, eles eram : <div> e . *Div* é um elemento reservado para se demarcar uma área ou um grupo de elementos; ou seja, uma divisão. *Span* é uma

tag utilizada para demarcar uma área dentro de uma linha.

Por si só, *div*s e *span*s não fazem nenhuma diferença em um arquivo HTML, mas são recursos, que utilizados em conjunto com CSS são fundamentais na formatação da página. Na próxima seção, indicaremos como utilizar elementos estruturais para diagramar arquivos.

O HTML 5.0 traz novos elementos estruturais, como <header>, <footer>, <article>, <aside> e <section>, que podem ser utilizados no lugar de <div>. Com esses elementos, a marcação estrutural ganha um sentido semântico maior.

4.4

Forma na estrutura: CSS

Como vimos, a (X)HTML é uma linguagem de marcação utilizada para estruturar documentos de texto e transformá-los em hipertexto. Através de suas *tags*, o (X)HTML marca semanticamente elementos do texto e cria grupos de elementos, mas sozinho, tem poucos recursos para dar forma aos documentos. Como comenta David McFarlane(2006):

Antes do CSS, os webdesigners eram limitados pelo esquema e as opções de estilo do HTML. E se você surfava na web em 1995, então você entende a ênfase em limitados. O HTML (...) ainda é o alicerce de todas as páginas na World Wide Web, mas simplesmente não é uma ferramenta de design. Claro, o HTML oferece as opções básicas de formatação de texto, imagens, tabelas e outros elementos da página Web e mestres webdesigners pacientes e meticulosos podem criar páginas que apareçam muito bem usando apenas HTML. Mas o resultado são muitas vezes páginas web lentas, imprevisíveis e carregadas de código enferrujado.¹

Nas versões anteriores da linguagem, até existiam *tags* especiais para determinar a apresentação dos documentos, como a *tag* <fonte>, mas como elas não têm nenhum significado estrutural, foram suprimidas das versões mais recentes, em favor da utilização de uma linguagem específica para isso. Na web, a linguagem mais popular para definição de estilo, embora não seja a única existente é o CSS, uma sigla para Cascading Style Sheets (folhas de estilo em cascata).

¹ Before CSS, Web designers were limited to the layout and styling options of HTML. And if you surfed the Web in 1995, then you understand the emphasis on limited. HTML (...) still forms the foundation of all pages on the World Wide Web, but it's simply not a design tool. Sure, HTML provides basic formatting options for text, images, tables, and other Web page elements and patient, meticulous Web masters can make pages look pretty good using only HTML. But the result's often sluggish, unpredictable Web pages laden with clunky code. McFarlane, David Sawyer. CSS: The Missing Manual. Sebastopol: O'Reilly, 2006

Vimos no capítulo anterior que o design gráfico pode ser considerado uma linguagem. O CSS é, de uma certa maneira, uma tradução da linguagem visual para uma linguagem funcional, ou seja, através de seus elementos, é possível dar características específicas de aparência para os elementos marcados pelo (X)HTML. Além disso, o CSS permite que haja uma independência entre forma e conteúdo, e permite a formatação de maneiras distintas para tipos diferentes de displays.

Existem três maneiras de se declarar estilo para um determinado documento: através de um *link* para um arquivo .css externo ao documento; inserindo um trecho de código CSS numa seção do documento através da *tag* <style> ou declarando o estilo de um determinado elemento através do atributo "style". O *Browser* lê as declarações nesta ordem, e vale a última declaração lida, isto é, se um determinado elemento têm uma declaração feita no arquivo CSS e outra o atributo *style*, vale a do atributo *style*.

É especialmente indicado a aplicação de CSS mediante a criação de um arquivo externo, pois podemos aplicar o mesmo arquivo a um número infinito de páginas. Assim, é mais fácil manter a coerência entre as páginas, mediante a aplicação dos mesmos estilos em todas páginas e é mais fácil também a realização de alterações de layout, pois uma vez determinados no arquivo central, todas páginas recebem os novos valores. Por essas facilidades, a utilização de CSS externo também facilita a customização de websites.

Uma experiência interessante que ajuda a entender as potencialidades desta linguagem, é o CSS *Zen Garden*, um website que reúne trabalho de vários designers ao redor do mundo em torno de um objetivo comum: criar layouts alternativos para uma mesma página HTML. Os *screenshots* a seguir ajudam a entender essa potencialidade.

fonte: <http://www.csszengarden.com/?cssfile=/212/212.css&page=0>

<http://www.csszengarden.com/?cssfile=/213/213.css&page=0>

<http://www.csszengarden.com/?cssfile=/209/209.css&page=0>

<http://www.csszengarden.com/?cssfile=176/176.css>

<http://www.csszengarden.com/?cssfile=181/181.css>

Em todas as imagens, o código HTML é exatamente o mesmo, só o CSS é diferente. Para conseguir tanta variedade formal, O CSS tem uma série de recursos, que inclui:

- Controle tipográfico (família, tamanho, cor, estilo, entrelinha e entreletra)
- Controle de posicionamento e dimensionamento dos elementos HTML
- Possibilidade de substituição de elementos por imagens
- Possibilidade de adicionar imagens ao fundo dos elementos
- Possibilidade de definir estilo para cada tipo de meio (impresso, monitor, celular, projetor, etc)

Assim como no design gráfico tradicional, o webdesign também é embasado na capacidade de se compor com texto, imagens e elementos gráficos. As possibilidades oferecidas para formatação dos elementos em CSS é bastante limitada se compararmos com os recursos disponibilizados

pelos softwares de edição e produção de imagens disponíveis no mercado. Por isso, muito do web-design contemporâneo é baseado na utilização massiva de imagens, principalmente para construir fundos ou botões, uma vez que efeitos *clichés*, muito utilizados, como sombras e *degrados*, só podem ser aplicados a websites mediante o uso de imagens.

Imagens podem ser inseridas no HTML, como vimos no capítulo anterior, mas também podem ser inseridas via CSS, como fundo ou em substituição a elementos HTML.

Sintaxe da Linguagem

Um código CSS é formado por uma série de declarações de estilo, atribuídas a elementos de um determinado documento. Toda declaração de estilo obedece a uma mesma estrutura que consiste em um seletor, que listaremos a seguir, seguido de um bloco de declarações, que é formado por duas chaves, aonde se inserem as propriedades desejadas para aquele elemento selecionado, separadas por ponto e vírgula. Cada declaração é constituída de um par formado por uma propriedade e um valor, separados por dois pontos, como por exemplo:

```
p {font-family: arial;
color:#FF0000;
font-decoration:underline}
```

É muito importante manter atenção quanto à ortografia e à estrutura da linguagem quando se escreve um código CSS, assim como na maioria dos códigos de computação em geral. Qualquer erro, como uma chave sem fechamento, ou a falta de dois pontos ou de ponto e vírgula, assim como qualquer erro de ortografia em qualquer parte,

pode causar um defeito e impedir o funcionamento do código².

Essas declarações de estilo podem ser aplicadas genericamente a todos elementos de um mesmo tipo, ou para elementos específicos dentro de um código. Para isso, os elementos tem de ter alguma característica particular, que pode ser um atributo *id* ou *class*, para que possam ser reconhecidos e formatados. Utilizamos uma *classe* quando queremos aplicar um estilo a vários elementos numa mesma página e *id* quando aquele elemento é único. O atributo *id* pode ser utilizado também para manipular dinamicamente aquele elemento através de *JavaScript*, como veremos no próximo item deste capítulo. Abaixo, exemplos de maneiras de selecionar elementos de um arquivo HTML (ou XML ou XHTML) para aplicarmos declarações de estilo.

Seletores

A primeira parte de uma declaração CSS, é o seletor. Ele determina a qual elemento da página a regra se aplica. A linguagem vai priorizar declarações dadas por *id*, em seguida por classe e posteriormente por elemento. Em caso de duas declarações para um mesmo elemento, vale a última escrita. A seguir, uma lista dos seletores possíveis, seguidos de exemplos de aplicação.

Para todos elementos de uma determinada página

* {*propriedade: valor*}

ex: * {*margin:0; padding:0*}

Esta declaração é muito comumente utilizada para zerear as propriedades *margin* e *padding* de todos elementos da página. Uma vez que *browsers* diferentes têm valores padrão diferentes para essas propriedades, é interessante zerá-los

² Isto pode ser bastante desagradável, principalmente para designers, que estão acostumados a trabalhar livremente a expressão visual. A percepção humana, pode relevar perfeitamente pequenos erros (Kofka, pg 150), mas os programas de computador são minuciosos, então, é extremamente aconselhável a utilização de softwares específicos para edição de código, que têm a capacidade de reconhecer, em alguns casos má formatação e erros de ortografia.

para que os elementos tenham igual aparência independente do meio.

Por elemento

elemento {*propriedade:valor; propriedade: valor valor*}

ex: *p* {*font-family: arial, verdana, sans-serif*}

Aplica esta fonte, em ordem de preferência a todos elementos «p» da página

Por classe e id

.classe {*propriedade:valor; propriedade: valor valor*}

#id {*propriedade:valor; propriedade: valor valor*}

Código exemplo:

```
<html>
<head>
<style type="text/css">
* {font-family: arial}
.borda {
border-top: solid 2px black;
width: 200px}
#fundo {
background-color: #dddddd}
</style>
</head>

<h1 class="borda">Exemplo</h1>
<div class="borda"> uma div</div>
<div class="borda" id="fundo">outra div</div>
<div class="borda"> mais uma</div>
<body>

</body>
</html>
```

Aparência:

Exemplo

uma div
outra div
mais uma

Para grupos de elementos

elemento, elemento {*propriedade: valor*}

ex. *p, h1, h2, h3* {*font-family:arial*}

Código exemplo:

```
<html>
<head>
<style type="text/css">
h1, h2, h3, p {font-family: arial}
span { font-family: serif}
</style>
</head>
<body>
<h1>Exemplo</h1>
<h2>grupo de elementos</h2>
<p>Podemos aplicar estilos a <span>grupos</span> de elementos</p>
</body>
</html>
```


Aparência:

Exemplo

grupo de elementos

Podemos aplicar estilos a grupos de elementos

Para determinados elementos em uma classe ou id

elemento.classe {propriedade: valor}

ex: *a.menu {text-decoration:none}*

Para elementos que estão contidos em outros elementos

elemento elemento {propriedade: valor}

ex: *ul li {list-style:none}*

Código exemplo:

```
<html>
<head>
<style type="text/css">
ul {font-family:arial}
ul ol li {font-size:0.8em}
</style>
</head>

<body>
<ul>
<li>item</li>
<li>item</li>
<ol>
<li>sub item</li>
<li>sub item</li>
</ol>
<li>item</li>
</ul>
</body>
</html>
```

Aparência:

- item
- item
 1. sub item
 2. sub item
- item

Para todos elementos que contêm determinado atributo:

elemento[atributo] {propriedade: valor}

Para um elemento que tenha um valor especificado

elemento[atributo="valor"] {propriedade: valor}

Código exemplo:

```
<html>
<head>
<style type="text/css">
img[alt] {border: 4px solid black}
img[src="bola.gif"] {border:4px dotted black}
</style>
</head>

<body>




</body>
</html>
```

Aparência:

Por relação hierárquica

elemento elemento-contido {propriedade: valor}

ou então, qualquer elemento determinado que esteja dentro de outro:

elemento > elemento-filho {propriedade: valor}

Pseudo-classes

Existem também seletores especiais chamados pseudo-classes, que aplicam um determinado estilo a um elemento que esteja em algum estado pré-determinado, como por exemplo:

elemento:link {propriedade: valor}

para links não visitados

elemento:visited {propriedade: valor}

para links visitados

elemento:hover {propriedade: valor}

para quando o mouse está sobre o elemento

elemento:active {propriedade: valor}

para quando o link está selecionado

Código exemplo:

```
<style type="text/css">
a {padding:2px; line-height:1.5em}
a:link {color: #000000}
a:visited {color:#aaaaaa}
a:hover {background-color:#ddddd}
a:active{color:#ffffff}
</style>

<a href="a.html">link</a> | <a href="b.html">visited</a> |
<a href="c.html">hover</a> | <a href="d.html">active</a>
```

Aparência:

[link](#) | [visited](#) | [hover](#) | [active](#)

elemento:before {propriedade: valor}

adiciona conteúdo antes de um elemento

elemento:after {propriedade: valor}

adiciona conteúdo depois de um elemento

Código exemplo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<style type="text/css">
h1:before {content:url(abreaspas.gif)}
h1:after {content:url(fechaspas.gif)}
</style>
</head>

<body>
<h1>Entre aspas</h1>
<p>Utilizamos os pseudo-seletores :before e :after para adic
<p>Para funcionar no IE 8, precisa haver uma declaraçãoccdil;
</body>
</html>
```

Aparência:

“Entre aspas”

Utilizamos os pseudo-seletores :before e :after para adicionar as aspas.

Para funcionar no IE 8, precisa haver uma declaração !Doctype

elemento:focus {propriedade: valor}
quando o input de teclado está funcionando

Código exemplo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"
>
<html>
<head>
<style type="text/css">
input:focus {background-color:#dddddd;}
</style>
</head>

<body>
<form action="form_action.asp" method="get">
login: <input type="text" name="fname" /><br />
senha: <input type="text" name="lname" /><br />
<input type="submit" value="Submit" />
</form>
</body>
</html>
```

Aparência:

login:

senha:

elemento:first-child {propriedade: valor}
define propriedades para o primeiro de uma série de elementos

Código exemplo:

```
<style type="text/css">
p > em {text-decoration: underline}
</style>

<p>Este parágrafo pode ter <em>uma</em>
<p>Neste parágrafo pode ter <span>a declara&ced
</span> funciona</p>
```

Aparência:

Este parágrafo pode ter uma palavra sublinhada

Neste parágrafo a declaração *não* funciona

elemento:first-line {propriedade: valor}
define propriedades para a primeira linha de elementos

elemento:first-letter {propriedade: valor}
define propriedades para a primeira letra

elemento:lang {propriedade: valor}
define propriedades para elementos com um atributo específico "lang"

É especialmente interessante utilizar alguns desses pseudo-seletores, como :hover :visited, :active e :focus pois eles dão ao usuário feedback sobre suas ações, ao indicar a ação que está ocorrendo ou pode ocorrer naquele momento.

Herança

Pensando que o documento (X)HTML se estrutura em forma de árvore, com elementos pais e filhos, algumas propriedades dadas aos elementos pai, são herdadas pelos elementos filhos. Se determinarmos, por exemplo que a cor da fonte do <body> é azul, a fonte de todos elementos contidos nele terão a cor azul. Como regra geral, propriedades como margens, fundos e bordas

não são herdadas pelos elementos filhos. Além disso, as cores padrão para links também prevalecem mesmo quando definidas cores específicas para algum elemento pai onde o link esteja inserido.³

Como regra geral também, o CSS vai representar a propriedade dada ao elemento mais próximo dele na hierarquia de árvore. Por exemplo, se definirmos que a cor padrão da fonte do <body> é vermelho e criarmos dentro dele uma <div> definirmos cor azul, um parágrafo que esteja dentro desta <div> será representado como azul.⁴

Diagramação

Agora que expusemos as formas de se selecionar elementos das linguagens de marcação para declararmos propriedades de estilo, cabe salientar alguns conceitos importantes para a diagramação de páginas da web.

Unidades de medida

Para dimensionar os elementos, existem alguns tipos de medidas aceitos em CSS. O único valor que não exige unidade de medida é o zero.

Unidade	Descrição
%	porcentagem relativa ao quadro onde o objeto está inserido
in	polegadas
cm	centímetros
mm	milímetros
em	unidade de medida relacionado ao tamanho de fonte do elemento.
pt	ponto 1/72 de polegada
pc	pica 1/6 de polegada ou 12 pontos
px	um pixel corresponde a um ponto da tela

Cores

³ McFarland, 2006 cap 4.3

⁴ Op. cit

As cores representadas na tela do computador, ou em qualquer tela colorida luminosa são compostas por meio de síntese aditiva, a partir do sistema conhecido como RGB. No sistema RGB, as cores são formadas a partir da soma de valores de vermelho, verde e azul. Para definirmos uma cor via CSS, podemos utilizar alguns nomes de cor aceitos pelos *browsers*, ou definirmos a cor em termos de seu valor RGB. A tabela abaixo mostra os meios possíveis de se definir cor em CSS.

Unidade	Descrição
nome	nome da cor em inglês nem todos nomes são suportados por todos <i>browsers</i> ex: red
rgb(x,x,x)	um valor RGB “x” pode variar de 0 a 255 ex: rgb(255,0,0)
rgb(x%,x%,x%)	um valor RGB definido por porcentagens de cada cor ex: rgb(100%,0,0)
#RRGGBB	neste caso, cada dupla de números equivale a um valor hexadecimal para cada cor, variando de 00 a FF ex: #FF0000

Apesar das tecnologias de display de telas terem evoluído muito desde o começo da web, Existem 256 cores consideradas seguras. A seguir, uma tabela com essas cores e seus respectivos valores hexadecimais:

000000	003333	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF
666600	666633	666666	666699	6666CC	6666FF
669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF
993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF
999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF

fonte: <http://www.cultura.ufpa.br/dicas/htm/htm-cor3.htm>.
Autor desconhecido. As cores impressas não correspondem às cores na tela.

Propriedades

Como explicamos no começo do capítulo, as linguagens de apresentação, como o CSS, são tentativas de traduzir a linguagem do design gráfico para uma linguagem funcional, para que possa ser transmitida por meio de signos discretos. Pelas restrições dessa linguagem, nem todos recursos gráficos possíveis de serem atingidos no meio impresso podem ser atingidos via CSS.

As possibilidades são aquelas variações permitidas dentro do escopo da linguagem, que separa uma série de propriedades que podem ser exploradas pelos designers e programadores.

A seguir listaremos e explicaremos como essas propriedades podem ser utilizadas para determinar a aparência das páginas.

Propriedades de texto

Domínio sobre as técnicas de formatação de texto é sem dúvida uma das habilidades fundamentais que um designer deve ter, qualquer que seja seu meio de trabalho. Para formatarmos texto na web temos, no entanto poucas possibilidades, se compararmos com as oferecidas pelos softwares comerciais voltados para mídia impressa. Para difundirmos texto pela internet, dispomos de algumas alternativas: a primeira, é formatarmos texto marcado por (X)HTML, contando com os recursos de formatação que o CSS propicia; a segunda é transformarmos um texto pré-formatado ou desenhado em imagem e dispor-lo como uma figura e a terceira é usar alguma linguagem mais desenvolvida como SVG⁵, *Flash* ou *Processing* e inserir o texto como um objeto externo no HTML.

Se inserirmos o texto como imagem ou como objeto externo, a ganhamos em recursos gráficos, mas perdemos em permeabilidade e acessibilidade das informações. Texto na forma de imagem, quer seja vetorial ou *bitmap*, só pode ser lida por humanos, enquanto texto marcado pode ser lida também por computadores.⁶ Por o HTML ser aberto, o texto inserido no código é permeável por uma série de mecanismos, como os *search engines*, conversores de texto em áudio

para surdos, ferramentas de indexação e outros mecanismos que lidam automaticamente com informação na rede.

Lista das Propriedades CSS relacionadas à Texto

Propriedade	Descrição	Valores	
font	Atalho para definir todas as propriedades de fonte em uma só declaração	font-style font-variant font-weight font-size/line-height font-family caption icon menu message-box small-caption status-bar	
font-family	Uma lista de prioridades de famílias de fontes ou estilos genéricos	family-name generic-family	
font-size	Dá o tamanho da fonte	xx-small x-small small medium large x-large	xx-large smaller larger length %
font-size-adjust	Especifica um valor para o tamanho e preserva a altura de x para a primeira escolha de fonte. É uma forma de manter as fontes de tamanho semelhante caso falte alguma das opções	none number	
font-stretch	Condensa ou expande a família corrente de fonte	normal wider narrower ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded	
font-style	Determina o estilo da fonte.	normal italic oblique	
font-variant	Dispõe o texto normal ou em versaletes	normal small-caps inherit	
font-weight	Dá o peso da fonte	normal bold bolder lighter 100 200 300	400 500 600 700 800 900

⁵ SVG é uma linguagem baseada em XML para descrição de figuras e desenhos. Está incluída nas especificações do W3C como uma linguagem destinada a exibição de gráficos vetoriais. <http://www.w3.org/Graphics/SVG/>

⁶ Na verdade, existem alguns algoritmos complexos de reconhecimento de texto, conhecidos como OCR, ou *optical caractere recognition*, que podem transformar texto apresentado na forma de imagem em texto alfanumérico, mas em geral, são mecanismos muito complexos para serem utilizados em larga escala. Não é à toa que muitos dos mecanismos anti-spam são embasados no *Captcha*, que oferece texto na forma de imagem para serem reconhecidos pelos usuários, distinguindo-os de robôs de todo o tipo. Além do mais, texto marcado também possui, quando marcado mais corretamente uma hierarquização das informações, por exemplo, separando o que é título do que é parágrafo, o que permite uma melhor indexação por mecanismos de busca.

color	Define uma cor a ser utilizada no texto contido no elemento	nome rgb(x,x,x) rgb(x%,x%,x%) #RRGGBB #RGB
direction	Define a direção que o texto corre	ltr rtl
letter spacing	Aumenta ou diminui o espaçamento entre as letras	normal valor
text-align	Define o alinhamento do texto	left right center justify
text-decoration	Define o tipo de decoração do texto	none underline overline line-through blink
text-indent	Identifica a primeira linha do texto	tamanho %
text-transform	Controla a capitalização do texto	normal capitalize uppercase lowercase
white space	Determina como o espaço em branco dentro do elemento é disposto. Pode ser usado para substituir o elemento pre	normal pre nowrap
word-spacing	Aumenta e diminui o espaço entre as palavras	normal tamanho
vertical-align	Determina o alinhamento vertical do elemento	baseline sub super top text-top middle bottom text-bottom tamanho %

Font-family

Talvez a pior restrição que o texto marcado pode oferecer aos designers em termos é a pouca variedade de tipos que se tem para trabalhar. Embora existam recursos para permitir a utilização de qualquer família de fonte na web, o que discutiremos mais adiante, vários autores (Como Zeldman, McFarlane e Haine, por exemplo) indicam a utilização apenas de fontes que são gratuitas ou que em geral já vem instaladas junto com os sistemas operacionais mais comuns⁷.

A definição de uma família de fonte pode ser feita de maneira específica, através do nome de alguma fonte, como arial, por exemplo, ou de maneira genérica, como “*serif*” ou “*sans-serif*”. A não ser que se utilize um recurso de embutir os arquivos de fonte nas páginas web, que pode além de aumentar o tempo de carregamento do site esbarrar em questões de direitos autorais, é impossível prever a exata aparência do texto em todos computadores, já que sistemas operacionais diferentes vem com fontes padrão distintas instaladas. Por essa razão, autores como McFarlane sugerem que sempre se declare um nome e uma família genérica, afim de evitar que a aparência do site seja completamente diferente em um sistema operacional que não possua a fonte desejada. A seguir, uma lista com as famílias de fontes mais comuns; extraída do livro “CSS: *The Missing Manual*”:

Internet Explorer 6/Windows XP Pro	Firefox 1.5/Mac OS X
Arial	Arial
Arial Black	Arial Black
Arial Narrow	Arial Narrow
Comic Sans MS	Comic Sans MS
Courier New	Courier New
Century Gothic	Century Gothic
Georgia	Georgia
Helvetica	Gill Sans
Lucida Console	Helvetica
	Lucida Grande
	Marker Felt
	Monaco
Tahoma	Tahoma
Times	Times
Times New Roman	Times New Roman
Trebuchet MS	Trebuchet MS
Verdana	Verdana
ⓧ ① ㄣ ㄣ ㄣ ㄣ (Webdings)	ⓧ ① ㄣ ㄣ ㄣ ㄣ (Webdings)
sans-serif	sans-serif
serif	serif
monospace	monospace
FANTASY	fantasy

famílias tipográficas mais comuns
fonte:McFarlane, 2006

Embed Fonts

O CSS reserva um recurso especial para permitir a anexação de uma fonte específica ao site, o recurso @font-face. Com ele você pode linkar um arquivo de fonte externo ao site, da seguinte maneira:

⁷ Considerem como sistemas mais comuns as plataformas Windows e Mac OS.


```
@font-face {
  font-family: "Helvetica thin";
  src: url(helvetica_thin.otf) format("opentype");
}

p {font family: Helvetica thin}
```

Usar fontes embebidas, no entanto, tem algumas consequências. A primeira é que nem todos browsers podem oferecer suporte adequado para este tipo de fonte. O segundo, é que se você usar uma fonte de licença proprietária, utilizar fontes embebidas podem oferecer risco de pirataria por parte dos usuários.

As *type foundries* ainda estão desenvolvendo meios de se disponibilizar fontes sem correr o risco de tê-las pirateadas pelos usuários. Uma forma é um aluguel, oferecido pela Typekit, que hospeda as fontes em um local seguro, onde só o site consegue acessar. Outra forma é a *.webfont*, um formato que ainda está sendo desenvolvido, e que daria permissão de acesso somente ao site também.⁸

Font-size

Font-size é a propriedade que define o tamanho da fonte. Podemos aqui utilizar todas as unidades de medida do CSS para definir o tamanho da fonte, ou também utilizarmos valores que vão de *xx-small* a *xx-large*. Utilizar tamanho de fonte em pixel é interessante pois garante que a fonte vai ter o mesmo tamanho independente do *browser* (McFalan, 2006 cap 6.2).

Existe também o recurso chamado *font-adjust*, que permite determinar um valor de altura de x, afim de evitar grandes discrepâncias causadas pela ausência da fonte determinada, e manter a fonte com uma altura de x determinada.

Caixas de conteúdo

Para diagramar elementos em uma página, é importante entendermos como funciona a determinação das caixas de conteúdo.

Tipos de elementos

Cada elemento que aparece na página gera para si um quadro. Existem quadros que são chamados *block-level* (em nível de bloco) e quadros *inline* (internos à linha) (Haine, 2006). Quadros *block-level*, como os gerados por *divs*, parágrafos e cabeçalhos, são renderizados, isto é, aparecem, em uma nova linha do documento; quadros *inline*, como os gerados por *span*, *em*, e *input*, por sua vez, são renderizados na mesma linha em que estão contidos. Na figura abaixo, desenhamos uma borda em torno dos quadros gerados pelos elementos. Adicionamos 10px de margin para facilitar a visualização.

Código exemplo:

```
<style type="text/css">
* {border: 1px solid black;
  font-family: arial;
  margin: 10px}
</style>

<div>
<p>&Eacute; a multiplica&ccedil;&atilde;o
<em>explosiva</em> dos meios, dos &oacute;digos
e das <span>linguagens</span> que contribui,
em boa <a href="parte.html">parte</a>,
para a incomunicabilidade</p>
</div>
```

Aparência no Safari:

Aparência no Firefox:

É interessante notar que, apesar de não estarem explicitamente indicados no código, o *browser* desenhou bordas para os elementos *html* e *body*, e também que *browsers* diferentes tem

⁸ Web fonts, where are we? I love typography. 2009. Disponível em: <<http://ilovetypography.com/2009/07/20/web-fonts-%E2%80%94-where-are-we/>>

interpretações diferentes para os valores desses elementos. Enquanto o Safari considera o tamanho deles como próximo das bordas da janela, o Firefox só leva em conta o tamanho dos elementos internos.

Display e visibility

A Propriedade *display* é importante pois pode transformar a aparência dos objetos, mudando seus tipos de renderização. Através dela, podemos por exemplo transformar um objeto do tipo *inline* em *block*, ou mesmo fazer com que um determinado elemento permaneça escondido.

A propriedade *visibility* pode também tornar um elemento oculto, mas neste caso, o elemento apesar de invisível, continua ocupando o mesmo espaço que ocuparia se estivesse visível. A tabela abaixo mostra os valores possíveis de *display* e *visibility*:

Propriedade	Descrição	Valores
display	especifica o modelo de caixa que o elemento gera	none
		table
		block
		table-caption
		inline
		table-cell
		inline-block
		table-column
		inline-table
		table-column-group
visibility	Determina se o elemento aparece ou não na página	list-item
		table-footer-group
		table-header-group
		run-in
		table-row
		inherit
		table-row-group
		visible
		hidden
		collapse (só para elementos de tabela)

Código exemplo:

```
<meta charset="UTF-8">
<style>
div {margin-top: 20px; border-top: 2px solid black;
width:400px;}
img {float:left}
</style>

<div>
  
  Deixa eu te levar pra ver um filme do Goddard<br>
  Deixa eu te levar pra gente jogar bilhar<br>
  Deixa eu te dizer você é foda na cama<br>
</div>
<div>
  
  Deixa eu te dizer <br>
  Eu te amo<br>
  Porra!<br>
</div>
<div>
  
  Deixa eu te dizer<br>
  Eu te amo<br>
  Porra!<br>
  Deixa eu te dizer <br>
  Eu te amo<br>
  Porra!<br>
</div>
```

Margin, Padding e Border

A representação dos elementos na janela do *browser* obedece a um modelo chamado box-model. Que representamos na figura abaixo:

1. Conteúdo; 2. Padding; 3. Border; 4. Margin

Cada uma dessas propriedades pode ser definida de maneira geral, ou de maneira específica, cada lado separadamente. A seguir uma lista com todas as propriedades definidas para trabalhar esses elementos.

Padding

Propriedade	Descrição	valores
padding	Define o espaçamento entre o conteúdo e a borda dos elementos, de todos os lados numa mesma declaração	padding-top padding-right padding-bottom padding-left
padding-top	Define o espaçamento para a parte superior do elemento	tamanho (em qualquer unidade de medida aceita) %
padding-right	Define o espaçamento para a parte direita do elemento	tamanho %
padding-bottom	Define o valor do espaçamento na parte de baixo do elemento	tamanho %
padding-left	Define o espaçamento para a parte esquerda do elemento	tamanho %

Border

Propriedade	Descrição	valores
Border	Define todas as propriedades de todas as bordas ao mesmo tempo	espessura estilo cor
border-color	Muda a cor de todas as bordas. Pode receber até quatro valores	cor
border-style	Define o estilo de todas as bordas em uma só declaração. Pode ter de um a quatro estilos	none hidden dotted dashed solid double groove ridge inset outset
border-width	Define a espessura de todas as bordas em uma só declaração. Pode ter de um a quatro valores	tamanho thin medium thick
border-top	Define as propriedades da borda do topo em uma declaração.	espessura estilo cor
border-top-color	Define a cor da borda do topo	cor
border-top-style	Define o o estilo da borda do topo	none hidden dotted dashed solid double groove ridge inset outset
border-top-width	Define a espessura da borda do topo	tamanho thin medium thick
border-right	Define as propriedades da borda da direita em uma declaração.	espessura estilo cor
border-right-color	Define a cor da borda da direita	cor

Propriedade	Descrição	valores
border-right-style	Define o o estilo da borda da direita	none double hidden groove dotted ridge dashed inset solid outset
border-right-width	Define a espessura da borda da direita	tamanho thin medium thick
border-bottom	Define as propriedades da borda de baixo em uma declaração.	espessura estilo cor
border-bottom-color	Define a cor da borda de baixo	cor
border-bottom-style	Define o o estilo da borda de baixo	none double hidden groove dotted ridge dashed inset solid outset
border-bottom-width	Define a espessura da borda de baixo	tamanho thin medium thick
border-left	Define as propriedades da borda da esquerda em uma declaração.	espessura estilo cor
border-left-color	Define a cor da borda de baixo	cor
border-left-style	Define o o estilo da borda da esquerda	none double hidden groove dotted ridge dashed inset solid outset
border-left-width	Define a espessura da borda da esquerda	tamanho thin medium thick

Código Exemplo:

```
<style>
div { width:75px; height:100px; margin:10px; float:left}
.a {border: 1px solid #000}
.b {border: 2px dotted #000}
.c {border: 2px dashed #000}
.d {border: 3px double #000}
</style>

<div class="a"></div>
<div class="b"></div>
<div class="c"></div>
<div class="d"></div>
```

Aparência:

Nem todos os valores de border-style são reconhecidos por todos browsers. Usa-se valores diferentes para cada borda para construir uma ilusão de volume. A propriedade border também

pode ser usada para criar figuras triangulares, como podemos ver no exemplo abaixo:

Código exemplo:

```
<style>
div {float:left}
.a {width:100px; height:100px;
background-color: #666;
border-right: 2px solid #000;
border-bottom: 2px solid #000;
border-left: 2px solid #aaa;
border-top: 2px solid #aaa;}

.b {border-right: 100px solid #000;
border-bottom: 100px solid #fff;}
</style>

<div class="a"></div>
<div class="b"></div>
```

Aparência:

Margin

Propriedade	Descrição	valores
margin	Define a margem externa do elemento de uma vez para todos os lados	margin-top margin-right margin-bottom margin-left
margin-top	A margem superior do elemento	tamanho (em qualquer unidade de medida aceita) %
margin-right	Define a margem direita do elemento	tamanho %
margin-bottom	Define a margem na parte de baixo do elemento	tamanho %
margin-left	Define a margem esquerda do elemento	tamanho %

É possível fornecer valores distintos mesmo utilizando as declarações-atalho. Para tanto, pode-se declarar mais de um valor separado por espaço. Em caso de dois valores, o *browser* aplicará o primeiro para o topo e para baixo, e o segundo para direita e esquerda. Em caso de três valores, Aplicará um para o topo, outro para direita e esquerda e o terceiro para a parte de baixa. Em caso de quatro valores, a ordem será topo, direita, baixo e esquerda.

ex: `div { border-color: #333 #666 }`

Width, height, overflow

Além dos espaçamentos e bordas, podemos definir via CSS o tamanho das caixas de conteúdo (que podem ser objetos e imagens também). Para isso, existem duas propriedades, *width* e *height*.

Ambas podem ser definidas em valores absolutos ou relativos. Em caso de valores absolutos, pode se usar qualquer unidade permitida no CSS. Valores relativos, são definidos em porcentagens, relativas ao tamanho do elemento aonde o segundo está inserido.

A largura total do elemento, no entanto, será a soma da largura definida mais os valores de *padding*, *border* e *margin*.

Código exemplo:

```
<style type="text/css">
li {list-style: none; background-color:#aaa;
width:100px; height:50px}
.a2 { padding:10px}
.a3 { padding:10px; border:10px solid #000;}
.a4 { padding:10px; border:10px solid #000; margin:10px;}
</style>

<ul>
<li class="a1">1</li>
<li class="a2">2</li>
<li class="a3">3</li>
<li class="a4">4</li>
</ul>
```

Aparência:

O valor da altura pode variar de acordo com o conteúdo interno do elemento. Caso haja um valor de altura definido, a propriedade *overflow* pode determinar o comportamento do conteúdo que ultrapassar o container.

Código exemplo:

```
<style type="text/css">
div {float:left; background-color:#aaa;
width:100px; height:100px; margin:10px}
.a1 { overflow:hidden;}
.a2 { overflow:scroll;}
.a3 { overflow:auto;}
.a4 { overflow:visible;}
</style>

<div class="a1">A experimentação a criação, a descoberta e a invenção devem constituir a forma moderna do ensino universitário, de sua instrução e de sua formação</div>
<div class="a2">A experimentação a criação, a descoberta e a invenção</div>
<div class="a3">A experimentação a criação, a descoberta e a invenção</div>
<div class="a4">A experimentação a criação, a descoberta e a invenção</div>
```

Aparência:

Background

As caixas de conteúdo criadas pelos elementos (X)HTML, tem o fundo transparente a priori. Podemos utilizar a propriedade *background*, bem como suas variantes para determinar cor ou textura de fundo.

Propriedade	Descrição	Valores
background	Um atalho para definir todas as propriedades do fundo em uma só declaração	background-color background-image background-repeat background-attachment background-position
background-attachment	Define se o fundo do elemento é fixo ou rola junto com a página	scroll fixed
background-color	Determina a cor do fundo do elemento	color-rgb color-hex color-name transparent
background-image	Define uma imagem para servir de fundo do elemento	url(endereço) none
background-repeat	Determina se e como a imagem do fundo repete	repeat repeat-x repeat-y no-repeat

Propriedade	Descrição	Valores
background-position	Determina a posição inicial da imagem do fundo	top left top center top right center left center center center right bottom left bottom center bottom right x-% y-% x-pos y-pos

A propriedade *background-image* é uma das mais utilizadas para inserir imagem em elementos via CSS. Assim que se criam fundos texturizados ou degradês, por exemplo. Podemos ver como se comportam as propriedades de *background* no seguinte exemplo.

Código:

```
<style type="text/css">
div {float:left; border:1px solid #000; width:100px;
height:100px; margin:10px}
.a1 { background-image:url('asterisco.gif') }
.a2 { background-image:url('asterisco.gif');
background-repeat: repeat-x}
.a3 { background-image:url('asterisco.gif');
background-repeat: repeat-y}
.a4 { background-image:url('asterisco.gif');
background-repeat: no-repeat}
.a5 { background-image:url('asterisco.gif');
background-repeat: no-repeat;
background-position: bottom}
.a6 { background-image:url('asterisco.gif');
background-repeat: no-repeat;
background-position: center}
</style>

<div class="a1"></div>
<div class="a2"></div>
<div class="a3"></div>
<div class="a4"></div>
<div class="a5"></div>
<div class="a6"></div>
```

Aparência:

Float e clear

Uma vez que conseguimos determinar o tamanho das caixas de conteúdo, é importante entender as formas com que as caixas de conteúdo podem ser posicionadas dentro do campo, que é a janela do navegador.

Já explicamos que os objetos podem ser do tipo *block* ou *inline*, e que elementos do tipo *block* são dispostos sempre em uma nova linha. Através da propriedade *float*, conseguimos fazer com que esses objetos se comportem de forma fluida. Os exemplos abaixo ajudam a compreender melhor esse recurso. Primeiro, vejamos como se comporta o seguinte exemplo, sem *float* definido.

Código:

```
<style type="text/css">
.quadrada {background-color:red; width:50px;height:50px}
</style>


<div class="quadrada">
</div>
<p>A transmissão&atilde;o passiva de conhecimentos adquiridos &
uma fal&aacute;a: s&oacute; se conhece verdadeiramente, de
mediante sele&ccedil;&atilde;o cr&iacute;tica</p>
<p>D&eacute;cio Pignatari, contracomunica&ccedil;&atilde;o</p>
```

Aparência:

A transmissão passiva de conhecimentos adquiridos é uma falácia: só se conhece verdadeiramente, de modo criativo, mediante seleção crítica

Décio Pignatari, contracomunicação

Como os elementos estão sem *float* definido, cada elemento aparece em uma nova linha. Agora se definirmos uma propriedade *float* para os dois primeiros objetos, temos o seguinte resultado:

Código:

```
<style type="text/css">
.quadrada {background-color:red; width:50px;height:50px}
img {float:left}
div {float:right}
</style>


<div class="quadrada">
</div>
<p>A transmissão&atilde;o passiva de conhecimentos adquiridos &
uma fal&aacute;a: s&oacute; se conhece verdadeiramente, de
mediante sele&ccedil;&atilde;o cr&iacute;tica</p>
<p>D&eacute;cio Pignatari, contracomunica&ccedil;&atilde;o</p>
```

Aparência:

Décio Pignatari, contracomunicação

Podemos notar que a imagem, com *float:left* permanece na esquerda, enquanto a div, com *float:right*, vai para a direita, e o texto preenche o espaço entre as duas de forma fluida. Para interromper o fluxo, existe uma propriedade chamada *clear*, que pode interromper o fluxo da direita, da esquerda e de ambos lados. As figuras abaixo, onde as divs estão parte com declaração *float:left* e parte com *float:right* mostram como a declaração *clear* influi no layout:

Código:

```
<style type="text/css">
div { border:1px solid #000; width:50px;
height:50px; margin:10px}
.left { float: left }
.clear { clear: both }
.right { float:right }
</style>

<div class="left"></div>
<div class="left"></div>
<div class="left"></div>
<div class="clear"></div>
<div class="right"></div>
<div class="right"></div>
<div class="right"></div>
```

Aparência:

Position

Além do controle via *float* e *clear*, é possível determinar a posição exata de um objeto no campo. Para isso, podemos determinar valores para a posição do topo, da esquerda, da direita ou da parte de baixo dos elementos, mas antes de definirmos os valores, precisamos decidir se o posicionamento dele é absoluto ou relativo.

Através de *position:absolute*, podemos definir os valores absolutos da posição dos elementos. Se o elemento aonde estiver contido também tiver definido o posicionamento, a posição dos elementos internos será relativa a ele:

Código:

```
<style type="text/css">
div { position:absolute;
    border:1px solid #000; }
.caixa { width:300px; height:300 px }
.posicionada {
    position:absolute;
    top:50px;
    right:50px;
    width:50px;
    height:50px;
    background:#000
}
</style>

<div class="caixa">
<div class="posicionada"></div>
</div>
```

Aparência:

Enquanto o posicionamento absoluto coloca os elementos numa posição determinada do plano cartesiano, o posicionamento relativo trata os elementos em relação à posição que ele ocuparia na página, como podemos ver abaixo:

Código:

```
<style type="text/css">
h4 { width:300px; |
    line-height:50px }
.deslocado {
    position:relative;
    top:-20px;
    color:red }
</style>

<h4 class="left">
Entramos na era da linguagem.
<span class="deslocado">Do código e da mensagem. </span>
Da informação e da comunicação
</h4>
```

Aparência:

Do código e

Entramos na era da linguagem.

da mensagem.

Da informação e da

comunicação

A frase destacada, neste caso, aparece 20px acima do lugar onde deveria estar, pois seu topo foi deslocado em -20px.

Z-index

Algumas vezes, dependendo do posicionamento dos elementos, pode ocorrer que um elemento fique sobre o outro. Para ordenar o posicionamento dos elementos no eixo vertical, existe a propriedade z-index. Na imagem abaixo, o quadrado vermelho deveria estar abaixo dos demais, por vir primeiro no código, mas devido a seu valor de z-index, ele se sobrepõe aos demais.

Código:

```
<style type="text/css">
div {width:150px; height:150px}
.z1 {background-color: red;
    position: absolute;
    top:20px; left:40px;
    z-index:3}
.z2 {background-color: #aaa;
    position: absolute;
    top:60px;
    z-index:2}
.z3 {background-color: black;
    position: absolute;
    top:100px; left:80px;
    z-index:1}
</style>

<div class="z1"></div>
<div class="z2"></div>
<div class="z3"></div>
```

Aparência:

Media Types

Agora que já fizemos um percurso pelas propriedades mais importantes do CSS, cabe ressaltar mais um importante recurso que a linguagem oferece aos programadores e designers, a capacidade de adaptação do *layout* de um site a diversos suportes de mídia, como explica o site *w3schools*:

Algumas propriedades CSS são projetadas para serem utilizadas apenas em certos meios

de comunicação. Por exemplo, a propriedade “voice-family” é projetada para browsers que transformam texto em áudio. Algumas outras propriedades podem ser utilizadas para diferentes tipos de mídia. Por exemplo, a propriedade “font-size” pode ser utilizada tanto para tela quanto para mídia impressa, mas talvez com valores diferentes. Um documento geralmente precisa de uma fonte de tamanho maior em uma tela do que no papel, e fontes sans-serif são mais fáceis de ler na tela, enquanto fontes com serifa são mais fáceis de ler no papel.⁹

Para designarmos uma determinada propriedade a um determinado tipo de mídia, precisamos usar a declaração @media, e definirmos a mídia a que esta declaração se aplica, como mostra o exemplo abaixo:

```
@media screen 000 {
  p {font-family:verdana,sans-serif;font-size:14px}
}
@media print {
  p {font-family:times,serif;font-size:10px}
}
```

A tabela a seguir lista os *media type* possíveis:

Media Type	Descrição
all	Para todos os tipos de mídia
aural	Utilizado para sintetizadores de voz (para cegos)
braille	Para dispositivos táteis de feedback em braile
embossed	Para impressoras de braile
handheld	para telas pequenas ou dispositivos portáteis
print	para impressoras
projection	para projeções
screen	para telas de computador

⁹ Some CSS properties are only designed for a certain media. For example the “voice-family” property is designed for aural user agents. Some other properties can be used for different media types. For example, the “font-size” property can be used for both screen and print media, but perhaps with different values. A document usually needs a larger font-size on a screen than on paper, and sans-serif fonts are easier to read on the screen, while serif fonts are easier to read on paper. (Refsnes Data, w3schools.com http://w3schools.com/css/css_mediatypes.asp)

Media Type	Descrição
tty	para dispositivos de texto com caracteres de largura fixa, como teletipo ou terminais
tv	para televisões e dispositivos afim

É possível se fazer declarações específicas dentro de um código CSS utilizando a sintaxe descrita anteriormente, ou construindo arquivos separados para cada tipo de mídia, que podem ser linkados pelo arquivo (X)HTML, da seguinte maneira:

```
<link rel="stylesheet" type="text/css"
media="screen, projection, handheld"
href="screen.css"/>
```

```
<link rel="stylesheet" type="text/css"
media="print" href="print.css"/>
```

É interessante desenvolver códigos específicos para mídia impressa, removendo elementos como fundos coloridos, menus de navegação e anúncios; buscando otimizar a impressão e evitar o desperdício de tinta, por exemplo. Também é interessante mudar o tamanho do texto, que pode ser menor do que o que normalmente é apresentado na tela.

Para dispositivos móveis, também é importante remover elementos desnecessários, bem como colunas, otimizando o site para o funcionamento do site em telas pequenas.

Boas práticas de código

Com isso, demos uma visão geral dos recursos mais importantes que um designer pode precisar se quiser se aventurar na produção de seus próprios códigos. Mas além do domínio das propriedades e recursos oferecidos pela linguagem, alguns autores de material técnico, como McFarlane, por exemplo, sugerem algumas “boas práticas” que podem ajudar no desenvolvimento de códigos.

A primeira sugestão é comentar os códigos, inserindo indicações sobre do que cada parte do código se trata, ajudando a identificar os arquivos e facilitando assim, a edição posterior quer seja pelo autor, quer seja por outros. Além dos comentários, é interessante também agrupar estilos que digam respeito às mesmas áreas, para que seja mais fácil de encontrá-los depois. Comentários, em CSS podem ser escritos da seguinte maneira:

```
/* este texto é um comentário, e não tem ne-  
nhum efeito sobre o resto do código */
```

É muito comum utilizarmos comentários para desativarmos trechos do código temporariamente.

Outra boa prática é de nomear de maneira clara as classes e *ids*, por propósito, e não por aparência. Se o programador utilizar uma classe chamada *vermelho*, para ser aplicada aos links, por exemplo, e depois decidir que os links vão ser azuis, o código ficará confuso. O Autor também sugere evitar nomes muito curtos, pois também dificultam a compreensão posterior.

Em sites muito complexos, o arquivo CSS pode ficar muito grande, dificultando a localização dos estilos mesmo se eles estiverem agrupados em blocos comentados. Para facilitar o trabalho, McFarlane sugere que se organize o trabalho em vários arquivos separados, que podem ser reunidos em um só arquivo mediante a declaração *@import*, como exemplificado abaixo:

Para isto ocorrer, bastaria que o conteúdo do arquivo *base.css* fosse o seguinte:

```
@import url(principal.css);
@import url(tabelas.css);
@import url(formularios.css);
```

Incompatibilidade entre browsers

Um dos maiores pesadelos de todo mundo que escreve código CSS é a inconsistência da interpretação dos dados entre os *browsers*. Embora cada vez mais os navegadores estejam oferecendo suporte adequado para CSS, ainda existem alguns elementos que são interpretados de maneira diferente em *browser* diferentes.

Em um projeto profissional, muitas vezes o programador recebe uma tela, que é uma imagem exata de como a interface deve ficar depois de desenvolvida, e seu trabalho é fazer com que o site seja visualizado exatamente igual em todos *browsers*, ou no máximo possível deles. Para isso, o programador deve testar o site em vários navegadores e plataformas diferentes¹⁰, e encontrando inconsistências, deve utilizar de alguns recursos para dar duas ou mais declarações diferentes para cada elemento, de modo com que cada programa reconheça apenas uma delas.

Uma das soluções possíveis é a criação de arquivos diferentes para cada *browser*, e a utilização de um script que reconhece o programa utilizado pelo usuário e aplica automaticamente o arquivo desejado. O código abaixo reconhece se o *browser* é Internet explorer 6.0 ou 5.0, por exemplo:

```
<link href="global_styles.css" rel="stylesheet"  
type="text/css" />
```

¹⁰ Como dificilmente alguém pode ter todos os browsers utilizados instalados, existem ferramentas que emulam a aparência de um determinado website em vários *browsers*. O site <http://browsershots.org> oferece um serviço desse tipo, ao gerar screenshots com a aparência de um determinado site em uma série de *browsers* diferentes.

```

<!--[if IE 6]>
<link href="IE6_styles.css"
rel="stylesheet" type="text/css" />
<![endif]-->
<!--[if lte IE 5]>
<link href="IE5_styles.css"
rel="stylesheet" type="text/css" />
<![endif]--

```

Outra forma de ajustar determinada declaração a determinado *browser*, sem ter a necessidade de se utilizar múltiplos arquivos é utilizando alguns tipos de declarações que só são compreendidas ou ignoradas por determinados navegadores. Na página seguinte, temos uma lista compreensiva executada por Paul Irish, que pode ser encontrada no endereço <http://paulirish.com/2009/browser-specific-css-hacks/>

A prática, que é um meio de hackear o código, é condenada por alguns designers, pois o código fica inválido, mas em caso de pequenas alterações, pode ser uma boa idéia, ao invés de se utilizar um arquivo diferente só para corrigir alguns pequenos erros.

Agora que já estudamos a principal linguagem utilizada para reger a forma das páginas web, passaremos ao estudo das linguagens que são utilizadas para dar interatividade às páginas. Com as informações aqui relacionadas, um designer consegue dar forma a um código HTML. Mas até aqui, a forma é estática, uma vez que são poucos os elementos CSS que respondem à interatividade. Para adicionar efeitos interativos, uma linguagem meramente de descrição ainda não é suficiente, e é preciso recorrer à agilidade conferida pelas linguagens de *script*.

Como vimos, a (X)HTML é uma linguagem de marcação utilizada para estruturar documentos de texto e transformá-los em hipertexto. Através de suas *tags*, o (X)HTML marca semanticamente elementos do texto e cria grupos de elementos, mas sozinho, tem poucos recursos para dar forma aos documentos. Como comenta David McFarlane(2006):

Antes do CSS, os webdesigners eram limitados pelo esquema e as opções de estilo do HTML. E se você surfava na web em 1995, então você entende a ênfase em limitados. O HTML (...) ainda é o alicerce de todas as páginas na World Wide Web, mas simplesmente não é uma ferramenta de design. Claro, o HTML oferece as opções básicas de formatação de texto, imagens, tabelas e outros elementos da página Web e mestres webdesigners pacientes e meticulosos podem criar páginas que apareçam muito bem usando apenas HTML. Mas o resultado são muitas vezes páginas web lentas, imprevisíveis e carregadas de código enferrujado.¹¹

Nas versões anteriores da linguagem, até existiam *tags* especiais para determinar a apresentação dos documentos, como a *tag* <fonte>, mas como elas não têm nenhum significado estrutural, foram suprimidas das versões mais recentes, em favor da utilização de uma linguagem específica para isso. Na web, a linguagem mais popular para definição de estilo, embora não seja a única existente é o CSS, uma sigla para Cascading Style Sheets (folhas de estilo em cascata).

¹¹ Before CSS, Web designers were limited to the layout and styling options of HTML. And if you surfed the Web in 1995, then you understand the emphasis on limited. HTML (...) still forms the foundation of all pages on the World Wide Web, but it's simply not a design tool. Sure, HTML provides basic formatting options for text, images, tables, and other Web page elements and patient, meticulous Web masters can make pages look pretty good using only HTML. But the result's often sluggish, unpredictable Web pages laden with clunky code. McFarlane, David Sawyer. CSS: The Missing Manual. Sebastopol: O'Reilly, 2006


```

/***** Hacks por seletores *****/

/* IE6 and below */
* html #uno { color: red }

/* IE7 */
*:first-child+html #dos { color: red }

/* IE7, FF, Saf, Opera */
html>body #tres { color: red }

/* IE8, FF, Saf, Opera (Everything but IE 6,7) */
html>*/body #cuatro { color: red }

/* Opera 9.27 and below, safari 2 */
html:first-child #cinco { color: red }

/* Safari 2-3 */
html[xmlns*=""] body:last-child #seis { color: red }

/* safari 3+, chrome 1+, opera9+, ff 3.5+ */
body:nth-of-type(1) #siete { color: red }

/* safari 3+, chrome 1+, opera9+, ff 3.5+ */
body:first-of-type #ocho { color: red }

/* saf3+, chrome1+ */
@media screen and (-webkit-min-device-pixel-ratio:0) {
  #diez { color: red }
}

/* iPhone / mobile webkit */
@media screen and (max-device-width: 480px) {
  #veintiseis { color: red }
}

/* Safari 2 - 3.1 */
html[xmlns*=""]:root #trece { color: red }

/* Safari 2 - 3.1, Opera 9.25 */
*|html[xmlns*=""] #catorce { color: red }

/* Everything but IE6-8 */
:root *> #quince { color: red }

/* IE7 */
*+html #dieciocho { color: red }

/* Firefox only. 1+ */
#veinticuatro, x:-moz-any-link { color: red }

/* Firefox 3.0+ */
#veinticinco, x:-moz-any-link, x:default { color: red }

/***** Hacks por atributos *****/

/* IE6 */
#once { _color: blue }

/* IE6, IE7 */
#doce { *color: blue; /* or #color: blue */ }

/* Everything but IE6 */
#diecisiete { color/**/: blue }

/* IE6, IE7, IE8 */
#diecinueve { color: blue\9; }

/* IE7, IE8 */
#veinte { color/*\**/: blue\9; }

/* IE6, IE7 -- acts as an !important */
#veintesiete { color: blue !ie; } /* string after ! can be anything */

```

4.5

Forma em movimento: linguagens para interatividade

A primeira geração de websites era, como já dissemos, estática. Com excessão dos *hiperlinks*, que adicionavam interação ao hipertexto, as páginas web eram, no começo dos anos 90, em geral, estáticas. Tirando os gifs animados, que eram recursos muito utilizados nos primórdios dos *browsers* gráficos¹, não havia movimento algum.

Em 1995, a *Netscape* que já havia licenciado a tecnologia *Java* para a inserção de *applets*² em páginas web, lançou uma linguagem mais simples, o *JavaScript*, que possibilitava a adição de conteúdo dinâmico às páginas:

JavaScript atraiu muita atenção de programadores, porque o que as pessoas queriam na época (e ainda querem) é a capacidade de ir um passo além do HTML e adicionar um pouco de código que torne uma página da web dinâmica — que faça mover as coisas, responder a entradas dos usuários, ou mudar de cor, que faça com que novas janelas pop-up surjam, ou que gere um caixa de diálogo para fazer uma pergunta, com uma resposta necessária para prosseguir — coisas que o HTML não pode expressar. Isso é quando você realmente precisa de uma linguagem de programação, mas algo mais simples do que Java ou C + +.³

1 Segundo Marco Aurélio Machado, no começo da década de 90, os gifs animados reinavam nas páginas, web. “quanto mais animação melhor. Era comum entrar em uma página com tantas animações e sentir-se perdido.” (Machado, 2003)

2 Um applet é um pequeno programa que pode ser inserido em uma página web e ser executado dentro da janela do browser. (Negrino e Smith, 2007)

3 *JavaScript* attracted a lot of developer attention, because what people wanted back then (and still want) is the ability to go one step beyond HTML and add a little bit of code that makes a web page dynamic - that makes things move, respond to user input, or change color; that makes new windows pop up; or that raises a dialog box to ask a question, with an answer necessary to proceed - things that HTML cannot express. That's really where you need a programming language, but something simpler than Java or C++. (Brendan Eich in Anderssen, 1998)

Baseado no *JavaScript*, foi criado o *ECMAScript*, um padrão de linguagem, que deu origem ao *JScript*, criado pela *Microsoft*, e ao *Action Script*, utilizado pelo *Flash*. Cada uma dessas linguagens tem características específicas, e implementam alguns recursos diferente umas das outras, mas não entraremos aqui em detalhes.

Através do uso desse tipo de linguagem, programadores conseguem ampliar os recursos de interatividade de uma página, criando pequenas animações, jogos ou validando formulários, por exemplo. A “mania de ter coisas mechendo” se espalhou entre os sites, segundo Marco Aurélio Machado, e o *JavaScript* servia bem a esse propósito. Milhares de códigos se propagaram pela rede e eram aplicados sem muito discernimento pelos programadores. Começaram também a surgir as páginas personalizadas, e isso foi aos poucos tornando a rede mais interativa.

A linguagem *JavaScript*, trabalhando em conjunto com as características do browser relacionado, é uma tecnologia amplificadora da Web. Quando empregada no computador do cliente, a linguagem pode ajudar a transformar uma página de conteúdo estático em uma experiência envolvente, interativa e inteligente. As aplicações podem ser tão sutis quanto acolher o visitante de um site com a saudação “Bom dia!” Quando é de manhã na hora do computador cliente, mesmo que seja a hora do jantar onde o servidor está localizado. Ou podem ser muito mais óbvias, tais como mostrar o conteúdo de um slide show em um download de uma página, enquanto o código *JavaScript* controla a seqüência de esconder, mostrar e “voar deslizando” durante as transi-

ções ao navegar pela apresentação. (Goodman e Morrison, 2004)⁴

Essa possibilidade de criar aplicativos reside na capacidade dos navegadores de alterar as propriedades do arquivo em tempo real, seja por interação do usuário ou pelo tempo, por exemplo. Para alterar o documento, essas linguagens se amparam em uma especificação do W3C chamada *Document Object Model* (DOM), que define eventos e uma maneira padrão de se acessar cada elemento existente na estrutura de um determinado documento. (tabela nas páginas a seguir)⁵

Scripts, assim como os arquivos CSS podem ser escritos em um arquivo separado, e anexados via a *tag* <link>, podem ser escritos dentro do arquivo HTML, dentro da *tag* <script> ou podem ser escritos no meio do código HTML, através de atributos especiais dos elementos. Através do DOM, um *script* consegue reconhecer certos “eventos”, que ocorrem quando por exemplo, o documento é carregado, ou quando o mouse passa sobre algum elemento. A partir desses eventos, é possível de se programar o resultado da interação do usuário com o site através de uma linguagem de script como *JavaScript*, e assim, construir aplicativos online.

O computador reconhece quando determinados “eventos” acontecem, como o clique de um

4 The JavaScript language, working in tandem with related browser features, is a Web-enhancing technology. When employed on the client computer, the language can help turn a static page of content into an engaging, interactive, and intelligent experience. Applications can be as subtle as welcoming a site’s visitor with the greeting “Good morning!” when it is morning in the client computer’s time zone—even though it is dinner-time where the server is located. Or applications can be much more obvious, such as delivering the content of a slide show in a one-page download while JavaScript controls the sequence of hiding, showing, and “flying slide” transitions while navigating through the presentation. Goodman e Morrison, 2004

5 W3C, Document Object Model Activity Statement <http://www.w3.org/DOM/Activity>

Objetos DOM

Objeto	Descrição
Anchor	Representa um elemento âncora (hiperlink) <a>
Applet	Representa um elemento <applet>
Area	Representa uma área em um image map <area>
Base	Representa um elemento <base>
Basefont	Representa um elemento <basefont>
Body	Representa o corpo do HTML
Button	Representa um botão em um formulário HTML <input type="button">
Checkbox	Representa um checkbox em um formulário <input type="checkbox">
Document	Utilizado para acessar todos elementos de uma página
Event	Representa o estado de um evento, que pode ser o elemento onde o evento ocorreu, o estado das teclas do teclado, a localização do mouse e o estado dos botões do mouse
FileUpload	Representa um input do tipo "file" <input type="file"> em um formulário
Form	Representa um formulário <form>
Frame	Representa um <frame>
Frameset	Representa um <frameset>
Hidden	Representa um campo oculto em um formulário <input type="hidden">
History	Um objeto pré-definido que pode ser acessado através da propriedade history do objeto window. É uma matriz das urls visitadas em uma determinada janela
Iframe	Representa um <iframe>
Image	Representa um elemento
Link	representa o elemento <link>
Location	Fornece informação sobre a url corrente
Meta	representa um elemento <meta>
Navigator	Fornece informações sobre o browser do usuário
Option	Representa um elemento <option>, uma opção em uma lista
Password	Representa um campo de senha <input type="password"> em um formulário
Radio	Representa um botão tipo radio em um formulário <input type="radio">
Reset	Representa um botão de reset em um formulário <input type="reset">
Screen	Contém informação sobre o tipo de tela do usuário
Select	Representa uma lista de seleção em um formulário. <select>
Style	Representa uma declaração de estilo individual
Submit	Representa um botão de envio em um formulário <input type="submit">
Table	Representa um elemento <table>
TableData	Representa um elemento <td>
TableHeader	Representa um elemento <th>
Text	Representa um campo de input de texto em um formulário <input type="text">
Textarea	Representa um elemento <textarea>
Window	Corresponde à janela do navegador

Lista de Eventos HTML DOM

Evento	Ocorrência
onabort	Quando o carregamento de um elemento é interrompido
onblur	Quando um elemento perde o foco
onchange	Quando o conteúdo de um campo muda
onclick	Quando o objeto recebe é clicado
ondblclick	Quando o objeto recebe um duplo-clique
onerror	Quando ocorre um erro no carregamento de um arquivo ou imagem
onfocus	Quando um elemento entra em foco
onkeydown	Quando uma tecla é pressionada
onkeypress	Quando uma tecla é pressionada ou segurada
onkeyup	Quando uma tecla é solta
onload	Quando uma página ou uma imagem termina de ser carregada
onmousedown	Quando um botão do mouse é pressionado
onmousemove	Quando o mouse se move
onmouseout	Quando o ponteiro do mouse sai de um objeto
onmouseover	Quando o mouse passa sobre um objeto
onmouseup	Quando um botão do mouse é solto
onreset	Quando o botão de reset é clicado
onresize	Quando a janela é redimensionada
onselect	Quando texto é selecionado
onsubmit	Quando o botão "submit" é clicado
onunload	Quando o usuário deixa a página

Atributos de eventos

Propriedade	Descrição
altKey	Indica se a tecla alt foi pressionada
button	Retorna qual botão do mouse foi pressionado
clientX	Retorna a coordenada horizontal do pointer do mouse
clientY	Retorna a coordenada vertical do mouse
ctrlKey	Indica se a tecla control foi pressionada
metaKey	Indica se a tecla "meta" foi pressionada
relatedTarget	Retorna o elemento relacionado ao elemento que desencadeou o evento
screenX	Retorna as coordenadas horizontais do mouse quando um determinado evento foi acionado
screenY	Retorna as coordenadas verticais do mouse quando um determinado evento foi acionado
shiftKey	Indica se a tecla "shift" foi pressionada
type	retorna o nome do evento

botão, o passar do mouse sobre um determinado campo, o envio de um formulário ou o fim do carregamento da página, por exemplo. O princípio da linguagem reside em criar certas funções, que podem ser aplicadas quando ocorrer algum desses eventos. Um webdesiner não precisa saber escrever funções para conseguir aplicar alguns *scripts* a suas páginas. Existe uma série de bibliotecas de *script* disponíveis na rede, que realizam as mais variadas funções. O designer só precisa saber como inserir as funções desejadas no código, e linkar a biblioteca desejada à sua página, para fazê-los funcionar.

Dada esta possibilidade, e dada a grande complexidade envolvida, não estudaremos a fundo aqui o *JavaScript*, mas apenas o suficiente para que um interessado possa se apropriar de uma dessas muitas bibliotecas disponíveis e aplicá-las a seus projetos.

Mas antes de partirmos para uma explicação mais técnica, cabe ressaltarmos um novo paradigma de utilização de *JavaScript* para adicionar interatividade às páginas.

Ajax

O modelo clássico de aplicativos web funciona assim: A maioria das ações do usuário na interface faz uma solicitação HTTP para um servidor web. O servidor realiza algum processamento — recuperando de dados, processando números, conversando com vários sistemas internos — e então retorna uma página HTML para o cliente. É um modelo adaptado do uso original da Web como um meio de hipertexto, mas como os fãs de *The Elements of User Experience* sabem, o que faz da Web boa para

hipertexto não necessariamente faz dela boa para aplicativos *software*.⁶

Isto significava que a cada interação que o usuário fazia, como clicar em algum link, a página recarregava, e um espaço (inclusive de tempo) em branco, de carregar a nova página ocorria. Programadores desenvolveram uma maneira de fazer com que só uma parte da página, aquela que realmente precisa mudar recarregue, evitando, esse espaço em branco entre o carregamento das páginas. A essa técnica, que ficou conhecida através do serviço *Google Maps*, Jesse James Garrett batizou de *Ajax* — uma sigla para *Asynchronous JavaScript + XML*. A técnica incorpora, segundo ele:

- representação de acordo com os padrões, utilizando XHTML e CSS
- interação e disposição interativa, utilizando o DOM
- Intercâmbio e manipulação de dados usando XML e XSLT
- recebimento assíncronico de dados usando XMLHttpRequest
- e JavaScript para costurar tudo⁷

Para que isso funcione, é criado um mecanismo especial que "serve de intermediário" entre o servidor e o usuário, fazendo com que a comunicação ocorra de maneira assíncrona. Com esse novo modelo de programação interativa, foi possível o desenvolvimento de toda uma nova leva de aplicativos online, baseadas em HTML e CSS, mas com recursos de interatividade que só eram

6 The classic web application model works like this: Most user actions in the interface trigger an HTTP request back to a web server. The server does some processing — retrieving data, crunching numbers, talking to various legacy systems — and then returns an HTML page to the client. It's a model adapted from the Web's original use as a hypertext medium, but as fans of *The Elements of User Experience* know, what makes the Web good for hypertext doesn't necessarily make it good for software applications. (Garret, Jesse James, 2005)

7 Ibid.

possíveis de ser alcançados até então por meio de aplicativos criados em *Flash*.

classic web application model (synchronous)

Ajax web application model (asynchronous)

Jesse James Garrett / adaptivepath.com

fonte: Jesse James Garret / adaptive path.com

Inserindo código JavaScript

Como já dissemos, a linguagem usa o DOM para alterar aspectos ou elementos de uma página (X) HTML. Para tanto, conta com algumas funções

básicas chamadas métodos, que podem ser combinados para criar funções mais complexas.

Antes de tudo, precisamos saber como se inserir um trecho *JavaScript* num documento (X) HTML. Podemos inserir trechos de código tanto na seção *head* como na seção *body* do documento, utilizando a tag `<script>`:

```
<script type="text/javascript">
function message()
{
  alert("This alert box was called with the onload
event");
}
</script>
```

Outra maneira de inserir um código externo em um arquivo (X)HTML, pode ser conseguida mo da seguinte maneira:

```
<script type="text/javascript" src="xxx.js"></
script>
```

Pode-se criar funções e chama-las quando ocorrer um determinado evento em algum dos objetos, como exemplificado abaixo:

```
<html>
<head>
<script type="text/javascript">
function displaymessage()
{
  alert("Hello World!");
}
</script>
</head>

<body>
<form>
<input type="button" value="Click me!"
onclick="displaymessage()" />
</form>
</body>
</html>
```

Utilizando bibliotecas prontas

Como já dissemos, a habilidade de programar *scripts* não é necessária para se fazer utilização de *JavaScript* em algum projeto. Existe uma série de bibliotecas pré-programadas na internet que facilitam o trabalho, oferecendo uma série de funções e efeitos especiais para aplicação em websites. Boa parte delas contém documentação explicando como podem ser utilizadas.

Abaixo, relacionamos e indicamos os principais recursos de algumas das mais importantes bibliotecas *Ajax*. O site <http://javascriptlibraries.com/> oferece links para mais uma série de bibliotecas de recursos possíveis, para quem se interessar em conhecer mais delas e estudá-las mais a fundo.

jQuery

http://docs.jquery.com/Main_Page

jQuery é uma das principais bibliotecas existentes que podem ser utilizadas pelos designers para enriquecer a interatividade da página sem exigir muitas habilidades de programação. É possível se utilizar diretamente as funções disponíveis na biblioteca, assim como criar novas funções a partir delas.

Eles têm inclusive um tutorial específico para designers que ensina a utilizar alguns recursos simples, mas efetivos. O exemplo a seguir, abastado deste tutorial mostra como utilizar a biblioteca para criar uma lista de artigos aonde o sumário fica aparente e o conteúdo só aparece quando desejado.

Código exemplo:

```
<html lang="">
<head>
  <meta http-equiv="content-type" content="text/html;
  charset="UTF-8">
  <script src="http://jquery.com/src/jquery-
  latest.js"></script>
  <script>
    // quando a página estiver pronta
    $(document).ready(function(){

      // a linha abaixo faz esconder tudo que tiver a classe
      "corpo"
      $(".artigo .corpo").hide();

      //este bloco cria o link para mostrar o conteúdo do
      artigo
      $("#container .artigo span")
        .prepend("<p class='leia'><a href='' title='Leia o
      artigo'>Leia o artigo</a></p>");
      //determina o que acontece quando clica no link
      $(".acao p.leia a").click(function(event){
        $(this).parents("span").prev(".corpo").toggle();

        // Impede o link de ter seu comportamento padrão
        event.preventDefault();
      });
      //aqui fecha o bloco de declarações
    });
  </script>

</head>
<body>

  <div id="container">
    <h2>Contracomunicação</h2>
    <div class="artigo">
      <h3>Volta à Tribo</h3>

      <p class="resumo">Os meios são extensões do
      homem</p>
      <p class="corpo">Os meios são extensões do homem:
      a roda são os pés em rotação, a palavra falada e a música
      são extensões do ouvido, a palavra escrita e a fotografia são
      extensões do olho, a escultura é extensão do tato, o perfume
      uma extensão do olfato etc.
      </p>
      <span class="acao"></span>
    </div>
    <div class="artigo">
      <h3>Arte como Antiambiente</h3>

      <p class="resumo">Os media em sucessão e
      atrito</p>
      <p class="corpo">Os media em sucessão e atrito
      criam um ambiente irritante, quase abrasivo, especialmente
      agora, quando sentimos que o implosivo, o integrado e o
      sintético se chocam com o explosivo, o fragmentário e o
      analítico. Para Neutralizar a pressão do ambiente, que
      produz o stress e a irritação, urge um antídoto, um contra-
      irritante. A arte é esse contra-irritante.
      </p>
      <span class="acao"></span>
    </div>
  </div>
</body>
</html>
```

Aparência:

Contracomunicação

Volta à Tribo

Os meios são extensões do homem

[Leia o artigo](#)

Arte como Antiambiente

Os media em sucessão e atrito

[Leia o artigo](#)

Contracomunicação

Volta à Tribo

Os meios são extensões do homem

O meios são extensões do homem: a roda são os pés em rotação, a palavra falada e a música são extensões do ouvido, a palavra escrita e a fotografia são extensões do olho, a escultura é extensão do tato, o perfume uma extensão do olfato etc.

[Leia o artigo](#)

Arte como Antiambiente

Os media em sucessão e atrito

[Leia o artigo](#)

Cabe notar que as funções utilizadas, *hide()* e *toggle()* estão no arquivo externo, *jquery.js*, e o que o designer precisa saber é como inserir as funções no arquivo, que pode ser aprendido seguindo a documentação do site, bem como substituir as variáveis, que no caso são os elementos e as classes utilizadas, para que corresponda ao seu documento.

Principais recursos oferecidos:

Mecanismo *Ajax*, adiciona e remove classes a determinados elementos, esconde e mostra determinadas classes, substitui conteúdo, lida com propriedades CSS, desencadeia eventos em elementos determinados, permite o redimensionamento de elementos; fornece uma série de efeitos especiais como animações e efeitos de *fade in*, *fade out* e *slide*; detecção de *browser*, utilização de filtros de busca para aplicação de determinadas funções, junção de matrizes; efeitos especiais para interface, como drag and drop, redimensionamento de objetos.

Oferece ainda uma série de *widgets* pré-programados, como calendários, *displays* que abrem e fecham com efeito sanfona, Caixas de diálogo arrastáveis e redimensionáveis, barras de progresso e barras de rolagem, entre outros.

exemplos:

O *plugin Galleria* é um exemplo de recursos possíveis de serem utilizados com a biblioteca *Jquery*. Ele cria uma galeria de imagens a partir de uma lista de imagens HTML. O código desse script pode ser baixado em <http://code.google.com/p/galleria/downloads/list>

A JQuery User Interface Library oferece uma série de exemplos de recursos de interface pré-programadas, bem como o acesso aos códigos fontes. Acima, o exemplo de input para seleção de datas. <http://jqueryui.com/demos/datepicker/#default>

Prototype

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

Outra biblioteca de uso disseminado, com larga documentação possível, utilizada por grandes companhias como *Apple*, *CNN*, *Espn*, *Twitter*, *Last.fm*, *scribd*, *ebay* entre outros. O *prototype* não disponibiliza *widgets* prontos como o *jQuery*,

mas oferece recursos para programadores desenvolverem efeitos e aplicativos.

Recursos oferecidos:

Mecanismo Ajax; Extensões do DOM, Validação de formulários, entre outros.

Script.aculo.us

<http://script.aculo.us/>

Uma biblioteca desenvolvida com base no *framework*⁸ *Prototype* que fornece efeitos pré-programados para desenvolvimento de interfaces. A documentação é bastante extensa e fornece vários exemplos de utilização dos recursos oferecidos. Muitos efeitos são de uso muito simples, e só necessitam da inclusão de algumas linhas de código no arquivo (X)HTML.

Recursos oferecidos:

Auto completa formulários; efeitos para animações de objetos, como opacidade, redimensionamento, transformação, movimento, destaque entre outros; efeitos para abertura e fechamento de abas, para tocar arquivos de áudio; possibilidade de tornar elementos arrastáveis; processamento de efeitos em fila; rolagem da página ou de elementos; mostra e esconde objetos utilizando vários efeitos e rearranja listas, entre outros.

Abaixo, um exemplo de como utilizar um efeito de *fade out*. A linguagem fica extremamente simplificada pela função \$ que substitui o método *getElementById* do Javascript (Talbot, 2006).

```
<head>
<script src="javascripts/prototype.js" type="text/
javascript"></script>
<script src="javascripts/scriptaculous.js"
type="text/javascript"></script>
```

8 Um *Framework* é um tipo especial de biblioteca que oferece uma interface para programação de novas funções, com comportamentos padrão e extensibilidade, mas que em geral, não pode ser alterada. Oferece uma interface para programação.

```
</head>
<body>
<div id="fade_demo" style="width:80px;
height:80px; background:#c2defb; border:1px
solid #333;"></div>
<ul>
<li><a href="#" onclick="$('fade_demo').fade();
return false;">Clique aqui e o objeto vai sumir</a></li>
<li><a href="#" onclick="$('fade_demo').show();
return false;">Reset</a></li>
</ul>
</body>
```

Exemplos:

Prototype JavaScript Image Cropper UI - Demo

This is a sterile demo file for the post [Prototype JavaScript Image Cropper UI](#) by Dave Spurr.

Acima, um aplicativo para cortar imagens, baseado em Prototype e Script.aculo.us, pode ser baixada no endereço <http://www.defusion.org.uk/demos/060519/cropper.php>. A seguir, um screenshot da biblioteca *Glassbox*, também baseada nessas bibliotecas que cria bordas translúcidas sobre um fundo determinado. O código-fonte pode ser baixado em <http://www.glassbox-js.com/#home>

Outra biblioteca construída com base nas bibliotecas acima citadas, cria um efeito sanfona, horizontal ou vertical, pode ser baixada no site <http://stickmanlabs.com/accordion/>

Yahoo User Interface Library

<http://developer.yahoo.com/yui/>

A Yahoo também provém um *framework* para desenvolvimento de interfaces, conhecida como YUI. O *framework* oferece, além de recursos em *JavaScript*, *layouts* em CSS, entre outros. É uma biblioteca bastante extensa, e está hoje em sua terceira versão. Também oferece uma documentação online bastante extensa.

Principais Recursos:

Mecanismo *Ajax*, Animações, barra de navegação, *cookies*, formatação de datas, *drag* e *drop*, carregadores de imagens, ajuste de *layout*, *sliders*, menus de navegação, editores de texto, ferramentas para criação de gráficos dinâmicos, editores de texto, carrossel de imagens, entre outros

Acima, um exemplo de utilização do *framework* YUI2 para criação de gráficos interativos. <http://developer.yahoo.com/yui/examples/charts/charts-legend.html>

Example of Color Picker Built in a Dialog via JavaScript

This example explores the implementation of a Color Picker Control within the context of a popup Dialog. The control can be called up in the floating Dialog, a color can be chosen, and upon submission of the Color Picker's value the color data is submitted to the server using Dialog's built-in support for [Connection Manager](#).

Click the "Show Color Picker" button to get started. Use the other buttons on the configuration dashboard to explore different configurations of the Color Picker Controls. When you submit your color choice, the example uses Connection Manager to echo back [the form fields that were passed to the server](#).

Entre os vários exemplos oferecidos no site da YUI2, está este seletor de cores, construído em Javascript, que pode ser utilizado por aplicativos online. <http://developer.yahoo.com/yui/examples/colorpicker/colorpicker-dialog-from-script.html>

Dojo

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

Dojo é uma biblioteca bastante extensa, que compreende inúmeros recursos. Existem três versões dessa biblioteca, cada uma com uma gama de recursos. Através do feature explorer (<http://dojocampus.org/explorer/>) é possível se explorar uma gama de recursos e ver exemplos em ação de cada um deles. É possível se fazer players de vídeo, jogos e se utilizar fontes vetoriais.

Principais recursos:

Dojo base:

Reconhece elementos em banco de dados, drag e drop, efeitos de animação variados, barras de progresso, administração de formulários, filtros de pesquisa de elementos e estilos, entre outros

Dijit:

Caixas de diálogo, menus, barras de slide, paleta de cores, barras de progress, árvore de arquivos, botões, seletores de números, entre outros.

DojoX:

Galerias de imagens, seletor de cor, ferramentas decriação de gráficos e tabelas, lente “olho de peixe e vários outros efeitos e aplicativos para lidar com gráficos.

Exemplos:

Exemplo de *lightbox* produzido com a biblioteca Dojo, encontrado no *feature explorer* do site. O script produz um quadro sobre a página, onde pode ser carregado um carrossel de imagens. <http://dojocampus.org/explorer/>

Um exemplo de galeria de imagens produzida com base em *Dojo*, também encontrada no *feature explorer* do site: http://dojocampus.org/explorer/#Dojox_Image_Gallery_Default

Um pequeno aplicativo que amplia uma determinada imagem dinamicamente. <http://demos.dojotoolkit.org/demos/cropper/>

Utilizar um *framework* como um desses apresentados, às vezes não é uma tarefa muito simples para um leigo, e um programador certamente consegue se apoderar mais de uma desses recursos do que um designer. Mas existem, como pudemos mostrar aqui, uma série de recursos pré-programados que podem ser apoderados com mais facilidade. Seguindo a documentação oferecida por muitos deles, alguém que tenha a mínima intimidade com linguagens de marcação consegue certamente fazer uso de alguns desses recursos oferecidos.

4.6

Forma em construção: Sistemas de gerenciamento de conteúdo

Expusemos no capítulo 2.7 desta dissertação uma nova forma como a web vem se organizando nessa última década, chamada de Web 2.0. Explicamos também que ela se ampara na possibilidade de se contruir aplicativos executados no servidor, que interagem com bancos de dados para a criação de páginas dinâmicas. Independente de serem amparados em conteúdo criados pelos usuários, boa parte dos websites construídos hoje me dia, são amparados por ferramentas de gerenciamento de conteúdo dinâmicas, conhecidos como CMSs (*Content Management Systems*).

No modelo de web anterior ao surgimento dessas ferramentas, a cada atualização era necessário a execução ou alteração de arquivos (X)HTML. Mas boa parte dos proprietários ou funcionários das empresas que contratavam os serviços de um webdesigner eram leigos em matéria de linguagens de marcação. Isto acabava por deixá-los sempre nas mãos dos profissionais, o que poderia retardar o processo de atualização do website ou intranet, além de ampliar os custos de manutenção do site.¹ Com a popularização do uso de linguagens de *server-side scripting* e bancos de dados, bem como o desenvolvimento de uma série de sistemas de gerenciamento de conteúdo *open-source* ou com preços acessíveis, cresceu a demanda por websites dinâmicos, que podem ser atualizados de maneira mais fácil por leigos. Nas palavras de James Robertson:

Um sistema de gerenciamento de conteúdo (CMS) dá suporte à criação, gestão, distribuição, publicação e descoberta de informação corporativa. Ele cobre o ciclo completo das páginas do seu site, desde fornecer ferramentas simples para criar o conteúdo, até a publicação e, finalmente, seu arquivamento. Ele

¹ Robertson, James. So, What is a CMS? Step two Designs., 2003 http://www.steptwo.com.au/papers/kmc_what/index.html

também fornece a capacidade de gerenciar a estrutura do site, o aspecto das páginas publicadas, e a navegação provida aos usuários.²

Com esse tipo de recurso, a web abre as portas para novos níveis de produtores, que são aqueles que têm poder de adicionar, remover ou editar conteúdo, as não têm acesso à estrutura propriamente dita. Podemos dividir as categorias de indivíduos envolvidos em um determinado site em três: uma dos usuários que têm acesso aos arquivos hospedados no servidor, que são programadores, designers (em alguns casos) e administradores; outra são os consumidores e leitores que têm acesso somente à interface pública do site e uma terceira os produtores de conteúdo, que têm acesso às ferramentas de publicação e edição de conteúdo.

Uma série de tecnologias diferentes podem ser utilizadas para a criação desses sistemas, PHP, Asp, .net, JSP e *Coldfusion* são as mais comuns, segundo Goodman (2004)³. Ultimamente *Ruby* é uma linguagem que vem se destacando entre elas, mas outras linguagens como *Perl*, *Python*, *Java*, *C++*, *C#*, *Visual Basic*, e até *server-side JavaScript* pode ser usado para isso. A maioria dos CMSs utilizam uma — ou mais — linguagem de programação, executada no servidor, em combinação com alguma tecnologia de construção de bancos de dados. As tecnologias mais utilizadas para a construção de bancos de dados na web são MySQL e XML.

2 A content management system (CMS) supports the creation, management, distribution, publishing, and discovery of corporate information. It covers the complete lifecycle of the pages on your site, from providing simple tools to create the content, through to publishing, and finally to archiving. It also provides the ability to manage the structure of the site, the appearance of the published pages, and the navigation provided to the users. Robertson, op. cit.

3 Goodman, Danny e Morrison, Michael. *JavaScript Bible*. Indianapolis: Wiley Publishing, 2004

James Robertson⁴ divide a estrutura de um sistema de gerenciamento de conteúdo da seguinte maneira: criação de conteúdo, gerenciamento de conteúdo, publicação e apresentação. Para a criação de conteúdo, é necessário o desenvolvimento de uma interface gráfica, normalmente organizada na forma de formulários, aonde os produtores de conteúdo inserem os dados. Esta seção normalmente é protegida por senha, para que não ocorram invasões. Para o gerenciamento de conteúdo, é desejável construir uma estrutura que permita o armazenamento de versões antigas, evitando danos permanentes em caso de erros dos usuários, e pode ser desejado também um sistema de aprovação do conteúdo por uma autoridade, por exemplo. O sistema de publicação é o mecanismo que reconhece os inputs dos usuários e o transforma em código para a apresentação das páginas. CMSs complexos podem ter mecanismos sofisticados de publicação, que incluam, por exemplo, mecanismos anti-ataques e avaliação dos dados digitados, para transformação de texto em links, por exemplo. Por fim, o sistema de apresentação é o que constrói a navegação do site automaticamente a partir de cada nova publicação, e organiza os conteúdos para a navegação dos usuários.

A adaptação do projeto, normalmente incorre em transformações do layout, via CSS e algumas vezes a edição dos *templates* de página. As páginas geradas por um CMS são compostas de acordo com as informações recebidas do usuário, em junção com as informações recolhidas dos bancos de dados. Para a formatação da página, no entanto, existem modelos criados que inserem as variáveis (criadas pelas linguagens de programação ou bancos de dados) dentro de uma marcação (X)HTML. Cada modelo de página necessita de um *template* diferente. Páginas semelhantes,

4 Robertson, op. cit

embora possam possuir conteúdo diferentes, são geradas pelos mesmos templates.

Alguns dos CMSs mais comuns disponibilizam várias opções de “temas” para aplicação no site. Existem temas gratuitos, que podem ser baixados por qualquer um, e temas “profissionais” que podem ser pagos. Um tema, é composto por uma série de templates que alteram a aparência final do site, normalmente mantendo uma certa consistência de design entre as páginas. Temas são como cascas de aparência para a estrutura de programação de um determinado sistema. Alguns designers vivem hoje da execução de temas “profissionais” para determinados CMSs.

Para utilizar um CMS em um site não é necessário desenvolver toda a programação de *server-side scripting* e a construção dos bancos de dados; existem muitos CMS já desenvolvidos que podem ser utilizados. O site *Cmsmatrix*, que reúne informações sobre CMSs pré-desenvolvidos lista por exemplo, mais de 1000 sistemas prontos, dos quais pelo menos 300 são open-source. Mesmo para criar um tema novo, em alguns casos tudo o que o usuário precisa saber é HTML e CSS.

Essa possibilidade de customização faz com que esses sistemas sejam completamente adaptáveis. Entendendo as necessidades do cliente, um designer pode facilmente adotar e adaptar algum desses sistemas disponíveis para seu projeto. A seguir, iremos expor alguns dos CMSs open source mais importantes e utilizados no mundo hoje.

Wordpress

O *Wordpress*, plataforma *open-source* para construção de *blogs*, é um dos CMSs mais utilizados no mundo. Qualquer pessoa que disponha de um servidor de internet pode baixar e instalar o software. O *software* tem uma interface para administração de onde se pode acessar uma série

de funções, que vão, de criar um *post*, editar os usuários, adicionar *plugins*, escolher um tema para o *layout* ou mesmo editar os *templates* que originam as páginas.

Página inicial de administração do Wordpress, que serve de entrada para toda administração do sistema

Para criar o *post*, o Wordpress conta com um editor tipo WYSIWYG, que facilita a inserção de arquivos multimídia e demais conteúdos de maneira fácil, mesmo para leigos.

Interface para criação de novas *posts*

Do gerenciador de temas, pode-se escolher entre centenas de temas disponíveis, produzidos pela comunidade de usuários e colaboradores do sistema. Uma vez selecionado o tema, o wordpress aplica-o automaticamente à página.

Seleção de temas; existem centenas de temas disponíveis para pronta utilização dos usuários do sistema

A interface para edição do tema inclui um *widget* de onde se pode acessar o arquivo CSS, que pode ser editado online, sem a necessidade de se acessar os arquivos via FTP. Além do CSS, é possível também se acessar templates feitos em PHP de toda páginas públicas do site.

No wordpress, os temas podem ser editados online, sem a necessidade de acesso via Ftp ao servidor

No editor de plugins é possível se acesar o diretório de plugins do wordpress, que contém milhares de add-ons desenvolvidos pela comunidade para as mais variadas tarefas.

Media Wiki

Desenvolvido originalmente para abrigar a Wikipédia, a maior enciclopédia livre, o media wiki é um sistema que provém grande capacidade de edição e customização, é um software oferecido para ser instalado em servidores privados.

Como foi desenvolvido para abrigar projetos multilíngues, tem grande ênfase na internacionalização do conteúdo.

A wikipédia é a maior enciclopédia online do mundo, e mudou o paradigma das enciclopédias, ao utilizar os usuários na criação do conteúdo dos artigos

Têm um sistema especial que permite a fácil criação de novas páginas, o que favorece o

desenvolvimento do hipertexto nos seus conteúdos. Além disso oferece a possibilidade fácil de criação de usuários-editores, e mantém em cache um histórico das atualizações feitas, para evitar possíveis vandalismos.

Editando a página wiki. Existe uma notação especial para definir o que é um link, o que é um cabeçalho ou uma barra

Para a instalação, é necessário acesso ftp ao servidor, mas para customizar o layout, é possível alterar um arquivo CSS online.

Nestes exemplos, a página foi editada através da interface online acima, que permite a alteração do CSS.

O website acima, embora não retenha nada da aparência, é uma customização de um sistema mediawiki, desenvolvido pela Cooperativa de Tecnologia Laudelina em 2008 para o coletivo Eia, Experiência Imersiva Ambiental

Drupal

Drupal é um CMS open-source escrito em PHP, que pode construir uma série de tipos de websites, de sites comerciais a portais de notícias.

O sistema também oferece uma gama de temas, que podem ser customizados, mas diferente do wordpress, o CSS só pode ser acessível via FTP.

Drupal teste we got the tools

Home > Administer > Site building

Themes [List](#) [Configure](#)

Select which themes are available to your users and specify the default theme. To configure site-wide display settings, click the "configure" task above. Alternatively, to override these settings in a specific theme, click the "configure" link for that theme. Note that different themes may have different regions available for displaying content; for consistency in presentation, you may wish to enable only one theme.

To change the appearance of your site, a number of [contributed themes](#) are available.

Screenshot	Name	Version	Enabled	Default	Operations
	007 niGraphic Studio 007 niGraphic is light-weight, 2-column, fixed width and uses CSS.	6.x-1.x-dev	☐	☐	Configure
	A3 Atlantis Tableless, multi-column, fixed width theme.	6.x-1.2	☐	☐	Configure
	aBeesParadise Tableless, multi-column, fixed width theme.	6.x-1.x-dev	☐	☐	Configure
	Ability Multi-column, 1 color style, 12 fully collapsible regions, fixed or fluid width theme (default), easy configuration.	6.x-2.x-dev	☐	☐	Configure
	About Aboutpeople theme with left sidebar first.	6.x-1.1	☐	☐	Configure
	Aboutpeople A fresh 3-column layout for Drupal.	6.x-1.1	☐	☐	Configure
	AD Lemon Twist W3C valid one column theme.	6.x-2.1	☐	☐	Configure
	AD Redoable A clean elegant tableless valid theme.	6.x-1.3	☐	☐	Configure
	Agregado Agregado is a free professional Wordpress theme created by Diemen Hoek and Hert Dawson of Categorix . Converted to Drupal theme by Lacew and Kah Theng .	6.x-2.0-beta3	☐	☐	Configure
	alchemist Alchemist turns your website to gold.	6.x-6.x-dev	☐	☐	Configure
	Amor_Azul div-based three-column theme.	6.x-1.x-dev	☐	☐	Configure
	Barron XHTML & CSS based two column theme.	6.x-1.0	☐	☐	Configure
	Baroness Sub-theme for Barron - sidebar on the left.	6.x-1.0	☐	☐	Configure
	Bluemarine Table-based multi-column theme with a marine and ash color scheme.	6.14	☐	☐	Configure
	Celju XHTML & CSS based, two columns, multiple footer regions.	6.x-1.3	☐	☐	Configure
	Chameleon Minimalist tabled theme with light colors.	6.14	☐	☐	Configure
	Colourise Dark colorful design with valid XHTML/CSS. Visit Webster.net for more information.	6.x-2.1	☐	☐	Configure
	Cool web Dev Based Theme.	6.x-1.x-dev	☐	☐	Configure
	Coolwater for Drupal 6.x Coolwater	6.x-1.0	☐	☐	Configure
	Dark Elegance Simple & Elegant Dark 2 Column Theme.	6.x-1.1	☐	☐	Configure
	Decayed Dark tableless design with valid XHTML/CSS.	6.x-1.x-dev	☐	☐	Configure
	Deco Modern, tableless, multi-column, fluid width theme with art deco looks.	6.x-1.1	☐	☐	Configure
	Delicious Fruit Beautiful theme with greenery and grapes. Fully XHTML and CSS standards compliant with a pinkish and green color scheme.	6.x-2.1	☐	☐	Configure
	Drupazine Drupazine is a magazine style drupal theme, based on Framework theme and inspired in a famous wordpress magazine theme.	6.x-1.x-dev	☐	☐	Configure
	EXERED Table-based multi-column theme with a marine and ash color scheme.	6.x-1.2	☐	☐	Configure
	Foliage Table-less, source-ordered and multi-column theme. Maintained by FreeCmsDesigns.com	6.x-4.x-dev	☐	☐	Configure

Drupal teste we got the tools

Home > Create content

Create Page

Title:

Menu settings

Body: [Split summary at cursor](#)

A comunicação é uma função do código e do repertório que, por sua vez, estão na dependência da informação. Informação não é uma "coisa", mas uma relação estatística entre o que se conhece, entre o previsível e o imprevisível. O grau de imprevisibilidade dos sinais – da informação portanto – vincula-se à capacidade de manipulação do código. Uma grande capacidade de manipulação do código implica em um repertório mais redundante, mais previsível e ambíguo, menos preciso: quer-se informar mais do que os sinais comportam.

Input format

☒ **Filtered HTML**

- Web page addresses and e-mail addresses turn into links automatically.
- Allowed HTML tags: <a> <code> <div> <div> <div> <div> <div>
- Lines and paragraphs break automatically.

☐ **Full HTML**

- Web page addresses and e-mail addresses turn into links automatically.
- Lines and paragraphs break automatically.

[More information about formatting options](#)

Revision information

Comment settings

Authoring information

Authored by:

Leave blank for Anonymous.

Authored on:

Publishing options

☒ **Published**

☐ **Promoted to front page**

☐ **Sticky at top of lists**

[Save](#) [Preview](#)

Cada página criada é um nó específico, e pode ser colocada no menu principal ou lateral

Tem suporte para uso de folksonomias e para múltiplos usuários. E oferece uma ampla gama de plugins para criação de fóruns, blogs, inserção de conteúdo multimídia, calendários entre outros. Possui também módulos pré-programados para e-commerce.

O drupal é utilizado em sites como o da Agência Brasil (<http://www.agenciabrasil.gov.br/>) site do MST, movimento dos trabalhadores sem terra (<http://mst.org.br>)

Joomla

Outro CMS *open-source* bastante popular, oferece uma interface completa para edição de conteúdo, mas não para customização, que tem de ser feita via CSS por meio de FTP.

Interface para criação de conteúdo no Joomla

É extensível também através de módulos e add-ons, que podem ser encontrados no site <http://extensions.joomla.org/>

Página para download de extensões para o CMS

Não vem com muitas opções de templates pré-instalados, mas é possível de se encontrar uma grande variedade deles para download na internet.

Tem suporte para diferentes tipos de usuários, que podem ter permissões diferentes para alteração do site e edição de conteúdo.

O website da Pró-reitoria de Graduação da Universidade de São Paulo é construído a partir deste CMS.

Existem muitos outros CMSs disponíveis na rede, cada qual com uma estrutura particular de organização, mas não nos estenderemos aqui em muitos exemplos. Para instalá-los, pode ser preciso algum domínio adicional sobre criação de bancos de dados, que não cobriremos aqui neste trabalho. Com as ferramentas que apresentamos aqui, é possível se contruir uma ampla gama de websites, mas customizá-los para que tenham a aparência desejada, exige uma certa prática de código, especialmente CSS. Com o que apresentamos neste capítulo, esperamos ter dado um referencial básico para aquele que deseje se aventurar nesta tarefa.

4

Forma aparente e forma oculta

5.1 Designers e Programadores

Processo de trabalho

A web e a consequente explosão do design de interfaces trouxe novos desafios para o campo profissional do designer gráfico. À tradicional preocupação com a forma, é somada uma nova preocupação que, como Alan Cooper aponta, "as disciplinas tradicionais não exploram com frequência: o design do comportamento".

A emergência de produtos com comportamento complexo deu origem a uma nova disciplina. O Design de Interação pega emprestado teoria e técnica de disciplinas tradicionais do design, da e da engenharia. Mas é maior do que a soma de suas partes, com os seus métodos e práticas próprias. E para ser claro — é muito uma disciplina de design, bem diferente da ciência e da engenharia. Ao mesmo tempo que deve ser sempre praticado de forma racional e ponderada, o design de interação é de sobre síntese e imaginação das coisas como elas poderiam ser, não necessariamente como elas são atualmente.¹

Surge, diante desta necessidade, uma nova leva de profissionais, chamados por Cooper de Interaction designers, que seriam especializados em novas disciplinas como Arquitetura de informação, e design de interação. Em grandes projetos, é possível se ter times explorando cada um dos campos necessários para o desenvolvimento do projeto:

¹ This emergence of products with complex behavior has given rise to a new discipline. Interaction design borrows theory and technique from traditional design, usability, and engineering disciplines. But it is greater than a sum of its parts, with its own unique methods and practices. And to be clear — it is very much a design discipline, quite different from science and engineering. While it should always be practiced in a rational and considered manner, interaction design is about synthesis and imagining things as they might be, not necessarily as they currently are. Cooper, Alan; Reimann, Robert; Cronin, David. About Face 3. The essentials of Interaction Design. Indianapolis: Wiley Publishing, 2007 pg XXVII-XXVIII

uma equipe de designers, que se responsabilizaria pela satisfação dos usuários com o produto

uma equipe de engenharia, que se responsabiliza pela implementação e desenvolvimento do produto

uma equipe de marketing, que se responsabiliza pela divulgação e venda

e uma equipe de gerenciamento, que é responsável por tomar decisões que envolvam as demais equipes de trabalho

Mas nem todos projetos no entanto, têm orçamentos grandes o suficiente para abarcar toda a gama de profissionais envolvidas no projeto. Muitos dos trabalhos de web design são pequenos sites, para pequenas empresas ou profissionais liberais que querem contratar um profissional para ajudá-los a expor seu trabalho na grande feira mundial que é a web.

Um designer que queira sozinho desenvolver um website contava sempre com a possibilidade de utilização de softwares do tipo WYSIWYG (*What you see is what you get*), que permitem, mediante a uma interface gráfica, criar websites de maneira intuitiva, mais próxima da prática profissional tradicional para mídia impressa. Alguns exemplos desse tipo de software são o *Dreamweaver* e o *Flash*, da Adobe e o *Frontpage*, da Microsoft.

Interface do Dreamweaver

Na época em que o CSS ainda não era amplamente suportado pelos browsers, e a forma de se fazer layout de websites possível era a utilização de tabelas e frames, essa escolha podia fazer sentido. Para se construir uma página web, bastava um certo domínio sobre algum desses softwares:

O resultado disso foi que os sites podem muitas vezes ser pesados e lentos, normalmente só funcionavam corretamente em um navegador, eram complicados para atualizar e manter, e era necessária a duplicação do conteúdo para se criar versões amigáveis para impressão e motores de busca tinham dificuldade em indexá-los, encontrar sentido e classificá-los. Isto, por sua vez, levou a uma proliferação de métodos obscuros de otimização de pesquisa pesquisa, elementos <meta> estudados com palavras-chave e desenvolvimento de páginas específicas para motores de busca. Apresentação (a aparência) e comportamento (normalmente JavaScript) eram misturados ao conteúdo, as páginas e não tinha qualquer significado ou estrutura lógica, a preocupação do dia era como páginas iriam aparentar, e não o que significavam.²

A questão é que organizar a informação, na web, não é somente colocar as coisas nos seus devidos lugares, é estruturar a informação de forma semanticamente correta, e além disso, colocar as coisas nos seus devidos lugares. Um documento gerado automaticamente por um editor WYSIWYG dificilmente terá a melhor estrutura semântica, pois o computador, ao contrário do que muitas pessoas acreditam, não pensa, ele só

2 The result of this was that websites could often be heavy and slow, usually only worked properly in one browser, were complicated to update and maintain, required duplication of content for “print-friendly” versions, and search engines had a hard time indexing, making sense of, and ranking them. This, in turn, led to a proliferation of shady search-engine-optimization tricks, <meta> elements overstuffed with keywords, and per-search-engine entry pages. Presentation (the look and feel) and behavior (usually JavaScript) were both mixed in with content, and pages had no meaning or logical structure—the concern of the day was how pages looked, not what they meant.

processa informação, e não consegue, automaticamente, criar a solução mais simples, mais acessível e mais adequada, e pode gerar problemas de manutenção futura. Além disso, esses programas geravam código fora dos padrões das linguagens definidos pelo W3C.

Mas nesta época, nem os browsers seguiam os padrões das linguagens. O movimento em prol do suporte aos padrões da W3C conseguiu aos poucos, forçar os fabricantes de navegadores a adotar os padrões da linguagem CSS, e hoje, tirando pequenas incompatibilidades, o CSS pode ser empregado com sucesso na construção de websites.

Diante dessa realidade, muitos designers têm preferido abandonar as ferramentas de automação, e partir para o mundo do código.

WYTIWYG

O Designer, como organizador da informação acima de tudo pensa. Quando o designer ganha o domínio das linguagens necessárias para fazer o seu trabalho, ele consegue ter o que ele pensa, e não o que ele vê, por isso, introduzimos aqui esse conceito, que já vem sendo praticado por milhares de indivíduos que se libertaram das ferramentas de criação automática: *What you think is what you get*.

Aprender a utilizar essas linguagens demanda, sem dúvida uma dedicação maior do que aprender a utilizar um software, mas a liberdade ganha com o domínio da técnica pode compensar, por uma série de razões, como aponta Marc Gugliuzza:

Uma vez que você pode escrever o código a mão com confiança e proeficiência, criar um novo layout levará menos tempo do que seria se você usar um editor. Controlar detalhes

extremamente pequenos, e fazer o layout ficar exatamente correto se torna muito mais fácil.³

Além disso, a manutenção e a facilidade de relacionar o código com linguagens de server side scripting é amplificada; assim como a acessibilidade em geral.

Mas a parte mais interessante, é a possibilidade de se livrar de muletas visuais e pensar na estrutura da informação como elemento de construção do layout, e não tratar a interface gráfica somente como uma casca. Traduzir um layout em código exige um esforço de abstração e um pensar na informação. Quando se pensa em termos de código, é muito mais fácil se jogar fora elementos meramente decorativos e pensar na página como uma estrutura portante da informação.

Designers e programadores

Designers gráficos são, historicamente, profissionais com um grande desenvolvimento da habilidade da intuição, e um domínio da capacidade de expressão mediante a linguagem visual. Programadores por outro lado, são vistos como técnicos com alto domínio da capacidade de resolver problemas mediante o uso de linguagens de programação.

A primeira vista, estão situados em campos opostos da cadeia produtiva. Programadores podem considerar o trabalho do designer superfluo, e designers considerar o trabalho dos programadores um tanto místico. A web, de uma certa maneira, aproxima esses dois campos profissionais.

³ Once you can hand code confidently and proefficiently, creating a new layout will take less time than it would if you used an editor. Controlling extremely small details, and getting the layout to look just right become much easier. Gugliuzza, Marc. Debate - hand coding: the ultimate freedom. Sitepoint, 2002. Disponível em <<http://www.sitepoint.com/article/hand-coding-ultimate-freedom/>>

O campo da programação, mesmo se estreitarmos somente às tecnologias para web ainda é um campo muito vasto: construção de bancos de dados, administração de servidores e sistemas, domínio de linguagens de marcação, script e server-side scripting; conhecimento das principais bibliotecas, de sistemas de geração de conteúdo, segurança, shell scripting. É um campo muito grande de possibilidades de trabalho, que abre margem para uma gama de perfis profissionais possíveis; desde o especialista em uma determinada tecnologia, como o generalista que consegue trabalhar mais ou menos com uma grande gama de tecnologias.

Em qualquer projeto grande, que vá além de customizar a aparência de um sistema pronto, ou de um pequeno site, é necessário se contar com o trabalho de um programador dedicado à construção da infra-estrutura de processamento, e mesmo da tradução da interface em código. Neste caso, é uma prática muito comum no mercado de web design, terceirizar o trabalho, mandando-o para um programador especialista externo à equipe. O contrário, programadores terceirizando o trabalho de desenho do layout também ocorre, como aponta Jay Hilgert:

Empresas de design gráfico raramente contratam programadores e empresas de programação raramente contratam designers. Eles terceirizam vez de contratar um funcionário em tempo integral, cujas habilidades são usadas apenas ocasionalmente, porque a maioria de seus negócios é derivado de seus serviços essenciais. Isto é parte do problema. Designers e programadores trabalham em diferentes campos mais frequentemente do que não.⁴

4 Graphic design firms rarely hire programmers, and programming companies rarely hire designers. They would rather outsource than hire a full-time employee whose skills are only used on occasion, because most of their business is derived from their core services. This is part of the problem. Designers and Programmers

Da mesma forma que os designers não costumam a aprender programação em cursos de design, programadores também não recebem nenhuma formação específica de design em seus cursos. Jay Hilbert defende que colocar esses dois tipos de profissionais para trabalharem juntos, enriquece a capacidade profissional de ambos:

Designers e programadores devem estar sempre de olho em novas maneiras de ajudar uns aos outros. E se você for uma empresa pequena e crescente e tiver demanda, contratar alguém pode complementar sua equipe. Seu designers irão aprender 100 vezes mais informações provenientes de um programador a bordo do que escrevendo e-mails um “amigo” terceirizado. O mesmo vale para programadores. Tendo um designer gráfico a bordo poderia poupar milhares de terceirizações, e ensinar a sua equipe de informações inestimáveis sobre o layout na frente do código, e como os scripts manipulam o que o usuário vê. Interfaces de websites devem ser projetadas para a usabilidade e acessibilidade, o que envolve tanto o que você vê, quanto programação que dá poderes a isso. Faça um esforço para entender a sua disciplina de complementar.⁵

Designers e programadores devem cada vez mais se aproximar, tendo cada mais um domínio maior sobre os campos uns dos outros. Mesmo

work in different arenas more often than not. Hilgert, Jay. Webdesign: The void between designer and programmer. Bittbox, 2007. disponível em <<http://www.bittbox.com/rants/web-design-the-void-between-designer-and-programmer>>

5 Designers and programmers should be constantly looking into new ways to help each other out. And if you're a small and growing company and have the demand, hire someone to compliment your team. Your designers will learn 100 times more information from an on-board programmer than emailing an outsourced “friend.” The same goes for programmers. Having one Graphic Designer on board could save you thousands in outsourcing, and teach your team priceless information about the layout in front of the code, and how the scripts manipulate what the user sees. Website interfaces should be designed for usability and accessibility, and that involves both what you see, and the programming that powers it. Make an effort to understand your complimentary discipline. Ibid.

que um designer não seja capaz de desenvolver toda a estrutura de programação de um projeto complexo, ele deve ser capaz de saber o que é possível de se fazer, e como criar layouts cuja implementação seja mais fácil.

Mas tanto programadores com habilidades na linguagem visual, como designers que tenham domínio maior das linguagens de programação, conseguem criar experiências que extrapolam os limites dos dois campos de conhecimento. A academia, como aponta Jon Maeda, no prefácio do livro *Processing*, tradicionalmente forma dois tipos de estudantes: estudantes de arte e estudantes de ciências, "mas nunca mistura os dois lados da equação na mesma pessoa". Agora, abre-se um campo na arte e na ciência para esses seres "híbridos" que trafegam por esses campos agora convergentes. Dessa abertura, surge uma nova produção artística e experimental.

5.2

Produção, Consumo e Produçssumo

Produção

Desde a criação do primeiro browser gráfico, o Mosaic em 1993, alguns artistas e coletivos passaram a explorar a plataforma da web como um espaço para expressão artística, amparada por um crescente surgimento de fóruns e listas de discussão, como aponta Jon Ippolito:

Em 1995, oito por cento de todos os sites eram produzidos por artistas, dando-lhes uma oportunidade sem precedentes para moldar um novo meio em seu princípio. Desde aquela época, a arte na internet tem gerado inúmeras discussões críticas em listas de e-mail como the Thing, Nettime, 7-11, e Rhizome.org. Incentivado por um entusiasmo crescente com a Internet como um fenômeno social e econômico, proliferam artigos e exposições do museu que passaram a trazer a arte online para a vanguarda do debate sobre o futuro da arte no século 21.¹

A internet (não somente a web) amplifica não só as possibilidades de lidar com linguagem visual, mas, como já dissemos, muda radicalmente o acesso aos meios de difusão das obras de arte, o artista pode se destacar do meio dos museus, galerias e curadores, e consegue, via rede, o acesso potencial indiscriminado a um público numa escala mundial, como apontam os autores de *Interaction: Artististic Practice in the Network*:

artistas nunca foram realmente capazes de acessar os espaços de transmissão como um meio. As possibilidades de intervenção em transmissões foram simplesmente

1 By 1995, eight percent of all Web sites were produced by artists, giving them an unprecedented opportunity to shape a new medium at its very inception. Since that time, art on the Internet has spawned countless critical discussions on e-mail-based communities such as the Thing, Nettime, 7-11, and Rhizome.org. Encouraged by a growing excitement over the Internet as a social and economic phenomenon, proliferating news articles and museum exhibitions have brought online art to the forefront of the discussion on art's future in the 21st century. (Ippolito, Jon. *Ten Myths of Internet Art. Digital art of our time, vectors*. Disponível em: <<http://www.nydigitalsalon.org/10/essay.php?essay=6>>)

incorporados como “videoarte” e seguramente enclausuradas dentro do museu. Terá a “net art” o mesmo destino? Com a internet, nós temos o veículo para a produção e distribuição artística a uma escala nunca antes conhecida. Sua relação com o sistema de museus/galerias não é tão urgente como a necessidade do desenvolvimento contínuo de Plataformas inclusivas para se tirar o máximo de proveito das redes — especialmente porque, com a consolidação da mídia e comercialização desenfreada, essas plataformas estão ameaçadas.²

A permeabilidade dos códigos HTML, que podem ser vistos por qualquer um, traz uma facilidade a mais para que artistas se apropriem do código como matéria prima, independente de grandes recursos e grandes desenvolvimentos científicos. Frequentemente nesse tipo de arte, o código não é só o suporte mas também o objeto da obra de arte. O coletivo Jodi, por exemplo, é um deles, que explora a estética do código e dos bugs de computador como referência para seus trabalhos online.

2 “artists were never really able to access broadcast space as medium. The possibilities of broadcast intervention were simply incorporated as “videoart” and safely enconced within the museum. Will “net art” meet the same fate? With the internet, we have a vehicle of artistic production and distribution on a scale we have never known before. Its relationship to the museum/gallery system is not as urgent as the need for the continual development of inclusive platforms that take full advantage of the networks - especially since, with media consolidation and rampant commercialization, those platforms are endangered. Johnson, John; Crandall, Jordan; Scholder, Amy. Interaction: Artistic Practice in the Network. Eyebeam Atelier, 2001 pg 3.

Screenshots do projeto <http://text.jodi.org/> do coletivo Jodi. Experimentações com a estética dos bugs de computador e do código

Outro tipo de experimentações são desenvolvidas pelo programador Yugo Nakamura no site Yugop.com. Ele trabalha com actionscript, levando ao extremo as possibilidades de interação que o software Flash proporciona, trabalhando com objetos em movimento e com simulações de gravidade e de movimento elástico.

screenshot da experiência jampack. Yougop.com

Numa tendência que pode ser vista entre vários artistas relacionados no banco de dados Rhizome.org, de trabalhos baseados em APIs de outros sites, ele elaborou o Amaztype, que produz tipografia a partir de capas de livros, cds ou dvds do catálogo da Amazon.

Numa outra ponta da cadeia de produção, estão os inventores e autores de aplicativos *online*, que criaram produtos de consumo de massa, como as comunidades virtuais e novas formas de comunicação e circulação de idéias. Os que conseguiram sucesso, transformaram seus empreendimentos em empresas valiosas, como é o caso dos fundadores do Youtube, Chad Hurley, Steve Chen e Jawed Karim, ou os próprios fundadores da Google, Larry Page e Sergey Brin, que fundaram a empresa enquanto ainda eram estudantes.

Plataformas como *Twitter*, *Myspace* e *Facebook*, angariam milhões de usuários e revolucionaram os modos de comunicação da internet, dando abertura para a realização plena do modo de comunicação muitos a muitos, ao permitir que usuários sem grandes conhecimentos técnicos se organizem em rede, publiquem conteúdo e acessem conteúdos publicados por outros.

Consumo e Produçsumo

Linguagem, comunicação e vida — eis o que há de novo, além do meramente estético-artístico. Nada de impingir à massa o que chamamos cultura. Mesmo porque a massa e os

meios de comunicação de massas é que estão destruindo essa cultura pré-selecionada. Quantidades à massa. Para que ela desenvolva a sua capacidade de opção e seleção — a sua capacidade de criar. Acaso e escolha. Chance & Choice. Invenção. Na linguagem, como na vida, como na máquina.³

Boa parte dos uso da web é somente passivo. Assistir um vídeo, ler e-mails, ler notícias. Está, de uma certa forma na esfera do consumo. Contudo, com a web dinâmica, o consumo se aproxima cada vez mais da produção. É um resultado claro da automação, como McLuhan já notava:

A automação não afeta somente a produção mas também o consumo e o mercado; pois, no circuito da automação, o consumidor se transforma em produtor.⁴

Podemos identificar dentro dos usuários da rede, uma gama de níveis de consumo e produçsumo, a partir dos níveis de uso:

- navegação passiva (esfera do consumo)
- retransmissão de conteúdos alheios (reprodução)
- produção de conteúdo (produçsumo)
- produção de pequenos aplicativos que funcionam dentro dessas redes (produção)

O *Facebook*, por exemplo, uma das maiores redes sociais em funcionamento hoje, permite que seus usuários criem pequenos aplicativos, como jogos, enquetes ou geradores de citação. Milhares de aplicativos foram criados nesse formato, e são utilizados por milhões em todo o mundo. Muitos aplicativos *online* têm interfaces para permitir a criação de subprodutos que rodem a partir dos seus sistemas.

3 Pignatari, Decio. Contracomunicação, pg 26

4 McLuhan. Os meios de comunicação, pg 392

A *Wikipédia* é também um grande exemplo de produssumo na rede, seus mais de onze milhões de artigos, escritos em mais de 240 línguas são todos criados pela comunidade de usuários, que adiciona, completa e edita seus conteúdos. Mais de 50.000 colaboradores⁵ ajudaram a criar este conteúdo. O sistema da wikipédia é amparado por um complexo sistema de hipertexto, de modo que a maioria dos artigos são inter-relacionados uns com os outros. De início, haviam dúvidas se uma enciclopédia criada pelos usuários poderia ter credibilidade, mas hoje, sua consolidação é inegável, e a maioria dos artigos escritos é amparado em fontes sólidas.

Outro grande expoente do produssumo na rede, é sem dúvida, o *Youtube*. Embora muito do seu conteúdo seja reprodução de conteúdos de meios anteriores, como a televisão, o cinema e mesmo o rádio, existe uma quantidade imensa de vídeos produzidos amadoristicamente com divulgação no site. São paródias, vídeos engraçados, coberturas de eventos, vídeos de auto-promoção ou clipes, de produtores independentes que conseguem agora uma cesso a um meio global de difusão de conteúdo. O *Youtube*, vem propiciando o surgimento de novos gêneros de filme, como aponta Virginia Heffernan, do New York Times:

Mas o que é surpreendente é quão pouco os vídeos caseiros lembram os profissionais. Claro, existem paródias de clipes mainstream aqui e ali, mas principalmente os amadores estão saindo por conta própria, incubando novos gêneros. Considere vídeos de “compra”, nos quais as pessoas mostram coisas que compraram recentemente, ou os populares “fail” vídeos, que mostram toda a espécie de esforços dando errado. Vídeos individuais desses tipos muitas vezes chegam a atrair

100.000 exibições ou mais — e ninguém tinha sequer imaginado gêneros como esses até recentemente. Ao mesmo tempo, ninguém em nenhuma empresa de produção parece estar lutando para atender as audiências desses tipos de novos gêneros (ou combinando-os?). E as pessoas da “compra” e do “fail”, evidentemente, não se sentem mal-servidas: eles estão ajudando a si-propios e criando o que só pode ser chamado de uma cena de arte, toda em torno de muitos, muitos vídeos de seus gêneros no YouTube.⁶

É como ela chama, uma nova “vanguarda vernacular”, que cria novas formas de “expressão criativa”. O aplicativo permite ainda que os usuários criem canais próprios, em páginas que podem ser visualmente customizadas, se adequando aos desejos pessoais de cada um.

6 But what’s surprising is how little the homemade videos resemble the pro goods. Sure, there are parodies of mainstream clips here and there, but mostly the amateurs are off on their own, hatching new genres. Consider “haul” videos, in which people show off the stuff they recently bought, or the popular “fail” videos, which show all manner of efforts gone wrong. Individual haul and fail videos often attract 100,000 views or more — and no one had even imagined such genres until recently. At the same time, no one at any production company seems to be struggling to serve the haul-fail audiences (or combine them?). And the haul people and fail people evidently don’t feel underserved: they are helping themselves and creating what can only be called an art scene, all around the many, many videos of their genre on YouTube. (Heffernan, Virginia. Uploading the Avant-garde. New York Times, Magazine, setembro de 2009. Disponível em: <http://www.nytimes.com/2009/09/06/magazine/06FOB-medium-t.html?_r=1>

5 BENKLER, Yochai - The Wealth of Networks: How Social Production Transforms Markets and Freedom. Disponível em: <www.benkler.org>

Um princípio que foi largamente empregado pela comunidade virtual *Myspace*, voltada para a divulgação de músicos. O Myspace permite que seus usuários criem código CSS que pode ser inserido diretamente na sua página pessoal, dando a eles um ampla escolha da forma com que seus conteúdos específicos são mostrados. Usuários comuns muitas vezes se apoiam em gráficos prontos, como gifs animados, textos com *glitter*, e imagens engraçadinhas, disponíveis aos montes pela rede, prontas para serem copiadas e coladas nos seus comentários, nos seus perfis ou para serem utilizadas como fundo. Essa facilidade, faz, segundo Olia Lialina, que os usuários "se sintam em casa":

Pode parecer paradoxal, mas, incentivando o usuário a "se sentir em casa" esses serviços criam mais distância entre os usuários e a si próprios. Simplistas, gráficos bobos, gadgets sem sentido, páginas personalizadas com cachorrinhos e gatinhos virtuais do dia, amontoados com notícias da CNN e pedaços de sabedoria de Oprah - tudo isso sutilmente serve para mostrar ao usuário o seu espaço próprio.⁷

Ou, como já preconizava Décio Pignatari:

A tecnologia chega a um tal ponto de requinte que passa a requerer o marco zero de uma NOVA BARBÁRIE para desobstruir os poros. Sociedade cada vez mais rica, vida cada vez mais pobre. O dinheiro é a leucemia. Os modelos de consumo de hoje são os modelos de produção de quarenta anos atrás: vide Oswald de Andrade e o tropicalismo do grupo baiano. É tempo de PRODUSUMO. (...) O mundo do consumo substituído pelo mundo da informação, onde se travarão as grandes lutas.

⁷ It might sound paradoxical, but by encouraging the user to "feel at home" services create more distance between the users and themselves. Simplistic, silly graphics, senseless gadgets, customized pages with virtual puppies and kittens of the day heaped together with CNN news and bites of wisdom from Oprah – all of that subtly serves to show the user his proper place.

NOVA BARBÁRIE: campo aberto para os novos modelos da batalha informacional.⁸ (pg 31-32)

Gráficos com "glitter" como esse, são segundo Olia Lialina, símbolo da nova web "vernacular" <http://www.glittergraphics-now.com/sexy/page/2/>

A cultura de massas já esteve bastante consolidada com a televisão, mas nunca, a massa teve acesso aos meios de *broadcast*. Hoje, vivemos um estágio inicial do desenvolvimento de um novo tipo de cultura de massa, que é o momento onde as massas tomam os meios de produção e de difusão. Boa parte desta produção, é somente reprodução de outros sistemas sociais, como criação de video clipes alternativos, ou auto propaganda, mas este estágio inicial pode ter outros desdobramentos futuros. Pode ser ponta de lança para uma perspectiva social maior, ou não. Quem viver verá.

⁸ Pignatari. Op. cit pg 31-32

6 Conclusões

Uma estética em escala global

Procuramos, ao longo deste trabalho, expor a web sob vários aspectos. É claro, que pela sua dimensão é impossível abordar todos os aspectos detalhadamente. Mas conseguimos, aqui, dar uma introdução básica para aqueles que desejam participar da construção da rede de uma maneira concreta, para além do uso dos novos aplicativos online.

O código, que expusemos aqui em parte no capítulo 4, é a forma unificadora dessa rede. Tudo que circula nela é mediado por ele. Até a invenção da web, em todas as esferas da produção dos sistemas de comunicação, sempre houveram diferenças culturais e tecnológicas nos meios de produção, quer por acesso a matérias primas e técnicas, quer pela diferença entre níveis de desenvolvimento dos países e regiões.

A web, pela primeira vez, cria um meio homogêneo, gerido por uma tecnologia aberta e de acesso comum a qualquer indivíduo. É claro que existem diferenças entre velocidades de rede e acesso universal, mas quanto à produção, não existem diferenças econômicas regionais para limitar o desenvolvimento e produção de tecnologia e produtos.

Sabemos que a forma aparente dos meios de comunicação tem de uma certa maneira, uma relação direta com as técnicas disponíveis para sua execução. Com a web, as técnicas disponíveis são as mesmas independente de qualquer constrição física. A necessidade básica, é um computador, um editor de texto e o domínio das linguagens de marcação e programação.

Não por acaso, podemos notar que uma estética em escala global, aonde não se notam diferenças de estilo regionais, vem se configurando na web. É claro que esta é uma hipótese que mereceria

um estudo dedicado, mas de uma maneira geral, quando se navega na web, o espaço informacional é um espaço único, só dividido pelas línguas faladas em cada região.

É claro que existem diferenças de estilo; existe o que se chama de web-design vernacular, existe uma estética comercial, existe um novo minimalismo, o kitch e uma série de possibilidades estilísticas diferentes, mas elas não estão, em geral, associadas a uma determinada cultura, a uma determinada identidade nacional.

Abre-se assim, também a possibilidade de trabalhar num mercado de mão de obra em escala global. Em uma série de websites, trabalhadores de nações tão distante quanto o Paquistão e o Brasil competem por trabalhos relacionados à web. O site rentacoder.com é um exemplo disso. lá, frequentemente, americanos (ou de qualquer país) contratam designers e programadores à valores muito inferiores praticados no seu mercado de trabalho.

Se a diferença de tecnologia necessária não ocorre entre as diferenças nações, o mesmo não se pode dizer da capacidade técnica da mão-de-obra. A diferença entre a produção digital entre os países é dada pela diferença de acesso à internet e pela diferença da formação de mão-de-obra qualificada e de pesquisa na área. Diferenças de substrato, de massa crítica no tecido de invenção, para poder superar a rigidez dos sistemas de educação tradicionais.

Um novo campo disciplinar

Vimos no primeiro capítulo, como o desenvolvimento das tecnologias em favor da ampliação do intelecto humano pode transformar o computador de uma simples ferramenta, cujo uso era especificamente principalmente militar, para o mais completo instrumento de ampliação do

intelecto, humano, numa escala geral, e como essa transformação abriu espaço para a disseminação da web.

A web, tendo se tornado o mais importante meio de comunicação em escala global, abre um leque de novos campos para estudo e pesquisa. Webdesign, como disciplina, já é incluído em uma série de cursos, tanto de design, como de computação

Muitas “escolas de tecnologia da informação” e departamentos de Informática oferecem cursos que se focam em aplicativos para web ou em tópicos como “Web 2.0”, mas os protocolos, arquiteturas e princípios subjacentes à web, por si só raramente são cobertos.¹

No entanto, sua complexidade e relevância econômica, faz com que alguns, como o próprio Tim-Berners Lee e James Hendler, entre outros, sugiram a necessidade da criação de um campo interdisciplinar novo específico para o estudo da web, que procure pensar na web como um todo, mesclando aspectos da ciência da física, da ciência da computação, e com abertura também para ciências sociais, para pensar a web como um fenômeno que deve ser compreendido mas também como algo que deve ser projetado para crescimento futuro e adição de novas capacidades.²

É importante um campo de estudo que pense na web como parte de um sistema social de comunicação humana, e que compreenda como a web vem afetando a sociedade e transformando o modo como que os indivíduos produzem e

1 Many “information schools” and informatics departments” offer courses that focus on applications on the web or on topics such as “Web 2.0,” but the protocols, architectures, and underlying principles of the web per se are rarely covered. Hendler, James; Shadbolt, Nigel; Hall, Wendy, Berners-Lee, Tim. Web Science: An interdisciplinary Approach to Understanding the Web. Communications of the ACM. no. 7 Julho 2008)

2 Hendler, op cit.

distribuem conhecimento. A figura abaixo, extraída do artigo de Hendler "Web Science: An interdisciplinary Approach to Understanding the Web" mostra um novo modo de observar o desenvolvimento web:

Um aplicativo é desenhado baseado em alguma tecnologia, sob certas circunstâncias sociais. Micro propriedades são testadas com um grupo de usuários. Quando mais e mais pessoas vão adotando um micro-sistema, acontece uma escalada, e o sistema se torna um macro-sistema, como por exemplo o que ocorreu com o Youtube. A nova escala, requer novos desafios, que vai gerar a próxima geração de tecnologia. (Hendler) Para pensar nas possíveis consequências numa macro escala, é necessário ter conhecimentos que vão além da ciência da computação, que ele cita:

Dada a amplitude da Web e sua natureza (social) inerentemente multi-usuário, sua ciência é necessariamente interdisciplinar, envolvendo pelo menos matemática, ciência da computação, inteligência artificial, sociologia, psicologia, biologia e economia.³

Apesar de não estar incluído explicitamente nesta lista, o design gráfico também pode fazer parte desse conjunto de disciplinas, uma vez que a qualidade da interface, e a forma como ela aparenta também pode gerar questões de uso de um modo mais geral.

Neste trabalho, procuramos apresentar um panorama geral da web, sob dois de seus aspectos, bem como sua formação histórica. Uma série de questões foram apresentadas, mas muitas, com certeza ficaram de fora. Como já havíamos declarado na introdução, é impossível de se abarcar

todos aspectos de algo com a dimensão da web. Procuramos aqui, por isso, fazer um recorte mais específico sobre alguns aspectos que seriam mais importantes para um designer que queria dar um passo além na compreensão do código, ou para um programador que queria saber algo mais de design.

Como salientamos no início, procuramos dar uma panorama geral, embora não muito profundo, da web. Algumas questões ficam para possíveis estudos futuros, como as consequências sociais e estéticas deste desenvolvimento, mas esperamos com este trabalho, de alguma maneira, colaborar no desenvolvimento desse novo campo de conhecimento que se configura, a ciência da web, do ponto de vista de onde estamos inseridos no presente, que é o campo de pesquisa em Design e Arquitetura, da Faculdade de Urbanismo.

Esperamos fornecer um material de referência que pode ser utilizado como base em futuros cursos de *webdesign* e colaborando para a difusão desse conhecimento, que é um assunto de grande interesse entre os jovens designers e assim, ajudar com mais um passo na inserção desse campo do saber na Universidade.

³ Given the breadth of the Web and its inherently multi-user (social) nature, its science is necessarily interdisciplinary, involving at least mathematics, CS, artificial intelligence, sociology, psychology, biology, and economics.

7. Bibliografia

- ANDREESSEN, Marc. Innovators of the net: Brendan Eich and JavaScript. Techvision. 1998. Disponível em <http://web.archive.org/web/20080208124612/http://wp.netscape.com/comprod/columns/techvision/innovators_be.html>
- ANDREW, Rachel; SHAFER, Dan. Html Utopia: Designing without tables using CSS. Estados Unidos: Sitepoint, 2006
- ARNEHEIM, Rudolf - Arte e percepção visual: psicologia da visão criadora. São Paulo, EDUSP. 1078
- ARNSTON, Amy. Graphic Design Basics. Belmont: Thomson Wadsworth, 2007
- ADG. O valor do Design. São Paulo: SENAC, 2004
- BARBROOK, Richard. Futuros Imaginários, das máquinas pensantes à aldeia global. São Paulo: Peirópolis, 2009
- BENJAMIN, Walter - Obras escolhidas : Magia e técnica, arte e política: ensaios sobre literatura e história da cultura. São Paulo : Brasiliense, 1996
- BENKLER, Yochai - The Wealth of Networks: How Social Production Transforms Markets and Freedom. Disponível em: <www.benkler.org>
- BERNERS-LEE, Tim – Information Management: A Proposal. Cern, 1989. Disponível em: <<http://www.w3.org/History/1989/proposal.html>>
- _____. World Wide Web: Proposal for a Hypertext Project. 1990. Disponível em: <<http://www.w3.org/Proposal>>
- _____. – Design issues, architectural and philosophical points. (vers 12 April 2005).

Disponível em: <<http://www.w3.org/DesignIssues/>>

_____. - The future of the Web - LCS 35th anniversary talk transcript. talk to the LCS 35th Anniversary celebrations, Cambridge Massachusetts, 1999/ April/14. Disponível em: <<http://www.w3.org/1999/04/13-tbl.html>>

_____. - Realising the Full Potential of the Web. Based on a talk presented at the W3C meeting, London, 1997/12/3. Disponível em: <<http://www.w3.org/1998/02/Potential.html>>

_____. - The web, past, present and future. 1996. Disponível em: <<http://www.w3.org/People/Berners-Lee/>>

_____. - Hypertext and our collective Destiny. 1995. Disponível em: <<http://www.w3.org/People/Berners-Lee/>>

_____. Weaving the Web. São Francisco: Harper Collins, 1999

_____. Ten Years Public Domain for the Original Web, 2003 Software. Disponível em: <<http://tenyears-www.web.cern.ch/tenyears-www/Welcome.html>>

_____. Web History and Web Science. IBM, London, UK, March, 2008

BERNERS-LEE, Tim e SHADBOT, Nigel. Web Science Emerges. In Scientific American, October 2008

BONSIEPE, Gui. Design: do material ao digital. Florianópolis: FIESC/IEL, 1997

BONSIEPE, Gui - A tecnologia da tecnologia - São Paulo, Edgard Blucher - 1983

BRINZAREA, B.; DARIE, C.; CHERCHES-TOSA, F.; BUCICA, M. Ajax and PHP

- Building Responsive Web Applications. Mumbai: Packt Publishing, 2006

BROCKMANN, Josef-Müller. Grid Systems in graphic design. Nigli, 2001

BUNCH, Bryan e HELLEMANS, Alexander. The Timetables of Technology : A Chronology of the Most Important People and events in the History of Technology. New York: Touchstone, 1993

BUSH, Vannevar. As we may think. in The Atlantic, Julho 1945. Disponível em: <<http://www.theatlantic.com/doc/194507/bush>>

CAILLIAU, Robert. A little history of the World Wide Web. 1995 Disponível em: <<http://www.w3.org/History.html>>

CASTELLS, Manuel. A sociedade em rede - volume I. tradução Roneide Venancio Majer. 6a. ed. São Paulo: Paz e Terra, 1999

_____. A Galáxia da Internet: Reflexões sobre a Internet, os negócios e a sociedade. Trad. Maria Luiza X. de A. Borges. Rio de Janeiro: Jorge Zahar, 2003

CARPENTER, Edmund. As Novas Linguagens in Carpenter, Edmund e McLuhan, Marshall. Revolução na Comunicação. Rio de Janeiro: Zahar Editores, 1968.

Color Usage Research Lab. Using Color in Information Display Graphics - Design Methods, Color Science, and Color Guidelines. Nasa Ames Research Center.

COOPER, Alan; Reimann, Robert, Cronin, David. About Face 3. The essentials of Interaction Design. Indianapolis: Wiley Publishing, 2007

- Computers: bits & bytes. Disponível em: <
<http://www.scribd.com/doc/19955/Computers-bits-bytes>>
- DONDIS, Donis A. Sintaxe da Linguagem visual. São Paulo: Martins Fontes, 2007
- ECO, Umberto - A Estrutura Ausente. São Paulo, Perspectiva, ed. USP, 1972
- _____. - As formas do conteúdo. São Paulo: Perspectiva, 1973
- _____. - Obra Aberta. São Paulo, Perspectiva, 1968
- ENGELBARTH, Douglas. C. Augmenting Human Intellect: A Conceptual Framework. AFSOR-3233, Summary Report Menlo Park: Stanford Research Institute, 1962. Disponível em: <http://www.invisiblerevolution.net/engelbart/full_62_paper_augm_hum_int.html>
- ERLHOFFM, Michael e MARSHALL, Tim; eds. Design Dictionary. Basel: Birkhauser Verlag AG, 2008
- Eye Tracking Study Shows Sponsored Ads Attract Social Media Searches. SearchEngineWatch.com. Julho de 2009. Disponível em: <<http://blog.searchenginewatch.com/090713-134945>>
- FITTING, Melvin - Computability theory, semantics, and logic programming. New York; Oxford : Oxford University Press: Clarendon Press, 1987
- FRIEDMAN, Daniel P. - Essentials of programming languages. Cambridge, Mass.; New York : MIT Press: McGraw-Hill, c1992
- GARRET, Jesse James. Ajax, A new Approach to Web Aplications. Adaptive Path, february 2005. Disponível em: <<http://www.adaptivepath.com/ideas/essays/archives/000385.php>>
- GALBRAITH, James K e HALE, Travis. Income Distribution and the Information Technology Bubble. University of Texas Inequality Project Working Paper, 2004
- GALITZ, Wilbert. The Essential Guide to user interface Design. An introduction to GUI principles and Techniques. Indianapolis: Willey, 2007
- GEHTLAND, Justin; Galbraith, Ben; Almaer, Dion. Pragmatic Ajax: a web 2.0 Primer. Raligh: Pragmatic Programmers, 2006
- GLEICK, James. The information Future: Out of Control (And its a good thing too.) . New York times, 1 Maio 1994. Disponível em: <<http://www.nytimes.com/1994/05/01/magazine/the-information-future-out-of-control-and-is-a-good-thing-too.html>>
- GOODMAN, Danny e Morrison, Michael. JavaScript Bible. Indianapolis: Wiley Publishi, 2004
- GORDON, Bob & Maggie. The Complete guide to digital graphic design. London: Thames & Hudson, 2005
- GORE, Albert Jr. Information Superhighways: The next information revoution. The futurist, 1991. Disponível em: <<http://www.allbusiness.com/technology/internet-technology/150025-1.html>>
- GOIN, Linda. Design fo web developers: Colour and layout for the artistically overwhelmed. PA Enchede: DMXzone, 2005
- GRAHAM, Paul. What Bussiness Can Learn From Open Source. Paul Graham. paulgraham.com: 2005. Disponível em: <<http://www.paulgraham.com/opensource.html>>

- GRANNELL, Graig. Exploding the myths of web design. .net magazine. www.netmag.co.uk, 2009. Disponível em: <<http://www.netmag.co.uk/zine/discover-culture/exploding-the-myths-of-web-design>>
- GREENE, Rachel "Web Work: A HISTORY OF INTERNET ART". ArtForum. FindArticles.com. 16 Feb, 2010. Disponível em <http://findarticles.com/p/articles/mi_m0268/is_9_38/ai_65649375/>
- GROMOV, Gregory. The Roads and Crossroads of Internet History. Disponível em <http://www.netvalley.com/cgi-bin/intval/net_history.pl?chapter=1>
- GUGLUIZZA, Marc. Debate - hand coding: the ultimate freedom. Sitepoint, 2002. Disponível em: <<http://www.sitepoint.com/article/hand-coding-ultimate-freedom/>>
- HADLOCK, Kris. Ajax for web applications Developers. Sams, 2006
- HAINE, Paul. HTML Mastery. Semantics, Standards and Styling. Berkeley: Friendsoft, 2006
- HARRIS, Andy. PHP/MySQL Programming for the absolute beginner. Boston: Premier Press, 2003
- HARRIS, Daniel, The Best faces for the screen. Typofile Magazine. Disponível em: <<http://www.will-harris.com/typoscrn.htm>>
- HAUBEN, Ronda and Michael. The Netizens and the Wonderful World of the Net. Columbia, 2004. Disponível em: <<http://www.columbia.edu/~hauben/netbook/>>
- HEGEL, Georg Wilhelm Friedrich - Enciclopédia das ciências filosóficas em compendio (1830) v.1. A ciência da lógica. São Paulo: Loyola, 1995
- HENDLER, James; SHADBOLT, Nigel; HALL, Wendy, BERNERS-LEE, Tim; Weitzner, Daniel. WebScience: An Interdisciplinary approach to understanding the web. Communications of the ACM, vol 51 no. 7, julho de 2008.
- HEINDERSON, Harry. Encyclopedia of Computer Science and Technology. New York: Facts on File, 2003
- HENDLER, James; Shadbolt, Nigel; Hall, Wendy, Berners-Lee, Tim. Web Science: An interdisciplinary Approach to Understanding the Web. Communications of the ACM. no. 7 Julho 2008
- HIRATSUKA, Tei Peixoto. Contribuições da Ergonomia e do Design na Concepção de Interfaces Multimídia. Dissertação submetida à Universidade Federal de Santa Catarina. Florianópolis, 1996
- HILGERT, Jay. Webdesign: The void between designer and programmer. Bittbox, 2007. Disponível em <<http://www.bittbox.com/rants/web-design-the-void-between-designer-and-programmer>>
- How Much Information? - Internet - Summary, 2000. Disponível em: <<http://www2.sims.berkeley.edu/research/projects/how-much-info/internet.html>>
- HOWE, Walt. Histories of the Internet - A Brief History of the Internet and Related Networks, 1998. Disponível em <<http://www.isoc.org/internet/history/cerf.shtml>>

- IAB. June 2008: Us Retail e-commerce. Industry Stats & Data by eMarketer. Iab, 2008.
Disponível em: <http://www.iab.net/insights_research/530422/1675/334589>
- Introduction to Digital Logic Circuits.
Disponível em: <<http://www.scribd.com/doc/6701127/>
Introduction-to-Digital-Logic-Circuits>
- Internet World Stats - Usage and Population Statistics. Internet Lybrary, 2009.
Disponível em: <<http://internetworldstats.com/stats10.htm>>
- Internetworking Technology Handbook.
Internet Protocols. Disponível em:
<<http://www.ciscosystems.ch/en/US/docs/internetworking/technology/handbook/Internet-Protocols.html#wp4155>>
- IPPOLITO, Jon. Ten Miths of Internet Art. Digital art of our time, vectors.
Disponível em: <<http://www.nydigitalsalon.org/10/essay.php?essay=6>>
- IRISH, Paul. Browser CSS hacks. Paul Irish. 2009. Disponível em: <<http://paulirish.com/2009/browser-specific-css-hacks/>>
- JAKOBSON, Roman - Linguística E Comunicação. São Paulo, Cultrix, 1969
- JOHNSON, Steven. Cultura da Interface: Como o computador transforma nossa maneira de criar e comunicar. trad. Maria Luiza de A. Borges. Rio de Janeiro: Jorge Zahar, 2001
- JOHNSON, John; Crandall, Jordan; Scholder, Amy. Interaction: Artistic Practice in the Network. Eyebeam Atelier, 2001
- KELLER, Peter, KELLER. Mary, "visual cues: practical data visualization". Ieee computer society press, 1993.
- KENNEDY, Bill; MUSCIANO, Chuck. HTML & XHTML: The Definitive Guide, 6th Edition. O'Reilly, 2006
- KEMPER, Michael; ROSSO, Guido; MONNONE, Brian. Advanced Flash Interface Design. Friendsoft, 2006.
- KLEE, Paul. The thinking eye. London: Percy Lund Hunphries, 1961
- KLEINER, Dimitri e WYRICK, Brian. Infoenclosure 2.0. Mute - Culture and politics after the net, 2007. Disponível em: <<http://www.metamute.org/en/InfoEnclosure-2.0>>
- KÖHLER, Wolfgang - Gestalt psychology. New York: H. Liveright, 1929
- KOFFKA, Kurt. Princípios da Psicologia da Gestalt. São Paulo: Cultrix, 1975
- KRUG, Steve. Dont Make me think — A Common Sense Approach to Web Usability. Berkeley: New Riders, 2006
- KOLTERJAHN, Guillermo. Tratado elemental de las artes gráficas. Buenos Aires: Editorial de du S. R. L., 1965
- KOYANL, Sanjal, Bailey, Robert, Nall, Janice. Research-based web design & Usability Guidelines. Maryland: Communication Technologies Branch, 2003
- LANGRIDGE, Stuart. DHTML Utopia: Modern Web Design Using JavaScript & DOM. VIC Australia: Sitepoint, 2005
- LATHI, B. P. Modern Digital and Analog Communication Systems. 1a.ed Sacramento: Holt, Rinehart and Winston, 1983
- LEARY, Timothy. Chaos & cyberculture. Berkeley: Ronin Publishing 1994

- LENTZ, Jon Warren; TURNER, Bill e CHIA, Ian. *Flash: The Future*. San Francisco: No Starch Press, 2002
- LIE, Hakon Wium & BOS, Bert. *Cascading Style Sheets - designing for the web*. 3rd ed. Addison-Wesley, 2005
- LILAIANA, Olia. *Vernacular Web 2*. Disponível em: <<http://contemporary-home-computing.org/vernacular-web-2/>>
- LIMA, Marcelo Oliveira Coutinho de - *A sociedade da informação e a world wide web : uma análise comparativa de nove países*. São Paulo, 2002
- LOTMAN, Iuri. *A estrutura do texto artístico*, Lisboa, Editorial Estampa, 1978
- LUPTON, Ellen. *Novos Fundamentos do Design*. São Paulo, Cosacnaify, 2008.
- MACHADO, Irene. *Semiosphera — Espaços semióticos compartilhados*. <http://www.usp.br/semiosphera/>
- MACHADO, Marco Aurélio. *Do tempo em que as gifs animadas reinavam*, 2003. Webinsider. Disponível em: <<http://webinsider.uol.com.br/index.php/2003/02/25/do-tempo-em-que-as-gifs-animadas-reinavam/>>
- MANOVICH, Lev. *The Engineering of Vision from Constructivism to Computers*. University of Rochester tese de pós doutorado, 1993
- _____. *An Archeology of a computer screen*. Kunstforum International. Germany, 1995. Disponível em <http://www.manovich.net/TEXT/digital_nature.html>
- _____. *The Language of New Media*. Cambridge: MIT Press, 2001
- MARX, Karl - *O capital. Crítica da economia política*. São Paulo, Abril Cultural.
- MARTINEZ, Maria Laura. *Um método de webdesign baseado em usabilidade*. São Paulo, 2002
- MEYER, Eric. *CSS: The definitive Guide*, 3rd ed. Sebastopol: O'Reilly Media, 2007
- MOLES, Abraham, A. *A criação científica*. São Paulo: EDUSP, 1971
- MCFARLANE, David Sawyer. *CSS: The Missing Manual*. Sebastopol: O'Reilly, 2006
- MCINTIRE, Penny. *Visual Design for the modern web*. Berkeley, New Riders, 2008
- MCLUHAN, Marshall. *Comunicações em Massa e Cultura Tecnológica in Carpenter, Edmund e McLuhan, Marshall. Revolução na Comunicação*. Rio de Janeiro: Zahar Editores, 1968.
- _____. *Os meios de comunicação como extensões do homem*. São Paulo, Cultrix, 1964
- _____. *A galáxia de Gutenberg : a formação do homem tipográfico*. trad. Leônidas Gontijo de Carvalho, Anísio Teixeira. São Paulo : Nacional/EDUSP, 1972
- MCLUHAN, Marshall e FIORI, Quentin. *Medium is the message : an inventory of effects*. New York : Bantam, 1967
- _____. *Guerra e Paz na aldeia global*. Rio de Janeiro: Redord, 1971
- MEGGS, Philip. *Historia del diseño gráfico*. Tradução de Martha Izaguirre I., Carlos Iriondo S., Elvia Vera S. 1a.ed. - México: Editorial Trillas, 1991

- _____. Type and Image: The Language of Graphic Design. Devers: John Wiley & Sons inc, 1992
- MEYER, Eric. CSS: The Definitive Guide. 3rd ed. Sebastopol: O'Reilly, 2007
- MOGGRIDGE, Bill. Designing Interaction. Massachusetts: MIT Press, 2007
- MOLES, Abraham A. A criação científica. São Paulo: EDUSP, 1971
- _____. - Teoria da informação e percepção estética. Rio de Janeiro : Tempo Brasileiro, 1969.
- MUNARI, Bruno. Design e comunicação visual Lisboa. Edições 1970
- MURGIO, Matthew. Communications Graphics. New York: Van Nostrand Reinhold Company, 1969
- MUSTARO, Pollyana Notargiacomo - Hackers : um estudo sobre linguagens, identidades e educações no ciberespaço. São Paulo, 2003 Tese (Doutorado)
- NAVARRO, Ann. Effective Web Design. 2nd ed. Alameda: Sybex, 1999
- NEGRINO, Tom e SMITH, Dori. Visual QuickStart Guide JavaScript and Ajax for the Web. Berkeley: Peachpit Press, 2007
- NIELSEN, Jakob. Homepage: Usabilidade - 50 website desconstruídos. Rio de Janeiro: Campus, 2002
- _____. Projetando Websites. trad. Ana Gibson. Rio de Janeiro: Campus, 2000
- _____. One Billion Internet Users. Alertbox, dezembro de 2005. Disponível em: <http://www.useit.com/alertbox/internet_growth.html>
- ONTARIO, Ministry of Health and Long-Term care. Building a website from A to Z. Ontario: Ministry of Health and Long-Term care, 2003. Disponível em: <<http://hdl.handle.net/1873/5449>>
- O'Reilly, Tim. What is Web 2.0 - Design Patterns and Bussiness Models for the Next Generation of Software. O'Reilly. oreilly.com, 2005. Disponível em: <<http://oreilly.com/web2/archive/what-is-web-20.html>>
- PAGANI, Margherita, org. Encyclopedia of Multimedia Tecnology and Networking. 2a.ed. New York: Hershey, 2009
- PEIRCE, Charles Sanders - Semiótica. São Paulo, Perspectiva, 1977
- PIGNATARI, Décio - As massas e as mídias, in Comunicação e novas tecnologias: reflexões. São Paulo, SP : Editora COM-ARTE, Escola de Comunicações e Artes, Universidade de São Paulo, 1984
- _____. - Contracomunicacao. São Paulo : Perspectiva, 1973
- _____. - Informação, linguagem, comunicação. São Paulo, Perspectiva, 1968
- Programming Languages. Scriptol. Disponível em <<http://www.scriptol.com/programming/languages.php>>
- Protocols.com. TCP/IP Reference Page. Disponível em: <<http://www.protocols.com/pbook/tcpip1.htm>>
- REAS, Casey; Fry, Ben. Processin: a programming handbook for visual designers and artists. Massachustts: MIT Press, 2007
- RHEINGOLD, Howard. Tools for thought. Howard Rheigold, 1985. Disponível em: <http://www.rheingold.com/texts/tft/>

- _____. The Virtual Community, 1998.
Disponível em: <<http://www.rheingold.com/vc/book/>>
- ROBERTSON, James. So, what is a CMS? Step Two Design. 2003. Disponível em: <http://www.steptwo.com.au/papers/kmc_what/index.html>
- ROGÉRIO, Pedro, 2009. O HTML 5 e o fim do Flash player. Disponível em: <<http://www.pinceladasda-web.com.br/blog/2009/06/17/html5-e-o-fim-do-flash-player/>>
- ROLLIS, Richard - Design gráfico. Uma história concisa. São Paulo. Martins Fontes. 2001
- ROSENFELD, Louis e Morville, Peter. Information Architecture for the World Wide Web. Sebastopol: O'Reilly Media, 2002
- RUSSEL, Bertrand Arthur William - Introducao à filosofia da matemática.
- SEGARAN, Toby. Programming Collective Intelligence. Sebastopol: O'Reilly Media, 2007
- SELDES, Gibert. Revolução nas Comunicações in Carpenter, Edmund e McLuhan, Marshall. Revolução na Comunicação. Rio de Janeiro: Zahar Editores, 1968.
- SAMUELSON, Pamela. Legally Speaking: IBM's pragmatic embrace of open source. Communications of the ACM, vol. 49 no. 10. New York: ACM, 2006
- SCHAUER, Brandon. Experience Attributes: Crucial DNA of Web 2.0. Adaptive Path, dezembro de 2005. Disponível em: <<http://www.adaptivepath.com/ideas/essays/archives/000547.php>>
- SCHONFELD, Erick, 2009 Demo: FireFox 3.5 Treats Videos Like Web Pages. Why Can't Flash Do That? Disponível em: <<http://www.techcrunch.com/2009/06/09/demo-firefox-35-treats-videos-like-web-pages-why-cant-flash-do-that/>>
- SHEDROFF, Nathan. Organizing Things. Information Interaction Design: A Unified Theory of Design. 1994. Disponível em <<http://www.nathan.com/thoughts/unified/6.html>>
- SIQUEIRA, Ethevaldo. Grandes Personalidades das Comunicações. São Paulo: Dezero Editorial, 2001
- SIMONDON, Gilbert - On the mode of existence of technical objects. University of Western Ontario, 1980
- SOARES, D. B. - Compreendendo a Web no ensino de design : reflexões de uma professora de arte em formação contínua. São Paulo, 1998
- STALLMAN, Richard. Freedom — or Copyright?. GNU Operating system, 2009. Disponível em: <<http://www.gnu.org/philosophy/freedom-or-copyright.html>>
- _____. Science must "put copyright aside". GNU Operating system, 2001. Disponível em: <<http://www.gnu.org/philosophy/freedom-or-copyright.html>>
- STOLFI, Ariane. Legibilidade e evolução das mídias. In finetanks.com/referencia. São Paulo, cnpq, 2001
- SUREAU, Denis. Hystory of Programming Languages. Scriptol. 2008. Disponível em: <<http://www.scriptol.com/programming/history.php>>

- TALBOT, David. Which Ajax Lybrary is Right for Me? DevX.com dezembro de 2006. Disponível em: <<http://www.devx.com/AJAXRoundup/Article/33142>>
- TIMBERCON, Inc. History of the Fiber Optics, 2009. Disponível em: <<http://www.timbercon.com/History-of-Fiber-Optics/index.html>>
- THIMBLEBY, Harold. User Interface design. reading: ACM Press, 1990
- THOMSON, John. Privatization of the New Communication Channel - Timeline, 2000. Disponível em <<https://mywebspace.wisc.edu/jcthomsonjr/web/j561/timeline.html>>
- TUBAK, Franklin e Gifford, David. design Concepts in Programming Languages. Massachsets: MIT Press, 208
- MONNA, Rogério t. Interfaces digitais: a construção da informação na Web. São Paulo, Universidade de São Paulo, Escola de Comunicação e Artes, 2004
- TURNBULL, Artur e BAIRD, Russell N. The graphics of Communication - Typography Layout Design. New York: Holt, Rinehart and Winston, Inc., 1968
- Usability.gov. Your guide for developing usable & usefull Web sites. Disponível em: <<http://www.usability.gov/>>
- WALLER, Robert - The typographic contribution to Language. Towards a model of typographical genres and their underlying structures. unpublished PhD thesis, Department of Typography and Graphic Communication, University of Reading, August 1987. Disponível em: <<http://www.fb10.uni-bremen.de/anglistik/lang-pro/projects/gem/robwaller.html>>
- WALSH, David. Artistic and cultural problems in the current situation – part one.
- Web fonts, were are we? I love typography. 2009. Disponível em: <<http://ilovetypography.com/2009/07/20/web-fonts-%E2%80%94-where-are-we/>>
- WSWS International Editorial Board meeting, 2006. www.wsws.org
- WHEILDON, Colin; WARWICK, Mal,ED.— Type and layout: How typography and design can get in your message across – or get in the way; Berkeley: Strathmoor, 1996
- WEILL, Allain. Graphic Design: A History. New York: Thames & Hudson, 2004
- WOLF, Gary. The Curse of Xanadu, wired Issue 3.06, Jun 1995. Disponível em: <<http://www.wired.com/wired/archive//3.06/xanadu.html>>
- WOYCHOWSKY, Edmonf. Ajax:Creating Web Pages with Asynchronous JavaScript and XML. Indiana: Prendice Hall, 2006
- WILLIAMS, Trevor I.. A Short History of Twentieth-Century Technology c.1900 - c.1950. New York: Clarendon Press, Oxford University Press, 1982
- ZAKON, Robert H. Hobbes' Internet Timeline v8.2. Zakon Group LCC, 1993-2006. Disponível em: <<http://www.zakon.org/robert/internet/timeline/>>
- ZELDMAN, Jeffrey. Designing with web standards. Berkeley: New Riders Publishing, 2003
- <http://google-opensource.blogspot.com/>
- RIXC. 2010, <http://rixc.lv/>