



o próprio e o comum propriedade e liberdade na produção digital ¹

Fábio Malini ²

Universidade Federal do Rio de Janeiro
Escola de Comunicação
Doutorando em Comunicação e Cultura

Resumo

Partindo do referencial teórico de Antonio Negri, este trabalho visa mostrar como a cooperação em rede vem constituindo novas formas de propriedade intelectual, baseadas na lógica de difusão e socialização pública do conhecimento. Trata-se de evidenciar tal afirmação a partir da análise dos movimentos de software livre, particularmente, como se caracteriza os bens digitais produzidos a partir do copyleft, licença de caráter público que permite a qualquer usuário um acesso irrestrito aos códigos-fonte dos programas (o segredo de fabricação), liberando modificações em softwares produzidos por algum programador (ou pares, como chamam).

Palavras-chave

Trabalho Imaterial; Propriedade Intelectual; Liberdade; Comunicação; Software Livre

Comunicação, Propriedade e Trabalho Imaterial

Na economia do século XXI, o trabalho fabril deixou de ser o modelo hegemônico a ser seguido pelas empresas capitalistas. O sistema fordista hoje se constitui muito mais como o “velho ainda não totalmente superado” do que o novo a ser alcançado. Para resistir às dinâmicas silenciosas e asubjetivas impostas pela racionalização *taylorista*, os trabalhadores constituíram, principalmente, a partir da década de 70, ações de rompimento dessa “forma de estar junto”. Foi uma ruptura ideológica do trabalho (que ainda está em curso): já que por anos a fio vivenciamos o predomínio da ideologia do emprego, da submissão à lógica do escravo. Então, no lugar do automatismo burro, a invenção. No lugar da repetição e do padrão (*standard*), a diferença e a customização. No lugar do trabalho silencioso, a comunicação.

A passagem para o pós-fordismo indica, antes de tudo, a instalação de novas relações entre a fábrica e o território, entre as forças de trabalho e a sociedade, entre os

¹ Trabalho apresentado ao NP 08 - Tecnologias da Informação e da Comunicação, do V Encontro dos Núcleos de Pesquisa da Intercom.

² Jornalista, Mestre em Ciência da Informação. Pesquisador do CyberIdea/ECO-UFRJ e do Laboratório de Território e Comunicação (LABTEC/UFRJ). Email: fabiomalini@uol.com.br



serviços e os usuários. O paradigma fundamental do pós-fordismo como modo de produção largamente socializado, baseado portanto sobre a comunicação social (esta é que alimenta a inovação, as tecnologias da informação e a chamada economia do conhecimento) de atores flexíveis e móveis, é o do trabalho imaterial. (COCCO, MALINI, 2003). Por trabalho imaterial entende-se aqui como aquele que “cria bens imateriais, como o conhecimento, a informação, a comunicação, uma relação ou uma resposta emocional” (NEGRI, HARDT, 2005, p.136).

Desde sempre o conhecimento, a ciência, a inovação, foram incorporados no desenvolvimento dos produtos, contudo, a novidade é que tais dimensões cognitivas não estão mais fixadas e presas no produto. Ganham autonomia do trabalho e do capital. “O conhecimento é, agora, ao mesmo tempo um recurso e um produto, desincorporado de qualquer recurso e de qualquer produto” (CORSANI in GALVÃO et al, 2003, p.26).

A emergência do trabalho imaterial determina, portanto, uma transformação radical dos papéis, das relações e das hierarquias que caracterizam e estruturam o sistema industrial e sistema de propriedade intelectual. Pois é a tradicional separação entre produção (tempo de trabalho) e reprodução (tempo de lazer) que entra em crise. O paradigma pós-fordista define-se como “paradigma social” exatamente porque o novo modo de produção integra estes dois momentos e faz com que a circulação e a comunicação se tornem imediatamente produtivas. O trabalho imaterial se constitui como a interface entre esses dois momentos. É uma atividade que institui o material simbólico de uma mercadoria. Passamos a adquirir produtos e mercadorias que contenham subjetividades. Produzir mercadoria é fundamentalmente produzir sentido e significados, que são materializados em forma de afeto e estilo de vida. Não compramos sabonete, mas beleza; não compramos jornal, mas credibilidade; não compramos um celular, mas conexões; não compramos um serviço de saúde, mas a manutenção da vida.

NEGRI & HARDT (2005) chamam a atenção que, antes, na fase de produção material, os produtos criavam os *meios de vida social*. Os carros, as roupas e os alimentos, por exemplo, eram os mediadores para fazer com que a vida se realizasse: deslocar-se, vestir-se e alimentar-se. Hoje, tais produtos criam a própria vida social. Para viver em um grupo é preciso ter estilos, gostos, afetos, formas de vida, que se encontra em gôndulas de supermercados ou em *shopping centers*. Os produtos contemporâneos são verdadeiros meios de comunicação: possuem mensagens, permitem a comunicabilidade (e até a identidade) entre os homens. Mais que consumir, precisa-se



reconhecer, ou melhor, decodificar as estratégias de discursivas presentes nos bens imateriais.

Quando o trabalho se transforma em trabalho imaterial e o trabalho imaterial é reconhecido como base fundamental da produção, este processo não envolve somente a produção, mas a forma inteira do ciclo “reprodução-consumo”: o trabalho imaterial não se reproduz (e não reproduz a sociedade) na forma de exploração, mas na forma de reprodução da subjetividade. (NEGRI & LAZZARATO, 2000, p.11).

Essa dimensão imaterial torna a produção de bens resultada de conhecimentos adquiridos em redes diversas. Penso no sujeito que precisa produzir um site, um software ou um projeto. Todos começam a produzir a partir do produzido: resgatam relações com outros sujeitos, obtêm informações existentes, solicitam ajuda aos colegas de trabalho, enfim, mergulham em relações e dados obtidos no cotidiano de suas vidas (o que engloba o trabalho). Buscam ativar uma memória. Essa circulação de conhecimento é portanto uma condição *sine qua non* da criação, da inventividade.

Por isso que o paradoxo do sistema pós-fordista é que, na era do conhecimento, a valorização do capital passa por um jogo de constituição do tempo que os economistas definem como o fato da oposição entre **aceleração da difusão e desaceleração da socialização dos produtos do trabalho cognitivo**. Por um lado, o capital pretende estabelecer o domínio do futuro, de um tempo constituído pela aceleração da difusão (que aumenta o valor). Pelo outro, o capital precisa enfrentar a potência do devir, desacelerar e fechar o tempo constituinte da socialização do trabalho vivo (que diminui o valor). (COCCO, MALINI, 2003)

A equação capitalista entre uma difusão cada vez mais rápida em face de uma socialização que deveria ser cada vez mais lenta se faz a custos incalculáveis. Os produtos do trabalho cognitivo (ou imaterial) não precisam da relação de capital para ser produzidos e não pertencem mais ao capital, pois eles coincidem com as próprias relações sociais de cooperação. A dimensão privada da riqueza como valor se sustenta na base da afirmação abstrata e arbitrária do direito de propriedade. Esta “sustentação” privada da riqueza para manter a extração de valor acaba reduzindo dramaticamente o potencial produtivo de riqueza. (COCCO, MALINI, 2003)

Para se tornar valor, a riqueza deve ser difusa (pública), mas não pode ser socializada (comum). O público e o comum são mantidos separados. Esta é a força (o público, a difusão: os celulares para todo o mundo; a internet grátis; o baixíssimo custo



dos hardwares e ainda mais baixo dos softwares) e a fraqueza (a discriminação na base do poder de compra do uso real dos serviços; os obstáculos à proliferação criativa dos usos das informações e de suas ferramentas impostos pela lógica proprietária do copyright) da proposta de universalização das condições de ter direito aos direitos via mercado. (COCCO, MALINI, 2003)

É nessa tensão do paradoxo do capitalismo cognitivo que deve ser o ponto de partida para a análise das formas de propriedade intelectual, notadamente, como essas formas de propriedade estão associadas com as liberdades de produção e cooperação social. Tomemos a produção de software livre como criação de uma novo regime de propriedade que embasa novos processos de cooperação produtiva.

A propriedade baseada no comum: a produção do software livre

É graças a forma de cooperação produtiva de software livres que se constitui um novo direito público: o copyleft. Livres porque permitem a qualquer usuário um acesso irrestrito aos códigos-fonte dos programas (o segredo de fabricação), liberando modificações em softwares produzidos por algum programador (ou pares, como chamam). Aqueles que produzem ou utilizam esses programas intitulam-se como membros do Movimento do Software Livre.

“O software que você usa é o terreno que pisa no ciberespaço. Viver com um software que não pode ser alterado é como viver em uma casa onde todos os móveis estão fixados no solo. A falta de liberdade produz danos materiais na sociedade, porque os usuários são prisioneiros dos seus softwares. Manter secreto o código dos programas é uma traição aos princípios de divulgação que regem à Ciência” (STALLMAN, 2000, p.01)

Se os modelos de comunicação p2p geram conflitos predominantemente no terreno da circulação, o Movimento do Software Livre fomenta uma luta pela produção na circulação. De dupla forma: negando-a e afirmando-a, ao mesmo tempo. Nega quando insere o processo de valorização somente na circulação (o consumo é produção). Afirma ao introduzir novas formas de cooperação produtiva, à medida que um programador disponibiliza uma invenção para que um determinado grupo de trabalho online a aperfeiçoe. Essa liberdade inunda a produção e a circulação, tornando-as uma coisa só.

“O ‘livre’ do software livre se refere à liberdade, e não ao preço; especificamente, isso quer dizer que você tem a liberdade de estudar, mudar e redistribuir o software que utilizar. Essas liberdades permitem que cidadãos ajudem a si mesmos e uns aos outros, e dessa forma participem de uma



comunidade. O software proprietário tem um funcionamento interno secreto, e você está proibido de compartilhar o programa com seu vizinho. Não podemos estabelecer uma comunidade de liberdade na terra do software proprietário, pois o programador tem sempre um senhor” (STALLMAN, 2001, p.02)

O desejo que atravessa a formação de todas as lutas do Movimento do Software Livre consiste na busca pela liberdade de estudar, alterar e redistribuir qualquer software. Na verdade, esse desejo só se tornou evidente quando a nascente indústria de informática, nos anos 70, impôs a chamada lógica (bastante autoritária) do “concordo em não revelar”, o que impedia o acesso ao segredo de fabricação (o bloqueio dos códigos-fonte), por conseguinte, a renovação da inovação pelos usuários.

“Nós não chamávamos nosso software de "software livre" porque este termo ainda não existia; mas isso é o que era (no anos70). Quando alguém de outra universidade ou companhia queria portar e usar o programa, nós permitíamos isto com prazer. Se você visse alguém usando um programa interessante e pouco conhecido, sempre poderia pedir para ver o código-fonte, de forma que poderia lê-lo, mudá-lo, ou canibalizar certas partes do mesmo para fazer um novo programa” (Stallman, 2001, p.02)

A prática da troca de saberes como elemento nuclear do processo de produção de software sempre foi um *habitus* dos programadores. A ameaça a esse *habitus* se iniciou em 1971. Na verdade, a história do Movimento do Software Livre pode ser marcada por dois momentos. O primeiro a irrupção do desejo de resistência. O segundo pela criação de instrumentos e produtos capazes de criar uma alternativa ao modelo de produção do mundo da informática.

A primeira história é datada dos anos 70, ligada ao surgimento do sistema operacional (SO) Unix, lançado em 1971, pela AT&T. Esta pretendia acabar com a incompatibilidade de vários processadores de texto em suas máquinas. Era um incômodo ver que o mesmo editor de texto não funcionava nos diferentes tipos de computador localizados na empresa. A gigante da telefonia resolveu investir na criação de um sistema operacional que, instalado em diferentes máquinas, possibilitasse a inserção de todo tipo de programa. Em 74, o Unix foi portado⁽³⁾ com sucesso em vários modelos de computadores. “O sistema Unix se apresentou, portanto, como um agenciamento de ferramentas concebidas para se combinarem, de modo útil, umas com as outras e constituírem, assim, um ‘ambiente comum’.” (MOINEAU & PAPTATHEODOROU, 2000, p.108).

³ **Portar** – modificar um sistema operacional a fim de fazê-lo funcionar em diferentes tipos de plataforma material (hardware).



No mesmo ano (1974), a Universidade de Berkeley, na Califórnia, comprou da AT&T o acesso ao código-fonte do Unix. Este passou a ser objeto de pesquisa, sofrendo uma série de transformações advindas de estudantes e pesquisadores. Em 1977, surgiu a versão Unix da Berkeley Software Distribution (BSD). Por se objeto de usuários dos quatro cantos do mundo, a cada ano tinha uma nova atualização. Esta é a história da fundação das práticas de produção livre de software.

“Muito rapidamente, duas versões do Unix, e sobretudo duas culturas Unix, passaram a coexistir e, depois, pouco a pouco, a se opor uma à outra. A do Unix System Labs, da AT&T, por um lado, e a do Computer Systems Research Group (CSRG), de Berkeley, por outro; de um lado, os mecanismos de inovação de empresa, do outro, o processo cooperativo de inovação no interior de uma comunidade (mesmo restrita). O sucesso do Unix BSD trouxe logo um problema para a AT&T, que, em 1992, levou o CSRG aos tribunais, por ter ultrapassado os termos de licença concedida e publicado ‘os segredos industriais’. Este foi o começo de uma longa batalha jurídica, que só terminou em 1994, com a venda da AT&T e dos Unix Systems Labs à sociedade Novell, que tudo fez para chega a um acordo rápido. Contudo, o projeto BSD não sobreviveu à aventura judiciária” (MOINEAU & PAPTATHEODOROU, 2000, p.104)

A segunda interpretação sobre a gênese do Movimento do Software Livre se remete à figura do seu fundador, o programador norte-americano Richard Stallman, que sentiu na pele o nó da propriedade intelectual na indústria da computação. Tudo começou devido a uma impressora matricial. No início da década de 80, quando o Laboratório de Inteligência Artificial do MIT (EUA) – onde Stallman estava empregado - comprou um aparelho que deixou boquiabertos os programadores e funcionários que ali trabalhavam.

“Tratava-se de uma impressora capaz de funcionar a uma velocidade assombrosa, e o alvoroço inicial que provocou entre os programadores do laboratório se transformou em desilusão ao comprovar que aquela máquina não só podia imprimir em segundos, mas também mastigar algumas folhas que passavam pelas entranhas eletrônicas” (STALLMAN, 2000, p.03)

O então programador Richard Stallman propôs acabar com o problema e solicitou à empresa fabricante o código-fonte (a seqüência das instruções) do programa que fazia funcionar a impressora. O objetivo de Stallman era corrigí-lo e melhorá-lo para evitar os gastos com papel e as conseqüentes irritações dos usuários. Contudo, a empresa se negou a disponibilizar o código, alegando razões comerciais, e os programadores tiveram que seguir cruzando os dedos para que o diabólico aparelho não triturasse seu trabalho. Este fato fez com que Stallman perdesse a paciência e o empurrou a sonhar



com uma nova ordem informática, em que ninguém pudesse se apropriar do código de um programa e que qualquer pessoa pudesse usar e modificar os programas de forma que quisessem.

Ele se empenhou, então, na criação de um sistema operacional – por ser este o software crucial para que um computador pudesse funcionar. “Com um sistema operacional livre, nós poderíamos ter uma comunidade de hackers cooperando novamente – e convidar qualquer um para unir-se a nós” (STALLMAN, 1998, p.04).

Stallman decidiu produzir um sistema operacional baseado no Unix, pois assim o sistema seria portátil e os usuários do Unix poderiam migrar para ele com facilidade. Em 1984, o programador norte-americano concluiu seu feito, batizando-o de GNU, seguindo a tradição hacker⁽⁴⁾ de utilizar acrônimos, no caso, “GNU is Not Unix”. Essa caminhada em busca de um sistema operacional foi denominada de Projeto GNU, o marco fundador do Movimento do Software Livre.

“Um sistema operacional não é só um kernel⁽⁵⁾ [núcleo], executando basicamente outros programas. Nos anos setenta, todo sistema operacional merecedor de ser chamado assim incluía processadores de comandos, montadores, compiladores, interpretadores, depuradores, editores de texto, programas de correio, e muitos mais. ITS os teve, Multics os teve, VMS os teve e Unix os teve. O Sistema Operacional GNU também os teria” (STALLMAN, 1998, p.05).

Logo após o lançamento do sistema operacional GNU, Stallman o nomeou de “free software”, não no sentido de “free” como gratuito, mas como liberdade de: (1) executar um programa, com qualquer propósito; (2) modificar o programa e adaptar às necessidades do usuário (isto significa ter acesso ao código fonte); (3) redistribuir cópias, tanto grátis como com taxa; (4) distribuir versões modificadas do programa, de tal modo que a comunidade possa beneficiar-se com as melhorias.

Como na ocasião estava desempregado, pois havia pedido demissão do MIT (com medo deste se manifestar, no futuro, proprietário do seu invento) e como algumas pessoas queriam utilizar o GNU, Stallman cobrou US\$ 150,00 pela distribuição pelo correio dos códigos-fonte. Isto era setembro de 1984. Vendo a ocorrência da massificação do seu invento, por meio de uma distribuição livre, o programador temeu

⁴ Para a comunicação de software livre, **hacker** é sinônimo de pessoas com grande habilidade em informática, capazes de romper com a lógica proprietária da informação por meio de produtos e inovações tecnológicas. **Cracker**, ao contrário, é um agente fundamentalista da informação, invasores de sistemas, destruidores de páginas, etc.

⁵ **Kernel** é o núcleo de um sistema operacional que permite a outros programas se inserirem no sistema. É o código mais importante do SO, que raramente sofre profundas transformações.



que alguém se manifestasse unilateralmente dono da sua criação. Para conter isto, surgiu o desejo de criar um dispositivo que assegurasse um GNU livre, antes de ser popular. “A meta do GNU era dar liberdade aos usuários, não só ser popular. Então, deveríamos usar condições de distribuição que prevenissem que alguém se apropriasse do software GNU, tornando-se proprietário. O método que nós usamos para isto foi denominado de copyleft(6)” (Stallman, 1998, p.07).

O copyleft utiliza o mesmo modelo da lei dos direitos autorais, mas invertendo-o em termo do propósito habitual: em vez de ser um meio de privatizar o software, se torna um meio de mantê-lo livre.

“A idéia central do copyleft é que nós damos a qualquer um a permissão para executar o programa, copiar o programa, modificar o programa e redistribuir versões modificadas --mas nós não lhe damos permissão para somar restrições de sua propriedade. Deste modo, as liberdades cruciais que definem o "software livre" são garantidos a qualquer um que tenha uma cópia; eles tornam-se direitos inalienáveis. Para um copyleft efetivo, as versões modificadas também devem ser livres. Isto assegura que todo o trabalho baseado no nosso fique disponível para nossa comunidade, se for publicado. É o copyleft que impede que os empregadores digam: "não se pode compartilhar essas mudanças, porque nós queremos usá-las para fazer nossa versão proprietária do programa" (STALLMAN, 1998, p.2001).

Não há brecha no copyleft para a permissão de uma combinação entre um programa livre com outro proprietário. Se for usado um código livre para modificar um software não livre, a versão final – obrigatoriamente – tem que ser copyleft. Para implementar o copyleft dos softwares produzidos a partir do GNU, criou-se a licença GNU General Public License (GNU Licença de Domínio Público), um termo de compromisso que um desenvolvedor assume concordando com as normas para copiar, distribuir e modificar um programa ou trabalhos derivados dele.

Abrir um código-fonte e torná-lo livre consiste na pulsão de vida desse militante da informática. Antes de ser uma provocação ao copyright, o copyleft cria um verdadeiro axioma ao preservar a propriedade intelectual do produto negando a propriedade do produto intelectual. A licença GNU GPL cria, então, dois dispositivos, o de proteção e o de produção de direitos. O primeiro focado no autor. O segundo, na inovação.

⁶ “Em 1984 ou 1985, Don Hopkins (um companheiro muito imaginativo) enviou-me uma carta. No envelope, ele tinha escrito várias declarações divertidas, incluindo esta aqui: ‘**copyleft – all rights reversed**’ [*Copyleft – todos os direitos invertidos*]. Eu usei a palavra **copyleft** para denominar o conceito de distribuição que estava desenvolvendo aquele tempo” (Stallman, 1998, p.17).

“A proteção dos direitos envolve dois passos: (1) copyright do software e (2) licença que dá permissão legal para cópia, distribuição e—ou modificação dos softwares. Não há garantias para softwares de livre distribuição. Caso o software seja modificado por alguém e passado adiante, este software não mais refletirá o trabalho original do autor podendo portanto ser garantido pela Licença Pública GNU. Qualquer programa de livre distribuição é constantemente ameaçado pelas patentes de softwares. Buscamos evitar o perigo de que distribuidores destes programas obtenham patentes individuais, tornando-se seus donos efetivos. Para evitar isso, foram feitas declarações expressas de que qualquer solicitação de patente deve ser feita permitindo o uso por qualquer indivíduo, sem a necessidade de licença de uso” (LICENÇA PÚBLICA GNU, 1989, p.01).

Para preservar a autoria, a GNU GPL exige que o detentor dos direitos autorais insira um aviso no programa ou em trabalhos derivados dele, contendo a informação que a produção do software (ou sua atualização) foi realizada por ele sob as condições da licença GPL, que exige a negação da propriedade do produto criado pelo autor. Isto permite que um outro usuário adquira o produto e imponha alterações, que também serão disponibilizadas com um aviso de negação de garantias sobre a propriedade do produto. Contudo, ao ser distribuído, o nome de quem fez a alteração é sempre divulgado.

Pode-se modificar a cópia do Programa de qualquer forma que se deseje, ou ainda criar-se um trabalho baseado no Programa, e copiá-la e distribuir tais modificações sob os termos:

- Deve existir aviso em destaque de que os dados originais foram alterados nos arquivos e as datas das mudanças;
- Deve existir aviso de que o trabalho distribuído ou publicado é, de forma total ou em parte, derivado do Programa ou de alguma parte sua, e que pode ser licenciado totalmente sem custos para terceiros sob os termos desta Licença;
- Caso o programa modificado seja executado de forma interativa, é obrigatório, no início de sua execução, apresentar a informação de *copyright* e da ausência de garantias (ou de que a garantia ocorre por conta de terceiros), e que os usuários podem redistribuir o programa sob estas condições, indicando ao usuário como acessar esta licença na sua íntegra (LICENÇA PÚBLICA GNU, 1989, p.02).

Um importante ponto a destacar é a permissão da cobrança pelo ato físico de transferência ou gravação de cópias, e por garantias e suporte em troca de valores em dinheiro. O que importa é manter o caráter livre do movimento, quer dizer, desvendar o segredo de informação dos programas e divulgá-lo. O político, pois, concentra-se na produção (local do conflito social), o econômico na circulação (local onde se gera valor).

“Por uma questão de cooperação, incentivamos as pessoas a modificar e ampliar os programas que publicamos. Por uma questão de liberdade, determinamos os critérios da licença que as modificações das versões originais devem respeitar.



Critérios também são exigidos para as versões originais. Nós incentivamos uma cooperação de mão dupla rejeitando parasitas: quem desejar copiar partes de nosso software em seu programa deve nos deixar utilizar partes desse programa em nossos programas. Ninguém é forçado a entrar para nosso clube, mas aqueles que desejam participar devem nos oferecer a mesma cooperação que recebem de nós. Isso torna o sistema justo” (STALLMAN, 2001, p.01)

Com o aumento do interesse pelo uso do GNU e seus softwares derivados foi criado, em 1985, a Fundação Software Livre (*Free Software Foundation*), uma organização isenta de impostos para o desenvolvimento do software livre. É a fundação que administra os recursos financeiros e preserva o *copyleft*. Sob a proteção da FSF, estão importantes programas como o sistema operacional Linux, o software para servidor web Apache (usado por 80% dos servidores na Net), o mais importante programa de banco de dados, MySQL, e o controle do código do browser Navigator, da Netscape.

O *Free Software Foundation* aceita doações, mas a maioria de suas rendas sempre veio das vendas --de cópias de software livre, e outros serviços relacionados. Hoje vende CD-ROMs de código fonte, CD-ROMs com binaries, manuais bem impressos (tudo com liberdade para redistribuir e modificar), e distribuições de luxo (onde nós incorporamos uma coleção inteira de software para a plataforma de sua escolha) (STALLMAN, 1998, p.08)

A FSF é um ator que, ao mesmo tempo, armazena diferentes programas, como ainda catalisa e amplifica novas formas de cooperação produtivas. *E como se caracteriza a cooperação na produção de software livre? Quais são os seus produtos? Seus valores?* Antes da resposta, há duas premissas que orientam o processo de produção do software livre. Uma consiste na constatação que **a produção só é possível, ou mais especificamente, só ocorre no ambiente da circulação**. Há circulação sem produção, mas não o contrário. Isto porque a circulação é o locus da troca de idéias e de distribuição de “versão beta” entre os usuários — que irão “re-vertê-las” em um sólido software.

Uma outra premissa trata de caracterizar essa “co-operação” como um **dever fundamentado mais no a favor de (criar um novo sistema produtivo) do que ser contra a (Microsoft, por exemplo)**. Isto porque, ao “co-operarem”, os usuários produzem bens, ao mesmo tempo em que organizam as lutas sociais, solidificando o modelo público de produção, preconizado pela Fundação do Software Livre. Neste caso, o que motiva as pessoas a produzir de forma cooperativa não é um altruísmo ou um revanchismo, mas a possibilidade de criar (re-existir) um novo modelo econômico,



não apenas uma nova cultura (Raymond, 2001). Ao produzir riqueza, o Movimento do Software Livre desloca o discurso da recusa (*contra a*) para as instituições caudatárias da “propriedade da informação”, mas sem tornar estas um pólo antagônico, mas em mais um “*plus* produtivo”. Um exemplo: em outubro de 2001, a IBM liberou um investimento de US\$ 3 bilhões para o desenvolvimento e ampliação do GNU—Linux. A Netscape, em meados dos anos 90, liberou o código-fonte do seu browser, Napigator, para que os usuários pudessem aperfeiçoá-lo e, assim, concorrer com o Internet Explorer, da Microsoft. O caráter constituinte do Movimento do Software Livre é alternativo ao modelo proprietário, e não dialético.

O valor que garante isto é a liberdade. Ser livre, garante Stallman (2001), é poder construir incessantemente expressões, inovar naquilo que é conservado como correto. Só há “co-operação” se houver liberdade de expressão. O verbo *give away*, bastante expressado por esse movimento, resume bem o seu *ethos*: por um lado, significa **fornecer** (dinheiro, oportunidade, dados), por um outro lado, é sinônimo de **revelar** (segredo, informação etc).

Duas características ainda são fundamentais para entender a “co-operação” produtiva do software livre. A primeira é a recusa da competição como valor produtivo, pois competir significa o predomínio da lógica de um só vencedor, monopolizando duplamente o conhecimento do produto e o produto do conhecimento. E nem sempre o vencedor possui mais competência ou habilidade técnica, e sim, capacidade de apreender (ou prender) a capacidade cognitiva de outros.

O paradigma da competição é uma corrida: recompensando o vencedor, nós encorajamos todos a correr mais rápido. Quando o capitalismo realmente funciona deste modo, ele faz um bom trabalho; mas os defensores estão errados em assumir que as coisas sempre funcionam desta forma. Se os corredores se esquecem do porquê da recompensa ser oferecida e buscarem vencer, não importa como, eles podem encontrar outras estratégias -- como, por exemplo, atacar os outros corredores. Se os corredores se envolverem em uma luta corpo-a-corpo, todos eles chegarão mais tarde. Software proprietário e secreto é o equivalente moral aos corredores em uma luta corpo-a-corpo. É triste dizer, mas o único juiz que nós conseguimos não parece se opor às lutas (*copyright*); ele somente as regula ("para cada 10 metros, você pode disparar um tiro"). Ele na verdade deveria encerrar as lutas, e penalizar os corredores que tentarem lutar (STALLMAN, 1985, p.07).

O paradigma da competição é estritamente ligado às empresas “pontocom’s”, simbolizada pela Microsoft. Segundo STALLMAN, esta utiliza uma estratégia denominada *abraçar e estender*. Ela começa um desenvolvimento com uma inovação



tecnológica produzida por outros, adicionam uma pequena informação adicional que é secreta, de modo que ninguém pode imitá-las. Em seguida, usa essa informação secreta de forma que apenas o software da Microsoft possa se comunicar com outro software Microsoft. Em alguns casos, isso torna difícil que se utilize um programa não-Microsoft quando outros com os quais se trabalha usam um programa Microsoft. Por isto, que o Movimento do Software Livre zela pelo *copyleft* – que impede a canibalização da inovação pelas empresas que compactuam com o paradigma da competição.

Nenhuma licença pode impedir a Microsoft de praticar *abraçar e estender* se ela estiver determinada a fazê-lo às nossas custas. Se eles escreverem seu programa do zero, e não utilizarem nada do nosso código, a licença sobre nosso código não os afetará. Mas uma reescritura total custa mais caro e é mais difícil, e mesmo a Microsoft não consegue fazer isto o tempo todo. Daí essa campanha deles para nos convencer a abandonar a licença que protege nossa comunidade, a licença que não os deixará dizer: ‘o que é seu é meu, e o que é meu é meu’. Eles querem que nós os deixemos pegar o que quiserem, sem nunca devolver nada. Eles querem que abandonemos nossas defesas (STALLMAN, 2000, p.03)

Fugindo do modelo competitivo, o Movimento do Software Livre se inspira no modelo de produção do conhecimento científico. Na verdade, o movimento nasceu dentro do ambiente acadêmico, em um laboratório do MIT. Foi uma resposta a forma de produção da ciência da computação feita pela nascente indústria da informática, que contratou os principais profissionais das universidades, aprisionando o conhecimento nos laboratórios empresariais de Pesquisa e Desenvolvimento.

Ao contrário disto, o processo de produção científica fornece um compartilhamento de informação e idéias. Na verdade, estas só ganham força se tiverem o reconhecimento dos pares. A produção de conhecimento não ocorre no isolamento. Por isto que cientistas e os programadores se assemelham tanto, pois o valor de cada um deles se concentra no “trocar com o outro”, não buscar lucro com suas invenções, mas publicá-las e compartilhá-las para todos beneficiar-se dela. Em vez de lucro, busca-se mais o reconhecimento. Esse regime em que a criação se dá de forma coletiva foi designado pelo hacker Eric Raymond como “modelo bazar”, em oposição ao modo de cooperação do “modelo catedral”.

Eu acreditava que os softwares mais importantes necessitavam ser construídos como as catedrais, habilmente criados com cuidado por mágicos ou pequenos grupos de magos trabalhando em esplêndido isolamento, com nenhum beta para ser liberado antes do tempo. O estilo de Linus Torvalds⁽⁷⁾ de desenvolvimento – libere cedo e freqüentemente, delegue tudo que você possa, esteja aberto ao

⁷ Criador do sistema operacional Linux, que complementou as lacunas do GNU.



ponto da promiscuidade – veio como uma surpresa. Nenhuma catedral calma é respeitosa, ao contrário, a comunidade Linux pareceu assemelhar-se a um grande e barulhento bazar” (RAYMOND, 1998, p.01)

Liberar cedo para evitar erros mais tarde é uma prática cotidiana desse movimento. Cada novo “trabalhador” inserido no processo de trabalho livre não elimina um outro e nem cria uma reserva. Isto caracterizaria a “lei de Linus”, que tem como formulação geral que todo problema será óbvio para alguém. “Alguém acha o problema e uma outra pessoa o entende. E eu acredito que promover esse encontro é o grande desafio”, afirma Linus Torvalds. Esse modelo baseia-se na idéia de “muitos olhos, poucos erros” é o que diferencia o bazar da catedral.

Na visão catedral de programação, erros e problemas de desenvolvimento são difíceis, insidiosos, um fenômeno profundo. Leva meses de exame minucioso por poucas pessoas dedicadas para desenvolver confiança de que você se livrou de todos eles. Por conseguinte os longos intervalos de liberação, e o inevitável desapontamento quando as liberações por tanto tempo esperadas não são perfeitas. Na visão bazar, por outro lado, você assume que erros são geralmente um fenômeno trivial -- ou, pelo menos, eles se tornam triviais muito rapidamente quando expostos para centenas de ávidos co-desenvolvedores triturando cada nova liberação. Conseqüentemente você libera freqüentemente para ter mais correções, e como um benéfico efeito colateral você tem menos a perder se um erro ocasional aparece (idem, p.08)

A comunidade software livre reforça esse *ethos* (“muitos olhos, poucos erros”) para mostrar uma saída econômica, que não só se manifesta na inovação constante no produto, mas também na diminuição dos custos do processo de produção.

“Depurar é paralelizável. Embora requeira que depuradores se comuniquem com algum desenvolvedor coordenador, não necessita de coordenação significativa entre depuradores. Assim não se cai vítima dos altos custos de produção que limita adicionar mais desenvolvedores. O custo total de manter um programa amplamente utilizado é 40% a mais que o custo de desenvolvê-lo. Mais usuários acham mais erros” (Raymond 1998, p.04).

O modelo bazar da economia da doação, entretanto, vive atualmente uma bifurcação. Uma fratura produzida pelo *Movimento Open Source*, que pretende manter toda a cultura do *free software*, mas dotando-a de uma veia mais comercial e empresarial.

O movimento Open Source tenta ganhar o suporte empresarial, apenas falando sobre os benefícios práticos em permitir que os usuários copiem e alterem o software. Eles nunca falam sobre a liberdade como benefício em si. Eu concordo com o que eles dizem, quanto a isso, porém eles não dizem o suficiente. A obtenção do suporte empresarial é



útil, mas ainda mais importante é a difusão entre os usuários da idéia de que a liberdade por si só tem seu valor. Para manter nossa liberdade, freqüentemente precisamos nos esforçar para mantê-la e resistir à tentação de desistir dela. As pessoas não farão isso por meros benefícios práticos. Mas farão, se acharem que algo mais importante está em jogo. Portanto, precisamos falar sobre liberdade. O movimento Open Source não faz isso. O movimento Free Software faz. Não acho que o movimento Open Source seja ruim ou que deve desaparecer. Mas o movimento Free Software também é necessário. (STALLMAN, 2001, p.02)

Referências bibliográficas

- CORSANI, A. Elemento de ruptura: a hipótese do capitalismo cognitivo. In: GALVÃO, A.P., SILVA, G., COCCO, G. **Capitalismo Cognitivo**. Rio de Janeiro: DP&A, 15-31
- COCCO, G, MALINI, Fábio. Circular para Produzir: novos mecanismos de socialização do conhecimento. **Revista Rio de Janeiro Trabalho e Sociedade**, Rio de Janeiro, p. 7-11, 2003
- MOINEAU, Laurent, PAPATHÉODORO, Aris. Cooperação e produção imaterial em software livres – elementos para um leitura política do fenômeno GNU/LINUX. **Revista Lugar Comum**. Rio de Janeiro: UFRJ, n.11, maio-agosto 2000.
- NEGRI, Antonio, LAZZARATO, Maurizio. **Trabalho Imaterial – formas de vida e produção de subjetividade**. Rio de Janeiro: DP&A, 1998
- NEGRI, A., HARDT, M. **Multitude: guerra y democracia en la era del Imperio**. Buenos Aires: Debate, 2005
- NEGRI, Antonio, HARDT, Michael. **Império**. São Paulo: Record, 2001
- RAYMOND, E. **A catedral e o bazar**. Acesso 12/06/2005: <http://www.geocities.com/CollegePark/Union/3590/pt-cathedral-bazaar.html?200512>
- STALLMANN, R. O manifesto GNU. Boston: Free Software Foundation, 1985
- _____. O projeto GNU. Boston: Free Software Foundation, 1998
- _____. A GNU e o American Way. Nova economia. URL recuperada: www.novaeconomia.inf.Brasil/marketinghacker/especial.htm, 2001
- _____. Por que o software deve ser livre? Documento recuperado da página: www.gnu.org, 1992
- _____. Liberdade ou poder? Documento recuperado na url: www.cipsga.org.Brasil/sections.php?op=viewarticle&artid=406, 2001
- _____. Da indignação à liberdade. Documento recuperado da URL: www.cipsga.org.Brasil/sections.php?op=viewarticle&artid=176