

## “Programa de computador implementado por programa de computador”: a inventividade dos juristas e a liberdade dos programadores

Juliano Maranhão<sup>1</sup>

---

**Resumo:** O regime jurídico para exploração do programa de computador no Brasil é o de direito autoral. Neste artigo, resgato o fundamento desse regime, observando as diferentes dimensões do software (ideia, obra, produto, funcionalidade). Diante da recente iniciativa do INPI, mostro que a hipótese de adoção do regime de patentes para software choca-se frontalmente com as diretrizes constitucionais e à política nacional de informática, que prezam a autonomia tecnológica nacional e a difusão do conhecimento informático. Tais objetivos apontam para um ambiente de liberdade e cooperação entre programadores. Interpreto então a expressão “invenção implementada por programa de computador”, expressão criada pela *ars inveniendi* jurídica como aquilo que não seria o “programa de computador em si” e que, portanto, seria patenteável. Alerto que essa distinção, embora conceitualmente possível, é de difícil aplicação prática e pode levar a uma série de situações em que o objeto da patente seria o “programa de computador implementado pelo programa de computador”, colocando-se em risco a liberdade dos programadores.

**Palavras-chave:** programa de computador, direito autoral, patentes

*Abstract: Software is protected in Brazil by the copyright system. In this paper, I recollect the foundation of this normative solution with attention to the different software dimensions (idea, content, product, functionality). Before the recent initiative by INPI, I show that the hypothesis of software patents is uncongenial to the principles of the Brazilian Constitution and the national policy for informatics, which protect technological autonomy and diffusion of knowledge. Such values are better satisfied in an environment of freedom and cooperation among software programmers. Then I interpret the expression “invention implemented by software” created by the legal *ars inveniendi* as a category distinct from the “software as such” and thus, subject to patents. I warn that though this distinction is conceptually possible it creates several hard cases of application where the object of patent may be the “software implemented by software”, thus hindering the freedom of software engineers and developers.*

*Keywords: software, copyright, patents*

---

<sup>1</sup> Professor Associado da Faculdade de Direito da USP e Coordenador Jurídico do Centro de Competência em Software Livre da USP – CCSL/USP

## 1. Introdução: dimensões do software, a proteção autoral e a provocação patentária

O regime jurídico pátrio de proteção ao *software* é o de direito autoral, 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, 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 ideia 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 (*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, sobleva-se uma ou outra dimensão do software, o que traz posicionamentos distintos sobre seu enquadramento jurídico, que oscilam dentre os dois extremos: tomar o programa como conhecimento, questionando-se até mesmo se o programa mereceria qualquer tipo de apropriação; privilegiar o aspecto de funcionalidade, 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 e afastando-se prudentemente dos extremos, a solução consagrada no âmbito internacional foi a do *software* como obra artística sujeita a proteção autoral.<sup>2</sup> Ou seja, a ideia não é apropriável, o suporte de armazenagem não importa e a aplicação

---

<sup>2</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

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 a mesma aplicação técnica e a 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 a mesma programação pode servir a utilidades distintas, notadamente quando for combinada com 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, a 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>3</sup> chama a atenção para o caráter cumulativo das criações nesse ramo de atividade: um mesmo software pode conter centenas de programas diversos e as criações não são iniciativas isoladas, mas sequências de derivações ou variações sobre um mesmo tipo de aplicação técnica, que pode se mostrar mais ou menos eficiente para esse ou para aquele fim. Assim, pode-se ter a criação de um caça-níquel eletrônico, mas que admite diferentes softwares para gerar as combinações de resultados, com

---

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.

<sup>3</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>. Acessado em 23 de novembro de 2012.

maior ou menor eficiência e que podem utilizar trechos de programação um do outro. Tornar cada criação objeto de patente seria impraticável muito provavelmente significaria 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>4</sup>

Mais recentemente, em alguns países, foi levantada a discussão sobre a possibilidade de se admitir 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.

O Escritório de Patentes (USPTO) dos E.U.A., pelo próprio pioneirismo da indústria naquele país foi o primeiro que se tem notícias a conceder tais patentes, que em certa medida receberam confirmações diante de questionamentos judiciais na década de 80. Com base em precedentes judiciais da Suprema Corte, o USPTO publicou “manual de exame” disciplinando situações em que se equipara o próprio programa de computador a invenção. O resultado foi o aumento vertiginoso dos pedidos e da concessão de patentes de software naquele país.<sup>5</sup> Porém, o exame do comportamento dos agentes nesse setor de atividade, após a concessão de patentes, revela que: a) a patente provavelmente não é o principal veículo de incentivo à inovação, pois poucas empresas de software buscam patentes, b) a patente leva a concentração de mercado, pois as patentes de softwares ficam concentradas nas mãos de grandes empresas, com índice muito baixo de registros por pequenas e médias empresas e c) eleva-se o custo e o risco da inovação, não só pelo custo para a patente mas principalmente em função das disputas judiciais em torno das patentes de software.<sup>6</sup> Por outro lado a proliferação de pedidos, com a concessão de patentes descabidas para uma série de tecnologias triviais, expôs as limitações do sistema de patentes, com a criação de um movimento nos E.U.A para questionamento de uma série de registros obtidos.<sup>7</sup>

Na Europa a proteção do programa de computador também é autoral e a Convenção de Munique sobre a Patente Européia os exclui expressamente daquilo que pode ser considerado

---

<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, p.43.

<sup>5</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>6</sup> Ver 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 23 de novembro de 2012.

<sup>7</sup> Para mais informações, veja o projeto “Patente Busting”. Disponível em <https://www.eff.org/patent-busting>. Acessado em 23 de novembro de 2012.

invenção (Art. 52, 2), exclusão que abrange apenas o programa de computador quando “considerado como tal”.<sup>8</sup> Essa expressão foi a saída para os defensores da proteção patentária, dentre os quais o Escritório de Patentes Europeu-EPE, desenvolverem uma exegese para que algum aspecto do software, que não ele “em si” pudesse ser abrangido pelas patentes. Esse aspecto, como sugerido pelo EPE em 1986, seria o “efeito técnico” (caso “VICOM” T 208/84), o que abriu as portas para o deslocamento da proteção para a *dimensão de funcionalidade* do programa de computador.

Seguindo este raciocínio, um processo determinado, mesmo que reportando a elementos não patenteáveis (algoritmos) poderiam gerar efeitos que, por sua vez, se traduziriam em 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, situação na qual desloca-se o tema para o Art. 56 da Convenção mencionada, que tem por objeto a “atividade inventiva”.<sup>9</sup> Com isso a proteção seria para a “invenção implementada por programa de computador” e não para o programa de computador “considerado como tal” ou “em si”.

---

<sup>8</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 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>9</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.

Com base nessa interpretação foram concedidos diversos registros de patente para programas de computadores, culminando com uma proposta de Diretiva relativa à patenteabilidade dos inventos implementados por programas de computador<sup>10-11</sup> dirigida ao Parlamento Europeu em 2002. Após cinco anos de aprofundado e amplo debate público, a proposta foi votada e rejeitada pelo Parlamento Europeu.<sup>12</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.

O Brasil começou a trilhar caminho parecido ao dos europeus. Após a concessão de patentes ao “efeito técnico” de um programa de computador, o INPI submeteu a consulta pública uma proposta de Diretrizes com o objetivo de “*auxiliar no exame técnico de pedidos de patente envolvendo invenções implementadas por programa de computador*”. O conteúdo das “Diretrizes do INPI” confere ampla patenteabilidade a tais invenções, abrangendo os efeitos técnicos do software tanto no mundo físico quanto no virtual, estando o programa embarcado em máquina ou não 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 ou “efeito técnico” é uma das dimensões de qualquer software, a *dimensão de funcionalidade*, o que mostra ser a distinção entre “*invenção implementada por programa de computador*”, patenteável, em oposição ao “*programa de computador em si*”, não patenteável, um jogo semântico perigoso, no qual a inventividade conceitual dos juristas, para demarcar com precisão uma distinção no mundo teórico, pode trazer riscos à liberdade dos programadores, dado que, na prática, a distinção é de difícil aplicação. O objetivo deste artigo é reler essa interpretação jurídica e discutir o alcance semântico dessas expressões à luz das diretrizes constitucionais para o desenvolvimento tecnológico do País, em particular da informática, e chamar a atenção para a dificuldade de separação, com base nos próprios exemplos trazidos pela proposta de Diretrizes do INPI.

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

---

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

<sup>11</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>12</sup> Common Position (EC) n. 20/2005.

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>13</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 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

---

<sup>13</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.

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>14</sup>

Em função desse efeito, os produtos inovadores vencedores acabam concentrando grande parte do mercado (*winner takes most*),<sup>15</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>16</sup> Em primeiro lugar, não

---

<sup>14</sup> Shapiro, Carl e Varian Hal R., *Information Rules: A Strategic Guide to the Network Economy*, Boston, Harvard Business School Press, 1999.

<sup>15</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.



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>17</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 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>18</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>19</sup>

---

<sup>16</sup> Economides (2001, *op.cit.*).

<sup>17</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).

<sup>18</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)

<sup>19</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>

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 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 objetivando estimular 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 desenvolvimento do software licenciado em regime 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, o mesmo pode trazer restrições excessivas em indústria marcada pela sequencialidade e cumulatividade, como já apontado acima. As patentes não tiveram um papel relevante no desenvolvimento da tecnologia relacionada ao software nos últimos 50 anos e a abertura para a patenteabilidade nas últimas duas décadas nos E.U.A. aparentemente não trouxe incentivos relevantes para as empresas de softwares, principalmente para as pequenas e médias empresas, o que indica ser esse mecanismo 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 concentrar 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. As Diretrizes do INPI**

#### **3.1 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*” (“Diretrizes do INPI”), submetido a consulta

pública e que deverá ser 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*: 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 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, o que se oporia ao programa em linguagem de máquina.

Mais adiante, porém, 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 estar, então, entre programas que produzam efeito técnico e programas que não tragam 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 afrontam 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, de um lado, a proteção autoral das obras e, de outro, a 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, uma vez que 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”.

Em suma, em contraste com as Diretrizes do INPI, 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. Ou seja, não pode haver “programa de computador implementado por programa de computador” sujeito a patente como invento industrial. A proteção disponível e constitucionalmente delineada, nesse caso, é autoral.

### **3.2 Dificuldades específicas nos exemplos trazidos nas Diretrizes do INPI<sup>20</sup>**

As Diretrizes do INPI apresentam uma série de exemplos que revelam a dificuldade e o risco de se patentear um “programa de computador implementado por programa de computador”. As Diretrizes, em uma passagem, interpretam os incisos do art. 10 da LPI, admitindo que *“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, lê-se, em seguida, 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, fica difícil identificar a que tipo de criação, aos olhos do INPI, não se aplicaria a proteção patentária.

É evidente que o legislador quis restringir o escopo de patentes sobre concepções abstratas, científicas, matemáticas e computacionais, dado o risco de se obstar o desenvolvimento científico e, por consequência, o próprio desenvolvimento tecnológico, na medida em que essas áreas têm papel propedêutico para toda criação e produção industrial. 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.

---

<sup>20</sup> A elaboração deste item foi feita em colaboração com Fábio Kon, professor titular do IME/USP e Presidente do CCSL/USP, Nelson Lago, gerente técnico do CCSL/USP e Paulo Meirelles, pesquisador do CCSL/USP.

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. 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, de uso de software para produção de um resultado, tomando o exemplo do sistema de previsão de tempo, poderia haver proteção patentária para outras partes do processo que não o gerenciamento da previsão pelo software, 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 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 do INPI, há discussão sobre a hipótese de um robô que implementa um procedimento previamente conhecido. Fica claro, naquele texto, que pode haver proteção patentária caso a introdução do robô para a realização do procedimento 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, simplesmente aumentando a eficiência daquele procedimento, dada a sua realização por um computador, a partir de um software. O mesmo ocorre se o robô citado utiliza o



software que traduz o procedimento conhecido. O robô para concretizar o procedimento, se a ele trouxer inovação, pode ser objeto de patente, mas não o software.

Ao longo do texto das Diretrizes do INPI é elencada uma série de exemplos de invenções implementadas por programas de computador supostamente patenteáveis. Comentamos abaixo alguns deles:

i) Grandezas físicas em um processo em que um produto físico é gerado:

A transformação ou a redução de um produto a um estado diferente ou para um novo produto pode ser um indicativo de que uma criação implementada por programa de computador constitua invenção, embora não seja o único indicativo.

Exemplos: controle da temperatura de um forno; estabilização do comportamento dinâmico de um veículo ao longo de uma trajetória pré-estabelecida; um sistema de transmissão automática em veículos; controle de impressão; controle de máquinas industriais; (Consulta Pública do Instituto Nacional da Propriedade Industrial (INPI) sobre os Procedimentos para o exame de pedidos de patentes envolvendo invenções implementadas por programas de computador, p. 10),

As hipóteses de manipulação de grandezas físicas para gerar ou transformar um produto físico permitem uma separação mais clara do papel acessório do software em relação à “invenção” propriamente dita, mas alguns cuidados devem ser tomados, valendo destacar, para cada exemplo, o que segue:

a) *Controle de temperatura de um forno*: poderia ser aceita a patente desde que dirigida a uma inovação no mecanismo de medição ou modificação da temperatura, mas não no software que serve como ferramenta para realizar a medição ou controlar a modificação, mesmo que o processo não possa ser desenvolvido de outra forma que não por um software. 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, nem o software eventualmente empregado para o cálculo nem o método de cálculo seriam patenteáveis, este último por se tratar de um processo matemático.

b) *estabilização do comportamento dinâmico de um veículo ao longo de uma trajetória pré-estabelecida.* A patente poderia ser admitida desde que a estabilização envolvesse uma técnica inovadora, que seria o objeto da proteção. Assim, por exemplo, 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. Tais sistemas provavelmente dependeriam de software para funcionar com eficiência, mas a inovação estaria no processo físico da asa ou do amortecedor e não no software, acessório, responsável por seu controle. Já no caso de avião ou carro convencional que tivesse sua trajetória controlada por computador, não se vislumbra de que forma poderia haver proteção patentária, já que a inovação nessa hipótese estaria no modelo matemático do movimento representado no programa de computador, não havendo algo além da interação normal entre o software e o computador inserido no veículo (a inovação teria que residir em algo externo a essa interação, na relação entre o comando dado pelo computador e mecanismos físicos no veículo). O mesmo vale para o sistema de transmissão automática em veículos.

c) *controle de impressão.* Da mesma forma, a patente somente poderia ser aceita se a inovação do processo de controle fosse baseada em aspecto externo ao software, como, por exemplo, no mecanismo de deformação do material sobre o qual se apóia o material impresso finalizado. O objeto da patente seria esse mecanismo e não o programa de computador que o gerencia. 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.

d) *controle de máquinas industriais.* O controle de máquinas por software é rotineiro na indústria, de forma que o software com tal função ou a própria função não podem ser objeto de patente. A patente somente poderia ser cogitada e, mesmo assim, com cautela, se o processo de controle operasse de maneira inovadora e referente a uma inovação externa à programação; por exemplo, o sistema, em vez 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 tomados como relevantes no

campo de aplicação industrial pelo equipamento, mas não no software usado para identificar ou controlar esses mecanismos.

“ii) Grandezas Físicas em um processo em que um produto virtual é gerado:

O processamento de dados que representam características físicas de um objeto (dimensão, cor, atraso) gerando um produto intangível (vídeo, música, imagem).

Exemplos: tratamento de imagem e de áudio envolvendo as grandezas físicas Amplitude e Fase;” (Consulta Pública do Instituto Nacional da Propriedade Industrial (INPI) sobre os Procedimentos para o exame de pedidos de patentes envolvendo invenções implementadas por programas de computador, p. 10),

e) *tratamento de imagem e de áudio envolvendo as grandezas físicas Amplitude e Fase:* aqui o texto parece 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. Sendo assim tais patentes não seriam aceitáveis.

“iii) Grandezas abstratas em um processo em que um produto virtual é gerado:

O processamento que não manipula diretamente forças da natureza ou proporciona a transformação da matéria, tal como em (i), tampouco representa dados físicos, tal como em (ii), que não se enquadrem nos incisos do Art. 10 da LPI e que proporcionem efeitos técnicos tais como métodos que otimizam recursos de *hardware* ou que confirmam maior confiabilidade e segurança.

Exemplos: compressão de dados, criptografia, gerenciamento de bancos de dados, sistemas operacionais, interfaces gráficas (desde que não sejam meras apresentações de informações), protocolos de comunicação de dados.” (Consulta Pública do Instituto Nacional da Propriedade Industrial (INPI) sobre os Procedimentos para o exame de pedidos de patentes envolvendo invenções implementadas por programas de computador, pp. 10-11),

f) *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 em vez 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, incorporados em programas de computador, são eminentemente matemáticos e, exceção feita à modificação na representação dos dados, independem totalmente do tipo de dado, de modo que não poderiam ser objeto de 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, desde que inovadora, de utilizar uma determinada forma simplificada da informação no lugar da original, reservando-se os efeitos almejados, pode ser objeto de proteção patentária, mas não o algoritmo computacional (ou o procedimento matemático) 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>21</sup>

g) *criptografia e gerenciamento de bancos de dados*: 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. O mesmo vale para gerenciamento de bancos de dados, o que exclui tais processos do objeto de proteção patentária.

h) *sistemas operacionais*: por definição, sistemas operacionais são totalmente restritos ao escopo do software e, portanto, não podem ser alvo de proteção patentária.

---

<sup>21</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”.

i) *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.

j) *protocolos de comunicação de dados*: da mesma forma que bancos de dados ou sistemas operacionais, protocolos desse tipo independem do campo de aplicação e, portanto, como processos matemáticos, não poderiam ser alvo de proteção patentária, a menos que o protocolo refira-se e seja implementado com inovação tecnológica no meio de comunicação (por exemplo, sinais eletromagnéticos) e não como uma camada abstrata de software.

Por fim, outro ponto duvidoso das Diretrizes do INPI 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*”. O problema descrito pertence à matemática (mais especificamente, à teoria dos grafos) e, portanto, qualquer solução para ele consiste na aplicação de um conceito matemático, que não poderia ser objeto de patente.

#### **4. Conclusão**

Embora seja possível separar conceitualmente “*invenções implementadas por programa de computador*” do “*programa de computador em si*” com referência ao “efeito técnico” do software quando aplicado a algum processo ou produto industrial, na prática, cria-se abertura para a patenteabilidade do próprio programa de computador, uma vez que a funcionalidade nada mais é do que uma dimensão de qualquer software, assim como o conceito matemático que lhe é subjacente ou sua expressão literal em linguagem natural ou de máquina. Isso é ainda mais crítico quando o “efeito técnico” considerado pode abranger também efeitos no mundo virtual, manipulando-se grandezas físicas ou abstratas, como fazem as Diretrizes do INPI, com o agravante de não definir com clareza o que significa “virtual”, limitando-se a indicar exemplos bastante duvidosos e ambíguos.

Com isso, corre-se o risco de se equiparar programa de computador à invenção, criando, portanto, a possibilidade de patentear um “*programa de computador implementado por programa de computador*”, ou seja, patentear o “*programa de computador em si*”, ao arripio da legislação. Na prática, vimos como é difícil manipular e aplicar a distinção, bastando observar as discussões judiciais no âmbito internacional, ou mesmo as hipóteses e exemplos duvidosos contidos no próprio texto das Diretrizes do INPI.

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 no *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. Assim, dado que as “Diretrizes do INPI”, em indicações bastante ambíguas, parecem equiparar, em algumas situações, a invenção implementada por *software* ao próprio *software*, o documento teria a pretensão de papel normativo, i.e. de alteração da Lei, o que é inadmissível em sede de resolução emitida por órgão da Administração Pública.

Importante considerar 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* que se afirme como invenção implementada por *software* é apreciado minuciosamente para que se reconheça uma exceção ao regime padrão de proteção autoral ou para que se circunscreva o que esta fora do âmbito do software em si. O ônus recai sempre sobre aquele que pretende a patente sobre o software. Assim, na medida em que o INPI admitir invenções implementadas por programas de computador “em regra” patenteáveis por resolução infralegal, 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.

É desnecessário lembrar que 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 Comunidade Européia debateu-o por cerca de cinco anos e definiu-o em sua esfera legislativa suprema, o Parlamento Europeu. Foram dois anos de discussão de texto teórico sobre a matéria e mais três anos de discussão de proposta normativa.

Assim, seja pela relevância do tema, seja pela complexidade de demarcação da linha entre o que seria e o que não seria patenteável em relação a programas de computador, discuti-lo em âmbito de documento normativo infralegal não parece ser o caminho apropriado.