

Trabalho imaterial e produção de software no capitalismo cognitivo

Giuseppe Cocco*

Gilvan de Oliveira Vilarim**

Resumo O artigo discute conceituações sobre a hegemonia do trabalho imaterial, focalizando sua produção cooperativa e a mobilização de subjetividades no capitalismo cognitivo; a produção de software é utilizada para identificar e analisar as características do trabalho imaterial. É feito um retrospecto da criação de software das suas origens até o momento atual da computação em nuvem e do software como serviço. Ressaltam-se as limitações da abordagem industrial para o software e a valorização e resistência de um trabalho vivo capaz de prover inovação nas condições de produção imaterial.

Palavras-chave trabalho imaterial; capitalismo cognitivo; engenharia de software; computação em nuvem

Immaterial labour and software production in cognitive capitalism

Abstract The article discusses conceptualizations about the hegemony of immaterial labor, focusing on its cooperative production and mobilization of subjectivities within cognitive capitalism; software production is used to identify and analyze the characteristics of immaterial labor. A historical review of software creation is made, from its origins to the present time of cloud computing and software as a service. We emphasize the limitations of an industrial approach to software and the valorization and resistance of a living labor which is able to provide innovation in terms of immaterial production.

Keywords immaterial labor; cognitive capitalism; software engineering; cloud computing

Introdução

* Doutor em História Social, Professor Titular da Escola de Serviço Social da UFRJ e Professor do Programa de Pós-Graduação em Ciência da Informação (IBICT-UFRJ). Av. Pasteur, 250, Campus da Praia Vermelha, Urca, Rio de Janeiro. CEP: 22290-240. Email: beppo1977@gmail.com.

** Doutorando em Serviço Social - UFRJ. Professor titular do Centro Universitário Serra dos Órgãos (UNIFESO) Av. Alberto Torres, 111 - Sala 402 - Prédio Arthur Dalmaso. 25964-001 - Teresópolis - RJ. Email: gilvan.vilarim@gmail.com

A configuração do capitalismo contemporâneo tem apontado, para muitos autores, a passagem de um modelo de capitalismo industrial, com um espaço de produção facilmente delimitado e controlado, para um modelo de capitalismo cognitivo onde a produção, circulação e o consumo se misturam. Neste capitalismo, o caráter imaterial da produção se torna hegemônico e resgata a existência de um trabalho vivo, que pode produzir sem necessidade de meios de produção fornecidos pelo capital. A acumulação de valor está cada vez mais situada do lado do “imaterial”.

Este artigo apresenta e discute algumas conceituações relacionadas a esta hegemonia do trabalho imaterial, tendo como foco sua dimensão altamente socializada e cooperativa, fortalecida pela proliferação das redes, e a constituição de uma força de trabalho cujas subjetividades envolvem a própria produção e reprodução de formas e estilos de vida.

A análise da produção de *software* nos permite melhor apreender as características do trabalho imaterial no capitalismo cognitivo. Para isso apresentamos um breve retrospecto da evolução do *software*, desde a programação feita em máquinas mecânicas, passando pelo seu *decoupling* do *hardware* propiciado pela eletrônica, até o momento atual onde as redes de comunicação já permitem sua utilização e produção à distância por meio da chamada *cloud computing*, aproximando o *software* muito mais de um serviço prestado do que de um produto manufaturado.

Nestas mudanças ocorridas no processo de produção, disseminação e “uso” do *software*, são ressaltadas as limitações de uma abordagem ainda industrial para o controle do processo de produção. O trabalho vivo inerente às atividades de produção de *software* possui uma subjetividade e uma imanência que inovam continuamente as condições da comunicação, do trabalho e do consumo, o que torna impossível uma plena apropriação por parte do capital, incapaz de controlar o *general intellect* e a horizontalização na cooperação trazidas pelas redes.

Do capitalismo industrial para o capitalismo cognitivo

No capitalismo industrial, as métricas para os padrões de produção podiam ser facilmente definidas a partir do momento em que se estabelecia um espaço de produção bem delimitado (o ambiente de produção fabril, o chão-de-fábrica) atrelado a uma unidade de tempo, um tempo que em si podia ser identificável e destacado do mundo da vida. O chão-de-fábrica, sobretudo em sua fase taylorista, era assim caracterizado por uma nítida disposição espaço-temporal de tipo hierárquico. A fábrica estava separada da sociedade e, ao mesmo tempo, subordinava seus espaços reprodutivos segundo a racionalidade de suas funções (produtivas): o espaço de produção industrial era aquele, funcionalista e moderno, das grandes metrópoles onde se concentravam os pólos de industrialização maciça e massiva. O tempo de vida, por sua vez, era nitidamente partido segundo essa mesma lógica funcional: tempo de trabalho e tempo livre e/ou de lazer, ao passo que a extração de tempo de trabalho excedente era sistematizada e intensificada pela rígida separação entre trabalho intelectual de concepção e trabalho manual de execução de tarefas, cada vez mais simples. O tempo de trabalho era assim aquele cronológico, pela dupla sequência da execução serial das tarefas e da subordinação do tempo livre (de reprodução) às necessidades do tempo de trabalho (de produção). A extração de mais-valia chegava a seu auge enquanto compressão-redução do tempo de trabalho necessário, ao passo que o tempo de trabalho excedente não conseguia transformar coerentemente o “sentido” do trabalho

na esfera de um consumo tanto massificado quanto unidimensional. Com isto era mais fácil estabelecer o tempo de trabalho e mobilizar meios para sua disciplinarização e apropriação.

Hoje, contudo, uma nova forma de produção, de caráter essencialmente *imaterial*, já não permite necessariamente a aplicação das mesmas regras e métricas. Tem-se delineado uma produção que extrapola os limites dos espaços funcionais da organização, que exige no chão-de-fábrica a recomposição de trabalho manual e trabalho da mente, e algum tipo de cooperação e colaboração constantes por parte de quem produz. Ao mesmo tempo, ela compreende um tipo de trabalho não mais petrificado e restrito a um ambiente fabril, ou confunde até mesmo quem é produtor e quem é consumidor. Nesse sentido, o trabalho vivo reencontra uma nova centralidade. Como sintoma dessas mudanças, podemos apontar a evolução das regras de contabilidade das empresas, feitas anteriormente para registrar uma acumulação de capital material, que se revelam inadequadas onde o problema agora é acumular um capital imaterial que se torna hegemônico, uma acumulação que é visível em um tipo de capitalismo informacional ou cognitivo. Fala-se assim de externalidades (BOUTANG, 2007) e de ativos intangíveis; é possível falar também de uma reversão tendencial do peso relativo de capital fixo e capital variável.

A hegemonia do trabalho imaterial não significa que se queira generalizar a idéia da imaterialidade em toda a produção, e nem mesmo assumir que o imaterial é predominante sob o ponto de vista quantitativo no mundo do trabalho. O que pode ser afirmado é que o trabalho imaterial tem imposto uma *tendência* a outras formas de trabalho e à própria sociedade, de certo modo comparável à hegemonia do trabalho industrial há cerca de 150 anos, mesmo que concentrado naquela época em uma pequena parte do globo.

Conforme Hardt e Negri (2005), podemos mobilizar quatro níveis de transformação que indicam a centralidade do trabalho vivo. Em primeiro lugar, a tendência da migração do emprego para ocupações mais relacionadas à prestação de serviços, circulação e distribuição – um processo de *terciarização* com relação ao qual se fala de um trabalho que se apresenta como uma “relação de serviço” ou uma “prestação de serviço”. Segundo, o fato de outras formas de trabalho estarem absorvendo características de uma produção imaterial, ainda que qualitativamente, dentro do próprio chão-de-fábrica. Terceiro, o crescimento de formas de propriedade de caráter imaterial, amparadas por um suporte jurídico de controle de acesso e de licenciamento. E, por último, a disseminação das redes como a forma típica da produção imaterial por toda a sociedade.

Tal como é percebido hoje, o trabalho imaterial resgata e justifica a utilização do conceito de Marx de um “trabalho vivo”, uma faculdade humana que todos nós possuímos para constituir o mundo e criar vida social. É fato que o trabalho vivo pode ser apropriado pelo capital e reduzido a uma força de trabalho. Mas isso hoje acontece segundo modalidades diferentes, novas! Não mais pela separação do tempo de vida *vis-à-vis* ao tempo de trabalho e, com isso, pela subordinação do trabalho à relação salarial (à relação de produção). Hoje em dia, o capital investe e subordina a vida como um todo: por um lado isso se traduz em um controle capitalista do conjunto das atividades vitais, produtivas e reprodutivas, intelectuais e manuais, instrumentais e comunicativo-efetivas: isso também se traduz em precarização e desemprego. Pelo outro, o trabalho se mobiliza dentro das próprias redes sociais; isso se traduz na proliferação de formas de cooperação social produtiva que independem do comando capitalista. O que está ocorrendo hoje é a possibilidade do trabalho tornar-se produtivo *sem dependência* do fornecimento dos meios de produção por parte do capitalista: e isso porque os meios de produção correspondem às próprias redes sociais.

Este trabalho vivo não pode mais ser decomposto e fragmentado com o objetivo de torná-lo um trabalho simples, facilmente subsumido – uma dinâmica que atingiu seu ápice com o modo de produção fordista e que se propagou durante boa parte do século XX. Na verdade, o que há no

trabalho vivo é uma convergência de articulações entre informações, determinações, subjetividades, que perpassam as fronteiras entre trabalho material e imaterial. Estudos como os de Virno (2003) e Marazzi (2002) identificam a *linguagem* como a chave comum para compreensão destas novas formas de trabalho; em inevitáveis relações entre pessoas no mundo do trabalho/vida, são as competências linguístico-comunicativas que atuam como os verdadeiros meios de produção. Os meios de produção, aqui, não são reduzidos meramente a máquinas cristalizadoras de trabalho morto (capital fixo), mas correspondem a um capital variável (trabalho vivo) cujas dimensões são imediatamente sociais, tendencialmente comuns.

A mecânica da crise do *subprime* (de final de 2008) nos mostra com nitidez essa mudança. Por um lado, a crise pode ser qualificada como o fato da contradição clássica entre desenvolvimento das forças produtivas e as relações de produção. Pelo outro, as forças produtivas e as relações de produção são completamente novas. As forças produtivas dizem respeito a esse trabalho de tipo novo cuja mobilização depende da densidade e da qualidade de sua vida social que tem que ser mobilizada em sua completa multiplicidade (e não mais pela partição da vida em dois “mundos” hierarquizados: aquele da “vida” e aquele “instrumental” da fábrica) que toda uma literatura passou a chamar de capital humano, capital social, etc. As relações de produção (o capital), ao mesmo tempo em que precisam dessa vida social do trabalho, operam para controlá-la, reduzindo sua multiplicidade à fragmentação regulada pelo mercado: a relação de emprego é individualizada no âmbito da execução de “projetos” cuja duração ideal é instantânea; a proteção social e os direitos reduzidos a mercadorias às quais teremos acesso por meio de relações de crédito/débito. Por um lado, o “empregado” ideal é aquele que é “empregável”. Pelo outro, a proteção social é aquela que não existe e obriga todo o mundo a ser “empregável”. A expansão do mercado financeiro construiu a solução ilusória do descompasso entre a queda do poder de compra de um trabalho vivo não reconhecido (apenas é pago quando se encontra uma situação de emprego que, por sua vez, é substituída pela condição precária de “empregabilidade”) e o fato que a “qualidade” desse trabalho de tipo novo (imaterial, vivo, altamente socializado) depende da qualidade de sua inserção no tecido social, quer dizer da qualidade de “acesso” aos serviços de educação, saúde, transporte, moradia!

Mas a relação de débito-crédito só funcionou na medida em que a expansão financeira do crédito dava a impressão de que o circuito se fechava. A crise do *subprime* não foi apenas o estouro de uma bolha, mas também a explicitação ingovernável da impossível equação capitalista entre a nova qualidade do trabalho vivo e social (das forças produtivas) e uma relação de produção que só consegue explorá-lo *a posteriori* (depois que a cooperação já aconteceu) e pela privatização dos direitos (transformados em mercadorias às quais se tem acesso com base no mercado).

Conceituação e tipologia do trabalho imaterial

De um modo geral, o trabalho imaterial diz respeito à produção que não resulta em bens materiais ou duráveis, ou seja, “trabalho que produz um bem imaterial, como serviço, produto cultural, conhecimento ou comunicação” (HARDT e NEGRI, 2006, p. 311). Essa definição não opõe os bens imateriais aos materiais, já que o valor destes é condicionado pelo seu conteúdo imaterial. Não se trata de uma concepção idealista ou psicologizante do trabalho imaterial, mas da apreensão da hibridação entre material e imaterial, onde o trabalho material se torna “cada vez mais interno ao trabalho intelectual e imaterial” (NEGRI, 2003, p.102).

Para poder colocar em discussão tal noção e legitimá-la, é preciso ir além da distinção feita por Marx entre um trabalho produtivo, gerador de uma obra desligada do seu produtor, e um trabalho improdutivo, um trabalho-sem-obra parecido com a atividade do artista virtuoso. Virno (2008) lembra que Marx define como produtivo o trabalho intelectual que gera uma obra (como um quadro), mas quando o trabalho intelectual tem um produto que é inseparável do ato de produzi-lo, Marx o associa diretamente a um trabalho improdutivo: temos aqui o exemplo da execução virtuosa de um balé, mas também a prestação pessoal de tipo servil do mordomo e de uma empregada!

O que caracteriza o trabalho contemporâneo é ser de tipo virtuoso e também produtivo, parte integrante das atuais dinâmicas do capitalismo cognitivo. Ele é caracterizado por atividades de comunicação, cooperação, portanto muito mais um “agir” do que um “fazer”, práxis mais do que poiésis.

É fato que este trabalho virtuoso pode ser tipificado dentro de um espectro de amplitude da produção imaterial. Para tanto Hardt e Negri (2006) distinguem três tipos de trabalho imaterial. O primeiro está envolvido em uma produção industrial que foi informacionalizada, incorporando tecnologias de comunicação de um modo que transforma o próprio processo de produção. O segundo é o trabalho imaterial de tarefas analíticas e simbólicas, que se divide na manipulação inteligente e criativa, de um lado, e os trabalhos simbólicos de rotina, do outro. E a terceira espécie envolve a produção e a manipulação de afetos e requer contato humano (virtual ou real), bem como trabalho de tipo físico. O trabalho imaterial é o fato da recomposição da mente e da mão e coloca, no cerne o corpo, sua capacidade de afetar e ser afetado.

Nesse sentido, o trabalho imaterial diz respeito a uma produção biopolítica. Mais recentemente, Negri (2009, p.7) voltou a citar o trabalho biopolítico como algo em que “praticamente toda a vida é submetida à exploração do valor”. Trata-se de um trabalho que não cria apenas bens materiais, mas também relações e, em última instância, a própria vida social; é a produção de formas de vida por meio de formas de vida:

A produção material – por exemplo, a produção de automóveis, aparelhos de televisão, roupas e alimentos – cria os meios da vida social. As formas modernas de vida social não seriam possíveis sem esses bens. A produção imaterial, em contrapartida, inclusive a produção de idéias, imagens, conhecimentos, comunicação, cooperação e relações afetivas, tende a criar não os meios da vida social, mas a própria vida social. A produção imaterial é biopolítica. (HARDT; NEGRI, 2005, p.194)

Se a produção no trabalho imaterial implica algum tipo de cooperação, é preciso enfatizar que se está falando de uma cooperação diferente daquela do paradigma industrial. Podemos pensar a cooperação social em dois modos distintos: na *cooperação objetiva*, cada indivíduo faz coisas diversas e específicas que são combinadas pelo capitalista – uma cooperação transcendente típica do fordismo; no fordismo, então, a cooperação é imposta aos trabalhadores e organizada de fora, ela é, pois, o resultado da relação de produção e acaba atribuindo ao capital a legitimidade de ser a condição prévia do acontecimento produtivo. No capitalismo cognitivo, no entanto, prevalece a *cooperação subjetiva*, quando o trabalho individual consiste em desenvolver, afinar e intensificar a cooperação *per se* de forma imanente. É por isso que se pode perceber que na cooperação subjetiva a relação com os outros, horizontal, é um elemento básico e não complementar do

trabalho virtuoso. Aqui, a cooperação social produtiva acontece antes e independentemente da relação capitalista de produção. O capital perde sua legitimidade e deve reimpô-la politicamente.

Diferentemente do *general intellect* de Marx, concebido como a capacidade científica objetivada em máquinas (capital fixo), e que reduz o aspecto virtuoso à aplicação tecnológica, é preciso entender a existência do *general intellect* como uma “partitura” geral, uma faculdade social que permite a execução de um trabalho-sem-obra – daí o virtuosismo, sem nenhuma conotação de elitização do trabalho, e sim como algo que é comum a todos (VIRNO, 2008).

Essa cooperação horizontal acaba por questionar a noção de força de trabalho como capital variável, que só seria acionada pelo capital. Se o trabalho afetivo é parte do trabalho imaterial e constrói relacionamentos, o próprio trabalho tende a produzir diretamente os seus meios de interação, comunicação e produção. São cérebros e corpos, juntos em cada singularidade, que necessitam uns dos outros, em afetação, para produzir valor, e agora esses cérebros e corpos não precisam mais ser fornecidos pelo capital. Na recomposição social da mente e da mão o capital variável engloba o capital fixo, sendo que este último é a própria rede hibridizada de relações sociais e tecnológicas, bem nos moldes das redes de telefonia móvel de 3ª geração ou da Internet.

Subjetividade: produtividade e produtibilidade

O ciclo do trabalho imaterial é constituído *ex-ante* por uma força de trabalho social virtualmente autônoma, capaz de organizar o próprio trabalho e as próprias relações com a empresa. Portanto, nenhuma organização científica, nenhuma métrica industrial do trabalho – leia-se, taylorista-fordista – pode mais predeterminar, mensurar, padronizar esta capacidade e a capacidade produtiva social. O que existe é uma virtualidade dessa capacidade que embute as características da subjetividade produtiva (LAZZARATO; NEGRI, 2001; GORZ, 2005).

O trabalho pode ser pensado de maneira imanente – não sendo mais estabelecida no interior de uma relação dialética (capital x trabalho), a nova força de trabalho autônoma passa a ser não apenas antagonista mas, além disso, constituinte de uma realidade social diferente; ela é também uma alternativa política. Nisso não há nenhuma teleologia, nem determinismo, mas um novo tipo de conflito.

É claro que a subjetividade do trabalho, sendo constituinte, também está inserida em um processo social onde o sujeito é ao mesmo tempo produto e produtor. Dessa forma, como lembram Negri e Hardt (2004), a subjetividade deve ser definida, simultânea e igualmente, por sua produtividade e sua produtibilidade, pelas suas atitudes para produzir e ser produzida. É por isso que, sob a ótica do capitalismo cognitivo, não interessa mais uma submissão individual a uma organização do trabalho científica (COCCO, 2001), mas uma submissão dos níveis de socialização por meio das capacidades comunicativas do trabalhador. Indo além, a organização pós-fordista encontra também na produção de subjetividade uma forma de interação com os seus consumidores que se assemelha sempre mais a uma ação política (LAZZARATO; NEGRI, 2001). E esta ação política empresarial permite identificar a figura de um empreendedor/empresário político como cúmplice do novo modo de produção, o que torna ultrapassada a separação entre economia e política (COCCO, 1999). A crise da lei do valor implica um regime de exploração que investe politicamente os processos de subjetivação e dessubjetivação. Diante da produção biopolítica organiza-se o controle do biopoder, do poder sobre a vida, do governo da população, como disse Foucault.

Entretanto, o motor da produção de subjetividade encontra-se no interior de relações de poder que são atravessadas por um desejo de vida, uma resistência ao poder; a resistência, portanto, pode ser encarada como *esse* verdadeiro motor da produção de subjetividade. A luta em rede passa a ter outros valores:

Como no caso da produção econômica pós-fordista, uma característica da luta em rede da multidão é que ocorre no terreno biopolítico – em outras palavras, ela produz diretamente novas subjetividades e novas formas de vida. (...) Mais uma vez, a luta em rede, como a produção pós-fordista, não recorre da mesma maneira à disciplina: seus valores fundamentais são a criatividade, a comunicação e a cooperação auto-organizada. (HARDT; NEGRI, 2005, p. 121)

Isso significa que lutas e produção tendem a coincidir. Com a luta em tal terreno biopolítico, o espaço delimitado das instituições, incluídas aí as empresas, é derrubado ao ponto de haver uma lógica comum entre empresa e sociedade. Se no capitalismo industrial tínhamos a figura da empresa-fábrica que interiorizava a produção e suas técnicas na sociedade (a sociedade-fábrica), o que agora se observa é a existência da empresa-sociedade: a substância de valor situa-se na sociedade, na população, nos saberes implícitos, nos processos cognitivos, na cooperação social (MOULIER-BOUTANG, 2003). Poderíamos dizer, com Lazzarato (2006), que o capitalismo cognitivo se caracteriza não pelo seu modo de produção, mas pela “produção de mundos”, quer dizer, de formas de vida. Quanto mais vasta a diversidade de atores e mais densa a rede de fluxos, maior é a substância de valor, pois esse depende das excedências vitais.

Trabalho imaterial e produção de software

Com base nas conceituações apresentadas até o momento, podemos agora situar a discussão do trabalho imaterial em um contexto que consideramos relevante para a clarificação de conceitos e desmistificação de determinadas práticas: a crescente produção e circulação de *softwares* no capitalismo contemporâneo, nascida originalmente da evolução da mecânica e da ciência da computação.

A discussão sobre as condições de produção e difusão de *softwares* diz respeito a quatro pontos relevantes. Em primeiro lugar, no fato dos *softwares* terem se espalhado em toda a esfera do mundo da vida/trabalho, em tal pervasividade que transborda a sua própria utilização original em máquinas computadoras, invadindo atualmente uma série de outros dispositivos eletrônicos (celulares, televisores, rádios, tocadores de MP3 etc).

Em segundo lugar, há uma característica própria do *software* em compreender e manipular a digitalização das informações que atravessam várias atividades quotidianas, o que nos leva a atrelar o processo de digitalização/numerização à existência de *softwares* em condições de processar tais informações e facilitar a geração de conhecimento. Diversos equipamentos tornaram-se digitalizados porque seus dados analógicos (sons, por exemplo) são convertidos em códigos binários facilmente processáveis pelo *software*.

Terceiro, a produção de software é emblemática para a compreensão do trabalho imaterial; de fato, ela até mesmo atravessa as quatro provas da hegemonia deste tipo de trabalho, citadas anteriormente. Ela está inserida na tendência de migração de atividades para os serviços; inclui as características mais significativas das atividades imateriais (colaboração, atividades simbólicas, criação de bem imaterial, etc); gera discussões sobre como seus produtos podem ser protegidos e cerceados juridicamente; e, nos últimos tempos, tem se manifestado nas redes de cooperação, com uma sinergia entre atores acelerada por novas tecnologias de comunicação e de informação.

Por último, há uma miríade de pessoas envolvidas na produção de *software*. Em consonância com as características do capitalismo cognitivo, essa multidão engloba tanto desenvolvedores quanto usuários, tanto organizações com fins comerciais quanto grupos interessados em disseminar conhecimentos para a sociedade, como veremos adiante.

O *decoupling* do software

Tradicionalmente, a computação é estudada a partir de diversas ramificações de áreas de conhecimento que procuram desenvolver máquinas capazes de efetuar cálculos (cálculos). Trata-se de estudos e pesquisas modelados ao longo do tempo nos pólos do *hardware* e do *software*. Até as primeiras décadas do século XX, o *hardware* – a parte física das máquinas, ou, em uma tradução literal, as *ferragens* – era justamente o que caracterizava o funcionamento da máquina, cuja arquitetura mecânica (e depois eletromecânica) estabelecia também a aplicação (por vezes única) do equipamento. Uma máquina para cortar chapas era construída para cumprir esta finalidade, e nada além.

Posteriormente, a possibilidade de construção de máquinas capazes de serem *programadas* a executar ações diferentes, gerou a disseminação do conceito de *software* – a parte lógica, não tangível, responsável por fornecer instruções sobre o que o *hardware* deve executar. O tear automático de Jacquard e as primeiras calculadoras programáveis (que evoluíram para os computadores) reforçaram esta separação entre *hardware* e *software* (FONSECA FILHO, 1999). Nos seus primórdios, contudo, fornecer instruções diferentes implicava em modificar *fisicamente* determinadas partes da máquina, para que a mesma pudesse “entender” as novas orientações. Isto é, para programar era necessário conhecer bem o funcionamento da máquina e fazer ajustes em determinados componentes físicos que modificavam o seu comportamento e também os seus resultados. Podemos dizer que o *hardware* subordinava e determinava o *software*, mantinha sua lógica de funcionamento separada.

Durante a década de 1940, com a intensificação do uso das tecnologias de base eletrônica e depois microeletrônica, a programação das máquinas começou a desgarrar-se do *hardware*, na medida em que não era mais necessário modificar fisicamente as “ferragens” das máquinas; o *software* se desenvolveu segundo sua própria dinâmica. O suporte eletrônico e o rigor matemático para este *decoupling*, concomitantemente, fortaleceram a possibilidade de se construir os computadores, mais genéricos que as calculadoras. A digitalização de dados e instruções, e sua total conversão em sinais elétricos, permitiram estabelecer um paradigma de arquitetura de computadores – vigente até hoje – onde diferentes ordens são dadas às máquinas sem necessidade de manipulá-las fisicamente. Com as pesquisas do matemático John von Neumann, vislumbrou-se a possibilidade de uma máquina conter, numa só memória, dados e também programas numerizados e capazes de serem manipulados eletronicamente.

Atualmente, portanto, dados e programas são armazenados e processados em memórias computacionais eletrônicas voláteis, o que permite trabalhar com o *software* sem mexer em componentes físicos. Cabe ressaltar que, com a digitalização de dados e instruções e o crescimento do uso da microeletrônica em outros equipamentos (rádios, televisores, eletrodomésticos em geral), foi possível dispersar a polarização *hardware/software* para além dos computadores, ao passo que o *software* ia ocupando um papel decisivo em termos de funcionalidades e mercado.

No paradigma industrial, a especificidade inerente às tecnologias mecânicas – e das formas de divisão e de organização do trabalho que caracterizam seu uso – repousa sobre o grau de especialização da máquina e, portanto, sobre uma heterogeneidade das máquinas que se opõe a um trabalho homogêneo, resolvido em simples dispêndio energético, em esforço muscular. Foi, como dissemos, o taylorismo que levou essa tendência às suas últimas consequências. As novas tecnologias de informação e de comunicação se apóiam em uma dissociação entre a máquina e o seu programa, o que faz com que a máquina se torne mais “desespecializada” e homogênea, uma vez que não há mais uma especificidade mecânica que a torne dedicada a certas tarefas. Estamos diante de uma mudança da própria relação homem-máquina.

Logo, sob essa perspectiva, não é exagero afirmar que o *software* torna-se um elemento nevrálgico na constituição de um novo espaço de produção, um espaço que abrange indivíduos-usuários, computadores, programadores, organizações, trabalhadores e outros atores. É a lógica embutida no *software* que permite reconhecer a homogeneidade das máquinas atuais e caracterizá-lo como um bom exemplo de produção inerentemente imaterial: o que faz a “diferença” é a inteligência (*software*) embutida nas máquinas (*hardware*). É ali que se concentra o “valor”. Mas a produção desse suplemento de valor não corresponde mais à economia do tempo de trabalho.

Software, fordização e trabalho imaterial

Se se quer frisar a mudança de um paradigma de produção fordista para outro pós-fordista, a passagem do paradigma industrial para o capitalismo cognitivo não se reflete apenas no papel crescente e independente dos *softwares*, que afeta e desmaterializa o processo de valorização. É o próprio processo de criação dos *softwares* que foi afetado por tal deslocamento. Por um lado, o peso crescente do *software* diz respeito à importância do trabalho imaterial. Pelo outro, sua produção nos indica a dimensão cooperativa do trabalho vivo.

Numa definição já clássica, um *software* “abrange programas que executam em computadores de qualquer tamanho e arquitetura, documentos que incluem formas impressas e virtuais e dados que combinam números e texto, mas também incluem representações de informação em figuras, em vídeo e em áudio” (PRESSMAN, 2002, p.3). Tal definição permite desmistificar de vez a visão de um construto tecnicamente atrelado a uma máquina; na realidade, embora abranja obviamente os programas de computador – os conjuntos de instruções simbólicas que determinam o que a máquina irá fazer – o *software* também contempla documentos e representações que interessam diretamente às pessoas envolvidas na sua criação. Assim, embora um *software* seja criado para ser executado por máquinas, ele é manipulado essencialmente por seres humanos: seu desenvolvimento é o fato do trabalho vivo!

Podemos então identificar dois tipos de atividade simbólico-linguística na produção de *software*. De um lado, a atividade de comunicação realizada com as máquinas, feita por meio de linguagens artificiais que procuram, em última instância, dar ordens a elas. Do outro lado, a atividade comunicativo-linguística que é feita entre as pessoas que participam da produção. Um desenvolvedor/produzidor de *software* precisa saber mobilizar competências linguístico-comunicativas nos dois tipos de atividade. Já com isso podemos dizer que, na figura desse trabalho, nós encontramos a interpenetração ou a determinação recíproca e híbrida daquelas esferas (ou mundos) que Habermas pensa que vão se colonizando reciprocamente. Aqui a esfera da racionalidade instrumental é mais “colonizada” – diremos hibridizada – do que colonizadora pelo agir comunicativo do “mundo da vida”.

Em seus primórdios, nos anos 1940, uma das primeiras consequências do uso da eletrônica na programação dos computadores foi deslocar a produção de *software* para além do mero alcance dos engenheiros, matemáticos e físicos, até então os únicos capazes de criar as instruções de cada equipamento: saber programar implicava conhecer literalmente as “entranhas” dos computadores e modificar fisicamente certos componentes. O *hard* (material) continuava dominando o *soft* (imaterial). Uma vez que a programação tornou-se eletrônica – instruções que em última instância são sinais elétricos nas memórias dos computadores – outras figuras técnicas e profissionais puderam contribuir para a criação de *software* além do círculo inicial de atores que foi mencionado. Um salto significativo foi a criação, nos anos 1950, de linguagens de programação mais genéricas, conjuntos simbólicos formalmente bem definidos, capazes de codificar instruções algorítmicas para os computadores sem necessidade de conhecimentos prévios do *hardware*. Estas linguagens mais genéricas são conhecidas como linguagens de alto nível (em contraposição a um nível de programação mais “baixo”, eletrônico); nelas, o conjunto de comandos é muito mais próximo da linguagem humana e distante dos detalhes específicos de um determinado equipamento. Não é por acaso que o estudo de tais linguagens envolve pesquisadores de computação e de linguística. Nessas linguagens, o conjunto dos comandos é mais próximo da comunicação humana e se apresenta meramente como a base genérica do saber, de um saber geral, do *general intellect* que Marx antecipou.

Neste breve histórico podemos apreender o crescimento da necessidade de comunicação pessoa-pessoa independente do desenvolvimento tecnológico. Isto é, as facilidades trazidas pela evolução das ferramentas de programação permitiram que mais pessoas pudessem participar da produção de *software* e desenvolvessem cada vez mais interação. Na década de 1970, com a disseminação dos microcomputadores, muitos outros puderam produzir *software*, uma vez que se rompia a dependência de equipamentos caros para executar os programas que estavam, até então, disponíveis apenas no ambiente científico-militar.

Tecnicamente, nas fases iniciais, ainda era necessária – e assim permanece até hoje – uma etapa de “tradução” eletrônica de instruções para códigos mais simples, capazes de serem compreendidos pela máquina. Esta tradução é chamada de compilação ou interpretação do *software*, a depender como é realizada algorítmicamente. Entretanto, a evolução das ferramentas de programação, num nível cada vez mais alto, permitiu estimular uma maior capacidade de abstração e de lógica para solução de problemas entre as pessoas, num coletivo de produção que podemos caracterizar como virtuoso. A computação não foi o vetor do saber, mas de sua difusão. Essa difusão e “genericidade” constituíram sua potência.

Desde então, um interessante fluxo de efervescência entre programadores, por vezes anárquico, foi permitindo a disseminação de milhões de produtos de *software*. No seu início, a atividade de programação embutia um caráter subjetivo e difuso, genérico, em cada processo de codificação de instruções; programas eram feitos para serem entendidos por seus criadores e exprimiam diretamente a sua criatividade. Com o aumento da quantidade e da complexidade de aplicações,

novas competências passaram a ser requisitadas aos seus produtores. Além de saber programar, isto é, possuir raciocínio lógico e capacidades de abstração para a criação de instruções simbólicas coerentes, tornou-se gradativamente necessário aumentar o nível de compreensão e abstração de recortes do mundo da vida, de modo a se poder encontrar soluções para problemas mais complexos por via do *software*. Ainda assim, cérebros e corpos dos criadores de *software* agiam por meio de uma cooperação subjetiva, tal como definida anteriormente.

A partir dos anos 1960, entretanto, as grandes corporações começaram a articular um processo de disciplinarização da programação. Naquele momento, o mundo corporativo já despertava para o potencial de uso dos computadores, que poderiam ser utilizados para aumentar ganhos por meio de mecanismos de automação e processamento eletrônico de dados. A mera codificação de programas já não era suficiente para representar o processo de solução computacional dos problemas apresentados, e outros recursos começaram a se difundir: diagramas para representação de fluxos de trabalho, modelos de teste de *softwares*, abstração e tipificação de dados, etc.

Do ponto de vista das organizações, um tipo de trabalho que, no seu início, corria à margem do espaço de gerenciamento corporativo precisava ser controlado. Em uma mútua sintonia entre meio científico e corporativo, surgiram e se solidificaram tentativas de criação de “camadas” de abstração para a construção de *software*. É sabido que grandes corporações de informática detinham/detêm forte influência na consolidação de determinadas metodologias e padrões; observe-se, a título de exemplo, o fato de um *byte* (uma medida de memória computacional) possuir oito *bits* (oito sinais elétricos binários), uma padronização que foi difundida por computadores da IBM, um gigante do setor.

Se, por um lado, tais camadas tinham uma intenção de facilitar a troca de conhecimentos entre os produtores, por outro, elas foram uma tentativa de aumentar gradativamente um exercício de poder capitalista, sob a alegação de que o desenvolvimento de *software* estava se tornando extremamente custoso e não-gerenciável, isto é, incontrolável. Houve então uma tentativa de se passar da cooperação subjetiva para a objetiva, com um esforço do capital em fornecer os meios de produção – no caso, ferramentas de desenvolvimento, suporte metodológico, controles de produção, etc – tal como num modelo fabril. Para justificar o aumento do controle, a chamada “crise do *software*” foi apresentada como um horizonte sombrio para uma tendência crescente de projetos de *software* que, vistos sob uma lógica meramente econômica, estouravam orçamentos e prazos; tornavam-se “ineficientes”, de baixa qualidade e não atendiam às necessidades prévias; eram “ingerenciáveis” e difíceis de se fazer manutenção.

Como resultado, forjou-se aos poucos um disciplinamento para a produção de *software* amparado em diversas técnicas, métricas e metodologias. Isto foi a base para o que se tornou a chamada *engenharia de software*, definida como: (1) Aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção do software; isto é, a aplicação de engenharia ao software. (2) O estudo de abordagens como as de (1). (PRESSMAN, 2002, p. 18)

Por que julgamos ser possível identificar uma forma de exercício de poder com a evolução do chamado processo de desenvolvimento de *software*? Porque é um exercício de poder disciplinar que procura dessubjetivar o processo de criação de software ou da acumulação em prol da racionalidade. Na realidade, assistiu-se aos poucos a uma disseminação de características de um modelo de produção fordista ao processo de desenvolvimento de *software*, com o programador multifuncional de antigamente sendo substituído por uma equipe de especialistas de *software*, com cada um se concentrando numa parte da tecnologia necessária para produzir uma aplicação de *software* complexa (PRESSMAN, 2002).

Podemos identificar, então, duas tentativas de fordização ocorridas na criação de *software*. Numa primeira, do ponto de vista da organização do trabalho, o padrão da administração científica foi disseminado na produção. O desenvolvimento do *software* foi fragmentado em tarefas e funções. Uma verticalização funcional cristalizou a existência de analistas, projetistas e programadores (por vezes subdivididos) que se, em última instância, participavam da geração do produto, não necessariamente conseguiam visualizar a produção como um todo. É o momento de uma racionalização econômica que, como lembra Gorz (2003), precisava quantificar o trabalho de novo tipo como uma grandeza material que descartasse a individualidade e as motivações do trabalhador. O desenvolvedor, no nosso caso, precisava estar envolvido no processo produtivo despido de sua personalidade e de sua singularidade, servindo a fins que lhe são estranhos e indiferentes. Um programador, por exemplo, poderia receber uma formalização de especificações feitas por outra pessoa na produção, e simplesmente codificar/programar a parte que lhe cabe no projeto de um *software*, ou até mesmo codificar partes de projetos diferentes numa mesma jornada de trabalho.

A segunda tentativa diz respeito à “componentização” do próprio *software* como um construto, tal como ocorreu com o automóvel. Se um dispositivo físico podia ser construído com peças e componentes padronizados e intercambiáveis, por que não utilizar supostamente a mesma lógica com o *software*? Foi com base neste argumento que surgiram metodologias e técnicas (ou até mesmo uma capacidade de abstração diferente no pensamento) capazes de permitir a criação de partes de *software* preferencialmente reutilizáveis, padronizadas, independentes entre si.

Um dos exemplos mais difundidos atualmente é a chamada Programação Orientada a Objetos, onde toda a solução para um problema é montada como sendo um conjunto de interações entre *objetos* concebidos abstratamente pelo desenvolvedor (representando objetos físicos do mundo da vida, ou não). Ela não se preocupa mais em definir apenas ordens para as máquinas, mas sim estabelecer uma representação de uma solução como uma comunicação entre objetos construídos por um pensamento abstrato – as ordens às máquinas se tornam uma etapa secundária, mais automatizada e menos importante para quem produz. Como resultado, muitos desenvolvedores passaram a executar tarefas repetitivas (com o cérebro!), e perderam a compreensão do que está sendo produzido no todo. Houve assim um processo de taylorização do trabalho intelectual.

A culminância deste modelo de desenvolvimento, para os tempos atuais, foi a aplicação da engenharia de *software*, sob a forma de um conjunto de métricas e métodos que procuram a todo momento quantificar e controlar o trabalho realizado; é daí que deriva a proliferação de diversas representações gráficas e numéricas para registrar aspectos das soluções elaboradas, a quantificação de instruções contidas nos programas, controle e garantia de qualidade, técnicas de teste, etc. Embora com um deslocamento temporal em relação ao que já estava ocorrendo nos ambientes fabris de produtos materiais (*hardware*), o desenvolvimento de *software* passou então a ser objeto de forte pressão de um paradigma de produção fordista, cristalizando-se a partir do final dos anos 1960.

Contudo, essa fordização do *software* não se tornou hegemônica do ponto de vista do trabalho vivo, e o próprio desenvolvimento do trabalho em rede numa dinâmica horizontal, colaborativa e subjetiva, aponta para a ineficácia desse projeto capitalista. Se considerarmos um *software* como uma manifestação de conhecimentos, e que os conhecimentos não possuem as mesmas características das mercadorias (VILARIM; COCCO, 2008), quatro situações intrínsecas ao *software* demonstram tais limitações: 1) ele pode ser “desencarnado” de um suporte material, desapropriando-se (é o caso de *softwares* “baixados” da Internet e que nem ao menos necessitam de um meio físico para sua distribuição); 2) não são intercambiáveis, pois não há perda ou alienação ao passá-los adiante; 3) não são consumíveis, isto é, o seu ato de consumo não é

automaticamente a sua morte; e 4) são inteligíveis, uma vez que dependem dos fluxos decorrentes do intelecto, tanto individual como coletivo.

Na realidade, se a produção de *software* passou por uma forte tentativa de codificação do conhecimento nele embutido, por meio de diagramas, modelagens, representações algorítmicas, cronogramas de execução de projetos, dentre outros, esta codificação nunca se tornou plena. O trabalho vivo inerente às atividades de programação nunca se deixou apropriar no todo, um trabalho vivo que só tem aumentado em função da horizontalidade trazida pelas redes e pela possibilidade dos desenvolvedores produzirem com meios que escapam da dialética capital-trabalho.

O conceito de trabalho imaterial nos permite assim identificar a existência de uma situação paradoxal na produção de *software*, ao passo que esse paradoxo confirma o interesse de uma abordagem em termos de trabalho imaterial. Por um lado, ainda é estimulada essa aplicação do aparato da engenharia (industrial) de *software* no processo de produção, embora agora sejam discutidos modelos de desenvolvimento de caráter mais evolutivo, isto é, capazes de permitir a maior interação produtor-produtor e produtor-usuário (os primeiros modelos eram uma reprodução fiel de um processo de desenvolvimento em “linha de montagem”, iniciando pela fase de análise e seguindo sequencialmente até a manutenção do *software* entregue aos usuários/clientes).

Por outro lado, e daí o paradoxo, a criação de *software* nunca foi plenamente subordinada à relação capital-trabalho, pois a criatividade implica uma liberdade e ao mesmo tempo uma cooperação que não cabem nos métodos industriais. A título de exemplo, embora criticada, ainda há uma métrica de *software* que leva em consideração a quantidade de linhas de código (*loc – lines of code*) criadas pelos programadores dentro de um processo de desenvolvimento. Uma quantidade maior indicaria mais produtividade; ora, mas se um programador consegue construir soluções criativas e lógicas com menos *loc's*, a métrica se torna totalmente distorcida! As facilidades trazidas pela colaboração em redes de comunicação só aumentaram o processo de “fricção social” entre seus atores, que contribuem com seus cérebros e corpos para a elaboração de produtos cada vez mais criativos e mais distantes de uma mercadoria, no seu sentido clássico. Mesmo as organizações já reconhecem isto e a engenharia de *software* tenta captar esta mudança:

O software é um elemento de um sistema lógico e não de um sistema físico. Assim, o software tem características que são consideravelmente diferentes daquelas do hardware: (1) o software é desenvolvido, ou passa por um processo de engenharia, não é manufaturado no sentido clássico. (...) 2) software não “se desgasta” (...) 3) Apesar da indústria estar se movendo em direção a montagem baseada em componentes, a maior parte do software continua a ser construída sob encomenda. (PRESSMAN, 2002, p. 6-8).

O comentário do autor indica que, sendo construído “sob encomenda”, ou seja, customizado para a realidade de uma pessoa/organização/cliente, o *software* não pode ser encarado como um produto de manufatura clássico, aproximando-se muito mais da prestação de um serviço. De fato, os últimos tempos têm apontado para a tendência do “*software* como um serviço”, e não se pode negar a existência de tentativas de apropriação por parte do capital e de mercantilização de determinadas atividades.

Lazzarato (2002), ao resgatar os estudos de Gabriel Tarde, lembra que a memória humana tem como particularidade o fato de poder se exteriorizar sem se alienar; os conhecimentos que são codificados e disseminados nos *softwares* também permanecem na memória corpóreo-intelectual. Dessa forma, a possibilidade de apropriação do *software* sempre esbarra na imanência do *general intellect* obtida com a profusão de cérebros e corpos e dos conhecimentos neles embutidos.

Não é por acaso, também, que o movimento do *software* livre tenha aumentado nos últimos tempos. Seus participantes sustentam que o *software* deve ter seus códigos e documentações liberados para que outros possam consultá-los, modificá-los ou recombina-los. Diferentemente do *software* proprietário, o *software* livre segue uma lógica baseada na doação ou dádiva. O livre é uma máquina recursiva cujo uso produz o “livre”, sendo ele mesmo sempre recombinaível (WEISSBERG, 2003).

As atividades realizadas são então pautadas em uma ética de cooperação voluntária. A coordenação e a comparação entre os seus membros são baseadas na qualidade e no valor de uso da contribuição de cada um. Na ética dos *hackers*, da qual faz parte o *software* livre, encontramos a aplicação de uma concepção de riqueza que não é associada ao sentido econômico clássico, mas sim a uma riqueza da dádiva.

Os participantes do movimento do *software* livre sempre se posicionam com um pé no campo dos adversários, expostos às garras dos exploradores do *software* comercial. Porém, ao mesmo tempo em que vivem perto de relações de dinheiro e troca, afastam-se delas na execução de seu trabalho virtuoso (GORZ, 2005). Afinal, o *general intellect* termina por se manifestar plenamente quando há o êxodo, a possibilidade das trocas ocorridas entre seus participantes independentemente da estrutura capital-trabalho, o que explica o envolvimento de muitos, mesmo dividindo seu tempo com o trabalho em determinadas organizações. Estas trocas gratuitas e a produção em redes interativas justificam a constituição de uma economia da doação (*gift economy*) muito mais amparada no *copyleft* – com a retirada de barreiras à utilização, difusão e modificação de uma obra criativa – do que no *copyright*. As diversas licenças de *software* livre, como a GPL, ou as de conteúdos e informação abertos, como a Creative Commons, são fortes iniciativas baseadas na troca.

Software como serviço e *cloud computing*

Com a facilidade da transmissão de dados trazida pelas redes digitais, a discussão mais recente em torno da produção de *software* tem rondado um novo fenômeno: a chamada *cloud computing*, ou computação em nuvem. A computação em nuvem parte da idéia de que, uma vez conectados em rede e com velocidade de acesso compatível, computadores podem acessar dados e principalmente programas, disponíveis remotamente, de modo muito mais disperso e fluido. Desta forma, os computadores que ficam nas bordas da rede irão apenas solicitar a execução do *software* à distância, sem que o mesmo esteja gravado localmente no computador que vai executá-lo (KNORR; GRUMAN, 2008).

Do ponto de vista de quem usa um *software*, torna-se nebulosa a fronteira de identificação entre o que está guardado e executado em um nó da rede (no disco rígido ou *pendrive* de um computador local, por exemplo) e o que está disponível na nuvem metafórica, ou seja, em um espaço virtual onde não se conhece exatamente uma referência territorial precisa, que indique

onde estão os dados. Um dos primeiros exemplos deste tipo de modelo foi o *webmail*; nos últimos tempos, não há mais necessidade de “baixar” as mensagens localmente. Um *site* provê o acesso às mensagens e permite o seu gerenciamento, e as mensagens ficam armazenadas em algum servidor “na nuvem”. O *webmail* pode então ser consultado a partir de um computador doméstico, de uma *lan house*, do trabalho, etc. Expandindo este conceito, torna-se agora possível consultar e alterar arquivos pessoais, marcar *sites* favoritos, fazer anotações particulares a partir de qualquer local de acesso.

Sob o ponto de vista corporativo, então, a computação em nuvem vem sendo divulgada para a sociedade como uma infra-estrutura tecnológica capaz de prover um conjunto de aplicações *on-line* que trazem mobilidade para seus usuários e otimizam recursos computacionais. Pois bem, sendo fortalecida essa tendência, do ponto de vista de corporações como a Google, teremos a manifestação plena do *software* como um serviço, onde a interação entre prestador e cliente é constante. Ou, ainda, os próprios usuários se tornam partícipes da produção e a cristalização do *software* como produto concreto se esvai para dar lugar às versões *beta* eternas (uma versão *beta* de um *software* já pode ser utilizada pelos usuários, mas é considerada incompleta e passível de modificações).

A Google, por exemplo, tem cancelado várias de suas aplicações *on-line*, como o GMail, como versões *beta* sem indicativo de convergência para uma versão “final”, acabada. O *site* de redes sociais Orkut, também da Google e bastante utilizado no Brasil, tem sido modificado ao longo do tempo a partir de uma série de sugestões trazidas pelos seus próprios usuários, permitindo dizer que os usuários também se tornaram parte do seu processo de desenvolvimento. Hoje o Orkut, mesmo sendo parte dos serviços de uma empresa norte-americana como a Google, possui quase 50% dos usuários no Brasil; diversos recursos foram adicionados ao serviço a partir dessa comunidade e a Google precisou estabelecer uma base de desenvolvedores no país. É daí que se pode falar de uma relação de serviço e da existência de um usuário-produtor (JOLLIVET, 2003).

Entretanto, não se pode dizer que essa tendência seja colocada como uma evolução naturalizada do *software* e como mero fruto do avanço proporcionado pelo aumento da velocidade de transmissão de dados e das novas ferramentas de desenvolvimento para a *web*. Richard Stallman, um dos pioneiros do movimento do *software* livre, considera perigoso o fenômeno da computação em nuvem, uma vez que tranca os usuários de lado de fora dos seus próprios dados (JOHNSON, 2008). Sem resistência, usuários podem se tornar mais dependentes de grupos organizacionais que passariam a ditar suas tecnologias em uma via de mão única. Na verdade, exemplos como os do Orkut demonstram que é possível haver resistência.

Considerações finais

Neste texto procuramos apresentar uma forma de produção que difere daquela típica de um modelo de capitalismo industrial, justificando a mudança contemporânea para um tipo de capitalismo cognitivo, onde são porosas as bordas existentes entre tempo de trabalho e tempo de vida e há um resgate da existência de um trabalho vivo de caráter imaterial. A centralidade do trabalho imaterial aparece sob atividades mais relacionadas aos serviços, realizadas em redes de cooperação horizontalizadas, sem que necessariamente o capital seja o fornecedor dos meios de produção.

A força de trabalho, sendo agora autônoma e hegemônica, passa a ser constituinte de uma realidade social diferente, situada no plano da imanência e de uma produção biopolítica onde as subjetividades tanto produzem como podem ser produzidas. É sobre esta força de trabalho que nos propusemos a efetuar um recorte e analisar a relevância da produção de *softwares* e também como essa produção foi afetada pelas mudanças trazidas com o capitalismo cognitivo.

O retrospecto histórico do processo de desenvolvimento de *software* mostra que uma cooperação produtiva seminal deu lugar a uma tentativa de fordização de sua produção, num momento onde se supôs que seria possível aplicar regras, métricas e métodos do capitalismo industrial a um produto imaterial. Todavia, no caso do *software* entendido como um construto que embute uma série de conhecimentos que o afastam de uma mercadoria de manufatura, sua criação por uma força de trabalho imaterial nunca foi totalmente subsumida ao capital. A “fricção” ocorrida entre seus atores em redes de cooperação subjetiva traz uma desmedida que carrega dentro de si a criatividade e a liberdade fundamentais para a criação de *software*.

Movimentos mais recentes como os do software livre têm apontado para a dimensão ilusória de um controle “industrial” do trabalho “pós-industrial”. Além de se constituir em um potente terreno de luta e produção, esse movimento explicita o fato de que a acumulação capitalista na era do trabalho em rede e da centralidade do *software*, enquanto inteligência social, *general intellect*, passa por dinâmicas imediatamente políticas, aquelas de construção de novos *enclosures* pela aplicação de uma legislação de propriedade intelectual e das patentes que não reconhece a dimensão social, cooperativa e criativa do trabalho humano. Ao mesmo tempo, novas formas de controle e acumulação são elaboradas pelos segmentos mais avançados do capitalismo cognitivo. É o modelo Google.

A disseminação da computação em nuvem, ainda em seus primeiros passos, mas como estágio último da produção de software, precisa ser desnaturalizada do avanço tecnológico e servir para problematizar ainda mais as possibilidades de luta em espaços desterritorializados onde o trabalho vivo é que permite a inovação.

Artigo recebido em 20/06/2009 e aprovado em 18/07/2009.

Referências

BOUTANG, Yann Moulier. *Le capitalisme cognitif: la nouvelle grande transformation*. Paris: Éditions Amsterdam, 2007.

COCCO, Giuseppe. *Trabalho e cidadania: produção e direitos na era da globalização*. São Paulo: Cortez, 2001.

COCCO, Giuseppe. A nova qualidade do trabalho na era da informação. In: LASTRES, Helena M. M.; ALBAGLI, Sarita (orgs). *Informação e globalização na era do conhecimento*. Rio de Janeiro: Campus, 1999.

- FONSECA FILHO, Cléuzio. *História da computação – teoria e tecnologia*. São Paulo: LTr, 1999.
- GORZ, André. *O imaterial: conhecimento, valor e capital*. São Paulo: Annablume, 2005.
- _____, André. *Metamorfoses do trabalho*. São Paulo: Annablume, 2003.
- KNORR, Eric; GRUMAN, Galen. *What cloud computing really means*. Infoworld, 2008. Disponível em: http://www.infoworld.com/article/08/04/07/15FE-cloud-computing-reality_1.html Acesso em 24.mai.2009.
- HARDT, Michael; NEGRI, Antonio. *Império*. Rio de Janeiro: Record, 2006.
- _____. *Multidão*. Rio de Janeiro: Record, 2005.
- JOHNSON, Bobbie. *Cloud computing is a trap, warns GNU founder Richard Stallman*. The Guardian, 2008. Disponível em: <http://www.guardian.co.uk/technology/2008/sep/29/cloud.computing.richard.stallman> Acesso em 24.mai.2009.
- JOLLIVET, Pascal. NTIC e trabalho cooperativo reticular: do conhecimento socialmente incorporado à inovação sociotécnica. In: Cocco, Giuseppe; Galvão, Alexander Patez; Silva, Gerardo. *Capitalismo cognitivo: trabalho, redes e inovação*. Rio de Janeiro: DP&A, 2003.
- LAZZARATO, Maurizio. *As revoluções do capitalismo*. Rio de Janeiro: Civilização Brasileira, 2006.
- _____, Maurizio. *Puissances de l'invention: la psychologie économique de Gabriel Tarde contre l'économie politique*. Paris: Les empêcheurs de penser en rond, 2002.
- LAZZARATO, Maurizio; NEGRI, Antonio. *Trabalho imaterial: formas de vida e produção de subjetividade*. Rio de Janeiro: DP&A, 2001.
- MARAZZI, Christian. *Capital and language: from the new economy to the war economy*. Los Angeles: Semiotext(e), 2002.
- MOULIER-BOUTANG, Yann. O território e as políticas de controle do trabalho no capitalismo cognitivo. In: COCCO, Giuseppe; GALVÃO, Alexander Patez; SILVA, Gerardo. *Capitalismo cognitivo: trabalho, redes e inovação*. Rio de Janeiro: DP&A, 2003.
- NEGRI, Antonio. *Cinco lições sobre Império*. Rio de Janeiro: DP&A, 2003.
- _____, Antonio. Os conflitos do Império. In: *Jornal da UFRJ*. Ano 4, n. 44, maio, 2009.
- NEGRI, Antonio; HARDT, Michael. *O trabalho de Dionísio: para a crítica do Estado pós-moderno*. Juiz de Fora: UFJF-Pazulin, 2004.
- PRESSMAN, Roger S. *Engenharia de software*. Rio de Janeiro: McGraw-Hill, 5.ed., 2002.
- VILARIM, Gilvan de Oliveira; COCCO, Giuseppe. *Produção de conhecimentos por meio de conhecimentos: a outra produção no capitalismo cognitivo*. In: IX Encontro Nacional de Pesquisa em Ciência da Informação, São Paulo-SP, 2008.
- VIRNO, Paolo. *Virtuosismo e revolução: a ideia de "mundo" entre a experiência sensível e a esfera pública*. Rio de Janeiro: Civilização Brasileira, 2008.

_____, Paolo. *Gramática da multidão: para uma análise das formas de vida contemporâneas*. Santa Maria, 2003. Disponível em: http://es.wikipedia.org/wiki/Paolo_Virno. Acesso 24.mai.2009.

WEISSBERG, Jean-Louis. Entre produção e recepção: hipermediação, uma mutação dos saberes simbólicos. In: COCCO, Giuseppe; GALVÃO, Alexander Patez; SILVA, Gerardo. *Capitalismo cognitivo: trabalho, redes e inovação*. Rio de Janeiro: DP&A, 2003.