

Do Napster e Gnutella ao projeto Freenet

O desenvolvimento cooperativo de software como exemplo de novas relações com as ferramentas semióticas trazidas pelo ambiente computacional.

André Vicente*

RESUMO

O Projeto Freenet aparecerá, aqui, como referência para pensar algumas das questões surgidas com a crescente utilização do ambiente computacional para a comunicação. Enfocaremos algumas consequências advindas da flexibilidade e replicabilidade da constituição parcialmente simbólica deste ambiente. Principalmente novas possibilidades de relação com os meios de comunicação dadas por conceitos como o de código aberto.

ABSTRACT

The Freenet Project will be reviewed as a indication for questions derived of the spreading of the communication process in computational environment. Will be focalize some consequences of the flexibility and reproductivity of the partiality symbolic structure of this environment. Include new possibilities of relation with the communication tools given by concepts like "open source".

Ao contrário do que possa sugerir o título, o foco principal desse trabalho não está na análise do fenômeno recente de explosão das ferramentas de compartilhamento de arquivos. Na verdade uma destas ferramentas está sendo tomada como exemplo de uma questão que se deseja contextualizar e discutir. O objetivo deste trabalho é levantar possíveis mudanças nas formas de construir e se relacionar com as ferramentas ligadas aos processos semióticos, entre as quais incluímos os meios de comunicação.

De início tentaremos ressaltar algumas especificidades do universo computacional. Principalmente os aspectos de plasticidade e capacidade de simulação. Em seguida discutiremos suas conseqüências, em termos do acesso à interferência na arquitetura das ferramentas semióticas, com balizamento na análise de experiências práticas. Veremos os aspectos sinérgicos trazidos pelas formas cooperativas associadas à progressiva interligação da computação. Principalmente, movimentos ligados à reunião de uma grande quantidade de pequenos esforços para a criação e transformação coletiva dessas ferramentas. Na terceira parte discutiremos como esses achados podem se relacionar com modificações nas relações entre o homem e seu universo tecno-cultural. E, por fim, analisaremos como o projeto de compartilhamento de arquivos Freenet pode representar efeitos práticos das novas possibilidades listadas.

44

Observamos ainda que este artigo apresenta resultados parciais de uma pesquisa em progressão, que aborda os aspectos econômicos, jurídicos e epistemológicos relacionados aos *softwares* de código aberto – vistos como ferramentas comunicacionais. Nesse artigo nos centraremos nos aspectos de virtualização do ferramental e suas conseqüências.

A máquina em aberto

No ano de 1937, o matemático inglês Alan Turing publicou um artigo chamado “Sobre Máquinas Computáveis”. Este trabalho foi considerado de grande importância no campo da matemática pura ao demonstrar que certos cálculos seriam irrealizáveis. Na época, passou relativamente despercebida uma proposição que mais tarde se mostraria fundamental. No artigo ele sugeria uma máquina que, sendo capaz de realizar seqüências programáveis de operações aritméticas simples, poderia simular qualquer sistema matematicamente modelizável, incluindo um número infinito de outras máquinas (Kay;2000:130). Por conseqüência essa máquina seria capaz de lidar com símbolos através de sua associação direta a sistemas numéricos. Poderia realizar com

eles qualquer operação desmembrável em passos de lógica simplificada como comparações e ordenamentos. (Pinker;1998)

Boa parte da engenharia da computação digital, do Z3¹ e ENIAC aos dias de hoje, foi baseada nesse tipo de proposta de cálculo programável e processamento lógico simbólico simplificado. Aquilo que conhecemos como o computador pessoal ainda pode ser entendido como uma máquina de Turing ligada a uma série de interfaces de entrada e saída. Desse ponto de vista é compreensível que se proponha que sua principal tarefa seja a de simular outras máquinas (Finneman;2000:142). Ou que ele seja visto como uma base de processamento com interface em aberto (Lévy;1993:101).

No entanto esse conceito pode ser entendido de forma mais ampla. Atualmente muitos dos processos não são executados por computadores isolados mas distribuídos por múltiplos níveis remotos dentro da rede mundial de computação Internet. Muitas requisições são parcialmente processadas nas máquinas em que se originam para então serem enviadas para servidores de ferramentas de busca, e-mail etc.

O projeto SETI@home exemplifica a questão de forma bem interessante. SETI (Search for Extra-terrestrial Intelligence) é o nome para o programa norte-americano de busca de vida inteligente fora de nosso planeta. Ele tem operado principalmente através da captação e análise de um largo espectro de radiofrequência em busca de sinais diferenciados das radiações típicas encontradas nos corpos celestes. Precisando de maior poder de processamento eles encontraram uma fórmula criativa para obter proveito de um fenômeno conhecido. Houve um contínuo desenvolvimento da capacidade de processamento dos computadores pessoais. O volume de instruções por segundo (MIPS) tem dobrado em média a cada dezoito meses. No entanto, as demandas das aplicações não tem acompanhado esse crescimento. Calcula-se que, hoje, o típico usuário caseiro use em média apenas 20% da capacidade de sua máquina dentro do total de tempo em que a mantém em funcionamento. A equipe do SETI encontrou uma forma de pedir um pouco da sobra. Criaram um protetor de tela que processa blocos de informação criados pelo servidor central do projeto. Na primeira conexão que o usuário faz à Internet, depois de instalar esse protetor, é recebido um desses blocos para análise. Nas conexões seguintes são enviados resultados de processamento e recebidos novos blocos. Assim o que é pedido aos usuários é que usem alguns hiatos nos quais não precisariam do processamento de suas máquinas, o momento em que pararam de trabalhar para tomar um café etc.

A busca de vida extraterrestre se mostrou uma causa com forte adesão. Atualmente o projeto conta com perto de 3 milhões de parti-

cipantes voluntários. Foi conseguido o maior poder supercomputacional já obtido. Já foi feito o processamento equivalente ao que um PC médio atual levaria seiscentos mil anos para realizar. Com o sucesso dessa iniciativa outras, usando o mesmo recurso, já começam a aparecer. A Universidade de Oxford está lançando um programa para o processamento de moléculas candidatas a remédios para o câncer. E existe um projeto para a previsão mais precisa do tempo. Um conceito interessante presente nessas idéias é que teríamos um conjunto de máquinas de Turing intercomunicantes que, por conseqüência, pode ser visto como uma máquina de Turing de maior porte. Como dizia um slogan da Sun Microsystems em meados da década de 90 "A Rede é o Computador" (O'Reilly;1999). Para Lévy (1993:102) ela seria a unidade de referência para a avaliação na informática.

Dada a capacidade intrínseca dos computadores para lidar com símbolos, e a facilidade cada vez maior da troca de dados entre eles, pode-se entender porque a simulação dos meios de comunicação feita por eles chama especialmente a atenção. Esta capacidade é enfatizada, por exemplo, em conceitos como o de unimídia (Rosnay;1997:95), voltado para a hipótese de que a capacidade principal dessa máquina, o conjunto da computação em rede, é o de condensar os mídias atualmente disponíveis - o que a levaria a se tornar "o" meio de comunicação.

46

No entanto, essa hipótese, não nos parece alcançar alguns aspectos fundamentais. As máquinas de Turing carregam possibilidades que vão além da simulação simples. Antes de tudo porque podem ser propostas hibridações. Se podemos simular este ou aquele meio de comunicação podemos, também, pensar nas possibilidades de combinação de suas características (Mayer;2000:321).

Contudo existem novas possibilidades para além das existentes nos meios de comunicação anteriores. Um exemplo está na capacidade de reagir interativamente ao "receptor" mesmo sem a presença do "emissor" (Goldberg e Kay;2000:112). Além disso a computação tem mostrado um forte papel como auxiliar na geração de conhecimento indicando que sua função vai além do transporte de informação (Finneman;2000:147). Lévy (1998:25) denomina o novo contexto de comunicação de era pós-mídia, caracterizada essencialmente pelo fato dos dispositivos de comunicação se tornarem ferramentas em aberto. Nesse sentido, talvez um conceito interessante seja o de metamídia (Kay;2000:136), onde o computador aparece como ambiente incubador de mídias, ou de suporte para a criação de mídias, muitas das quais sem correlatos possíveis no espaço físico.

O aspecto material da metamídia seguiria os padrões de outros produtos da era industrial. Engendrado e desenvolvido em laboratórios de alta tecnologia, sua construção envolveria métodos pouco acessíveis para a grande maioria. No entanto, ele vem sendo culturalmente formatado para desempenhar predominantemente duas funções: a execução cada vez mais eficiente de processos tipificados e a troca de dados. E esses procedimentos estão desenhados num código (código de máquina) que pode ser traduzido em outros mais próximos da compreensão humana, as linguagens de programação de alto nível. E a tradução de volta ao nível da máquina é feita caso a caso em cada plataforma. É interessante notar que, com isso, pode-se pensar o *software* com grande independência em relação à plataforma de hardware de que se dispõe (Pedro;1996:64). A maleabilidade do ambiente computacional viria do fato dessa camada simbólica ser o principal determinante em sua arquitetura final (Finneman;2000:146).

Essa camada traz uma série de conseqüências em conjunto com a elasticidade. Algumas das mais importantes estariam em novos padrões de acesso à construção das ferramentas contidas no ambiente computacional em comparação com as que se utilizam unicamente suporte material.

Arquitetura para todos?

Podem ser encontradas referências, em análises da história das técnicas, de como tecnologias de comunicação de grande impacto nos desenhos sociais humanos, como o alfabeto e a imprensa, teriam se vulgarizado com sua própria estrutura pouco evidenciada (Kerckhove;1995:142). Tecnologias mais recentes, surgidas no âmbito da revolução industrial, carregariam características ligadas às formas de produção em massa (Castels;1999:366). Uma delas é que seu projeto e sua construção se dão dentro dos padrões de alta especialização de grande escala típicos do fordismo. Ou seja, esses processos estariam distanciados da vida cotidiana e restritos a um pequeno número de participantes. Parte em função disto essas tecnologias teriam ganho uma aura de inquestionabilidade, de tecnicamente recomendadas, de intrinsecamente desejáveis (Lévy;1993).

Em geral essas constatações foram acompanhadas de exortações a que se tornem as tecnologias visíveis e domináveis, de que se desmistifique os frutos da ciência e se discuta como dirigi-los para o interesse público. Dessa forma as possibilidades de mudança no acesso à arquitetura, trazidas pelo aspecto simbólico da metamídia, podem ganhar um novo sentido. Principalmente depois de o matemático hún-

garo John Von Newman ter proposto que os programas fossem armazenados na memória eletrônica e não mais em cartões perfurados, dando a eles a altíssima velocidade, replicabilidade e plasticidade dos dados digitais. Gurgel (1/1/2001:4) considera que ele deu, assim, um grande presente a toda a humanidade.

Rosnay (1997:50) enfatiza a importância da liberdade de construção do computador pela programação. E Kay (2000:134) chega a dizer, pensamos que com um certo exagero, que as possibilidades dessa construção têm sido limitadas apenas pela imaginação de programadores e usuários e não pelo suporte físico.

O desejo de que esse processo seja intensificado aparece freqüentemente, em argumentos diversos. Seja por considerá-lo um desafio essencial na disponibilização do processo autoral na nova mídia (Murray;1997) ou por considerar que o ciberespaço deve ter sua construção feita de baixo para cima, para a preservação de um certo senso público no seu desenho (Johnson;1997:62); seja também para que seja cultivada a variedade de propostas de interface (Johnson;1997:224) ou mesmo por se acreditar que atingimos uma tal velocidade de transformação da mídia que é intrínseco que engenheiros e usuários comecem a se confundir. (Johnson;1997:6)

Mas, dentre estes, o argumento que aborda o problema colocado pelo excesso de informação é um dos principais. Ressalta-se, assim, a necessidade de uma “ergonomia para mentes” semelhante à que demandamos para a adaptação dos produtos industriais a nossos corpos (Rosnay;1997:204). Em função do excesso, o filtro de que dispomos se torna uma questão essencial. E quanto mais pudermos conformá-lo melhor. (Johnson;1997:235)

48

Prática

Alan Kay, o proponente da metamídia, não fazia apenas a defesa teórica de que todos pudessem transformar a máquina que propunha. Ele foi um dos que mais realizou experiências práticas na tentativa de criar uma linguagem de programação de fácil acesso. Seu principal trabalho nesse sentido foi o lançamento da Smalltalk, voltada para que crianças desenvolvessem suas ferramentas de desenho. Mas além de simples essa linguagem se mostrou poderosa. Seu uso se expandiu para áreas que iam de *softwares* de música feitos por músicos até o auxílio a tomada de decisões. Foram desenvolvidos mais de 200 aplicativos. Kay ressaltou o quanto o envolvimento com a programação mudava o ponto de vista dos profissionais sobre suas áreas de atuação (Goldberg e Kay;2000:116-117).

Uma das primeiras histórias bem sucedidas de tradução do aspecto simbólico da arquitetura computacional para algo mais próximo da compreensão humana coincide com a história do modelo de desenvolvimento cooperativo de *software* open source. Em 1974 foi liberada pelo Bell Labs a primeira versão do sistema UNIX. Até então todos os sistemas operacionais eram desenvolvidos em Assembler, linguagem de baixo nível, próxima aos códigos de máquina. Esse sistema foi desenvolvido em “C”, uma linguagem de alto nível, mais próxima das linguagens naturais humanas. Posteriormente era compilado, traduzido para o código de máquina. Esse processo lhe deu uma propriedade inédita até então. Podia ser instalado em vários tipos diferentes de computador, era transportado no formato de alto nível e depois compilado para cada máquina. Assim se poderia trabalhar em máquinas diferentes com o mesmo sistema e parque de aplicativos.

O UNIX foi desenvolvido através de fundos públicos de pesquisa e por esta razão seu código fonte, a versão em linguagem de alto nível, foi publicado (Castels;1999:376). As demandas de pequenas adaptações nesse processo de transporte, junto com a maior facilidade da linguagem de alto nível, levaram um grande número de programadores a conhecer a essência desse sistema. Esse fato tem alguns desdobramentos de grande importância para este trabalho.

Antes de tudo se tornou corriqueiro que atores com poucos recursos a mão implementassem suas próprias modificações. Assim, por exemplo, três estudantes universitários norte-americanos, que se ressentiam da falta de acesso à, então restrita, rede ARPANET, improvisaram uma adaptação de protocolos do sistema operacional UNIX que permitia a comunicação de computadores através de linhas telefônicas convencionais. Com ela a base instalada de microcomputadores com modems poderia ser integrada em uma nova rede de comunicação, com um esforço mínimo de instalação do novo *software*. Deste desenvolvimento surgiu a rede USENET, que integra milhões de usuários. E, também, os Sistemas de Boletins Informativos (Bulletin Board System - BBS), pequenas redes de informação local que se espalharam por todo mundo. (Raymond;1999b)

Com o código fonte disponível o *software* ganhava uma espécie de vida própria, uma autonomia em relação ao grupo que o criou. Com o tempo a AT&T, incorporadora do Bell Labs, começou a tentar obter retorno comercial sobre sua criação. Passou a cobrar pelas novas versões implementadas que lançava do UNIX. No entanto grupos de universitários com perspectivas contraculturais desenvolveram e lançaram novas versões gratuitas do sistema, a partir do código fonte das

versões anteriores. As mais conhecidas foram as Berkeley *Software Distribution*, como o 3BSD e o FreeBSD. (Edwards;26/08/00)

Esses foram considerados alguns dos primeiros passos de articulações maiores que apareceram posteriormente como o movimento Open Source ou o Free Software. Dentro do segundo, destacou-se a Free Software Foundation (FSF), fundada, no início dos anos 80, e dirigida até hoje por Richard Stallman. Esse programador trabalhava na área de inteligência artificial do MIT mas abandonou tudo para lutar pelo que chama de *software* livre. Ligado à contracultura californiana dos anos 70, ele afirma que sua causa é eliminar os efeitos perniciosos dos monopólios no *software*, buscando gerar oportunidades para dividir com toda a humanidade o prazer da programação e seus frutos. (Stallman;7/9/00)

O formato de licenciamento defendido pela FSF, a General Public License (GPL), se tornou referência para a maioria dos *softwares* de código aberto da atualidade. O conceito básico por trás dele é o de *copyleft* (uma brincadeira com o termo *copyright*): Um dos maiores receios daqueles que produzem programas com código aberto é o de que suas idéias sejam apropriadas por produtores de *software* comercial. O conceito de *copyleft* prevê, ao se assinar a licença de uso de um programa, que se possa fazer qualquer tipo de modificação ou utilização, a não ser por agregá-los a um produto comercial. Uma tentativa, por via legal, de impedir a apropriação empresarial. (Machado;1/5/2000:5)

O principal projeto da FSF é chamado de GNU (acrônimo de si mesmo - Gnu Not UNIX - visando enfatizar o afastamento das versões comerciais do sistema). Ele envolve a elaboração do ferramental necessário para o desenvolvimento de sistemas e aplicativos de código aberto baseados no UNIX. Estas ferramentas incluem compiladores, tratadores de erros, geradores de documentação de ajuda e outros. Elas foram empregadas para a criação da maior parte do *software* de código aberto disponível atualmente. (Machado;1/5/2000:5)

A rede - nova distribuição do simbólico

O aspecto simbólico da arquitetura do universo computacional foi potencializado pelas novas capacidades de distribuição e operação simbólicas trazidas pela Internet. A rede adicionou uma certa onipresença à altíssima replicabilidade dos dados digitais. Permitiu a publicação num nível mundial a baixos custos. Possibilitou trocas assíncronas. É o receptáculo de grandes repositórios simbólicos vasculháveis. Antes dela a distribuição de produtos de informação

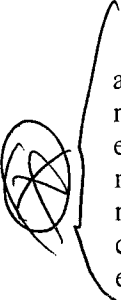
como o *software* era determinada pela distribuição de seus suportes físicos. Era necessária uma estrutura industrial e comercial proporcional ao mercado que se poderia atingir. Com o novo ambiente de comunicação começam a aparecer alguns casos que parecem fugir desse padrão.

Em 1991 Phil Zimmerman, norte-americano de convicções anarquistas, criou o PGP (Pretty Good Privacy) programa de criptografia bidirecional baseado num sistema de chaves duplas. Com sua disponibilização gratuita na Internet foi franqueada a qualquer usuário da rede uma ferramenta de comunicação isenta de “escuta”. Houve uma grande adesão, a criptografia está vulgarizada em muitas aplicações corriqueiras da rede. Ocorreram, inclusive, choques com as tradições de controle governamental das comunicações. A China proibiu por lei o uso deste tipo de ferramenta e a França criou pesadas restrições. No entanto o controle efetivo sobre esse tipo de prática é considerado muito difícil nas atuais configurações tecnológicas. Na medida em que se queira disponibilizar à população as vantagens do acesso a rede, é complicado restringir as ferramentas de *software* poderão ser usadas. (Lévy;1999:207)

Em 1992 um pequeno grupo de estudantes do Laboratório de Aplicações Computadorizadas de Illinois criou o Mosaic, aplicativo que permitia a navegação multimídia pela Internet através do protocolo HTML. Poucos anos mais tarde uma parte desses estudantes disponibilizou gratuitamente na rede o Netscape, uma evolução da estrutura do Mosaic (Castels;1999:377). Ele chegou a ser usado por mais de 70% dos usuários da Internet com dezenas de milhões de downloads de sua cópia de instalação.

Em outro projeto, de 1996, 4 estudantes israelenses criaram o ICQ (aplicativo que permite sessões de chat diretas, usuário a usuário). Novamente foi disponibilizado gratuitamente e alcançou a marca das centenas de milhões de usuários apenas três anos depois de ser criado.

O ponto que desejamos ressaltar é que essas iniciativas foram tomadas por pequenos atores, com algum conhecimento de programação, que não precisaram senão de acesso a terminais de computação. Indivíduos ou pequenos grupos que, com um mínimo de recursos, foram capazes de experimentar e promover a implantação de novas ferramentas de comunicação, em nível mundial, num espaço de poucos anos. Teria havido uma sinergia entre as características de comunicação da Internet e a possibilidade de construção simbólica de novas ferramentas, dentro do ambiente computacional. Ela permitiu prescindir da maior parte da estrutura de distribuição e comunicação necessária anteriormente.



De certo modo todo momento de mudanças aceleradas apresenta, em maior ou menor escala, oportunidades para grupos que não eram privilegiados dentro do jogo de forças anteriormente estabelecido. Mas duas questões nos parecem peculiares neste momento. De um lado a velocidade com que os pioneiros conseguem reconfigurar o espaço. E de outro o fato de que não precisam da construção progressiva de um grande suporte financeiro, institucional e material para isso. (Johnson;1997:224).

Potencialização do código aberto

No caso do modelo de desenvolvimento de aplicativos com código aberto os grandes repositórios simbólicos, trazidos pela Internet, trouxeram uma segunda consequência importante. Eles se mostraram excelentes bibliotecas e campos de troca para as linhas de código. Em torno deles foram reunidos programadores com interesses comuns, que passaram a contar com a possibilidade de comunicação assíncrona sem restrição geográfica. Ela teve um forte efeito sinérgico, principalmente no caso dos *softwares* de estrutura naturalmente compartimentalizada, caso dos baseados na plataforma UNIX. (Bezroukov;1999)

52

Uma das demonstrações dessa sinergia vem do fato do modelo de código aberto ter sido utilizado na engenharia da maioria dos programas usados na operação da própria Internet. Dentre eles estão incluídos o Sendmail e o Apache, os servidores mais usados na rede, o primeiro para e-mail e o segundo para HTML. Inclui ainda a linguagem Perl, muito usada para criar aplicações CGI, as mais comuns requisitadas via HTML. E os próprios protocolos TCP/IP e DNS que foram desenvolvidos por sistema de contribuições.

Todos eles foram desenvolvidos dentro do âmbito da IETF, Internet Engineering Task Force. Essa entidade agrupa instituições de pesquisa privadas e públicas, incluindo universidades, e recebe aportes de verba estatal. Sua principal função é definir padrões para os protocolos da Internet e a geração dos *softwares* surge com função agregada natural. Os códigos fonte desses programas foram distribuídos a toda a comunidade de pesquisa mesmo porque a IETF é, também, uma entidade de fomento. (Raymond;1999b)

Também é atribuído a esse efeito sinérgico, entre a Internet e o código aberto, boa parte do sucesso de Linus Torwalds em usar uma versão do UNIX para a família IBM-PC, o MINIX, como base para criar o sistema operacional GNU / LINUX. Através da rede ele conseguiu agregar milhares de desenvolvedores voluntários

(Raymond;1999a). O sistema LINUX vem sendo adotado por gigantes como Boeing, Disney, NASA. A China optou por sua utilização em toda a rede estatal e o México em sua rede de ensino. Ele tem recebido investimentos de empresas como IBM, Intel, Hewlett-Packard, Corel, Sun, Silicon, Oracle, Cisco, Netscape, principalmente através do apoio ao desenvolvimento de uma base de aplicativos (também de código aberto). O total de seus usuários foi estimado entre 16 e 40 milhões em 1999. Em função desses fatos, ele tem sido visto, por vezes, como a mais séria ameaça, em muitos anos, ao monopólio que a Microsoft estabeleceu sobre os sistemas operacionais e aplicativos na família de processadores x86. (Machado;3/1/2000:13)

Esse sistema vem sendo utilizado em redes governamentais e de educação como no México e na China, por exemplo (Bezroukov;1999). No Brasil, a base de usuários foi estimada em quatrocentos mil ao final do ano de 1999. O deputado Federal Walter Pinheiro PT-BA apresentou projeto de lei, semelhante ao aprovado recentemente na França, que torna obrigatório seu uso em órgãos do governo, inclusive em empresas estatais e de economia mista(Machado;3/1/2000:13).

A experiência brasileira mais significativa no uso do *software* de código aberto vem do Rio Grande do Sul. Lá, o Governo do Estado lançou o Projeto Software Livre RS em maio do ano passado, durante o 1º Fórum Internacional Software Livre 2000. Através da Companhia Estadual de Processamento de Dados do Rio Grande do Sul (Procergs), o governo propôs um grande movimento em parceria com a Prefeitura de Porto Alegre para instituir o uso do *software* de código aberto em todo o estado. Assim, foi criada uma rede de laboratórios para o estudo do GNU/Linux e outros *softwares* abertos, além de elaborados um curso de suporte para estas ferramentas e um Consórcio Editorial para a publicação de livros, apostilas e outros materiais sobre o assunto. O projeto também prevê a realização de um grande evento anual – o Fórum Internacional *Software* Livre. (N.A.;12/2/2001)

Circuito funcional

Independentemente de seu sucesso, os *softwares* de código aberto parecem carregar alguns diferenciais específicos. Um deles é a tendência a ser sistemas evolucionários, como o conceito proposto por Licklider e Taylor (2000:104). São aplicativos que não chegam a ter uma face acabada mas apenas caminhos de mutação. Uma vez que os usuários têm acesso às minúcias da arquitetura possibilita-se uma grande adaptabilidade permitindo a solução de problemas bastante es-

pecíficos, por vezes, individuais. Foi amplamente divulgado, na comunidade dedicada ao código aberto, um artigo de uma revista especializada do Butão enfatizando como esse formato de desenvolvimento permitiu que pequenas comunidades lingüísticas pudessem adaptar ferramentas de informática. Esse tipo de projeto dificilmente movimenta a área do *software* comercial (N.A.;25/3/2000)

E soluções como esta podem ser disponibilizadas em repositórios públicos na Internet. Com eles se estabelece um circuito de trocas, aglomeração, análises, aprimoramentos e colocação de novos problemas. No universo do código aberto é comum observar e aproveitar o trabalho já feito. Como já foi dito aqui, o próprio Linux partiu de um sistema anterior mais simples, o MINIX. Nele havia sido feito a maior parte do trabalho de portar o UNIX para a plataforma x86 (Raymond;1998). Uma vez que se percebe a demanda de uma alteração específica em um aplicativo desse tipo, é freqüente que se encontre algo próximo num repositório de código e apenas se precise fazer pequenos ajustes. E ainda, quando é preciso desenvolver a solução desde o início, é comum que ela seja disponibilizada, deixando a solução pronta para quem precise.

54

Essa forma cumulativa parece responder a uma parte do paradoxo de como foi desenvolvido um grande parque de aplicativos, com milhões de linhas de código, contando principalmente com a contribuição voluntária de milhares de programadores. Através dela, esses desenvolvedores passaram a contar com programas altamente personalizáveis, adaptáveis e controláveis. Além disso a possibilidade de interagir com os repositórios de código se mostrou um excelente campo para treinamento em programação. Muitos dos programadores voluntários vêm, inclusive, dos bancos universitários da informática (Hermann, Hertel e Niedner;26/08/00).

Tudo isso talvez ajude a entender o reconhecimento da troca de programas como parte importante da comunicação por computador (Licklider e Taylor;2000:102). Como todo esse circuito é paralelo às próprias discussões, por chats e maillists, sobre a engenharia coletiva dos aplicativos fala-se de uma comunicação complementada em dois níveis. De um lado, a linguagem natural humana e, de outro, as trocas em linguagem de programação (Edwards;26/08/00).

Essas características específicas dos aplicativos de código aberto parecem implicar mudanças nas próprias relações dessas ferramentas com seu ambiente sócio-cultural. Diversos autores têm demarcado que aspectos tipicamente associados ao comportamento inteligente não dependem apenas de atividades cerebrais. Está freqüentemente em jogo a habilidade para o agenciamento do meio circundante para realização de ta-

refas cognitivas. Para Dennet (1997:122) a grande capacidade de descarregar tarefas cognitivas para o meio ambiente é um diferencial importante do Homo Sapiens. Ela já aparece em operações simples como a mera separação física de itens processados um a um. No entanto, vários mamíferos já realizam atividades como a demarcação do território com urina. Estariam substituindo a complicada memorização visual de uma grande área pela simples detecção de odores. Mesmo formigas são capazes de utilizar sinalizações químicas. E não apenas para uso individual mas até para comunicação, no que Rosnay (1997:367) denominou escritura de uma memória coletiva.

Porém Dennet (1997:93-126) vê uma possibilidade fundamental de desdobramento dessa questão. Citando o trabalho de Richard Gregory ele descreve a capacidade de reincorporar a seqüência de ferramentas acumuladas no meio ambiente cultural nos próprios modelos mentais de elaboração, no que ele chama de imaginação proteticamente reforçada. Para ele a auto-recursividade que se estabelece nesse caso abre novas possibilidades com limites desconhecidos. As ferramentas cognitivas, como as fórmulas de cálculo, seriam as que mais intensamente reforçam este ciclo. Dentro delas a palavra seria uma das mais operantes.

Leroy-Gourhan (1965:11-22) também demarca que o homem e outros seres têm capacidade de projetar no mundo seqüências de cadeias operatórias. Mas o primeiro se diferencia por recorrer a um sistema simbólico elaborado para isso, deixando o aspecto instintivo em segundo plano. Através da simbolização o homem pode simular diferentes cenários e aumentar muito sua gama de reações. Mas, principalmente, ela permite que sua memória individual seja transferida para o meio social. E esse sistema se configurou com tal intensidade que a maior parte das cadeias operatórias das novas gerações viria da educação e não mais da programação genética. Isso teria permitido o estabelecimento do que chamamos de sociedade humana. Gourhan a categorizou como uma espécie de ser social cuja evolução seria muito mais rápida do que os padrões puramente biológicos permitem, evolução que modifica constantemente o próprio homem.

A linguagem aparece como a forma mais evidente de simbolização humana. No entanto, Gourhan (1965:30-37) procura enfatizar o aspecto simbólico também da atividade técnica. Antes de tudo pelo fato das áreas do cérebro relativas à articulação da linguagem e à atividade manual serem próximas e entrelaçadas. Ambas seriam formas de expressão. Para ele a linguagem se baseia numa capacidade de destacar aspectos do objeto designado e realizar sua substituição por um conceito que apenas o representa. Essa relação se repetiria dentro da téc-

nica da seguinte maneira: os utensílios primitivos funcionariam como extensões das mãos, unhas, dentes etc. O seu aspecto simbólico estaria na capacidade separar algumas das funções específicas dessas partes da anatomia e incorporá-las num objeto exterior. No seu exemplo o martelo não é uma nova mão artificial mas apenas representa a sua função de percussão.

Gourhan (1965:38-44) procura estabelecer uma seqüência histórica da exteriorização de tarefas cada vez mais complexas para a técnica. Primeiramente teriam sido exudadas algumas funções específicas de certos aspectos da anatomia. Em seguida a mão se torna apenas impulsionadora de artefatos que emulam encadeamentos de atividades mais complexas. Posteriormente mesmo a força motriz da mão é substituída pela força hidráulica, eólica, animal e, mais tarde, pelo vapor. Passou a caber ao homem o papel de controlador, regulador de processos. Por fim fala da externalização da memória com o aparecimento do conceito de programação já em máquinas como a pianola. O tear percorreria toda a série descrita por ele desde sua versão manual, passando pela de pedais, depois pela de acionamento externo e por fim em versões como a de cartão perfurado de Jaquard. Para Gourhan a programação estaria ligada à capacidade de fazer executar diferentes seqüências de tarefas complexas de acordo com instruções previamente codificadas. A “memória” do maquinário teria como base um código derivado da linguagem humana. Haveria uma conversão de nossas cadeias operatórias no que ele chama de programa operatório, uma exudação mais direta da capacidade simbólica.

56

Retomando a abordagem antropológica de Gourhan, Lévy (1996:98) descreve à sua maneira uma seqüência deste padrão. Para ele as ferramentas incorporam a memória da humanidade. Na medida em que as utilizamos estaríamos recorrendo à inteligência coletiva. Mas, principalmente, elas nos servem com novas metáforas para a análise do mundo que nos cerca, resultando em novos engendramentos técnicos. Assim existiria uma espécie de diálogo lento e constante, travado pela humanidade, inscrito nessa semântica material dos objetos técnicos.

Dentro desses objetos, é dado um destaque especial aos sistemas semióticos como sistemas de escrita, de cálculo, meios de comunicação. Um diferencial importante do homem seria uma capacidade mais específica para criá-los. Eles são ferramentas especialmente poderosas em termos da alimentação desse circuito. E já existiria uma certa percepção social disso. Por exemplo, muito do que se pretende com a educação em nível primário seria a introjeção desse tipo de ferramenta.

É interessante notar que, como sinalizou Martino (9/2000), as fer-

ramentas especificamente dirigidas para o auxílio ao processo de simbolização já possuíam uma certa duplicidade dentro da relação homem/técnica retratada por Leroy-Gourhan. Seu uso social reside nas operações de significação e, ao mesmo tempo, colocam seus próprios significados dentro do universo tecno-cultural.

Como dissemos a capacidade de simbolização foi tomada por Gourhan como um diferencial importante da espécie humana. Nesse sentido ele atribui grande importância aos significados de reconfiguração do homem e de seu aspecto social que as ferramentas de manipulação simbólica trazem. Nesse sentido ele fala de diferentes períodos de “memória coletiva” de acordo com as mudanças nesses suportes usados para preservá-la. Períodos que parecem relacionados com grandes modificações nas características do que ele chama de sociedade humana. Assim ele faz uma listagem de diferentes padrões sociais que podem ser associados aos diferentes aparatos disponíveis para a manipulação simbólica. Começando por relacionar a oralidade com o modo de vida caçador / coletor e os sistemas mitológicos de conhecimento. Também a escrita com os processos de controle de sociedades agrícolas mais extensas. E a mecanografia com a sistematização científica do saber. Indo até o que ele chamou de “seriação eletrônica” vista como necessária para lidar com a massa de conhecimento já produzida (Leroy-Gourhan;1965:59-66).

No entanto, nos parece que um dos grandes diferenciais das ferramentas de comunicação do ambiente computacional é que parte dos processos de significação interna que descrevemos parece estar efetivamente codificada, e não numa semântica material difusa. Em nossa capacidade típica para engendrar sistemas semióticos teríamos criado um que se confunde com a essência material do que representa. O código de programação é parte da constituição final de nossos ambientes computacionais. Criamos máquinas que são de constituição simbólica e são capazes de representar parcialmente o próprio sistema de representações

E o código em questão é do universo do digital. Para Rosnay (1997:96), o digital estabeleceu uma linguagem universal, uma espécie de esperanto das máquinas destinadas a comunicar. Ele compara sua invenção à da moeda, que permitiu comparar e estocar valor, em função de sua capacidade de conversão dos mais diferentes formatos de informação. E esse mesmo sistema semiótico serve tanto para a configuração da arquitetura dessa ferramenta quanto pode ser traduzido para nosso uso. Segundo Johnson (1997:14) a tarefa intrínseca das camadas superiores dessa arquitetura seria a de estabelecer uma tradução desse esperanto em novas metáforas

compreensíveis para o homem.

E como ficaria a posição dos aplicativos desenvolvidos coletivamente dentro da questão das técnicas até aqui detalhada? Existe um outro setor, além do de código aberto, onde essa questão parece ter trilhado um caminho inverso.

Dentro do conceituário de gestão do conhecimento, a configuração do ambiente e das ferramentas de trabalho é vista como depositária de importantes conhecimentos a respeito de uma dada função. Por isso alguns dos aplicativos criados nesta área têm esta questão em vista. Oferecem ambientes de trabalho altamente personalizáveis pelo usuário e buscam o máximo de simplificação para a confecção de novas ferramentas.

O que se pretende, num certo sentido, é que o *software* absorva uma parte do conhecimento necessário para desempenhar as funções de trabalho que o utilizam. Principalmente as tarefas mais simples e repetitivas. Os passos discretos das operações são traduzidos para a computação e automatizados. Deixa de ser necessário levá-los em conta bastando apenas requisitar tal processo à máquina.

No entanto, essas ferramentas podem envelhecer ou precisar de modificações. Nesse momento o código de programação passa a funcionar como referência de consulta para que se recuperem as tarefas incorporadas no aplicativo para modificá-las. Por essa razão aparece uma busca de simplificação do desenvolvimento.

58

Ou seja: qualquer nova idéia sobre o descarregamento de tarefas para os aplicativos pode ser aproveitada, convertida em ferramenta e esquecida pelos funcionários envolvidos. Ela volta a ser lembrada, com a retomada de seu código de construção, somente caso sejam necessários aprimoramentos.

Haveria aqui, então, um duplo armazenamento: um funcional, implícito, com o descarregamento de processos padronizados, longos e repetitivos para a execução pela máquina. A representação parcial do próprio sistema de representações simbólicas que somos nós (Pedro;1996:63). E outro explícito, com o armazenamento da sequência de instruções dadas para referência futura. Maximizado pela capacidade do universo computacional de manipular os próprios códigos que o constituem.

Algumas empresa de vulto estão começando a testar esses conceitos. Existe, por exemplo, um sistema em uso experimental na própria gerência da IBM chamado Competency Networks (N.A.; 22/10/00). A Xerox Company também vem fazendo experiências pioneiras com o sistema mundial de dados sobre soluções de manutenção. Mas, aparentemente, esses conceitos se radicalizaram em propostas de empresas menores como

Intraspect, Kanisa, Primus e outras (N.A.; 25/11/00) (N.A.; 27/11/00).

O que interessa ressaltar aqui é que uma espécie de potencialização da transferência de operações para as técnicas, no sentido descrito por Gourhan, parece ser uma preocupação nesses projetos. No caso a finalidade seria a fixação de tarefas cognitivas no patrimônio das empresas, com premissas de apropriação. No entanto, esse fenômeno também parece ocorrer numa escala maior no modelo de código aberto, numa forma menos intencional.

No caso do *open source* também é prática a reutilização do trabalho já feito anteriormente com pequenos ajustes. Isso pode ocorrer de uma forma integral, com o uso de um velho programa como base para a geração de um novo. Podem-se aproveitar apenas as partes de outro programa que executam uma função parcial do interesse do programador. Com o aparecimento de grandes repositórios de código o programador está se tornando um pouco um “bricoleur”.

É importante demarcar a possibilidade de analisar os trabalhos feitos anteriormente. Em meio às comunidades em desenvolvimento isso costuma ser feito usando parcialmente uma linguagem tradicional escrita como o inglês. Mas as comunidades mais avançadas trocam, por vezes, cadeias de código com apenas alguns comentários. Supõe-se que boa parte do que está sendo proposto vai ser entendido a partir da simples análise do código. Ao mesmo tempo, muitos programadores usam as bibliotecas de códigos como as do Linux como fonte de aprendizado.

Lévy (1996:122) cita a possibilidade da existência de muitas formas de pensamento e de seu registro. Ele coloca que um aplicativo desenvolvido coletivamente pode ser tomado como objeto de conhecimento, portador de saber, no sentido de veicular uma problematização dentro de um grupo de trabalho. Se a interface entre homem e máquina pode ser vista como uma nova forma cultural (Johnson;1997:38) ou uma forma discursiva significativa (Mayer;2000:322), as linguagens de programação são os códigos que estão tornando esse discurso mais claro e compreensível e permitindo seu registro.

Esta codificação pode estar trazendo uma série de novas consequências. Entre elas o fato de tornar essas significações mais explícitas no sentido aplicado por Nonaka (1997) e Stewart (1998). Fato que indica que elas poderão ser mais comunicáveis, serão mais facilmente estocadas e, também, analisadas. E, em função disso, talvez possamos falar de uma tendência à aceleração e complexificação das relações homem / técnica, como vistas por Gourhan, em comparação ao momento em que estavam em jogo apenas os suportes mate-

riais da técnica. Principalmente no sentido de um aprofundamento da afinidade, descrita por ele, entre o simbólico e o técnico.

Freenet

É sob a ótica dessas novas possibilidades da construção social de ferramentas de comunicação que queremos focar nosso principal objeto de estudo. O projeto Freenet surgiu, como várias outras propostas de desenvolvimento de aplicativos de código aberto, como resultado da insatisfação de um estudante universitário com as opções de *software* disponíveis para uma atividade que lhe interessava. No caso o compartilhamento de arquivos.

O uso das ferramentas de compartilhamento de arquivos via Internet viveu um grande crescimento recentemente. Principalmente em função da vulgarização de novos algoritmos de compressão de áudio (como MPEG Layer 3 - MP3 - e Yamaha VQF) que permitiram que arquivos contendo músicas inteiras ganhassem dimensões compatíveis com as taxas de transferência encontrados atualmente na rede. De início surgiram repositórios contendo listagens completas com o trabalho de músicos de sucesso. Como estes esbarravam em problemas de direito autoral foram sendo restringidos em processos movidos por selos musicais, empresários etc. O compartilhamento de arquivos surgiu como atalho para o problema judicial. Em princípio, não existiriam restrições legais a essa prática. Surgiram então ferramentas que catalogam os arquivos disponibilizados pelos seus usuários permitindo automatizar as trocas. O Napster é a ferramenta mais conhecida e mais usada atualmente na Internet para esse fim. No entanto, a sua estrutura é dependente da atividade de um servidor central. É nesse aspecto que aqueles que se sentem prejudicados por esse tipo de aplicativo, como os grandes selos musicais, têm trabalhado em termos judiciais. Conseguindo-se impedir o funcionamento desse servidor paralisa-se toda a ferramenta.

O aplicativo considerado a melhor opção tem uma história muito peculiar. O Gnutella foi desenvolvido pela mesma equipe criadora do executor MP3 mais utilizado atualmente, o Winamp. A idéia era oferecer uma ferramenta para que os usuários compartilhassem suas músicas. No entanto ele ficou disponível na rede por apenas algumas horas. A direção da AOL, possuidora dos direitos do programa, ordenou que fosse retirado do ar com receio de problemas judiciais. Ocorre que os poucos usuários que obtiveram uma cópia estabeleceram um circuito de redistribuição que elevou grandemente a base instalada. Ao mesmo tempo foi feita a engenharia reversa permitindo que programa-

dores independentes continuassem o desenvolvimento do aplicativo. Surgiram muitas versões, várias delas de código aberto. O Gnutella teria como grande diferença uma estrutura descentralizada. O programa funciona ponto a ponto entre as máquinas de cada um de seus usuários. No entanto, esse padrão dificulta muito a localização de arquivos específicos. Ao mesmo tempo, como as requisições são replicadas entre os nós, estabelece-se um tráfego muito intenso e desnecessário com uma queda proporcional no desempenho.

Descobriu-se, posteriormente, que nenhum dos dois aplicativos é seguro quanto ao anonimato de seus usuários, a principal proteção com que eles supunham contar. Os números de IP dos envolvidos (identificação única de qualquer computador ligado à Internet) podem ser obtidos por ações simples de rastreamento. (Clarke e Sandenberg; 18/9/00:)

Em função de sua insatisfação com esses problemas Ian Clarke, estudante de análise de sistemas na universidade de Edinburgo, dedicou boa parte de seu tempo ao desenvolvimento de uma nova ferramenta de compartilhamento de arquivos. Utilizou o ferramental do GNU e a licença GPL. Escolheu o JAVA como linguagem para maximizar a portabilidade entre as plataformas de *hardware*. Em função do fato de ele ter escolhido o formato de código aberto, esse projeto se tornou um foco de discussão sobre esse tipo de ferramenta, e as demandas de seus usuários.

Desde sua instalação, em dezembro de 1999, o site de referência do projeto Freenet já somou mais de seis milhões de *page views*. No mês de outubro de 2000 esse número estava crescendo numa magnitude de cerca de um milhão por mês. Já foram feitos mais de 350 mil *downloads* do código fonte diretamente do site. Esse número não inclui os sites secundários onde esse código é disponibilizado.

Já existe um projeto paralelo de interface gráfica. Alguns dos outros aplicativos *open source* como *browsers* e multi-clientes para download estão incorporando seu protocolo ao leque clássico TCP/IP, DNS, FTP etc. Ele parece tender ser mais bem sucedido do que qualquer todos os outros softwares de compartilhamento de arquivos incluindo o Gnutella. No entanto seu sucesso em maior escala parece estar vinculado à capacidade do Napster resistir ao ataque judicial. Ou da realização das intenções do selo musical alemão BMG, que recentemente o adquiriu, de passar a cobrar pelas músicas obtidas.

Construção social

O projeto Freenet foi amadurecendo em meio às discussões dos desenvolvedores/usuários. Um aspecto interessante para sua análise é

que, tanto o diálogo travado sobre seu desenvolvimento em *mailist*, quanto as contribuições para o código, estão bem documentadas em seu site. Esse fato permite que se possa acompanhar os paralelismos. E, também, de que forma um aspecto pode interferir no outro.

É interessante notar que mesmo nas listas de mensagens podemos encontrar uma grande quantidade de linhas de programação, que aparecem com poucos esclarecimentos. Supõe-se que os envolvidos entenderão dicas, soluções e correções, sugeridas dessa forma, sem necessidade de recorrer a palavras.

Esse diálogo em código por vezes ganha proporções mais amplas. Eventualmente é sugerido o uso de partes de outros aplicativos redigidos na mesma linguagem JAVA. Outras vezes são propostos encaixes de múltiplos delas que resultarão na função esperada (N.A.;14/9/00:). Essas possibilidades sugerem a associação da função do desenvolvedor quase à de um *bricoleur* de ferramentas digitais.

Muitas das discussões giram em torno de simples erros de construção, os “bugs”, e suas correções. A capacidade de detectar erros e eliminá-los é muito associada aos projetos de código aberto. Principalmente por conta da confusão e comunicação permanente entre usuários e programadores.

62

O tipo de ferramental que deve ser agregado ao programa é uma questão bastante recorrente. Normalmente ela aparece interligada com o que se espera do aplicativo. Discute-se se serão afinal necessárias ferramentas de busca e a dificuldade para implementá-las.

Mas a principal meta dos desenvolvedores seria dificultar a imposição de responsabilidade legal aos autores do conteúdo, aos que o obtém e aos que dão o suporte material para a sua publicação na rede.

Para esse fim foi proposto o uso de um sistema de servidores descentralizados. Qualquer pessoa pode disponibilizar um deles. E que o conteúdo desses servidores não seja controlável por esses operadores. Eles replicam os dados entre si baseados nas requisições feitas por seus clientes diretos. E descartam automaticamente as informações que não são procuradas a mais tempo.

Cada servidor detecta apenas os outros diretamente ligados a ele. Qualquer requisição de uma informação que esteja além deles é repassada aos servidores seguintes sem que se identifique sua fonte. Os protocolos Internet atuais funcionam de forma contrária. A informação e as requisições circulam com as marcas de sua origem e destino. Na rede Freenet cada servidor funciona como se apenas recebesse ou enviasse requisições diretamente de seus pares, mesmo que algumas requisições e respostas percorram muitos deles.

A tentativa é que essas técnicas dificultem atos como o de traçar a origem de uma informação. Ela sempre parece vir dos servidores mais próximos. Também que seja complicado descobrir a identidade daqueles que requisitaram determinados dados. Sempre parece que os pares diretos o fizeram. Não se pode responsabilizar quem disponibiliza um servidor pelo seu conteúdo visto que ele é colocado lá automaticamente. Também se torna pouco útil o impedimento judicial de um deles, ou mesmo algum ataque técnico. A informação, quase sempre, estará replicada em outros nós da rede.

Os próprios ataques são difíceis de realizar. Uma das formas ofensivas mais comuns na Internet é o excesso de requisições. Nesse caso, no sistema Freenet, os dados seriam apenas replicados por mais servidores. Além disso, o problema da ineficiência pelo excesso de tráfego do Gnutella fica reduzido. A informação tende sempre a ser replicada próximo de onde é mais requisitada, reduzindo muito o trânsito gerado pelo transporte e buscas (N.A.:16/9/00:).

É interessante notar que está sendo construído um conjunto de respostas funcionais para uma seqüência de desafios vindos de imposições judiciárias. Estão sendo desafiadas formas de controle social através de um esforço coletivo de modulação do ambiente tecnológico. Esse contexto parece sugerir a imagem de uma espécie de guerrilha. Um conflito travado pela contínua criação de novas ferramentas de informação, que tentam tornar inoperantes as formas de imposição de ordem presentes. Talvez ele se torne uma parte relevante no jogo de poderes e resistências do que chamamos de era da informação.

Considerações finais

Tentamos demarcar nesse texto algumas diferenças do ambiente computacional em relação ao padrão histórico das demais ferramentas presentes na cultura humana. Nosso enfoque principal deteve-se na discussão sobre a plasticidade e a capacidade de simulação. Destacamos a relação entre essas características e o múltiplo aspecto semiótico desse ambiente. De um lado ele vai sendo configurado como uma das principais ferramentas para engendrar, armazenar e fazer circular o simbólico. De outro, grande parte de sua arquitetura final é formada por cadeias de comando cujo aspecto simbólico procuramos destacar. Teríamos, então, símbolos processáveis pelo próprio ferramental a que servem de base. Eles dariam à arquitetura desse ferramental um importante diferencial, a auto-recursividade.

Esse aspecto estaria sendo potencializado pela altíssima replicabilidade típica do digital e o aumento progressivo das facilidades para sua circulação. Entre as possibilidades novas trazidas por ele estaria o fato de a evolução e a distribuição do ferramental de tratamento simbólico ganharem uma nova dimensão e velocidade. Nos interessou principalmente como isso parece ampliar o acesso à engenharia de novas ferramentas de comunicação, trazendo conseqüências que começamos a retratar.

Num certo sentido procuramos seguir aqui a sugestão epistemológica associada ao próprio universo do código aberto. Nele tem sido usual que as novas proposições sejam publicadas em um estágio ainda inicial. Supõe-se que a testagem, a depuração, o acabamento sejam feitos lado a lado com a comunidade interessada. Desejamos que esse artigo também seja encarado da mesma forma. Esperamos que ele possa ter trazido algumas proposições interessantes para o início de um diálogo.

Notas

- 64 1 Ao contrário do que se divulga com mais freqüência o primeiro computador do mundo não seria o ENIAC, de 45, mas o Z2 que Konrad Zuze construiu com relés eletromecânicos na Alemanha em 1940. Também é de sua autoria o primeiro a usar o código binário, o Z3 feito em 1941. (Cruz Filho;1995:148)

Bibliografia

- BEZROUKOV, Nikolai - Some vulnerabilities of the key ideas of the Cathedral and the Bazaar - site da Organização Softpanorama, <http://www.softpanorama.org>, 1999a
- BEZROUKOV, Nikolai - Open Source Development as a Special Case of Academic Research - site da Organização Softpanorama, <http://www.softpanorama.org>, 1999b
- CASTELS, Manuel - Era da Informação, A: Economia Sociedade e Cultural vol 1 - A Sociedade em Rede - Paz e Terra, São Paulo, 1999
- CLARKE, Ian e SANDENBERG, Oskar - Freenet: A Distributed Anonymous Information Storage And Retrieval System - Site da Zeropaid, <http://www.zeropaid.com/freenet/intro.shtml>, 18/9/00
- DENNET, Daniel C. - Tipos de Mentres - Rocco, Rio de Janeiro, 1997

- EDWARDS, Kasper - Ph. D. Outline - Paper Acadêmico - Universidade de Lingby, http://www.dtu.dk/its/ansat/ke/out_work.txt, 26/08/00
- FINNEMAN, Niels - Modernity Modernised - The Cultural Impact of Computerisation. publicado em: Computer Media and Communication: a Reader - Oxford University Press. New York, 2000
- GOLDBERG, Adele e KAY, Alan - Personal Dynamic Media. publicado em: Computer Media and Communication: a Reader - Oxford University Press, New York, 2000
- GURGEL, André - A Libertação do *Software* e a Odisséia Final. publicado em: Jornal "O Globo" - Caderno Informática Etc - , Rio de Janeiro, 1/1/2001
- HERMANN, Stefanie; HERTEL, Guido e NIEDNER, Sven - Linux Study First Results - Site da Universidade de Kiel. <http://www.psychologie.uni-kiel.de/linux-study/writeup.html>. 26/08/00
- JOHNSON, Steven - Interface Culture - Basic Books. São Francisco, 1997
- KAY, Alan - Computer *Software*. publicado em: Computer Media and Communication: a Reader - Oxford University Press. New York, 2000
- KERCKHOVE, Derrick - Pele da Cultura, A - Relógio D'Água Editores, Lisboa, 1995
- LEMOS, André - Imaginário da Cibercultura. O - Paper Acadêmico - UFBA - Salvador, 1999
- LEROY-GOURHAN, André - O Gesto e a Palavra: vol 2 - Memória e Ritmos - Edições 70, Lisboa, 1965
- LÉVY, Pierre - Cibercultura - Editora 34, Rio de Janeiro, 1999
- LÉVY, Pierre - Inteligência Coletiva, A - Por uma antropologia do ciberespço - Edições Loyola, São Paulo, 1998
- LÉVY, Pierre - O Que é o Virtual? - Editora 34, Rio de Janeiro, 1996
- LÉVY, Pierre - Tecnologias da Inteligência, As - O futuro do pensamento na era da informática - Editora 34, Rio de Janeiro, 1993

- LICKLIDER, J.C.R e TAYLOR, Robert - The Computer as a Communication Device. publicado em: Computer Media and Communication: a Reader - Oxford University Press, New York, 2000
- MACHADO, André - Plataforma Linux Ganha Espaço no País. publicado em: Jornal "O Globo" - Caderno Informática Etc - , Rio de Janeiro, 3/1/2000
- MACHADO, André - *Software* Livre não é Questão de Preço - Entrevista com Richrd Stallman (criador fsf). publicado em: Jornal "O Globo" - Caderno Informática Etc - , Rio de Janeiro, 1/05/2000
- MARTINO, L. C. - Contribuições para o estudo dos Meios de Comunicação - Paper Acadêmico apresentado no INTERCOM 2000, Manaus, 4/9/2000
- MAYER, Paul - Epilogue - Computer Media Studies: A Emerging Field. publicado em: Computer Media and Communication: a Reader - Oxford University Press, New York, 2000
- MURRAY, Janet - Hamlet on the holodek: the future of narrative in cyberspace - The Free Press, New York, 1997
- Não Assinado - Arquivo da lista de discussão sobre tecnologia do projeto Freenet - Site do projeto Freenet, freenet-tech.pre0.gz, 14/9/00
- Não Assinado - Freenet Features - Site do projeto Freenet, <http://freenet.sourceforge.net/index.php?page=features.htm>, 16/9/00
- Não Assinado - IBM Knowledge Management – site internacional da IBM, <http://www-4.ibm.com/software/data/knowledge/>, 22/10/00
- Não Assinado - Kanisa Inc. Knowledge Management – site da empresa Kanisa, <http://www.kanisa.com/index.html>, 27/11/00
- Não Assinado - KDE News Page - Site dos Desenvolvedores da Interface KDE, http://www.kde.org/news/news_1999_03.html, 25/3/2000
- Não Assinado - Primus Knowledge Solutions – site da empresa Primus, <http://www.primus.com/>, 25/11/00
- Não Assinado - *Software* Livre: Uma alternativa democrática – site da Rede de Informações para o Terceiro Setor <http://notitia.rits.org.br/pub/newstorm.notitia.apresentacao>

- .ServletDeSecao?codigoDaSecao=3&dataDoJornal=atual. 12/2/2001
- NONAKA, Ikujiro - Empresa Criadora de Conhecimento, A. publicado em: Como as Organizações Aprendem - Futura. São Paulo, 1997
- O'REILLY, Tim - Ten Myths about Open Source Software - Site da O'Reilly and Associates Inc, <http://www.opensource.oreilly.com>, 1999
- PEDRO, Rosa M. L. - Cognição do Híbrido. publicado em: Comunicação e Sistemas de Pensamento - Contemporaneidade e Novas Tecnologias - Livraria Sette Letras, Rio de Janeiro, 1996
- PINKER, Steven - Como a mente funciona - Companhia das letras, São Paulo, 1998
- RAYMOND, Eric S. - Open Source - Site da Open Source Initiative. www.opensource.org, 1999a
- RAYMOND, Eric S. - Brief History of Hackedom, A - Site da Open Source Initiative, www.opensource.org, 1999b
- RAYMOND, Eric S. - The Catedral and the Bazaar - site de Eric Raymond, <http://www.geocities.com/CollegePark/Union/3590/pt-cathedral-bazaar.html>, 1998
- RIDEAU, François-Rene – About ESR's Articles - site do The Tunes Project from a Reflectiving System. http://www.tunes.org/~fare/articles/about_esr.html, 1999
- ROSNAY, Joël - Homem Simbiótico, O - Vozes. Petrópolis, 1997
- STALMAN, Richard - The GNU Manifesto - Site da Free Software Foundation, <http://www.gnu.org/gnu/manifesto.html>, 7/9/00
- STEWART, Thomas - Capital Intelectual - Campus, Rio de Janeiro, 1998
- TURNER, Michael – When A Bazaar Is No Longer A Bazaar - Remarque.org - <http://www.remarque.org/~turner/bazaar-not.html>, 1999
- ZAWINSKY, Jamie – Resignation and Postmortem - site próprio – <http://www.jwz.org/gruntle/nomo.html>, 1999

Palavras-chave:

1. projeto Freenet
2. código aberto
3. software Gnutella
4. Napster
5. novas tecnologias