

Estudo sobre Propriedade Intelectual do Software

Maycon Viana Bordin, Helton Eduardo Ritter

¹Bacharelado em Sistemas de Informação – Sociedade Educacional Três de Maio
Caixa Postal 153 – CEP 98.910-000 – Três de Maio – RS – Brasil

{heltonritter,mayconbordin}@gmail.com

Resumo. *A propriedade intelectual faz parte da vida de muitas pessoas, sua proteção é de grande importância a sociedade, pois estimula a inovação e melhora a vida das pessoas. O software também faz parte deste cenário, sendo protegido pelos direitos autorais, assim como obras literárias, artísticas e científicas. Nas últimas décadas muito tem se falado sobre patentes de software, uma forma de se proteger as funcionalidades de um software, algo não compreendido na lei de direitos autorais. E por mais que alguns dos motivos para a proteção de softwares por patentes possam ser legítimos, em grande parte dos casos elas tem apenas servido como entrave para a inovação e benefício dos seus detentores.*

Abstract. *The intellectual property is part of many peoples life, and its protection has great importance in society, because it stimulates innovation and improve peoples life. Software is also part of it, being protected by copyrights, just like works in literature, arts and science. In the last decades there has been much talk about software patents, a way to protect software's functionality, something excluded from the copyrights law. And as much as some reasons for software patents are legitimate, in the majority of cases they're just slowing innovation and helping the patent holders.*

1. Introdução

A propriedade intelectual compreende qualquer criação da mente, como invenções, obras literárias e trabalhos artísticos, além de símbolos, nomes e imagens utilizadas no comércio [World Intellectual Property Organization 2011].

Ela pode ser dividida em dois ramos do direito: direitos autorais e a propriedade industrial. O primeiro compreende obras literárias, artísticas, programas de computador e a cultura como um todo. Já o segundo compreende principalmente as marcas, patentes, indicações geográficas e nomes de domínio, possuindo um caráter mais utilitário, com a finalidade de solucionar problemas [Lemos 2011].

Para a [World Intellectual Property Organization 2011], a promoção e proteção dos direitos autorais garantem o progresso e bem estar das pessoas, através da criação de novos trabalhos tecnológicos e culturais, do comprometimento com o investimento contínuo na inovação e o crescimento da economia.

A propriedade intelectual também serve para o propósito de dar o controle sobre o acesso às criações aquele que detém o exercício do direito das mesmas. O detentor dos direitos pode ser tanto o seu criador como uma empresa que adquiriu os direitos sobre a criação, dando a este o direito de usá-la da forma que desejar, desde que não esteja infringindo a lei, e de impedir o uso da criação por terceiros [Lemos 2011].

2. Patentes

Segundo a Lei da Propriedade Intelectual [Brasil 1996], “é patenteável a invenção que atenda aos requisitos de novidade, atividade inventiva e aplicação industrial” e também “é patenteável como modelo de utilidade o objeto de uso prático, ou parte deste, suscetível de aplicação industrial, que apresente nova forma ou disposição, envolvendo ato inventivo, que resulte em melhoria funcional no seu uso ou em sua fabricação”. Patentes servem como incentivo para inovação, garantindo ao criador dela exclusividade sobre a criação, recompensa merecida pelo tempo e recursos investidos.

É importante ainda deixar clara as diferenças existentes entre descoberta, criação e invenção. A invenção consiste na criação intelectual de efeito técnico ou industrial, mas o simples ato de criar algo através do intelecto não pode configurar invenção, para que isto ocorra esta criação precisa necessariamente ser uma nova solução para um problema técnico existente, como fora descrito no parágrafo anterior. Já a descoberta se difere de uma invenção pelo fato de não criar algo antes inexistente, mas sim revelar algo existente na Natureza.

3. Patente de Software

Em primeiro lugar, a Lei nº 9.279/96 (Lei da Propriedade Industrial) deixa claro no artigo 10º o que não é considerado invenção nem modelo de utilidade, e especificamente no item V deste artigo está escrito “programas de computador em si” [Brasil 1996].

Ainda no artigo 10º da mesma lei, está definido como não sendo uma invenção ou modelo de utilidade “descobertas, teorias científicas e métodos matemáticos”, e para [Rezende and Lacerda 2005] os métodos matemáticos também dizem respeito a programas de computador, pois as tecnologias de informação e comunicação são acentadas em fórmulas e métodos matemáticos.

Deixando de lado os benefícios econômicos por hora, que outro motivo poderia levar um indivíduo a patentear seu software, tendo em vista que a Lei do Direito Autoral (lei brasileira nº 9.610/98) e a Lei de Software (lei brasileira nº 9.609/98) já o protegem?

A verdade é que a Lei do Direito Autoral protege a expressão de uma idéia, no caso o código fonte em si, e não a funcionalidade que este programa de computador provê. Enquanto que uma patente protege a aplicação prática de uma idéia.

A Lei de Software considera como violação dos direitos autorais a “reprodução, por qualquer meio, de programa de computador, no todo ou em parte, para fins de comércio, sem autorização expressa do autor ou de quem o represente” [Brasil 1998]. Entretanto, no Artigo 6º da mesma lei, a respeito do que não constitui ofensa aos direitos do titular de programa de computador, no inciso III, está escrito:

[...] a ocorrência de semelhança de programa a outro, preexistente, quando se der por força das características funcionais de sua aplicação, da observância de preceitos normativos e técnicos, ou de limitação de forma alternativa para a sua expressão

Ou seja, se um programa é desenvolvido e possui funcionalidades semelhantes a outro programa que já existia no mercado, isto não é considerado infração dos direitos autorais, mesmo que o primeiro programa esteja pegando emprestado idéias deste último

programa. E ainda que estes sejam motivos suficientes para que um indivíduo busque proteção do seu software através de patentes, a Lei da Propriedade Industrial deixa claro que idéias não são patenteáveis e nem mesmo a Lei do Direito Autoral protege ideias [Barbosa 2011].

Entretanto, o INPI (Instituto Nacional da Propriedade Intelectual) tem concedido o registro de patentes para softwares sob a seguinte argumentação:

[...] o programa de computador em si, isto é, aquele que não apresenta um efeito técnico, é excluído de proteção patentária, ao passo que se tal programa altera tecnicamente o funcionamento da máquina em que é executado, este processo de controle ou a máquina resultante, pode configurar uma invenção patenteável [INPI 2011].

O problema desta diretriz seguida pelo INPI está no trecho que torna passível de patente o software que “altera tecnicamente o funcionamento da máquina em que é executado”, pois não há uma especificação clara do que se configura como alteração técnica de uma máquina. [Rezende and Lacerda 2005] fazem duras críticas com relação a esta diretriz seguida pelo INPI, inclusive destacando que foi através de uma interpretação semelhante que nos EUA, no caso *Diamond versus Diehr* em 1981 e em subsequentes casos, que se considerou que “o uso de um programa de computador num processo físico era insuficiente para tornar sua essência impatenteável”.

O principal motivo de debates com relação a interpretação que tem sido dada ao que pode ou não ser patenteado com relação ao software está exatamente no trecho que foi exposto acima a respeito da alteração técnica do funcionamento de uma máquina. [Rezende and Lacerda 2005] fazem uma relação entre uma máquina e o algoritmo nela executado com um violino e uma partitura, afirmando que mudanças na partitura não alteram o funcionamento técnico do violino, e o mesmo ocorre com a máquina.

[Rezende and Lacerda 2005] ainda colocam 16 pontos que confirmam suas idéias a respeito da relação entre o hardware de um computador e o software nele executado, e aqui é dado destaque aos dois principais:

1- Nenhum programa que faça uso de hardware, ou seja, que não esteja imerso (embedded) no hardware, altera tecnicamente o funcionamento do hardware onde executa. Tais programas apenas manipulam e interpretam símbolos representados por sinais binários (bits) que trafegam por e/ou se armazenam nesse hardware.

2- A função técnica do hardware é processar sinais. Sinais per se nada significam, porque não são símbolos. Sinais veiculam símbolos através de códigos. Portanto, o significado das seqüências de bits manipuladas por esses programas é função semiológica das camadas de código que a lógica e ontologia dos demais programas envolvidos permitem aos seus autores representar.

Uma interpretação mais sensata e um pouco mais clara é dada em [UFRGS 2011]:

Os programas de computador desenvolvidos estritamente para funcionar “embarcados” em máquinas ou equipamentos, normalmente gravados em “chips” integrantes das estruturas destes, podem ser objeto de proteção

via patente. Neste caso, o mercado não estará demandando o programa de computador “em si” e sim a máquina ou equipamento.

O trecho acima deixa mais claro que softwares só poderiam ser patenteados quando embarcados em um hardware, como roteadores, aparelhos meteorológicos ou médicos. Ainda assim, o que pode ser considerado um sistema embarcado também pode dar margem para interpretações.

O que se pode notar, entretanto, é que no Brasil a concessão de patentes de software ainda é muito tímida se comparada com países como os Estados Unidos, o que é um bom sinal, mas pode vir a mudar no futuro devido a pressões de empresas detentoras de patentes.

4. Patentes de Software e a Inovação

Nos EUA assim como no Brasil, programas de computador eram e ainda são protegidos por direitos autorais. Entretanto, até a década de 70, ideias, programas lógicos, algoritmos, sistemas, métodos e layouts não eram passíveis de proteção. Em 1981, no caso citado anteriormente, a Suprema Corte concedera uma patente para a empresa Diehr envolvendo um método de processamento de borracha controlado através de software. Através desta decisão criou-se uma regra para a concessão de patentes de software, regra essa que previa a existência de uma transformação física de matéria [Andrade et al. 2007].

Na década de 90 a situação ganhou nova interpretação da Suprema Corte, considerando patenteável softwares que ao invés de apresentarem idéias abstratas, manipulassem números que representassem grandezas concretas do mundo real, tendo como exemplo uma invenção relacionada à interpretação de sinais de um eletrocardiograma para prevenir arritmia [Andrade et al. 2007].

Em 1998, mais uma vez o conceito do que era ou não patenteável havia mudado após decisão sobre o caso *State Street Bank & Trust Co. vs. Signature Financial Group Inc*, fazendo com que os juízes negassem patentes apenas para leis da natureza, fenômenos naturais e idéias abstratas. E atualmente nos EUA entende-se que qualquer invenção capaz de produzir resultado útil, concreto e tangível pode ser patenteada [Andrade et al. 2007].

Com o relaxamento das regras para o patenteamento de software nos EUA estima-se que anualmente sejam aprovadas mais de 15.000 patentes de software [Ventura 2011]. Um dos problemas colocados por Dan Ravicher, diretor-executivo da Public Patent Foundation está no fato da USPTO (*United States Patent and Trademark Office*) ser financiada pela aplicação de patentes, ou seja, quanto mais patentes aprovadas maior o repasse de verbas [Lee 2011].

Em outro artigo ([Lee 2009]) é feita uma comparação com as obras literárias, imaginando uma situação onde prosas em inglês poderiam ser patenteadas, onde um autor poderia patentear um romance por ele possuir determinada reviravolta. Desta forma qualquer escritor teria de submeter suas obras a advogados especializados em patentes para garantir que a obra não infrinja nenhuma patente. Apesar das alegações de que essa limitação forçaria escritores a inovarem, a necessidade de compreender o sistema de patentes ou contratar alguém que o seja é inaceitável, para que seus pensamentos possam ser publicados.

O pior de tudo é que nos EUA existe uma verdadeira guerra de patentes, onde

empresas procuram concentrar o máximo de patentes como forma de se proteger de processos contra outras empresas. A inexistência de um índice para pesquisa de patentes também se tornou um problema grave, onde a maioria das empresas deixou de se preocupar com patentes na hora de desenvolver softwares e preferiu estocar patentes ou assinar acordos com várias outras empresas (como um acordo mútuo para não processarem umas as outras) [Lee 2009].

O problema é que tudo custa dinheiro, e para empresas pequenas que estão entrando no negócio não é viável arcar com licenciamento de patentes, assim como fazem outras empresas de maior porte. E no final das contas, o dispositivo que fora criado para incentivar a inovação está na verdade limitando ela ao colocar todas as empresas sob um campo minado de patentes e longos processos.

Com as brechas criadas nos EUA para a concessão de patentes de software um novo personagem surgiu, conhecido por *patent troll*. Estes são conhecidos também como entidades não-praticantes, ou seja, eles não fazem nada além de acumular patentes e procurar vítimas para processar. Segundo artigo do *The Washington Post* [Plumer 2011], um relatório da OECD (Organização de Cooperação e de Desenvolvimento Económico) mostrou uma queda na qualidade das patentes ao redor do mundo nas últimas duas décadas, e em contrapartida os escritórios de patentes estão cada vez mais abarrotados de pedidos de patentes, estas com invenções duvidosa e de baixa qualidade.

É aí que entram os *patent trolls*, pois eles não tem do que reclamar, algo comprovado em estudo da Universidade de Boston que estima que os *patent trolls* tenham custado a economia algo em torno de meio trilhão de dólares, isso apenas em ações judiciais. O estudo foi além e calculou o total em perdas nas ações das empresas processadas por entidades não-praticantes chegando a uma média de \$122 milhões por empresa ou um total de \$80 bilhões por ano. O mais interessante deste estudo é que 62% destas ações judiciais envolviam patentes de software [Plumer 2011].

5. Conclusão

A situação atual dos EUA mostra por si só as consequências de se permitir patentes de software, principalmente em casos como a guerra de patentes entre Google, Apple, Microsoft e Oracle. Programadores há um longo tempo alertaram sobre os riscos de patentes de software ([Epperly 1991]), pois elas desestimulam a inovação através da criação de custos proibitivos para empresas que estão iniciando, sem contar nos riscos que elas podem correr ao desenvolverem softwares.

Ao que tudo indica, patentes de software só estão beneficiando aqueles que estocam patentes em busca de lucros extraordinários através de ações judiciais. O Brasil, ao menos por enquanto, não está seguindo o mesmo caminho dos EUA e para o bem da inovação e da economia como um todo, isto deveria se manter como está.

Softwares não são iguais a outras invenções, os custos incorridos para o desenvolvimento de softwares são muito menores se comparado com carros, processadores ou outras máquinas. E mais, linguagens de programação são utilizadas para expressar conceitos matemáticos de forma semelhante que escritores utilizam o português para expressarem outros tipos de ideias.

O que não pode acontecer é a concessão de algo que tem se mostrado, na maioria

dos casos, nocivo e contrário aos ideais originais (de incentivar a inovação) simplesmente por pressões políticas e ganância, em detrimento da inovação e conseqüente benefício da sociedade, melhorando a qualidade de vida e desenvolvendo a economia.

Referências

- Andrade, E., Tigre, P. B., Silva, L. F., Silva, D. F., de Moura, J. A. C., de Oliveira, R. V., and Souza, A. (2007). Propriedade Intelectual em Software: o que podemos apreender da experiência internacional? *Revista Brasileira de Inovação*, 6(1):31–53.
- Barbosa, D. B. (2011). Quem é o dono da idéia? Denis Borges Barbosa Advogados. Disponível em: <<http://nbb.com.br/pub/propriedade18.pdf>>. Acesso em: nov. 2011.
- Brasil (1996). Lei nº 9.279, de 14 de Maio de 1996.
- Brasil (1998). Lei nº 9.609, de 19 de Fevereiro de 1998.
- Epperly, T. (1991). Against Software Patents. February 28, 1991. The League for Programming Freedom. Disponível em: <<http://www.progfree.org/Patents/against-software-patents.html>>. Acesso em: nov. 2011.
- INPI (2011). Perguntas Mais Frequentes - Patentes: Posso Patentear um programa de computador? Disponível em: <<http://www6.inpi.gov.br/faq/patentes/patentes.htm#m7>>. Acesso em: nov. 2011.
- Lee, T. B. (2009). The Case against Literary (and Software) Patents. August 28, 2009. TechKnowledge no. 125. Disponível em: <http://www.cato.org/pub_display.php?pub_id=11448>. Acesso em: nov. 2011.
- Lee, T. B. (2011). Ask Ars: Is serious patent reform on the horizon? Ars Technica. Disponível em: <<http://arstechnica.com/tech-policy/news/2011/08/ask-ars-is-serious-patent-reform-on-the-horizon.ars>>. Acesso em: nov. 2011.
- Lemos, R. (2011). Propriedade Intelectual. Fundação Getúlio Vargas.
- Plumer, B. (2011). Innovation's down, but patent trolls are thriving. The Washington Post. Disponível em: <http://www.washingtonpost.com/blogs/ezra-klein/post/innovations-down-but-at-least-patent-trolls-are-thriving/2011/09/21/gIQABGdKlK_blog.html>. Acesso em: nov. 2011.
- Rezende, P. A. D. and Lacerda, H. F. M. (2005). Computadores, Softwares e Patentes. II Conferência Latino-americana e do Caribe sobre Desenvolvimento e Uso de Software Livre da UNESCO. Setembro de 2005. Disponível em: <http://www.cic.unb.br/~pedro/trabs/lacfree_rezende_lacerda.pdf>.
- UFRGS (2011). Registro de Softwares - Manual de Procedimentos. Disponível em: <http://www.sedetec.ufrgs.br/pagina/eitt/download/formularios/manual_proc_softwares.doc>. Acesso em: nov. 2011.
- Ventura, F. (2011). Por que software não deveria ser patenteado. Gizmodo Brasil. Disponível em: <<http://www.gizmodo.com.br/conteudo/por-que-software-nao-deveria-ser-patenteado/>>. Acesso em: nov. 2011.
- World Intellectual Property Organization (2011). What is Intellectual Property? Acesso em: nov. 2011.