

Contribuição do **Centro de Competência em Software Livre da Universidade de São Paulo CCSL/USP** em conjunto com **Centro de Tecnologia e Sociedade da Fundação Getúlio Vargas do Rio de Janeiro - CTS-FGV** acerca do documento:

*“Procedimentos para o exame de pedidos de patentes envolvendo invenções implementadas por programa de computador”*

submetido à Consulta Pública pelo **Instituto Nacional de Propriedade Industrial do Ministério do Desenvolvimento, Indústria e Comércio Exterior- INPI/MDIC**

**Colaboradores:**

**Eros Roberto Grau** (Prof. Titular da Faculdade de Direito da USP e ex-Ministro do Supremo Tribunal Federal)

**Tercio Sampaio Ferraz Junior** (Prof. Titular da Faculdade de Direito da USP)

**Juliano Souza de Albuquerque Maranhão** (Prof. Associado da Faculdade de Direito da USP e Coordenador Jurídico do CCSL/USP)

**Fernando de Magalhães Furlan** (ex- Presidente do Conselho Administrativo de Defesa Econômica e Pesquisador Visitante da Universidade de Macau)

**Fabio Kon** (Prof. Titular do Departamento de Computação do Instituto de Matemática e Estatística da USP e Diretor do CCSL/USP)

**Nelson Lago** (Mestre pela USP e Gerente técnico do CCSL/USP)

**Paulo Meirelles** (Mestre pela UFRGS, sócio da Associação Software Livre.org e Pesquisador do CCSL/USP)

**Joaquim Falcão** (Diretor da Escola de Direito da FGV)

**Ronaldo Lemos** (Professor Titular da Escola de Direito da FGV e Diretor do CTS/FGV)

**Carlos Affonso Pereira de Souza** (Professor Titular da Escola de Direito FGV e Vice-Diretor do CTS/FGV)

**Bruno Magrani** (Mestre por Harvard e Professor do CTS/FGV)

**Joana Varon Ferraz** (Mestre pela FGV e Professora CTS/FGV)

**Karin Grau-Kuntz** (Doutora e Mestre pela Ludwig-Maximilian Universität de Munique, Coordenadora Acadêmica do Instituto Brasileiro de Propriedade Intelectual-IBPI)

## **1. Introdução: ilegalidade da proposta do INPI**

O documento *“Procedimentos para o exame de pedidos de patentes envolvendo invenções implementadas por programa de computador”* submetido à Consulta Pública pelo INPI propõe-se a responder quais tipos de programas de computador seriam patenteáveis e delimita então “Diretrizes de Exame”.

Ao fazê-lo, esquiva-se da questão fundamental a ser posta: programas de computador são patenteáveis de acordo com o direito brasileiro?

Como se sabe, o regime jurídico pátrio de proteção ao *software* é o de direito de autor, no qual o software é equiparado a obra literária, seja ele no seu código-fonte (linguagem de programação), seja em código-objeto (linguagem de máquina), em suporte físico de qualquer natureza, para emprego em qualquer máquina automática de tratamento de informação, na definição ampla que o termo “programa de computador” encontra no art. 1º da lei 8609/98. Sendo assim, a proteção é dirigida à expressão da solução computacional na estruturação e redação dos comandos do programa ao computador e não ao efeito ou utilidade que produz com seu emprego na máquina.

Como a legislação pátria não dá relevância jurídica à aplicação técnica do software, cuja proteção não se confunde e não pode se confundir com a proteção do programa de computador, que é de natureza autoral, percebe-se de imediato e tem-se como regra que a proteção patentária, voltada essencialmente para conferir exclusividade ao inventor na exploração de determinado produto ou processo com aplicação técnica (industrial), não é congênere ao programa de computador.

É possível vislumbrar no programa de computador quatro dimensões: (i) um computador em abstrato (máquina de Turing) é uma construção matemática, de modo que cada programação constitui uma solução para um problema matemático, residindo, no programa, uma idéia ou conhecimento informático (*programa como ideia*); (ii) o programa de computador em sua expressão literal (diretamente na linguagem de programação ou indiretamente na linguagem de máquina) veicula o conhecimento informático de determinada forma artística (*programa como obra*); (iii) o programa de computador impresso ou armazenado em determinado suporte constitui determinado produto que pode ser “consumido” pelo usuário em seu computador (*programa como produto*); (iv) sua operação na máquina traz um resultado útil ao usuário, podendo trazer efeitos ou aplicações técnicas no mundo físico ou virtual (*programa como funcionalidade*).

Em determinados contextos e na medida em que evoluem a tecnologia e os modelos de produção econômica, sobreleva-se uma ou outra dimensão do software, o que traz posicionamentos distintos sobre seu enquadramento jurídico, que oscilam dentre os dois extremos, em se tomar o programa como conhecimento, de um lado, em que se coloca mesmo em questão se o programa mereceria qualquer tipo de apropriação e, de outro, em se privilegiar o aspecto de funcionalidade ou utilidade, que inclina alguns para a proteção patentária do programa, como se o mesmo se confundisse com a própria invenção ou contribuição técnica industrial. Dentre essas diferentes dimensões, a solução consagrada no âmbito internacional foi a do *software* como obra artística sujeita a proteção autoral.<sup>1</sup> Ou seja, a ideia não é apropriável, o suporte de armazenagem não importa e a aplicação técnica é algo independente daquilo que constitui o “programa em si”, que somente é protegido em sua expressão literal. Foi essa a conclusão de comitê de experts da OMPI e da UNESCO, formado em 1985 e que foi referendado pelo TRIPS de 1994, estabelecendo-se que o programa de computador será protegido como obra literária no sentido da Convenção de Berna, o que significa que *o propósito* e *a mídia* para os quais a informação é criada são juridicamente irrelevantes.

Especificamente em relação à dimensão de funcionalidade do software, houve a clara percepção de que pode haver diversas expressões literais do programa para uma mesma aplicação técnica e uma mesma expressão literal pode receber aplicações técnicas distintas, seja de forma isolada ou combinada com outros programas. Daí o acerto na escolha do regime autoral. Vale lembrar que o regime patentário, em sua essência, cria uma reserva de mercado para o inventor explorar o produto ou processo em regime de monopólio, de forma que, se aplicada ao software e especificamente ao conferir privilégio sobre a utilidade que esse venha a produzir por sua operação no computador ou máquina automática, a patente pode abranger e, assim, obstruir uma série de soluções informáticas e expressões literais distintas que poderiam alcançar aquela utilidade. Mais do que isso, como uma mesma programação pode servir a utilidades distintas, notadamente quando for combinada com

---

<sup>1</sup> Esse regime é regido por acordos multilaterais e internacionais que conferem o regime autoral ao software, valendo destacar: a Convenção da União de Paris, de 1883, a convenção de Berna, de 1886, O Acordo Sobre Aspectos dos Direitos de Propriedade Intelectual Relacionados ao Comércio (Trade-Related Aspects of Intellectual Property Rights- TRIPS), no âmbito da OMC, de 1994, e o Tratado da Organização Mundial da Propriedade Intelectual (OMPI) sobre direito autoral de 1996.

outras em um mesmo programa, a patente sobre uma utilidade que abranger o software pode obstaculizar até mesmo a produção de outras utilidades. Ou seja, o software pode ser meio para invenções, mas não é uma invenção industrial em si. Na medida em que a proteção da invenção alcançar também o meio, outras invenções que dele poderiam fazer uso são comprometidas. Pensar de modo diverso, bloqueando-se a ponta final da utilidade, implicaria potencialmente a obstrução de uma série de processos criativos independentes.

Aliás, em relação à apropriação do software pelo direito de patentes, Stallman<sup>2</sup> lembra que um mesmo software pode conter centenas de programas diversos, de modo que o registro de uma patente muito provavelmente significará apropriação de criações intelectuais alheias, fator que é particularmente verdadeiro com a atual disseminação da produção de programas de computador em regime de software livre. A abrangência ao se apropriar do software na ponta da utilidade é tal que pode implicar absurdos acerca da propagação do conhecimento que está na base da produção de software, como foi o caso nos E.U.A. em que cientistas da computação e professores universitários foram presos por suposta violação de patentes quando nada mais faziam do que desenvolver suas atividades acadêmicas.<sup>3</sup>

Mais recentemente, em alguns países, foi levantada a discussão sobre a possibilidade de se admitir, *excepcionalmente*, a patente de software como “invenção” em casos nos quais o efeito técnico ou a aplicação industrial seria difícil de distinguir da execução do próprio programa. Nos E.U.A. o Escritório de Patentes chegou a publicar “manual de exame” disciplinando situações em que se equipara o próprio programa de computador a invenção, com base em precedentes judiciais da Suprema Corte. O resultado foi o aumento

---

<sup>2</sup> “[some fields, such as pharmaceuticals,] fits in with the naive idea [...] that there is one patent per product [...] but when one product corresponds to many different ideas combined, it becomes very likely your new product is going to be patented by somebody else already. [...] A physical system whose design has a million different pieces in it is a mega project. A computer program whose design has a million pieces in it [...] is not a particularly giant program. [...] If you look at a word processor today, you would find, I think, hundreds of different features. If you develop a nice new innovative word processor, that means there are some new ideas in it, but there must be hundreds of old ideas in it. If you are not allowed to use them, you cannot make an innovative word processor.” Stallman, R. (2002). Software patents: Obstacles to software development. Disponível em: <http://www.cl.cam.ac.uk/~mgk25/stallman-patents.html>

<sup>3</sup> “Propriedade Intelectual em Software: o que podemos aprender da experiência internacional?” Andrade et.al. *Revista Brasileira de Inovação*, Rio de Janeiro, 6(1), p.31-53, jan/jun 2007, p.43.

vertiginoso dos pedidos e da concessão de patentes de software naquele país.<sup>4</sup> Na Europa, a Comissão Europeia, a partir de pedidos e patentes concedidas em alguns países membros, gerando problemas de uniformização, e após anos de estudo sobre o tema, com a submissão de texto sobre a patenteabilidade de programas de computador no ano de 2000,<sup>5</sup> chegou a publicar uma proposta de diretiva em 2002, na qual um programa de computador *poderia* ser patenteado, após análise, se fosse demonstrado que o mesmo trazia uma contribuição técnica, sendo referenciado a um produto ou processo industrial no mesmo pedido de patente.<sup>6</sup> Após *cinco anos de aprofundado e amplo debate público*, a proposta foi votada e rejeitada pelo Parlamento Europeu, a mais alta esfera legislativa em âmbito comunitário.<sup>7</sup> Um dos principais fundamentos para a rejeição foi o risco de entrave ao desenvolvimento de software na Europa, em particular para pequenas e médias empresas de software e para comunidades de código aberto, dado risco de concentração de mercado em grandes grupos produtores de software, em particular de origem norte-americana.

A proposta do INPI teria como objetivo tão-somente “*auxiliar no exame técnico de pedidos de patente envolvendo invenções implementadas por programa de computador*”; porém, ao fazê-lo, já admite, em diploma infra-legal (Resolução de órgão Ministerial), que o software está sujeito ao regime de patentes, o que é o avesso do regime adotado pela legislação pátria. O conteúdo das “Diretrizes” é ainda mais surpreendente ao conferir ampla patenteabilidade aos programas de computador abrangendo tanto efeitos técnicos no mundo físico quanto no mundo virtual, estando o programa embarcado ou não em máquina e alcançando inclusive processos computacionais, como otimização de processadores de texto ou benefícios na interface com usuário. Tudo isso assentado na interpretação de que “*programa de computador em si*” (que não pode ser “invenção” por força do art. 10, inc. V da Lei de Propriedade Industrial- 9279/96) constituiria apenas o *software* sem efeito técnico, sendo os demais, todos aqueles com aplicação ou utilidade técnica, patenteáveis. Mas a utilidade prática é uma das dimensões de qualquer software, o que confere às

---

<sup>4</sup> “Propriedade Intelectual em Software: o que podemos aprender da experiência internacional?” Andrade et.al. *Revista Brasileira de Inovação*, Rio de Janeiro, 6(1), p.31-53, jan/jun 2007.

<sup>5</sup> “*The patentability of computer-implemented inventions: consultation paper by the services of the Directorate-General for the Internal Market*” (19/10/2000).

<sup>6</sup> “*Proposal for a Directive of the European Parliament and of the Council on the patentability of Computer Implemented Inventions*” (COM (2002) 92-final- 2002/0047(COD)).

<sup>7</sup> Common Position (EC) n. 20/2005.

Diretrizes do INPI um impacto potencial em termos de patenteabilidade de software ainda maior do que o encontrado nos E.U.A, abrindo o espaço para a descrição de qualquer software como invenção em pleito de patente.

Na verdade, o documento do INPI parece ter recolhido na jurisprudência estrangeira, em particular dos E.U.A., onde, por razões óbvias, há forte inclinação patentária, situações em que foi admitida a patente de software, utilizando-as como modelo para análise de pedidos de patentes no Brasil. Como se sabe, os E.U.A. constituem uma jurisdição baseada na *common law*, de modo que os julgados sobre o tema naquele país equivalem a *alterações no direito norte-americano*. Portanto, um manual de exame adotado pelo escritório de patentes dos E.U.A. com base em julgados da Suprema Corte tem função meramente descritiva do direito vigente naquele país. A situação é totalmente diversa no Brasil, que emprega o sistema de direito estatutário: um manual de exame deve estar baseado na legislação pátria, não em pronunciamentos judiciais (que não alteram o direito), muito menos em pronunciamentos judiciais estrangeiros. Mas como o manual de exame do INPI pretende introduzir a patente de software no Brasil, sendo que sequer há um entendimento jurisprudencial nacional sobre o tema, suas “Diretrizes” tem a pretensão de papel normativo, i.e. de alteração da Lei, o que é inadmissível em sede de resolução de órgão da Administração Pública.

Considere-se também que a jurisprudência norte-americana é baseada em *case-law* em sede do regime autoral para proteção de programas de computador, de modo que cada caso de patente de software é apreciado minuciosamente para que se reconheça uma exceção ao regime padrão de proteção autoral. O ônus recai sempre sobre aquele que pretende a patente sobre o software. Assim, na medida em que o INPI admite programas “em regra” patenteáveis (ainda que sob a veste de aparentemente inocente documento de auxílio técnico), por resolução infra-legal, em primeiro lugar, joga-se o ônus para aquele que contesta a patente e, em segundo, executa-se essa inversão à margem do debate democrático exigido pela relevância do tema.

Com efeito, o tema toca em questão estratégica para o país sobre o desenvolvimento da produção nacional de software, estímulo a pequenas e médias empresas e domínio de tecnologia da informação. Exatamente por perceber a sensibilidade do assunto, a *Comissão Européia debateu-o por cerca de cinco anos e definiu-o em sua esfera legislativa suprema*. Foram dois anos de discussão de texto teórico sobre a matéria e mais três anos de discussão de proposta normativa. Ao contrário, no Brasil, pretende-se partir diretamente para a proposta normativa, a ser discutida em *exíguos 60 dias*. Tudo isso em nível infra-legal!

Importante lembrar que o estímulo ao desenvolvimento e a autonomia tecnológica do país são diretrizes firmadas constitucionalmente no Brasil, o que vem reforçado na política nacional de informática, preocupada especificamente em vedar iniciativas que possam criar situações monopolísticas (Lei 7232/84, art. 2º, inc. IV). Por outro lado, como a Lei do Software (Lei 9609/98, art. 1º) adota definição ampla de “programa de computador” sujeitando-o à proteção autoral, fica nítido que qualquer proposta de patente de software deve envolver debate público a ser travado em esfera legal, sob pena de *patente ilegalidade*, valendo ainda considerar que eventual alteração da Lei do Software deveria ainda se coadunar com as diretrizes constitucionais, da política nacional de informática e, em última análise, com os interesses do País.

A presente manifestação organiza-se da seguinte maneira. Na próxima Seção abordamos o perfil constitucional e a política nacional de informática, examinando suas implicações no que se refere à proteção do software e ao desenvolvimento desse setor no Brasil. Em seguida, examinamos a experiência internacional, analisando a experiência dos E.U.A, em que houve larga abertura para as patentes de programas de computador, da Comunidade Européia e da Índia, que constitui país emergente e com forte potencial de crescimento na produção de software, assim como o Brasil. Tanto a Europa quanto a Índia têm preocupação semelhante à do Brasil, no sentido de estímulo do desenvolvimento econômico local, frente ao domínio de companhias norte-americanas de software. Passamos então a uma análise técnica e jurídica do documento do INPI e suas Diretrizes de Exame do pedido de patentes, voltada à interpretação do art. 10 da Lei de Patentes de modo a precisar o sentido da expressão “programa de computador em si”.

## 2. Diretrizes Constitucionais e a Política Nacional de Informática

A constituição Federal baseia a ordem econômica na livre iniciativa (art. 170, caput) na constituição de mercado soberano (inc. I), orientado pelos princípios de livre concorrência (inc. IV) e proteção a pequenas e médias empresas (inc. IX), entendendo que a interação competitiva dos agentes, sem limitações em função do poder econômico de uns diante dos outros, será mais apta a trazer benefícios ao consumidor (protegido em sua relação com os ofertantes de produtos e serviços - inc. V) e propiciar o pleno emprego (inc. VIII). Essa proteção da liberdade de iniciativa como fundamento não impede e, ao contrário, exige do Estado uma atuação indutora na definição de políticas públicas voltadas à proteção do meio ambiente, do desenvolvimento regional e para o desenvolvimento de mercados que possuam um significado estratégico, tanto em termos sociais como econômicos ou culturais.

Assim é que, em seu art. 216, a CF88 inclui no patrimônio cultural brasileiro a criação tecnológica e artística. Obviamente, não quer a norma, aqui, “estatizar” toda criação tecnológica ou artística, material ou imaterial, referindo-se, antes, ao conhecimento e aos elementos que apontem uma determinada identidade cultural. No § 3º desse dispositivo estabelece que a Lei não só incentivará a produção desses bens e valores culturais como proporcionará o *conhecimento* dessa produção. Por sua vez, no Capítulo IV, referente à Ciência e Tecnologia, a Constituição prevê, tendo em vista a promoção da capacitação tecnológica (art. 218, *caput*), uma atuação positiva do Estado com relação ao mercado de tecnologia, que considera patrimônio nacional, e no qual o Estado deve incentivar o desenvolvimento cultural, o bem estar e a autonomia tecnológica do país (art. 219)<sup>8</sup>. Ela, assim, incumbe o Estado de criar, por meio de legislação específica, os mecanismos para que se desenvolva a autonomia tecnológica do país, dando-se uma conotação especial ao

---

<sup>8</sup> Art. 219. O mercado interno integra o patrimônio nacional e será incentivado de modo a viabilizar o desenvolvimento cultural e sócio-econômico, o bem-estar da população e a autonomia tecnológica do País, nos termos de lei federal.



mercado de tecnologia, como patrimônio nacional, o que exige uma atuação positiva no sentido de sua preservação e promoção.

Essa atuação positiva não se confunde, obviamente, com intervenção direta, dado que o setor deve ser organizado pelas forças de mercado, cabendo ao Estado uma atuação somente supletiva, conforme art. 2º, inc. II, da Lei 7232/84, que fixa a política nacional de informática. Mas também não se limita a coibir abusos e irregularidades dos agentes. O mesmo art. 2º, no inc. VI, prevê a orientação de cunho político das atividades de informática, levando em conta a sua *natureza estratégica*. Essa consciência do legislador da importância estratégica da informática reflete-se na previsão de um *direcionamento* (não dirigismo) do esforço no setor para o desenvolvimento econômico e social e fortalecimento nacional. Assim, cabe ao Estado estabelecer mecanismos, apontar e *induzir comportamentos*, para que os objetivos constitucionalmente elencados de desenvolvimento e autonomia nacional do conhecimento informático sejam alcançados.

Essa preocupação com a produção e propagação do conhecimento, mas principalmente com a autonomia do país no setor, reflete-se em importante dispositivo legal, que prevê, dentro da política de informática, a “*proibição à criação de situações monopolísticas, de direito ou de fato*” (Lei 7232/84, art. 2º, inc. IV). Proibir a *criação de situações* de monopolísticas não é o mesmo que reprimir monopólios, nem mesmo proibir a concentração de empresas (ambas formas negativas de atuação). Essa forma de atuação repressiva e preventiva da dominação de mercado, exercício abusivo de posição dominante e elevação arbitrária de lucros, já tem lugar na legislação geral de defesa da concorrência para qualquer mercado. O bem protegido nesse dispositivo não é a *concorrência* no mercado de software. A proibição tem a ver, antes, com a centralização do conhecimento e reflete a preocupação do legislador com a importância estratégica do setor e a consciência de que o poder e a liberdade na sociedade de informação dependem do controle das novas tecnologias.

Portanto, proibir a criação de situações significa atuar positivamente nas causas da formação e preservação de monopólios no mercado de software para impedir a centralização de poder sobre a informação em algumas corporações. Para entender o

alcance e sentido dessa atuação prevista na política nacional de informática é importante destacar algumas peculiaridades do mercado de software.

O mercado de software é baseado na alta tecnologia e na inovação. Nesse ambiente, o programa inovador, na medida em que ganha popularidade e dada a necessidade de comunicação entre usuários através do produto e a importância da compatibilidade de padrão tecnológico nessa comunicação, passa a ter um elemento adicional de atratividade que cria um círculo virtuoso: quanto mais pessoas o usam, essa é mais uma razão para mais pessoas o utilizarem. Esse é o chamado “efeito de rede”.<sup>9</sup>

Em função desse efeito, os produtos inovadores vencedores acabam concentrando grande parte do mercado (*winner takes most*),<sup>10</sup> o que faz com que o equilíbrio natural desse mercado tenha a empresa líder com uma participação aproximada ou igual ao mercado como um todo.

Dentro desse contexto, Economides ressalta que a análise do grau de concentração no mercado de software pode superestimar a existência de poder de mercado.<sup>11</sup> Em primeiro lugar, não é possível concluir que práticas anticompetitivas foram responsáveis pela criação de tal desigualdade, afinal é esse o equilíbrio natural em mercados com externalidades de rede e incompatibilidade de padrão tecnológico. Em segundo lugar, a despeito de sua participação de mercado, práticas de preços de monopólio são controladas pelo risco de se estimular a inovação por concorrentes que, a qualquer momento, podem redefinir o mercado como um todo.<sup>12</sup>

Na verdade, a avaliação tradicional pautada pelo modelo antitruste estrutura-conduta-desempenho é inadequada para o mercado de software, que segue um padrão de competição

---

<sup>9</sup> Shapiro, Carl e Varian Hal R., *Information Rules: A Strategic Guide to the Network Economy*, Boston, Harvard Business School Press, 1999.

<sup>10</sup> Economides, Nicholas. *Competition and Vertical Integration in the Computing Industry*, in *Competition, Innovation, and the Role of Antitrust in the Digital Marketplace*, Jeffrey A. Eisenach and Thomas M. Lenard (eds.), Kluwer Academic Publishers 1999; e *The Impact of the Internet on Financial Markets*, *Journal of Financial Transformation*, vol. 1, no. 1 (2001), pp. 8-13.

<sup>11</sup> Economides (Agosto 2001).

<sup>12</sup> Economides, N. *The Microsoft Antitrust Case*, *Journal of Industry, Competition and Trade: From Theory to Policy* vol. 1, no. 1, pp. 7-39 (Agosto 2001).

inteiramente distinto. Não haveria, portanto, razão imediata para uma proibição pela Lei 7232/84 pura e simples para situações de monopólio em nome somente da concorrência.

Todavia, o monopólio pode se tornar perverso na medida em que o controle do produto monopolista e sua exploração como saber próprio, inacessível a terceiros, pode preservar a supremacia do monopolista, ainda que surjam inovações competitivas. É essa *situação de monopólio*, resistente à dinâmica competitiva do mercado, que se torna um risco para o setor (além das restrições aos concorrentes), pois centraliza o conhecimento informático e sujeita os demais agentes ao detentor desse conhecimento, que controla, assim, como coloca Boyle, *o curso da inovação tecnológica*.<sup>13</sup>

Nesse sentido, a combinação da participação de mercado da solução dominante com os efeitos de rede e o *sistema de proteção dos direitos sobre o software* pode ter efeito pernicioso na dinâmica de mercado no sentido da consolidação de posições dominantes. Como argumenta Zittrain, as conhecidas medidas antitruste de repressão e prevenção de monopólio são ineficazes diante de modelos de proteção dos direitos sobre o software que crie reservas de mercado e que limitem o acesso ao conhecimento informático (como mostram os casos de abuso da Microsoft no sentido de impedir o desenvolvimento da concorrência por bloquear o acesso a informações necessárias para a interoperabilidade com seus produtos). Sendo assim, a política de proteção ao mercado de software deve se concentrar justamente na discussão do alcance da propriedade intelectual que possa obstaculizar a produção de soluções informáticas alternativas ou tornar inacessível a concorrentes o conhecimento informático.<sup>14</sup>

A abrangência da proteção jurídica do software pode barrar ou mesmo desincentivar o esforço de produtores de inovação de forma que a posição de monopólio se perpetue independentemente do efetivo benefício que a empresa dominante proporcione aos

---

<sup>13</sup> Ao comentar o caso Microsoft, Boyle ressalta que: “*The complaint against Microsoft of the world is not so much that they keep their prices high- though that is sometimes alleged. Instead it is the claim that their intellectual property rights over fundamental standards with strong positive network effects give them too much power to **control the course of innovation***” (Boyle, James. The second enclosure movement and the construction of the public domain, in *Law and Contemporary Problems*, v. 66:33, p. 64, **grifamos**)

<sup>14</sup> Zittrain, Jonathan. The Um-Microsoft Um-Remedy: Law can prevent the problem that it can't patch later. *Connecticut Law Review*, 1999 disponível em: <http://cyber.law.harvard.edu/zittrainmsdoj.pdf>

consumidores. Exatamente por essa razão, os grandes produtores de software norte-americanos procuraram ampliar o escopo de proteção da exploração do software do regime de direito autoral para o regime de patentes, por meio do registro de patente do software seguido do esforço em fazer valer as patentes nos tribunais. O regime autoral protege a expressão intelectual do programa, abrindo-se espaço para outras formas de expressão da ideia contida no programa com o mesmo efeito ou utilidade na sua execução computacional, que podem ter sua exploração questionada pelo autor na medida em que se demonstrar a existência de plágio, seja da redação de partes do programa seja de sua estrutura ou arquitetura para solução do problema informático ao qual se dirige. Já o regime patentário, ao conferir exclusividade para aquela utilidade técnica produzida pela execução do programa, cria reserva de mercado sobre aquela aplicação, o que, como visto acima, obstrui a produção intelectual de formas alternativas de programação que poderiam provocar o mesmo efeito.

Fica claro, portanto, que qualquer política que estimule a patente de software no país anda em sentido contrário à orientação constitucional de desenvolvimento do mercado nacional de tecnologia. A política de Estado, firmada em Lei Federal, que visa a evitar a criação de situações de monopólio, é acertada, pois ajuda a impedir a manutenção da posição dominante dos líderes, posição essa que limita a inovação e o acesso ao conhecimento por concorrentes. Por outro lado, uma vez constatado que a posição dominante de mercado é de fato detida em grande parte por companhias estrangeiras de software, em particular norte-americanas, também é evidente que o regime patentário para programas de computador frustra o objetivo constitucional e legal de estimular e alcançar a autonomia tecnológica do Brasil nesse setor.

Com a percepção de que a ampliação do alcance dos direitos proprietários sobre o software limitam o desenvolvimento do país no setor, o Governo brasileiro segue orientação diametralmente oposta ao do regime de patentes. Vale dizer, não só reconhece o regime autoral como busca incentivar a difusão do conhecimento informático pela abertura do código-fonte dos programas de computador, ao adotar um conjunto de iniciativas para implementação do software livre no Brasil. Assim, até como reação aos enormes custos

gerados à administração com licenças de software, o Decreto Presidencial 18/00 instituiu o Comitê Executivo do Governo Eletrônico, no intuito de racionalizar os gastos com software. Para efetivar esse objetivo, instituiu o Comitê Técnico de Implementação de Software Livre, por meio do Decreto 29/03. Dentro desse programa, estuda-se a forma (livre) pela qual o Estado deveria se posicionar na aquisição e distribuição de seus programas.

Propõe-se, de um lado, que a Administração Pública somente adquira programas cujas cláusulas de licenciamento sejam compatíveis com o software livre, ou seja, que lhe permitam usar, estudar o código fonte, copiar, modificar, e redistribuir o software e, de outro, que o software de propriedade da Administração seja distribuído a terceiros nesses termos, obrigando, ademais, os licenciados, a licenciar todas as derivações no regime livre (o chamado *copyleft*). Fazendo uso de seu largo poder de compra e de sua ampla relação com os administrados, a opção da Administração pelo regime livre de uso de software pode criar, assim, um forte incentivo e induzir comportamentos no sentido de adoção desse modelo de uso o que dificultará o estabelecimento e preservação de monopólios privados.

O licenciamento do software em regime livre e a produção em regime de *creative commons* não só reduz barreiras à entrada diretamente no mercado de software, como também torna mais competitivo o mercado de prestação de serviços que é a ele adjacente. Uma vez que a plataforma de software padrão no mercado seja construída em regime livre e, portanto, permita a todos o acesso ao conhecimento informático subjacente, o mercado de prestação de serviços técnicos sobre esse software permanece aberto a todos os interessados, excluindo-se a possibilidade de condutas abusivas por parte de empresa que domine o software padrão no sentido de reduzir a concorrência ou obter lucros abusivos nos mercados adjacentes (tais como práticas discriminatórias a prestadores de serviços, acordos de exclusividade, etc.). Com isso, estimula-se a economia, o desenvolvimento tecnológico nacional e a oportunidade para a constituição e desenvolvimento de pequenas e médias empresas (também um dos objetivos orientadores da ordem econômica constitucional, conforme art. 170, inc. IX da CF88).

A admissão de patentes para software ameaça essa política de incentivo a pequenas empresas. De fato, enquanto empresas grandes têm acordos mútuos de licenciamento de patentes, as microempresas podem ser consideradas infratoras a qualquer momento. Ao mesmo tempo, dado o grande número de patentes existentes e sua não-especificidade a uma única área (em virtude da grande versatilidade do software), os custos para o levantamento de possíveis patentes em uso chega a ser proibitivo. Isso significa que o risco para a entrada de novas empresas no mercado de software é muito maior que em outras áreas, o que pode levar à concentração e mesmo ao monopólio.

Ademais, o mecanismo de patentes constitui verdadeiro inimigo do movimento de software livre. De um lado, a proteção pelo direito de autor é favorável ao compartilhamento e ao software livre, pois permite que várias expressões diferentes de uma mesma ideia coexistam e, assim, possibilita a criação de software livre para tratar qualquer tipo de problema, mesmo que já haja programas restritos similares. As patentes, por outro lado, pressupõem a existência de apenas uma ou algumas poucas soluções baseadas na mesma abordagem, todas formalmente autorizadas pelo detentor da patente, em geral com base em um acordo comercial. Esse tipo de organização evidentemente não se adequa ao software livre, onde pode haver múltiplas implementações e variações de um mesmo código sem que haja uma entidade central responsável.

Além disso, como não é viável cobrar *royalties* de programas livres, patentes de software podem impedir totalmente a existência de vários tipos de software livre. O licenciamento mútuo não serve como opção para minimizar esse problema: como o interesse da comunidade é o compartilhamento do conhecimento e os custos de registro são altos, não há vantagem em registrar patentes sobre quaisquer técnicas desenvolvidas no ambiente do software livre. Também não existe uma entidade com poderes para licenciar patentes de terceiros em nome da comunidade para uso em qualquer contexto. Dado que o software livre permite a reutilização do código em diferentes situações, uma única patente pode ser infringida em um sem-número de ambientes de software diferentes por conta de um único trecho de código. E, como raramente há uma entidade central responsável pelo software, não há verificações sobre possíveis violações de patentes, e eventuais processos a respeito colocam toda a comunidade, ao invés de uma única empresa, em xeque.

Assim, as patentes de software não se prestam ao ambiente de compartilhamento do conhecimento típico do software livre. Ao mesmo tempo, se é verdade que o mecanismo de patentes tem funcionado adequadamente em outras áreas, pode-se observar que o mesmo não se aplica ao software em geral. Patentes não tiveram um papel relevante no desenvolvimento da tecnologia relacionada ao software nos últimos 50 anos, o que mostra que esse mecanismo é um custo desnecessário a pagar.

Vê-se, assim, que a orientação política presente na Constituição e na Lei 7232/84, no sentido de promover a produção e conhecimento informático como patrimônio cultural, além da preocupação com a autonomia tecnológica e a oposição ao controle desse conhecimento por monopólios privados fornece supedâneo a ações administrativas no sentido de integração e disseminação do conhecimento sobre programas de computador, como a política de incentivo ao software livre, ao mesmo tempo em que impede ações no sentido de limitar o acesso a conhecimento informático ou limitar sua produção, como é o caso da iniciativa do INPI em abrir a oportunidade de proteção patentária do software no Brasil, ao arrepio da legislação pátria.

### **3. Análise internacional: experiência nos E.U.A, Europa e Índia**

#### **3.1. Estados Unidos da América<sup>15</sup>**

Os E.U.A. têm concedido patentes para proteção a software ao menos desde 1981<sup>16</sup>. Tendo permitido esse tipo de proteção para o software por tanto tempo, a análise da sua experiência é muito útil para a presente consulta.

---

<sup>15</sup> Agradecemos ao Koichi Kameda e ao Pedro Belchior pelo auxílio na pesquisa para este item.

<sup>16</sup> A primeira patente de software validada pela Suprema Corte Americana data de 1981.

Um dos estudos mais recentes sobre a patente de software foi realizado por James Bessen<sup>17</sup>. Ele aponta que ao longo da década de 1990 tanto empresas de software<sup>18</sup> quanto desenvolvedores nos EUA se opunham à proteção patentária para software<sup>19</sup>. Indo além, Bessen, após analisar tanto dados de pesquisas e análises realizadas na década de 1990 quanto dados recentes que vão até o ano de 2006 demonstra que:

- a) poucas empresas de software escolhem patentear suas invenções (apenas 24% o faziam em 1996 e apenas 33% o faziam em 2006);
- b) em empresas novas (*startups*) de software, esses montantes são ainda menores (17% em 1996 e apenas 14% em 2006);
- c) o grande número de patentes de software atualmente sendo concedidas nos E.U.A. combinado com o baixo percentual de empresas de software que decidem patentear seus programas indicam que grande parte das patentes estão sendo concedidas para empresas de outro setor que não o de desenvolvimento de software;
- d) se o setor de desenvolvimento de software, que é um dos mais inovadores nos E.U.A. no momento, decide não proteger seus produtos por patente, isso significa que outros incentivos para a inovação provavelmente são mais relevantes do que a expectativa de lucro que pode advir de uma patente;
- e) as patentes de software concedidas para a indústria de software geralmente ficam concentradas em poucas empresas de grande porte (75% em 1996 e 81% em 2006);

---

<sup>17</sup> BESSEN (2011).

<sup>18</sup> Conforme pode ser visto nas transcrições das audiências públicas realizadas pelo escritório de patentes dos EUA. Disponível em <http://www.uspto.gov/web/offices/com/hearings/index.html>. Acessado em 12 de Maio de 2012.

<sup>19</sup> Ver OZ, Effy. Acceptable protection of software intellectual property: A Survey of Software Developers and Lawyers, 34 Info. & Mgmt. 161, 161-173 (1998); SAMUELSON, Pamela et al., Developments on the Intellectual Property Front, 35 Communications of the ACM 33 (1992).



- f) o índice de patentes de software envolvidas em processos judiciais tem crescido exponencialmente (mais do que triplicando desde 1999), aumentando assim os riscos para os investimentos em inovação (devido aos altos custos dos processos judiciais, que podem significar a falência de novas empresas);

Em consonância com o que a literatura especializada tem apontado, as conclusões do estudo colocam em dúvida – ou no mínimo demonstram uma grande incerteza – a correlação a proteção do software através de patentes e o estímulo à inovação. Como demonstra o autor, a indústria de software possui características que não se adequam à proteção patentária, tais como a “sequencialidade” – que significa que a inovação se constrói cumulativamente sobre outras invenções, diferentemente da inovação estanque limitada a determinados produtos em muitas outras áreas. Dado que a grande maioria das empresas de software não vislumbra o sistema patentário como determinante para a inovação em seu setor (como visto nos dados acima), as patentes de software tornam-se instrumento de mera prática anticompetitiva.

Ao longo das últimas duas décadas tem havido um processo de ampliação na concessão de patentes em geral nos E.U.A., que tem se efetivado através da diminuição nos critérios de análise das patentes, bem como na ampliação do seu escopo. Isso levou o escritório de patentes dos E.U.A. (USPTO) a conceder patentes absurdas que têm sido utilizadas para ilustrar falhas em seu processo de análise de patentes. Uma dessas patentes tem por finalidade proteger um método para exercitar gatos através do emprego de raio laser<sup>20</sup>. A combinação desses dois elementos – baixos critérios de análise e ampliação do escopo das patentes – tem afetado consideravelmente a indústria de software, que na última década viu, além do aumento no número de patentes concedidas, um grande uso dessas patentes no judiciário<sup>21</sup>.

---

<sup>20</sup> Patente número 5443036 expedida pelo Escritório de Patentes dos E.U.A.

<sup>21</sup> BESSEN, James. A generation of software patents. Boston University School of Law Working Paper No. 11-31. Disponível em <http://www.bu.edu/law/faculty/scholarship/workingpapers/2011.html>. Acessado em 12 de maio de 2012.

A concentração das patentes de software nas mãos de empresas que não as desenvolvem e, muitas vezes, nem mesmo atuam na área geraram o aparecimento de uma figura no mercado, que personifica os defeitos e problemas do sistema: o *patent troll*. A expressão em inglês identifica empresas que compram patentes com a finalidade primordial de ameaçar outras empresas com caros processos judiciais, caso elas não concordem em pagar valores exorbitantes de licenciamento pelo uso das patentes. Aqui percebe-se o ciclo vicioso que os baixos critérios de análise conjugados com a amplitude da cobertura das patentes acaba por gerar: As patentes desse tipo podem ser obtidas facilmente, pois têm pouco valor comercial no sentido de promover a criação de produtos inovadores e, portanto, detêm valor apenas nessa forma deletéria, onde quanto mais patentes esses *trolls* conseguirem, maiores os custos para as novas empresas.

Algumas das patentes de software mais amplas e perigosas para a inovação da indústria têm sido alvo de ações judiciais de grupos dedicados à promoção da inovação. Organizações como a Eletronic Frontier Foundation (EFF) tem atacado essas patentes no judiciário com o objetivo de invalidar ou ao menos diminuir o escopo de sua proteção. Patentes que protegem *hiperlinks*, *streaming* de vídeo, janelas *pop-up*, cartazes de propagandas online direcionadas, pagamentos através de cartão de crédito online, compras através de um só clique, cesta de compras online, serviço de voz sobre IP e muitos outros são exemplos de tecnologias relativamente triviais que obtiveram proteção patentária nos E.U.A<sup>22</sup>.

A EFF tem obtido relativo sucesso no judiciário norte-americano. Por exemplo, recentemente invalidou uma patente para “um sistema que usa processamento de linguagem natural para responder a perguntas online feitas por clientes por email”<sup>23</sup>. Em outro caso, foi invalidada uma patente que cobria “o envio e recebimento de *streaming* de áudio e vídeo através da Internet”<sup>24</sup>. No entanto, essas vitórias são evidentemente pontuais e não podem reverter o malefício causado pelas falhas da USPTO.

---

<sup>22</sup> Para mais informações, veja o projeto “Patente Busting”. Disponível em <https://www.eff.org/patent-busting>. Acessado em 13 de maio de 2012.

<sup>23</sup> <https://www.eff.org/patent-busting/firepondpolaris>. Acessado em 14 de maio de 2012.

<sup>24</sup> *idem*.

O sistema de patente de software norte-americano tem sido alvo de críticas tanto de empresas bem estabelecidas de tecnologia quanto de acadêmicos e usuários<sup>25</sup>. Apesar disso, a reforma da lei de patentes aprovada pelo Congresso americano em 2011 pouco fez para resolver alguns dos problemas apontados acima. O caso recente envolvendo patentes sobre a linguagem de programação Java, da empresa Oracle, contra a Google, reacendeu a polêmica em torno das patentes de software nos E.U.A. e é provável que vejamos mais modificações na legislação ou na jurisprudência para restringir tais patentes. Dado o cenário que se desenha nos E.U.A., a expansão dos critérios de avaliação de patentes pelo INPI para incluir as patentes de software no Brasil nos parece ser um retrocesso para a indústria de software nacional.

### 3.2. Europa

Nos termos da Diretiva de 14 de maio de 1991<sup>26</sup> a proteção de programas de computador no âmbito da União Europeia ocorre por meio do Direito de Autor. Por sua vez, a Convenção de Munique sobre a Patente Europeia estabelece em seu Art. 52, 2 não serem os programas de computador considerados invenções, o que lhes nega a possibilidade de proteção por meio do Direito de Patentes. Por sua vez, o número 3 do mesmo dispositivo indica no sentido de que a proibição estaria vinculada ao programa de computador quando “considerado como tal”.<sup>27</sup>

<sup>25</sup> Veja por todos ZITTRAIN, Joathan. Disponível em

<http://www.guardian.co.uk/commentisfree/2012/apr/18/software-patents-shoulders-of-giants> Acessado em 14 de maio de 2012.

<sup>26</sup> **Directiva 91/250/CEE do Conselho, de 14 de Maio de 1991**, relativa à protecção jurídica dos programas de computador. Jornal Oficial nº L 122 de 17/05/1991 p.

<sup>27</sup> **Convenção de Munique sobre a Patente Europeia** - Artigo 52.º - Invenções patenteáveis:

1 - As patentes europeias são concedidas para as invenções novas que implicam uma actividade inventiva e são susceptíveis de aplicação industrial.

2 - Não são consideradas como invenções no sentido do parágrafo 1 particularmente:

a) As descobertas assim como as teorias científicas e os métodos matemáticos;

b) As criações estéticas;

c) Os planos, princípios e métodos no exercício de actividades intelectuais, em matéria de jogo ou no domínio das actividades económicas, assim como os programas de computadores;

d) As apresentações de informações.

3 - As disposições do parágrafo 2 apenas excluem a patenteabilidade dos elementos enumerados nas ditas disposições na medida em que o pedido da patente europeia ou a patente europeia apenas diga respeito a um

19

Tendo em vista os dispositivos mencionados e, especialmente, a expressão “considerado como tal”, o Escritório de Patentes Europeu, partidário da proteção patentária, desenvolveu uma linha interpretativa própria do dispositivo, uma prática que culminou por colocar em pauta de acirrada discussão a questão patenteabilidade dos programas de computador no âmbito europeu.

Em 1986 a Câmara Técnica do Escritório de Patentes Europeu elegeu, no caso “VICOM” (T 208/84), o efeito técnico como fator decisivo na determinação se o pedido de patente cairia no âmbito da exclusão do Art. 52, 2 e 3 (programa de computador considerado como tal”) ou não. Nesse sentido, decisivo para a concessão da proteção seria o critério do oferecimento de uma solução técnica a um problema específico e não, ao contrário do que dispõe a letra do dispositivo legal mencionado, a exclusão de proteção em consideração ao objeto – programa de computador – como tal considerado:

“Generally speaking, an invention which would be patentable in accordance with conventional patentability criteria should not be excluded from protection by the mere fact that, for its implementation, modern technical means in the form of a computer program are used. Decisive is what technical contribution the invention as defined in the claim when considered as a whole makes to the known art.”<sup>28</sup> (grifo nosso)

Com o caso VICOM o Escritório de Patentes Europeu abriu as portas para o entendimento de que não poderia persistir a máxima de que algoritmos – a base dos programas de computadores, só porque algoritmos, e portanto não passíveis de proteção autoral, não poderiam produzir efeitos técnicos posteriores que justificassem a proteção patentária.

Seguindo este raciocínio, um processo determinado, mesmo que reportando a elementos não patenteáveis (algoritmos) poderiam gerar efeitos que, por sua vez, se traduziriam em desses elementos considerado como tal.

4 - Não são considerados como invenções susceptíveis de aplicação industrial no sentido do parágrafo 1 os métodos de tratamento cirúrgico ou terapêutico do corpo humano ou animal e os métodos de diagnóstico aplicáveis ao corpo humano ou animal. Esta disposição não se aplica aos produtos, especialmente às substâncias ou composições, para utilização num desses métodos.

<sup>28</sup> Caso VICOM (T 208/84).

uma contribuição ao estado da técnica e, deste modo, desde que satisfeitos todos os requisitos legais (novidade, atividade inventiva e aplicabilidade industrial) seriam patenteáveis.

No fim dos anos 90 a Câmara Técnica de Recurso do Escritório de Patente Europeu expressou de forma inequívoca no caso IBM Program Product (T 1173/97), gozarem os programas de computador de proteção patentária como invenções, desde que comprovado um resultado técnico novo, cujo efeito deveria ser encontrado necessariamente além da mera interação física normal entre programa e computador.

Em outras palavras, e especialmente desde a decisão „Controlling Pension Benefits System (T931/95), a letra do Art. 52, 2 e 3 estaria circunscrita ao âmbito da mera interação entre programa e máquina (aqui a essência da disposição referente aos programas de computador “considerados como tais”), âmbito onde não caberia qualquer consideração relativa a possíveis efeitos técnicos. Porém, os resultados desta interação, portanto efeitos para além dos limites da mera interação física, poderiam gerar um progresso técnico (incremento do estado da técnica), o que possibilitaria a proteção patentária. Nesse sentido não se tratariam de patentes de programas de computador, mas antes de patentes implementadas por programas de computador.

Por meio deste artifício argumentativo o exame procedido pelo Escritório de Patentes Europeu deslocou-se da disposição restritiva do Art. 52, 2 e 3 para a letra do Art. 56 da Convenção mencionada, um dispositivo que tem por objeto a atividade inventiva.<sup>29</sup>

Com base neste artifício interpretativo foram concedidos diversos registros de patente para programas de computadores.

A linha interpretativa adotada pelo Escritório em tela gerou grande insegurança jurídica que, por sua vez, não sem o auxílio de um forte lobby do setor econômico interessado, acabou inaugurando um movimento a favor da supressão no artigo 52 da Convenção de Munique da exclusão de proteção a programas de computador. Em consequência foi apresentado ao Parlamento Europeu uma proposta de Diretiva relativa à patenteabilidade dos inventos que implicam programas de computador que, por sua vez, após cinco anos de

<sup>29</sup> Artigo 56.º Actividade inventiva Uma invenção é considerada como envolvendo actividade inventiva se, para um perito da técnica, não resultar de uma maneira evidente do estado da técnica. Se o estado da técnica abranger documentos citados no artigo 54.º, parágrafo 3, estes não são tidos em consideração para a apreciação da actividade inventiva.

acirrada discussão, acabou por ser rejeitada em 2005 pela maioria esmagadora dos parlamentares europeus.<sup>30</sup>

As considerações que levaram a rejeição da Diretiva em tela reportam a preocupações de cunho político-econômico. Possibilitar a proteção do programa de computador por meio do direito de patente significaria ampliar um espectro de proteção que, ao invés de incentivar o processo concorrencial – este é o fim imediato da proteção garantida ao programa de computador: fomento da concorrência de superação inovadora – acaba por fomentar tendências de concentração no mercado específico, o que vai em oposição ao fim político-econômico almejado.

Ao rejeitar a Proposta de Diretiva o Parlamento Europeu reagiu de forma determinada contra uma tendência real do mercado de programas de computador europeu: trata-se de um mercado concentrado, caracterizado pelo emprego da propriedade intelectual não só como estratégia de defesa de mercado, mas ainda como instrumento de desincentivo dos esforços inovadores por parte de pequenos e médios agentes econômicos.<sup>31</sup> Some-se a isto o fato da União em questão ter objetivos político-econômicos bem delineados, quais sejam a integração econômica dos mercados e a livre circulação de mercadorias, o que inclui, especialmente na forma de organização econômica moderna da sociedade de informação, bens intelectuais do tipo programas de computador.

### **3.3. Índia<sup>32</sup>**

Na Índia, o setor de software tem sido responsável pela atração de grandes volumes de investimento estrangeiro. A implementação de uma indústria tecnicamente sofisticada como a do software, que exige um ambiente propício à inovação, em um país em desenvolvimento como a Índia pode ser atribuída a estratégias de livre mercado concomitantes a uma série de políticas estatais. Essa emergência de novos atores desenhando trajetórias que diferem amplamente do que se recomendava pelas estratégias de

---

<sup>30</sup> Vide o processo em [http://ec.europa.eu/prelex/detail\\_dossier\\_real.cfm?CL=pt&DosId=172020](http://ec.europa.eu/prelex/detail_dossier_real.cfm?CL=pt&DosId=172020)

<sup>31</sup> Vide aqui o caso Microsoft.

<sup>32</sup> Agradecemos à Marília Monteiro pelo auxílio na pesquisa para este item.

desenvolvimento das instituições internacionais e do que se viu em países desenvolvidos é de fato algo a se ressaltar e observar.

Há mais de trinta anos, assim como o Brasil, a Índia tem desenvolvido políticas para o setor de informática. No país, o objetivo de auto-suficiência levou a uma forte valorização do papel da ciência e da tecnologia em sua estratégia de desenvolvimento desde a independência, preocupação enfatizada na política industrial e no campo da educação, bem como expressa na criação de novas instituições e na implementação de novas diretrizes de políticas públicas, como as estabelecidas na “*computer policy*” do Departamento de Eletrônica (*Department of Electronics - DOE*). A partir de 1984, políticas de liberalização econômica passaram a ser implementadas, afetando também as diretrizes para fomento das Tecnologias de Informação e, já na década de 90, a Índia foi o primeiro país em desenvolvimento a considerar o software como uma “área de confiança”, devido a sua demanda por habilidades, intensidade de trabalho e potencial de lucro no comércio internacional<sup>33</sup>.

Frente a esse mercado vibrante, a questão da proteção da propriedade intelectual também tem sido alvo de debate nas políticas indianas. Na Índia, assim como o Brasil, o “software *per se*” é considerado como uma obra literária, protegido pelo regime de direito autoral, abrangendo, portanto, a proteção de formas de expressão, não de ideias, planos ou conceitos<sup>34</sup>. Assim dispõe o artigo 10, parágrafo primeiro do TRIPS: “Computer programs, whether in source or object code, shall be protected as literary works under the Berne Convention (1971)”<sup>35</sup>. Apesar de que a expressão “*per se*” não seja prevista em nenhum tratado internacional, é um conceito similar ao da Convenção Europeia de Patentes, que exclui “programas de computador como tal” do escopo da patenteabilidade.

---

<sup>33</sup> Lateef, Asmah. Linking up with the global economy: A case study of the Bangalore software industry. Discussion Paper. New Industrial Organization Programme. International Institute for Labour Studies. 1997. <http://www.oit.org/public/english/bureau/inst/papers/1997/dp96/index.htm>

<sup>34</sup> Denis Borges Barbosa. A protecao do software, 2001. p10.

<sup>35</sup> Programas de computador, seja em código fonte puro ou em suporte físico, devem ser protegidos como obras literárias sob a Convenção de Berne” (tradução própria).

A partir desse conceito, por exclusão, tem-se inferido erroneamente que programas de computador que alcançam “soluções técnicas” são sujeitos à proteção patentária. Sob essa hipótese, da mesma forma que acontece hoje no Brasil, o escritório de patentes indiano lançou uma consulta visando revisar o Manual de Práticas e Procedimentos de Patenteamento referentes a patentes de programas de computador. O processo gerou amplas críticas apontando que a concessão de patentes de software no cenário indiano significaria que a indústria local passaria a ser privada do acesso ao conhecimento em que ela se baseou<sup>36</sup>. Críticas pontuais focavam principalmente na amplitude do termo “soluções técnicas” e sua relação com o conceito de software *per se*, mas também em razão do Escritório de Patentes estar tentando traçar diretrizes interpretativas para uma questão que deveria ser estabelecida em lei, frente a ampla consulta pública, dadas as consequências reais desse tipo de determinação, que tem grande potencial de afetar a produção de novos programas e a vida cotidiana de programadores, engenheiros, usuários e empreendedores do setor.

De fato, o esboço de manual indiano combinou princípios obtidos sem métodos de interpretação confiáveis a partir de análises de casos europeus e americanos reunindo todos os casos que mencionavam “soluções técnicas” e concluiu que o termo seria um fator chave para avaliar pedidos de patentes de software<sup>37</sup>. Destaca-se, contudo, que a concessão de patentes baseada nesse “efeito técnico” não se enquadra nos requisitos de patenteabilidade e o escritório foi fortemente criticado por não considerar um padrão mais rigoroso, de forma a evitar uma enxurrada de patentes de software. Não obstante, no Brasil o INPI veio, por meio do documento “auxiliar no exame técnico de pedidos de patente envolvendo invenções implementadas por programa de computador”, trazer a mesma abordagem ao tema, usando o conceito amplo de solução técnica para permitir a proteção patentária do software, excluindo apenas a proteção de “programa de computador em si” e, assim, assumindo que um programa de computador pode ser considerado invenção se “a solução proposta apresentar um efeito técnico e venha a resolver um problema encontrado na técnica, que não diga respeito unicamente ao modo como este programa de computador é

<sup>36</sup> <http://cis-india.org/a2k/blog/arguments-against-software-patents>

<http://cis-india.org/openness/publications/software-patents/ALF%20Position%20Paper%20Draft%20Patent%20Manual.pdf/view>

<sup>37</sup> Alternative Law Forum position paper Draft Patent Manual



escrito, isto é, ao programa de computador em si”. Observamos, portanto, o escritório brasileiro de patentes incorrendo na mesma solução que foi fortemente criticada no caso do escritório indiano e abrindo margem, portanto, a uma enxurrada de patentes de software não justificáveis pelos requisitos de patenteabilidade, com o potencial de criar um campo minado para a inovação do setor.

Por fim, o processo indiano colocou 2 perguntas chaves em pauta: porque deveríamos permitir o patenteamento de software? Quem se beneficiaria desse tipo de patentes? A conclusão apontada pelo Centre for Internet and Society, centro de pesquisa indiano preocupado com o pluralismo digital, é que o patenteamento de software auxilia três categorias de entidades: 1) grande empresas de software que já detêm um portfólio significativo de patentes de software; 2) empresas que não desenvolvem software, mas que apenas comercializam patentes ou ajuízam ações em razão de suas patentes (“*patent trolls*”) e 3) advogados da área de patentes<sup>38</sup>. Ou seja, exclui-se da lista as razões de ser do sistema de patentes, fundamentada no interesse público e fomento à inovação.

#### **4. As diretrizes do INPI**

##### **4.1 Interpretação do inc. V do art. 10 da Lei de Propriedade Industrial**

O texto “procedimentos para o exame de pedidos de patentes envolvendo invenções implementadas por programa de computador” (ao qual nos referiremos por “diretrizes”), de uso interno do INPI para orientar seus examinadores, apresenta-se como mero resumo e uma explicação tanto dos termos da Lei 9279/96 quanto do Ato Normativo 127/97.

O documento busca legitimar suas orientações para pedidos de patentes relativas a programas de computador o inc.V do Artigo 10 da LPI que exclui da proteção patentária o “*programa de computador em si*”. A interpretação é baseada em raciocínio a *contrario sensu*: se o que se exclui é o programa de computador em si, então, a contrario, há alguma dimensão do programa de computador, que não aquela do programa em si, que poderia ser

---

<sup>38</sup> <http://cis-india.org/a2k/blog/arguments-against-software-patents>

objeto de patente. Ao buscar essa dimensão, o documento não é claro e aponta para hipóteses distintas.

Primeiramente, parece apontar que o inciso V do art. 10 da LPI “refere-se aos elementos literais da criação, tal como o código fonte, entendido como um conjunto organizado de instruções escrito em uma linguagem computacional. Enquanto conjunto de instruções, código ou estrutura, o programa de computador em si não é considerado invenção e portanto não é objeto de proteção por patente por ser mera expressão de uma solução técnica, sendo intrinsecamente dependente da linguagem de programação”. Em seguida, afirma que “conjunto de instruções em uma linguagem, código fonte, ou estrutura de código fonte, mesmo que criativas, não são consideradas invenções, ainda que proporcionem efeitos técnicos”. Ou seja, aqui o INPI parece delimitar como “programa de computador em si” a expressão literal em código fonte do programa ou estrutura do código fonte.

Mais adiante fala em criação industrial implementada por programa de computador e a considera invenção “se a solução proposta apresentar um efeito técnico e venha a resolver um problema encontrado na técnica, que não diga unicamente ao modo como este programa é escrito, isto é, ao programa de computador em si”. A linha divisória parece então estar entre programas que produzam efeito técnico e programas que não trazem um efeito técnico, que seriam os programas em si. E considera efeito técnico não só aquele que manipula grandezas físicas e gera um produto físico, mas também o controle de grandezas físicas para gerar produtos virtuais, ou ainda o controle de grandezas abstratas para gerar produtos virtuais. O documento não esclarece o que seriam produtos virtuais, passando a elencar uma série de exemplos, muitas vezes desconexos, que serão comentados mais adiante.

Ora, ambas as hipóteses de interpretação são inadmissíveis frente a legislação pátria, a começar pela incorreta aplicação do raciocínio a contrario sobre o inc. V do art. 10 da LPI.

O raciocínio a contrario, como alerta Ulrich Klug, deve ser empregado com cautela pelo jurista, uma vez que pode implicar a falácia de afirmação do antecedente e apenas constitui inferência válida quando se está diante de exceção a uma regra, de modo que, ausente a condição de exceção, vale a solução normativa geral. Mas o inc. V do art. 10 da LPI, que exclui a patente do software em si, não pode ser lido como exceção à “regra geral de patente de programa de computador”, simplesmente porque não há tal regra. Pelo contrário, a regra geral, como vimos, é de proteção do programa de computador pelo regime autoral. Na medida em que a Constituição Federal em seu art. 5º, incs. XXVII e XXIX (que separam a proteção autoral das obras da proteção patentária da invenção) adota o princípio de especificidade da proteção à criação intelectual, não se pode admitir que o inc. V do art. 10 estabeleça incidência concomitante de patentes ao software, que já goza de proteção autoral em legislação específica (Lei do Software). Portanto, em consonância com o princípio constitucional de especificidade, a única interpretação aceitável para o inc. V do art. 10 é a exclusão de toda e qualquer forma de proteção patentária ao programa de computador, em qualquer de suas dimensões. Nessa leitura, o art. 10, inc. V, ao mencionar que o “programa de computador em si” não é patenteável apenas separa e distingue os sistemas de proteção quando se está diante de invenções que possam envolver programas de computador. Vale dizer, um programa de computador pode fazer parte de processo que leve a um efeito técnico industrial, o que significa, então, que há dois objetos a serem protegidos: a invenção que leva a um efeito técnico e o programa de computador. Para o inc. V do art. 10 da LPI esses dois objetos não se confundem: de um lado, a invenção ou criação com efeito técnico industrial pode ser protegida pelo sistema de patentes e, de outro, o programa de computador é protegido pelo regime autoral.

Fica claro, assim, que o legislador, ao incluir a expressão “em si” após “programa de computador” no inc. V do art. 10 da LPI, opôs o programa de computador ao efeito técnico industrial que possa ser advindo da aplicação do programa em algum processo inventivo e não opôs uma dimensão do programa de computador em expressão literal a alguma outra dimensão que poderia ser abrangida por patente.

Com isso, exclui-se a primeira hipótese de interpretação segundo a qual “o programa de computador em si” abrangeria apenas o código fonte ou a estrutura do programa, até porque a Lei de Software, já em seu primeiro artigo, inclui na própria definição de programa de computador a sua expressão seja em código fonte, seja em código de máquina (lei 9609/98, art.1º).

Também a segunda hipótese não pode ser admitida, pois de todo programa de computador se espera uma aplicação prática e solução para um problema. Quando esse “efeito técnico” não se limita ao mundo físico, mas alcança efeitos virtuais então se abre espaço para a descrição de qualquer programa de computador como não sendo o “programa em si”. Programa de computador com aplicação técnica física ou virtual, mormente quando não se define com clareza o significado de “virtual”, pode ser qualquer programa de computador, o que, na prática, tornaria letra morta a vedação de patente para o programa de computador em si.

As diretrizes trazem, ademais, algumas incoerências que também dão abertura para que os peticionários de patentes explorem a obtenção de patentes indevidas. Afirma-se, por exemplo, que *“se o método matemático é aplicado para se obter uma solução para um problema técnico, este objeto (produto ou processo) poderá ser considerado invenção desde que os efeitos resultantes sejam técnicos e não puramente matemáticos ou abstratos”* (Seção 2.2). Ao que parece, as diretrizes querem aqui introduzir a ideia que o que se restringe no Artigo 10 não são as invenções naqueles campos, mas sim aquelas que têm efeito técnico nesses campos. Aceitando-se essa visão, no entanto, o mesmo deveria valer para o software, o que automaticamente excluiria várias patentes citadas como supostamente válidas no texto, tais como aquelas envolvendo otimização, gerenciamento de arquivos, transmissão de dados, compressão de dados, gerenciamento de bancos de dados e protocolos de comunicação de dados (pg. 11).

O texto ainda procura interpretar esses incisos de outra forma: *“Criações que envolvam métodos matemáticos podem ser consideradas invenções quando aplicadas a problemas*

*técnicos, por manipularem dados que constituem a representação de objetos concretos”.* Aqui, a “aplicação a problemas do mundo físico” seria a justificativa para a interpretação de patentabilidade. No entanto, as diretrizes parecem se contradizer, pois afirmam que *“um método de criptografia que utiliza dados abstratos de forma específica e tem como resultado um produto virtual, as chaves de segurança, é aceitável como invenção, pois resolve um problema de garantia de segurança a um dado em um canal de comunicação”.* Dado que uma chave de segurança é um resultado abstrato obtido através de um método matemático implementado por um programa de computador, resta-nos apenas imaginar a que tipo de criação, aos olhos do INPI, não se aplica a proteção patentária.

É evidente que o legislador quis restringir o escopo de patentes sobre concepções abstratas, científicas, matemáticas e computacionais porque elas dariam margem a patentes de âmbito exageradamente amplo, na medida em que essas áreas têm papel propedêutico em quase todas as áreas da criação e produção industrial. Ainda assim, no entender exposto nas Diretrizes do INPI, a concessão de proteção relacionada a essas áreas perde seu caráter pernicioso quando se limita a apenas uma aplicação de conceitos desse tipo a um dado contexto, identificado pelo efeito técnico fora dessas áreas “problemáticas”. Além de inconsistente com o texto da lei, essa visão ainda sofre de um mal maior, que é o de ignorar a necessidade de ato inventivo para a concessão de patente.

De fato, as teorias científicas e as concepções abstratas e matemáticas, por sua própria natureza, são rotineiramente utilizadas para solucionar os mais diversos problemas. A transposição de conceitos abstratos, científicos ou matemáticos já conhecidos para as mais diversas áreas faz automaticamente parte do estado da técnica de qualquer área do conhecimento. A aplicação desse princípio é, praticamente por definição, isenta de atividade inventiva.

O mesmo se aplica ao software. As funções típicas de um software são:

1. Utilizar um método abstrato ou matemático de forma a produzir algum resultado. Por exemplo, calcular trajetórias de projéteis, estimar o volume de água a ser consumido no próximo ano em uma dada cidade ou prever as condições climáticas.
2. Implementar um processo qualquer em software. Por exemplo, gerenciar o controle de estoque de uma empresa, auxiliar no atendimento de clientes via telemarketing, identificar cadeias genômicas similares em diferentes espécies ou controlar os movimentos de um braço mecânico em uma fábrica.

No primeiro caso, não pode haver proteção patentária ao processo utilizado (já que se trata de método abstrato ou matemático). Tomando o exemplo do sistema de previsão de tempo, poderia haver proteção patentária para outras partes do processo, tais como um mecanismo de medição de temperatura atmosférica à distância usando raios laser para detectar a amplitude da vibração das moléculas do ar. Observe-se que, nesse caso, a proteção se aplicaria a parte do invento totalmente estranha ao software.

No segundo caso, só pode haver proteção patentária se o processo em si for inovador dentro do estado da arte da área de aplicação em questão, e não dentro do estado da arte no campo do software. Não se trata de transposição de um conceito abstrato ou de um método computacional para outro campo de aplicação, mas do contrário: trata-se da implementação em software de uma ideia inovadora pertencente a outro contexto.

É mister observar que, evidentemente, pode haver atividade inventiva sobre como realizar os objetivos almejados através do software. Mas a criação nesse caso reside no campo abstrato, e não no campo da aplicação; assim sendo, está fora do escopo da proteção patentária. Na página 13 das diretrizes há discussão sobre a hipótese de um robô que implementa um procedimento previamente conhecido. Fica claro, naquele texto, que pode haver proteção patentária caso o procedimento implementado pelo robô seja inventivo, mesmo que o procedimento seja previamente conhecido. No entanto, como discutido aqui, uma situação análoga no contexto do software não poderia ser alvo de proteção, pois a inovação estaria restrita ao campo do software e das idéias abstratas.

Em suma, em forte contraste com as diretrizes, é nosso entendimento que o inciso V do Artigo 10 da LPI pretende apenas esclarecer que o regime de proteção ao software é o direito autoral, mas que o uso do software em invento não elimina a proteção do invento como algo independente do software que o implementa, daí a expressão “em si”. A proteção pode existir para inventos onde o software tem papel acessório, como ocorre também com fontes de energia ou outros componentes necessários ao invento; mas não pode existir quando o invento consiste apenas no software, ou quando o software está no cerne do invento. A proteção disponível e constitucionalmente delineada, nesse caso, é autoral.

#### **4.2 Equívocos específicos nos exemplos trazidos nas Diretrizes do INPI**

Ao longo do texto das Diretrizes do INPI são elencados uma série de exemplos de invenções implementadas por programas de computador supostamente patenteáveis.

Comentamos abaixo alguns os exemplos citados que, segundo o INPI “provavelmente” seriam objeto de proteção patentária:

- a) **controle da temperatura de um forno** - só poderia ser aceitável se a inovação está no mecanismo de medição ou modificação de temperatura. Por exemplo, se o sistema prevê o aumento da temperatura do forno através de pulsos de energia que elevem a temperatura momentaneamente a valores elevados de maneira a gerar um crescimento controlado da temperatura média. Esse sistema pode ser controlado por software (e, se os pulsos forem suficientemente rápidos, essa seria a única possibilidade), mas a inovação não se encontra no software. Por outro lado, se o controle da temperatura dependesse da estimativa da temperatura em diferentes momentos do tempo a partir de cálculos realizados com medições realizadas no passado, tratar-se-ia de um processo matemático e, portanto, não poderia haver proteção.

- b) **estabilização do comportamento dinâmico de um veículo ao longo de uma trajetória pré-estabelecida** - só poderia ser aceitável se a estabilização fosse obtida através de técnicas inovadoras, como uma asa de avião com formato moldável em tempo real ou um sistema de amortecedores que ativamente modificasse a posição das rodas em um carro. Sistemas desse tipo provavelmente dependeriam de software para funcionar, mas a inovação estaria no processo físico da asa ou do amortecedor, e não no software responsável por seu controle. Um avião ou carro convencional que tivesse sua trajetória controlada por computador não poderia ser objeto de proteção, já que a única inovação nesse caso consistiria em um modelo matemático do movimento implementado no software, o que invalidaria uma eventual patente tanto pelo conteúdo eminentemente matemático quanto pela implementação da inovação na forma de software.
- c) **um sistema de transmissão automática em veículos** - aplica-se aqui argumentação similar ao item anterior.
- d) **controle de impressão** - só poderia ser aceitável se a inovação do processo de controle fosse baseada em aspecto externo ao software, como por exemplo na deformação do material sobre o qual se apóia o material impresso finalizado. O software pode ser responsável pelo cálculo identificando a quantidade de cópias impressas a partir da deformação medida, mas a inovação está no uso do fenômeno físico para a medição e não no processamento realizado pelo software, independentemente de sua complexidade.
- e) **controle de máquinas industriais** - só seria aceitável se processo de controle operasse de maneira inovadora; por exemplo, o sistema, ao invés de monitorar o funcionamento da máquina durante todo o tempo, identifica alguns estados como forçosamente levando a máquina ao próximo estado desejado. A partir dessa identificação, só esses estados precisam ser monitorados. Nesse caso, a inovação estaria na identificação desses mecanismos causais no equipamento, e não no software usado para identificar ou controlar esses mecanismos.



- f) **tratamento de imagem e de áudio envolvendo as grandezas físicas Amplitude e Fase** - aqui o texto parece querer fazer referência a operações matemáticas como transformadas de Fourier, que permitem tratar imagens e áudio como entidades matemáticas. Existem inúmeras patentes nos E.U.A. que, em última instância, se baseiam em técnicas desse tipo. No entanto, trata-se claramente de aplicar um modelo matemático (funções matemáticas como senos, no caso da transformada de Fourier) a um objeto do mundo físico, como uma imagem ou som. No nosso entender, patentes desse tipo simplesmente não são aceitáveis.
- g) **compressão de dados** - no caso de compressão de dados sem perdas, qualquer sistema consiste em possivelmente encontrar uma representação dos dados mais favorável à compressão (por exemplo, 7 bits ao invés de 8 para texto em inglês, ou aplicar um vocoder de fase sobre um sinal de áudio), identificar redundâncias e representar essa redundância de forma mais compacta. Todos esses procedimentos são eminentemente matemáticos e, exceção feita à mudança na representação dos dados, independem totalmente do tipo de dado; portanto, não fazem jus à proteção patentária. No caso de compressão de dados com perdas, existe um passo adicional que é o de avaliar quais partes dos dados podem ser descartadas. Por exemplo, partes de um sinal de áudio podem ser descartadas sem causar diferenças perceptíveis para um ouvinte humano, ou as cores de uma imagem podem ser irrelevantes para um sistema em um carro que detecta objetos em aproximação acelerada e notifica o motorista. Nesses casos, a ideia de utilizar uma determinada forma simplificada da informação no lugar da original pode ser objeto de proteção patentária, mas não o algoritmo computacional (ou o procedimento matemático, já que os dois são a mesma coisa) responsável por identificar a informação a ser descartada. Assim, não se pode oferecer proteção patentária a um novo algoritmo capaz de implementar uma forma conhecida de compressão já em domínio público.<sup>39</sup>

---

<sup>39</sup> Um algoritmo desse tipo também se aproxima de um modelo de utilidade, o que também o excluiria da proteção patentária, já que o Artigo 9 da LPI considera que só pode ser patenteado como modelo de utilidade “objeto de uso prático ou parte deste”.

- h) **criptografia** - tal como no caso da compressão de dados sem perdas, a criptografia não só é um processo eminentemente matemático como também é totalmente independente do tipo de dado a ser criptografado e, portanto, não tem nenhuma relação com o contexto de sua utilização. Qualquer uso de criptografia, portanto, não pode ser objeto de patente.
  
- i) **gerenciamento de bancos de dados** - qualquer sistema de gerenciamento de banco de dados opera de forma independente do contexto da aplicação e, portanto, não é passível de proteção patentária.
  
- j) **sistemas operacionais** - por definição, sistemas operacionais são totalmente restritos ao escopo do software e, portanto, não podem ser alvo de proteção patentária.
  
- k) **interfaces gráficas** (desde que não sejam meras apresentações de informações) - por não estarem diretamente dentro do escopo do programa de computador, podem ser objeto de patente. No entanto, convém observar que patentes desse tipo podem ter forte impacto sobre a interoperabilidade entre sistemas, o que pode ter resultados desinteressantes para a população em geral. Uma eventual adição à Lei tratando desse assunto (não apenas no contexto de sistemas computadorizados) poderia ser benéfica.
  
- l) **protocolos de comunicação de dados** - tal qual bancos de dados ou sistemas operacionais, protocolos desse tipo independem do campo de aplicação e, portanto, não podem ser alvo de proteção patentária, a menos que o protocolo seja implementado no nível do meio (por exemplo, sinais eletromagnéticos) e não como uma camada abstrata de software.

Uma outra amostra da inadequação das diretrizes é o exemplo da página 13: “No caso de um programa de CAD que a partir de uma lista de componentes eletrônicos determina qual o melhor traçado da trilhas condutoras em uma placa de circuito impresso que implementa

um circuito eletrônico desejado, uma reivindicação que pleiteie o método de roteamento destas trilhas baseado na hierarquia dos componentes otimizando o traçado constitui matéria patenteável”. Esse problema pertence à Matemática (mais especificamente, à teoria dos grafos) e, portanto, qualquer solução para ele consiste na aplicação de um conceito matemático, o que é vedado pela LPI.