

■ Tecnologias, hacks e liberdade

.....Gilvan Vilarim

Declarações: um retrospecto

Nos últimos anos, no Brasil, o final do mês de abril tem se consolidado como o prazo máximo definido pelo governo para entrega da declaração de Imposto de Renda de pessoas físicas, sob risco de multa para os que fizerem a entrega atrasada. Tal prazo tem provocado um pico de envio de declarações nos dias que antecedem o prazo final e também uma dispersão das responsabilidades de transmissão eletrônica entre milhões de declarantes, uma atividade antes restrita aos órgãos da Receita Federal (RF).

Em um breve retrospecto das décadas mais recentes, podemos identificar três momentos relacionados à entrega das declarações e o seu processamento. No primeiro, formulários eram utilizados para se registrar as declarações por escrito (ou por datilografia), depois do que eram entregues nos locais de competência. Naquele momento, cabia à Receita digitar os dados e tratar de sua tabulação, fazendo uso interno de processamento eletrônico dos dados já neste momento.

Numa segunda etapa, a disseminação do uso dos microcomputadores levou o governo a desenvolver um software que permitisse a digitação e conferência de certos dados já em meio eletrônico. Este software, liberado gratuitamente, permitiu que os declarantes assumissem o controle de determinados cálculos mais complexos, que antes dependiam, em última instância, da atividade de contabilistas. As declarações puderam então ser gravadas em uma mídia física (em geral, disquetes) e entregues também nos locais de competência. Estes locais enviavam os dados a serem processados pela RF, que já passava a trabalhar com estes dados de input conforme padrões de registro definidos pelo software usado pelos declarantes.

Em um terceiro momento, a conectividade permitida pelas redes elevou o grau de automação do processo. Tendo agora algum tipo de acesso à Internet, o declarante pode enviar eletronicamente para o sistema do governo a sua declaração, sem necessidade de uso de qualquer mídia física. O software oferecido pelo governo dispõe de um módulo que reconhece no computador as declarações que foram montadas e oferece a opção de envio.

Mesmo que os três momentos apresentados atualmente se sobreponham – ainda é possível, por exemplo, entregar a declaração por escrito – é inegável o esforço da Receita em estimular o envio eletrônico, uma vez que diminui custos e interliga o software dos declarantes diretamente com as bases de dados de todo o país, agilizando o processamento e o intercruzamento com outras bases, como já está acontecendo.

Nossa intenção é utilizar o caso do software de envio de declarações como base para a discussão sobre a potência estimulada pelas tecnologias livres no ambiente de ciberespaço, do qual este software faz parte indiretamente, e reforçar a importância desta liberdade para a existência de ambientes de produção colaborativos.

Problemas, soluções e um relato

A título de registro e servindo como base para o levantamento de alguns pontos para discussão, faremos uso de um relato de fatos reais ocorridos em 30/04/2009, último dia para entrega da declaração do ano corrente. Este relato será entremeado por comentários relacionados a questões tecno-políticas dos fatos descritos, que serão discutidas adiante; os tempos citados foram ajustados em intervalos de cinco minutos.

Como situação inicial, considere-se a existência de dois computadores domésticos A e B, conectados em uma rede local, acessando a Internet. O computador A usa o sistema operacional Windows e possui o software da Receita instalado. O computador B usa Windows e Linux com o recurso de dual boot, isto é, permite a escolha de um ou outro ao ser ligado. O computador B também possui o software da Receita, no Windows.

01:15

O computador A possui uma declaração encerrada e já enviada dias antes. Encerrou-se a declaração do computador B, mas a conexão com a Internet está instável e decide-se enviar durante a manhã. As declarações estão gravadas no disco rígido de cada computador.

08:55

O computador A é ligado normalmente. Ao ligar o computador B, o Windows não se inicia e trava; tenta-se religar a máquina várias vezes e o problema permanece. Com isto não há como ter acesso ao software da Receita para fazer o envio; não havia backup da declaração. Não haven-

do solução, a declaração precisaria ser totalmente refeita, conferida e enviada no mesmo dia.

Um sistema operacional (SO), como Windows ou Linux, é um software vital para o funcionamento do computador, sendo executado em uma camada “inferior” a qualquer outro software; se há algum problema com o mesmo, não há como usar o computador com qualquer software atrelado a ele. Com dois sistemas operacionais instalados no mesmo computador, pelo menos um deles precisa estar em execução para tornar a máquina funcional.

09:05

O computador B é religado, mas desta vez optando-se pelo uso do Linux como sistema operacional a ser carregado. Com o Linux em execução, seria possível tentar acessar os arquivos da declaração já encerrada, e também tentar copiá-los para o computador A. O Linux parece funcionar, mas também trava ao abrir a tela principal; várias tentativas de iniciar o Linux também não dão certo. Com o travamento ocorrendo nos dois sistemas, isto é, independente de software, algum defeito no hardware parece ser a causa.

O Linux não pode, pelas vias normais, executar o programa da Receita feito para Windows, pois, grosso modo, os dois sistemas possuem recursos de programação diferentes, o que faz com que os programas de um não funcionem no outro⁵⁸.

Contudo, o Linux é capaz de compreender como os dados estão armazenados no espaço de uso do Windows: ele permite então executar operações com os arquivos gravados pelos softwares para Windows. Isto acontece porque um SO só entende os dados gravados em uma mídia se for capaz de aceitar o seu sistema de arquivos – o conjunto padronizado de especificações que define como estes dados estão gravados; o Linux compreende muitos sistemas de arquivos diferentes, numa escala de diversidade muito maior do que o Windows. A recíproca não é verdadeira: o Windows não necessariamente consegue compreender arquivos gravados por softwares de Linux.

⁵⁸ A execução de um programa em um sistema operacional diferente só é possível se este sistema possuir um software chamado *emulador*, isto é, capaz de fazer um programa ser executado como se estivesse no ambiente do seu sistema operacional original. Este recurso não é 100% compatível com todos os softwares.

09:10

Optou-se por ligar o computador B com uma versão de Linux em live CD. Desta vez o computador funcionou normalmente. O Linux do CD pôs o computador em execução e permitiu o acesso aos arquivos gravados no disco rígido do computador.

Um live CD permite carregar um sistema operacional para a memória principal de um computador a partir de um CD-ROM, ignorando eventuais sistemas operacionais já instalados no disco rígido do mesmo. A execução do live CD torna o computador totalmente funcional, embora com carregamento mais lento: o CD possui uma velocidade de leitura menor que a do disco rígido.

09:15

Abriu-se a pasta usada internamente pelo software da Receita. Há uma pasta com os dados de cada declaração digitada no computador B; percebe-se que cada declaração digitada gera uma subpasta com o número do CPF do declarante e um arquivo com o mesmo número. Um outro arquivo isolado com o nome “iddeclaracoes” é aberto; é um índice que contém dados referentes às declarações contidas nas subpastas. Os arquivos estão gravados no formato XML.

Este arquivo, bem como a subpasta da declaração que precisa ser enviada, são copiados para um pendrive a ser usado no computador A. A intenção é tentar manipular estes dados no software da Receita do outro computador, e com isto evitar a redigitação de todos os dados.

O software da Receita é gratuito, porém proprietário; não se tem acesso ao seu projeto nem à sua codificação. Mas, como descrito, um pouco de conhecimento técnico permite analisar quais padrões o software utiliza para gravar os dados dos declarantes. Arquivos em formato XML possuem uma padronização de conteúdo com acesso aberto, que pode ser manipulada.

Ao se pesquisar o software da Receita numa abordagem no estilo “caixa-preta” (observando o que o software gravou referente aos dados, mas sem entender necessariamente como), tem-se uma primeira etapa de possibilidade de interferência no comportamento deste software, sendo possível fazer tentativas de alteração na estruturação dos dados – ou nos dados em si – e com isso afetar a forma com que ele irá gerar informações.

09:25

No computador A, com Windows, também foi aberta a pasta usada internamente pelo software da Receita (que já continha uma declaração enviada) e ficou confirmado o mesmo padrão de gravação usado no outro computador. Arriscou-se então fazer duas modificações: primeiro, o arquivo de índices “iddeclaracoes” do computador A foi alterado manualmente, incluindo-se dados referentes à declaração que estava no computador B e que estava copiada no pendrive. Segundo, a subpasta da declaração do computador B, copiada para o pendrive, foi transferida para o mesmo local do computador A onde já havia uma declaração. Com este mix esperou-se “enganar” o software da Receita, fazendo o mesmo supor que as duas declarações estavam todo o tempo no mesmo computador A.

Um arquivo em formato XML pode ser alterado separadamente do software que o gerou. Ele é como um texto que possui padrões para definir como os dados serão manipulados por um software. Para arquivos mais simples, é possível entender o que estes padrões estão representando (por exemplo, o número de recibo da declaração, nome do declarante, ou o CPF do declarante).

09:40

Ao abrir o software da Receita no computador A, as duas declarações foram reconhecidas normalmente sem erros. A declaração vinda do computador B foi encerrada e enviada pela Internet; o problema foi resolvido.

Uma vez “enganado”, o software da Receita comportou-se como se as declarações tivessem sido geradas no mesmo computador, e tratou cada uma sem diferenciação da forma original de criação, o que se supunha como provável.

Cabe observar que, como a programação interna do software não pode ser vista, haveria ainda uma possibilidade de mais algum arquivo interno, ou outro meio algorítmico de verificação, confirmar se houve alguma modificação externa nos arquivos do software, como foi o caso. Entretanto, o reconhecimento, checagem e envio normal da declaração demonstraram que, pelo menos para a situação descrita, a solução funcionou.

Tecnologias livres e resistência

Com base nos fatos e comentários anteriores, gostaríamos de tecer algumas observações sobre determinados pontos, em especial quanto à abertura a intervenções e mudanças permitidas por determinadas tecnologias.

Ao apresentar as formas de enfrentamento do controle da propriedade no terreno da cibernética e da Internet, Hardt e Negri (2005) mencionam o movimento do código-fonte aberto como o seu exemplo mais radical. Para os autores, os defensores deste movimento alegam que não podem ver como um software de modelo proprietário funciona, e tampouco são capazes de identificar seus problemas ou modificá-lo para que funcione melhor. Os softwares deveriam então ter seu código-fonte liberado para que outros pudessem consultá-lo: “o código dos softwares é sempre um projeto colaborativo, e quanto mais pessoas puderem vê-lo e modificá-lo, melhor ele se tornará” (Hardt e Negri, 2005, p. 380).

Podemos aqui extrapolar a importância deste tipo de movimento para algo além, algo que representa as potencialidades do uso de tecnologias livres. Se é importante, como descrevem os autores, a possibilidade de consulta e modificação de um código-fonte liberado, capaz de expor as lógicas e modelagens construídas para resolver determinados problemas, melhor ainda é considerar também que determinadas tecnologias despertem uma potência em seus usuários capaz de torná-los tão produtores quanto os chamados técnicos informáticos. Para isso, essas tecnologias necessitam de liberdade de uso e de criação de novas combinações.

Lazzarato (2006), ao lembrar que hoje a riqueza pode ser pensada também sob uma lógica da abundância, reforça a importância desta liberdade no caso do software:

Se a economia é a ciência da otimização dos recursos escassos, e se hoje em dia a escassez não é mais uma condição natural, mas um produto do direito, parece-nos necessário lançar as bases de reflexão para pensar a riqueza a partir da lógica da abundância própria aos bens comuns.

A ambigüidade que encerra o termo inglês que distingue o software livre do software proprietário – free software – pode ser um bom ângulo de aproximação dessas questões, que ultrapassam amplamente a própria questão do software livre. O termo free software remete a dois conceitos diferentes: liberdade e gratuidade. As comunidades do software livre insistem no fato de que um software livre se define, antes de mais nada, pela liberdade, mais do que pela gratuidade. (...) O software livre coloca o usuário em uma situação potencial – ao demandar um engajamento específico por parte deste mesmo usuário – de liberdade e independência. Já o software proprietário, mesmo que tenha sido adquirido

gratuitamente, deixa o usuário em uma condição de dependência e passividade (Lazzarato, 2006, p.138).

Um software livre pode não ser necessariamente gratuito, mas seus códigos devem ter acesso liberado a quem quiser consultá-lo; um software gratuito não é também necessariamente livre, na medida em que o modelo proprietário dá margem para que seus produtos sejam oferecidos a preço zero. Se a questão é centrada na liberdade, o que se quer então é a promoção de uma cooperação fluida entre cérebros e a eliminação de um usuário meramente passivo.

É daí que se pode supor que a liberdade embutida em determinadas tecnologias permite confundir beneficentemente usuários e produtores/desenvolvedores, uma vez que, no caso dos softwares, minimiza-se uma hierarquia de saberes entre os dois e maximiza-se a natureza reticular das inovações, como aponta Jollivet:

Esta natureza coletiva, cooperativa e reticular do processo de inovação é particularmente pronunciada na dinâmica da indústria informática, através da importância, recentemente afirmada, das comunidades informais de informaticistas de software livre. (Jollivet, in Cocco et al, 2003, p. 89)

O formato XML (eXtensible Markup Language), citado na narrativa, já nasceu com um grau de liberdade nele embutido. Ele é uma tecnologia aberta por permitir que se criem estruturas para representar conteúdos capazes de serem compreendidos por softwares de diferentes plataformas; ele estabelece uma clivagem entre estética e conteúdo, concentrando-se neste último. Tendo uma estrutura de conteúdo padronizada por meio deste formato, diferentes softwares podem fazer uso da mesma para acessar determinados dados, e multiplica-se a possibilidade de interação “lingüística” entre softwares, sejam eles desenvolvidos em modelos livres ou proprietários, sejam eles para um sistema operacional ou outro, ou ainda programados em diferentes linguagens computacionais.

Um aspecto a se observar, portanto, é que tais tecnologias livres podem ser utilizadas transversalmente tanto em modelos de software livre como em software proprietário, o que foi justamente o caso do exemplo narrado. Os desenvolvedores do software da Receita, mesmo mantendo seu código-fonte fechado, optaram por representar determinados conteúdos e configurações no formato XML, o que foi crucial para que o problema citado na narrativa pudesse ser solucionado. No mundo dos softwares livres, esta opção é ainda mais comum, daí o fato de

tantos usuários e programadores conseguirem personalizar determinados módulos dos produtos livres⁵⁹.

É importante destacar que a liberdade embutida em tais tecnologias deve ser reforçada como potência e criação, como lembra Lazzarato:

A potência de cooperação do software livre tem menos a ver com a natureza cognitiva da atividade dos 'colaboradores' do que com a capacidade de abrir o espaço-tempo da invenção, ou, melhor dizendo, a proposição dos problemas e a criação de respostas se dá a despeito das lógicas da empresa ou do Estado, ao implicarem uma multiplicidade de sujeitos (Lazzarato, 2006, p. 125).

A invenção citada pelo autor se manifesta justamente quando há uma troca/circulação de conteúdos entre sistemas e pessoas facilitada pelo XML, e a troca cresce geometricamente com a multiplicidade de sujeitos envolvidos – a “xemelização” ao extremo, como já foi cunhada no Brasil (Fonseca, 2007).

É claro que nada impede que o modelo de software proprietário se beneficie de tal sinergia, para o bem ou para o mal. Um dos exemplos mais emblemáticos é o novo formato de arquivos adotado pela Microsoft para sua suíte de aplicativos Office, a partir da versão 2007. Os arquivos gravados neste formato, como, por exemplo, o DOCX, possuem na realidade uma estrutura interna que faz uso do XML. Contrapondo-o ao formato DOC fechado, a Microsoft então propala que o novo formato é aberto.

Entretanto, o uso do XML neste caso só serviu para amarrar mais os arquivos à Microsoft, visto que a empresa não seguiu propostas que procuraram estimular a existência de uma transparência total na definição da estrutura dos seus arquivos. Como resultado, discussões regulares têm apresentado os problemas de tal opção, como aquelas provocadas por Taurion (2009) e que refletem o jogo de poder corporativo sobre o uso das tecnologias livres.

A opção político-tecnológica feita pelo XML no software da RF, nos últimos anos, reflete o interesse em compartilhar os dados e oferecer, ainda que de forma básica, a possibilidade de compartilhamento dos dados por terceiros – incluindo usuários – o que terminou por acontecer com o “truque” da modificação descrito anteriormente. Berardi (2005) valoriza tal compartilhamento:

A concepção do shareware, de que Linux representa uma aplicação sistemática, pressupõe que a relação entre produtores e consumidores tenha as características

59 A existência de tantas versões do Linux – chamadas de *distribuições* – é um bom exemplo desta abertura para modificações.

da cooperação, da participação num mesmo ciclo produtivo, e não as características do mercado. (...) Com o *shareware*, estamos diante de um modelo de funcionamento do processo produtivo que começa a fugir concretamente às regras da economia capitalista e que prefigura a possibilidade de criar comunidades produtivas culturalmente e de expansão (Berardi, 2005, p. 103).⁶⁰

Um outro ponto a se mencionar diz respeito às possibilidades de resistência trazidas pelas tecnologias livres, da qual o Linux faz parte. O Linux nasceu a partir de pessoas inconformadas com o status quo consolidado dos sistemas operacionais nos anos 80, marcado pela hegemonia do Windows e de seus aplicativos derivados. O compartilhamento de códigos de programação e a possibilidade de construção de um “outro mundo” nos softwares, constituíram um movimento inicialmente promovido pela comunidade hacker:

A atividade do hacker repousa numa ética de cooperação voluntária, na qual cada um se compara aos outros pela qualidade e pelo valor de uso da sua contribuição para seu grupo, coordenando-se livremente com eles. Nada se produz com a finalidade de trocas comerciais. O valor de troca nunca é levado em conta; considera-se apenas o valor de uso que, na essência, não é mensurável (Gorz, 2005, p. 67).

Para além da correta lembrança de Gorz sobre a ética hacker, o que podemos afirmar é que as redes de cooperação e colaboração estimuladas por tecnologias livres, como as redes de software livre, dão margem para que nós mesmos sejamos os hackers. O próprio termo *hack* embute a idéia de uma modificação criativa, uma solução ou ajuste fora-do-padrão para determinada situação e que freqüentemente é realizada no mundo dos softwares, mas não se entende aqui que seja necessariamente determinado ou executado por uma suposta elite técnica.

Quando o Linux se tornou um artefato tecnológico factível de ser utilizado de forma amigável por pessoas comuns, liberando-as do pagamento de licenças de uso bastante caras, o enfrentamento do monopólio de facto do Windows gerou soluções criativas para apresentar aos usuários novas opções de uso.

60 Ressalte-se que o autor utiliza o termo *shareware* como concepção potente, o que concordamos, mas no Brasil este termo tem sido mais utilizado para representar uma forma de distribuição de software, em geral proprietário, onde o usuário pode copiar legalmente o produto, mas com limitações (de uso, de tempo, etc). O Winzip, um programa bastante conhecido e que serve para comprimir arquivos, é um bom exemplo: é proprietário, precisa ser comprado, mas pode ser distribuído “livremente”.

O live CD utilizado nos fatos narrados é um outro exemplo da liberdade trazida pelo software livre. Normalmente, um SO precisa ser instalado (isto é, copiado e configurado) dentro do disco rígido dos computadores, e é dali que será posto em funcionamento na memória principal. A não ser que sejam instalados em divisões (partições) diferentes do disco rígido, dois sistemas não podem ficar misturados no mesmo lugar.

Com a velocidade de acesso dos leitores de CD (e agora DVD) tornando-se razoável, o live CD surgiu no mundo do software livre como solução para apresentar e circular o Linux em diversos computadores sem afetar o que já estivesse instalado no disco rígido, potencializando a sua difusão e aumentando a sua base de usuários⁶¹. Além disso, determinadas ferramentas podem ser utilizadas pelos próprios usuários para que construam suas próprias versões recombinantes de um Linux⁶².

Outra opção de liberdade, constante no relato, é a diversidade de sistemas de arquivo (file systems) reconhecidos pelo Linux. Ele é capaz de compreender como os arquivos estão armazenados nas partições do Windows, e muitas outras. Como parte da solução para o problema relatado, foi tal diversidade de reconhecimento que permitiu a cópia dos arquivos do software da Receita para o outro computador.

Os dois sistemas Linux citados na narrativa serviram então para demonstrar que: 1) O Linux instalado na outra partição do disco rígido do computador B, não tendo funcionado, provou que o problema não era de software, e sim no hardware – um computador que só tivesse Windows não permitiria discernir tão rapidamente a localização do problema; 2) o Linux utilizado no live CD funcionou como um artefato que canaliza a resistência ao Windows como produto dominante; e 3) a diversidade de sistemas de arquivos reconhecidos pelo Linux foi decisiva para se recuperar o que estava gravado no computador com problemas.

O momento seguinte?

As discussões mais recentes sobre a maximização do uso das redes técnico-digitais dizem respeito ao fenômeno da chamada cloud computing, ou com-

61 Atualmente também há possibilidade de se utilizar *live pendrives* – o mesmo conceito de se ter um sistema operacional inteiro sendo executado de uma mídia móvel, só que agora em memórias *flash*.

62 No mundo do Linux, versões feitas por pessoas ou comunidades também são chamadas de distribuições. Hoje há ferramentas automatizadas para geração de distribuições personalizadas para usuários.

putação em nuvem. A idéia é que a conexão dos computadores em rede e o crescimento da velocidade de transmissão dos dados já permitem o acesso remoto tanto aos dados como, mais ainda, aos softwares, de um modo muito mais disperso e fluido. Assim, as máquinas disponíveis nas bordas da rede irão apenas solicitar a execução de programas de modo on-line, sem que o software esteja gravado localmente no computador que vai executá-lo (KNORR e GRUMAN, 2008). Este espaço virtual para dados e programas, a “nuvem”, permite que usuários possam acessar seus mesmos dados de qualquer parte da rede, bastando para isso ter um ponto de acesso. O navegador web passa a ser o software-mor do computador local, pois é dentro dele que outros softwares são chamados para execução.

A despeito da facilidade de acesso aos dados independentemente da máquina, o paradigma da cloud computing tem demonstrado que grandes corporações podem mudar rapidamente para um modelo de software como serviço, com usuários sendo chamados a participar do processo de produção. É o caso da Google, que tem investido fortemente em ferramentas que funcionam “na nuvem” e que fez de seus softwares versões “beta” eternas – o que implica um desenvolvimento constante e uma versão final que nunca se define⁶³.

Não é difícil considerar, portanto, que o momento seguinte para a Receita seria tentar abarcar tal paradigma e mudar o conceito de entrega da declaração; ao invés de instalar um software e entregar a declaração por meio dele, o usuário manipularia um software on-line dos computadores da própria Receita, e a digitação dos dados já estaria na realidade atrelada aos bancos de dados do governo⁶⁴. É o que já ocorreu com a plataforma Lattes, utilizada no Brasil para cadastro e atualização de currículos acadêmicos. Se anteriormente havia um software a ser instalado no computador dos usuários para realizar o cadastro e fazer envio eletrônico, hoje tudo é feito de forma on-line a partir do site da plataforma. No caso do Lattes, o uso da plataforma por meio do software instalado nem é mais possível.

A questão é que o modelo da “cloud computing” pode trazer um desbalanceamento entre quem provê e quem usa. Além das possibilidades de produção de escassez por meio do acesso às aplicações on-line (controle por meio de um login) e das questões de segurança dos dados pessoais, dispostos em um lugar que não se conhece, a computação na nuvem pode ofuscar o potencial de compartilha-

63 Um *login* no Google permite, atualmente, a utilização de diversos recursos como *webmail*, agenda de compromissos, redes sociais, edição de documentos, dentre vários outros, todos eles sendo executados na nuvem.

64 Algumas declarações mais simples já podem ser fornecidas de forma *on-line* pelo site da RF.

mento de conhecimentos entre os seus usuários, justamente um dos pontos mais importantes para a prática de atividades colaborativas.

Um dos pioneiros do movimento do software livre, Richard Stallman considera o fenômeno da “cloud computing” como uma armadilha (JOHNSON, 2008), pois é capaz de amarrar as pessoas a sistemas proprietários e trancá-las do lado de fora dos locais dos seus próprios dados. Se organizações como a Google têm oferecido serviços de forma gratuita, não se pode garantir que não haja cobrança futura nem o que irá ocorrer com os dados, isto num momento em que os usuários já estarão em um nível muito mais elevado de dependência.

Na realidade, a computação em nuvem pode tolher o acesso livre ao software necessário ao trabalho vivo, como lembra Boutang:

No capitalismo cognitivo, para ser produtor de riqueza, o trabalho vivo deve dispor do acesso às máquinas (hardware), aos programas, à rede e às condições de execução de sua atividade *réseauxale* (ambientais em particular). O acesso livre suplanta o conceito de propriedade exclusiva. Trata-se de acessar juntas e ao mesmo tempo as informações, os conhecimentos para produzir outros conhecimentos (Boutang, 2007, p. 159).

Voltando ao caso da Receita, uma mudança para tal paradigma da nuvem, da forma como está começando a acontecer, torna impossível a execução das etapas de solução descritas anteriormente. A transferência do espaço de configuração dos softwares para um lugar desconhecido na nuvem restringe os “hacks” àquelas corporações que estão promovendo e estimulando tal deslocamento.

Considerações finais

Neste texto foi apresentado um relato de uma situação relacionada ao envio das declarações do imposto de renda por meio do software fornecido pela Receita Federal, cujo problema para entrega de uma declaração serviu de mote para a discussão. A partir deste relato, foram acrescentados alguns comentários que descrevem o papel de determinadas tecnologias que foram fundamentais para se resolver a situação.

Procuramos reforçar alguns pontos relacionados às possibilidades produtivas de intervenção na produção e utilização de software, por meio de certas dinâmicas que aproximam produtores e usuários e os confundem sob a mesma prática hacker. Foi o caso do acesso aos arquivos gravados por um programa em Windows, usando como ferramenta o Linux, e as modificações feitas nos arquivos do software da Receita Federal por meio do XML.

Estas práticas, contudo, dependem do grau de liberdade embutido em determinadas tecnologias, indo além da polarização entre artefatos proprietários e livres. A liberdade de modificação trazida pelas tecnologias apresentadas anteriormente foi necessária para se chegar a uma solução, uma liberdade que é impossível de ser alcançada em determinados paradigmas correntes. O momento da computação em nuvem, ainda embrionário em comparação com as muitas possibilidades de espaços de produção, ainda tem se mostrado desequilibrado quanto à liberdade embutida em suas tecnologias. Faz sentido, portanto, perceber que o movimento de software com código-fonte aberto, justamente por ser livre, tem feito ressalvas quanto ao futuro deste paradigma.

O general intellect é o fundamento de uma cooperação social, subjetiva, cada vez mais ampla e heterogênea. A ação em concerto, conforme escreve Virno (2003), move-se na direção do compartilhamento de aptidões comunicativas e cognitivas. O êxodo de um virtuosismo servil é como “uma invenção desrespeitosa que altera as regras do jogo e enlouquece a bússola do adversário”. Esta invenção, demonstrada pelas “gambiarras produtivas” realizadas no caso analisado, os hacks desrespeitosos que foram efetuados, se produz criando seu próprio plano de imanência.

Referências

- BERARDI, Franco. *A fábrica da infelicidade: trabalho cognitivo e crise da new economy*. Rio de Janeiro: DP&A, 2005.
- BOUTANG, Yann Moulier. *Le capitalisme cognitif: la nouvelle grande transformation*. Paris: Éditions Amsterdam, 2007.
- FONSECA, Felipe. *Xemeliza isso aí...: conhecendo a história do termo xemelê*. Disponível em: <http://xemele.net/wikka.php?wakka=Xemele>, 2007. Acesso 20.jun.2009.
- GORZ, André. *O imaterial: conhecimento, valor e capital*. São Paulo: Annablume, 2005.
- HARDT, Michael; NEGRI, Antonio. *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 20.jun.2009.
- JOLLIVET, Pascal. *NTIC e trabalho cooperativo reticular: do conhecimento socialmente incorporado à inovação sociotécnica*. In: COCCO, G.; GALVÃO, A. P.; Silva, G. (orgs.) *Capitalismo cognitivo: trabalho, redes e inovação*. Rio de Janeiro: DP&A, 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 20.jun.2009.

LAZZARATO, Maurizio. As revoluções do capitalismo. Rio de Janeiro: Civilização Brasileira, 2006.

TAURION, César. Eliminando os DOCX, 2009. Disponível em: https://www.ibm.com/developerworks/mydeveloperworks/blogs/taurion/entry/eliminando_os_docx Acesso 20.jun.2009.

VIRNO, 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 20.jun.2009.

■.....**Gilvan Vilarim** é professor do UNIFESO, doutorando em Serviço Social pela UFRJ e participante da Rede Universidade Nômade. Possui como temas de interesse as novas tecnologias, trabalho imaterial, ciência da computação, educação em computação.