


A IMPOSSIBILIDADE NORMATIVA DE PATENTE DE CÓDIGO FONTE NO BRASIL E O PROBLEMA DO PLÁGIO DE *SOFTWARE*

THE NORMATIVE IMPOSSIBILITY OF SOURCE CODE PATENT IN BRAZIL AND THE SOFTWARE PLAGIARISM PROBLEM

 **Charles Emmanuel Parchen**
Doutor/PUCPR e Unicuritiba/PR
charlesadv@gmail.com

 **Cinthia Obladen de Almendra Freitas**
Doutora/PUCPR-PPGD
cinthia@ppgia.pucpr.br

Resumo: Discute-se a ausência da possibilidade normativa de patente de programa de computador no Brasil e os efeitos decorrentes da falta de uma adequada proteção a tal obra imaterial que não pode ser comparada às demais obras artísticas e literárias, como atualmente ocorre na legislação brasileira. Como o *software* em si não é adequadamente protegido, há um incentivo ao plágio e contrafação. Isto submete a pessoa do originador do programa de computador a riscos e incertezas de ter que se valer dos meios comuns para proteção do seu patrimônio, sem que haja, contudo, garantia de exclusividade e primazia sobre sua obra intelectual, que é o código-fonte. Usando o método dedutivo e a pesquisa bibliográfica, propõe-se a necessidade de atualização da lei, para que esta se adeque, por exemplo, a países que possuem enfoque no empreendedorismo, tal como os Estados Unidos. Conclui o artigo que não é possível se alcançar um salutar empreendedorismo se às pessoas com olhar inovador e inventivo não forem asseguradas, pela lei, a exclusividade e primazia de sua criação por razoável lapso de tempo.

Palavras-chave: Direito de autor. Patente de *software*. Plágio.

Abstract: It is discussed the absence of the normative possibility of patenting a computer program in Brazil, along with the resulting effects of the shortage of an adequate protection to that immaterial work which cannot be compared to other artistic and literary works as it currently occurs in Brazilian legislation. Once the software itself does not have adequate protection, there is an incentive to plagiarism and counterfeiting. This submits the person of the originator of the computer program to risks and uncertainties of needing to draw upon common means to protecting their patrimony, without there being, however, guarantee of exclusivity and primacy over their intellectual work, which is the source code. By using the deductive method and the bibliographic research, the study proposes the necessity of updating the law, so that it will conform to, for example, countries with emphasis on entrepreneurship, such as the United States. The article concludes that it is not possible to achieve healthy entrepreneurship if, to people with an innovative and inventive eye, are not guaranteed by law, the exclusivity and primacy of their creation for a reasonable time lapse.

Keywords: Intellectual property. Software patent. Plagiarism.

Para citar este artigo (ABNT NBR 6023:2018)

PARCHEN, Charles Emmanuel; FREITAS, Cinthia Obladen de Almendra. A impossibilidade normativa de patente de código fonte no Brasil e o problema do plágio de *software*. *Revista Thesis Juris – RTJ*, São Paulo, v. 9, n. 1, p. 29-52, jan./jun. 2020. <http://doi.org/10.5585/rjt.v9i1.13169>.

1 Introdução

No Brasil, o programa de computador, sua propriedade intelectual e direito de autor são protegidos conforme dispõe a Lei nº 9.609/98 (Lei do *Software*), que versa sobre a proteção da propriedade intelectual de programa de computador e sua comercialização no país, bem como altera, atualiza e consolida a legislação que regula os direitos e obrigações relativos à propriedade industrial (Lei nº 9.279/96), trazendo o mecanismo da patente como proteção a tais direitos.

Conforme dispõem os artigos 8º e 9º da Lei 9.279/96, são patenteáveis tanto a invenção quanto o modelo de utilidade (BRASIL, 1996, p.1). Estes mesmos dispositivos da lei também impõem requisitos para a patente, tais como os da “novidade, atividade inventiva e aplicação industrial” para invenções, ou ainda que o objeto seja “susceptível de aplicação industrial”, ou “que apresente nova forma ou disposição, envolvendo ato inventivo, que resulte em melhoria funcional no seu uso ou em sua fabricação” para modelos de utilidade (BRASIL, 1996, p.1).

É interessante notar que a referida lei não menciona o conceito de invenção, mas apenas o que seria modelo de utilidade e também aquilo que não seria caracterizado como tal. É o que se observa da leitura do artigo 10º, inciso V que diz: “Não se considera invenção nem modelo de utilidade: V - programas de computador em si” (BRASIL, 1996, p.1).

Logo, no Brasil, embora protegidos no âmbito da propriedade intelectual e do direito de autor, programas de computador não são patenteáveis. Isso porque a expressão adotada pelo referido inciso V serve ao entendimento de que só é possível a patente de *software* se este servir ao funcionamento de um *hardware*. Portanto, a Lei 9.279/96 só permite a patente de programas de computador se este for a base inseparável para um dispositivo informático funcionar, sendo impossível que este desempenhe suas funções sem que o *software* o comande:

Porém, o modelo jurídico brasileiro só permite a patenteabilidade de programas de computador que possuam além do caráter de objeto técnico um efeito técnico, ou seja, uma atuação técnica interna ou externa que exceda à simples satisfação de funcionalidade em um sistema específico, mesmo assim deve ser dotada de novidade e atividade inventiva. Ainda, podemos dizer que a patenteabilidade é concedida somente às criações que combinam características de processos ou de produto com etapas de programas de computador (SALES, 2018, p.1).

Ao atrelar a necessidade de uma atuação do *software* sobre um aparato tecnológico, está a lei da propriedade industrial mitigando e limitando a proteção para aquele que origina o programa de computador. Para corroborar tal afirmação, pode-se salientar que a possibilidade de registro do código-fonte no Instituto Nacional da Propriedade Industrial (INPI) só assegura a comprovação de autoria sobre ele, mas não serve a tutelar corretamente situações em que haja

deturpações, uso indevido e modificações do programa de computador valendo-se de código-fonte de *outrem*.

A falta da possibilidade de patente sobre o *software* em si favorece a concorrência desleal, sem falar na ocorrência de plágio e contrafação. Como exemplo de tais práticas vedadas pelo direito brasileiro, pode-se citar situações em que ex-colaboradores, mais especificamente, programadores, analistas e gerentes de projetos de informática, apropriam-se dos códigos-fonte da empresa a qual pertenciam e utilizam-no em outra com setor de atuação idêntico ou semelhante, plagiando assim sistemas completos ou partes de sistemas computacionais.

Outro exemplo é a costumeira ocorrência de cópias ou modificações baseadas em personagens famosos de jogos digitais, como a que ocorreu entre os aplicativos *Flappy Bird* e o mundialmente conhecido jogo *Mario Brothers* da empresa de tecnologia Nintendo, caso este abordado adiante no presente artigo.

A impossibilidade de patente sobre o *software* em si, portanto, sujeita a pessoa do originador ao provável risco de não conseguir provar sua “paternidade” sobre o código-fonte, haja vista que pela Lei nº 9.609/98 é facultativo o registro do programa de computador (parágrafo 3º do artigo 2º). Outra dificuldade advém do fato de que a tutela que a lei dá, de 50 anos¹, ao programa informático conta-se, conforme dispõe o parágrafo 2º do artigo 2º da referida Lei: “[...] a partir de 1º de janeiro do ano subsequente ao da sua publicação ou, na ausência desta, da sua criação” (BRASIL, 1998, p.1). Mas sem obrigação de registro, como comprovar com certeza e segurança jurídica, quando houve a publicação ou ainda, a efetiva criação do *software*?

Desta maneira, sempre haverá dúvida: quem originou o programa de computador antes? Logo, quem plagiou quem? Sem registro, como determinar facilmente qual é o programa original e qual a cópia, portanto, qual deve ser eventualmente condenado pelo Judiciário a pagar indenização ou *royalties* (direito de uso) e compelido a retirar do mercado o *software* plagiado?

As vias comuns da proteção ao direito de autor e de propriedade, previstas na Constituição da República, nas leis especiais e no Código Civil brasileiro claramente não são aptas a ensejar a melhor interpretação a estes tipos de casos modernos em que se discute propriedade de bens imateriais e tecnológicos que mudam tão rapidamente quanto a velocidade do mundo digital.

Diversos e nefastos efeitos socioambientais são oriundos não só da falha da legislação em vigor querer equiparar o programa de computador às demais obras artísticas e literárias, mas

¹ Art. 2º, § 2º: “Fica assegurada a tutela dos direitos relativos a programa de computador pelo prazo de cinquenta anos [...]” (BRASIL, 1998, p. 1).

também da inexistência de uma norma que dê a devida e especializada proteção ao programa de computador. Inclusive as razões pela qual não se deve equiparar o programa de computador às demais obras artísticas e literárias são técnicas e discutidas ao longo deste texto.

Os prejuízos materiais e morais decorrentes do uso sem autorização de *software* e de suas indevidas modificações são problemas que envolvem o plágio e a contrafação. E a gravidade dos danos perpetrados à sociedade contemporânea pela falta de uma adequada proteção normativa ao programa de computador passa, inclusive, pelo inadequado método pelo qual tais práticas são detectadas no Brasil e também pelo desincentivo que as leis existentes – na medida em que há pouca segurança jurídica e proteção – ocasionam a empreendedores de tecnologia, *startups* de aplicações e outros. Isto sem falar na falta de sustentabilidade do modelo normativo vigente, que necessita ser urgentemente revisto para se adequar às legislações de países como os Estados Unidos e Alemanha, nos quais ao *software* é permitida a concessão de patentes.

O presente artigo utiliza do método dedutivo e o procedimento da pesquisa bibliográfica, jurisprudencial e de legislação para alertar para o fato de que ao Estado brasileiro incumbe uma mudança de paradigma normativo para se atualizar ou modificar a Lei 9.279/96, tornando-a mais adaptada à realidade da sociedade contemporânea. Entende-se ser necessário que a lei garanta de forma expressa e inequívoca a possibilidade de patente de programas de computador por si só, ou seja, independentemente de estarem atrelados a outros processos ou servirem de base ao uso de *hardware*. Isso porque na medida em que é o código-fonte a razão pura, técnica e simples do *software* existir, é necessário assegurar ao seu criador, a exclusividade e primazia, por intermédio da patente.

Somente desta forma um programa de computador estará completamente protegido contra a concorrência desleal, o plágio e a contrafação, pois não só estas práticas vilipendiam o direito de autor e de propriedade, mas também, pela falta de patente, surgem outros problemas, tais como a dificuldade de comprovar quando o *software* foi publicado ou quando foi efetivamente criado e utilizado pela primeira vez, ou ainda, entregue à empresa que o contratou, entre outras diferentes situações.

2 Os programas de computador, a proteção da lei brasileira e uma breve descrição da proteção nos Estados Unidos da América

O conceito normativo de programa de computador pode ser encontrado no artigo 1º da Lei 9.609/98, que diz:

Art. 1º Programa de computador é a expressão de um conjunto organizado de instruções em linguagem natural ou codificada, contida em suporte físico de qualquer natureza, de emprego necessário em máquinas automáticas de tratamento da informação, dispositivos, instrumentos ou equipamentos periféricos, baseados em técnica digital ou análoga, para fazê-los funcionar de modo e para fins determinados (BRASIL, 1998, p.1).

Um ponto de inflexão é cabível aqui, para criticar o fato de que referido conceito encontra-se claramente defasado e necessita ser atualizado, na medida em que, contemporaneamente a maioria dos *softwares* tem por base o *download* do arquivo para dentro de um dispositivo eletrônico com acesso ao programa de forma instantânea e automática a partir de servidores espalhados pelo globo (por meio da nuvem computacional) que tem a capacidade de prover, de maneira totalmente imaterial, a possibilidade de aplicação prática do programa de computador sem que haja suporte físico para o mesmo. É claro que este suporte físico existe na origem e passará a existir no destino a partir da instalação “remota”, mas não haverá transferência, entrega, venda ou outra modalidade de suporte material específico que contenha o *software* a ser instalado.

Um melhor conceito de programa de computador, baseado nas Convenções de Berna e Paris diz: “*software* é entendido como o programa de computador, ou escrito destinado a processamento de dados, compreendendo todo o conjunto de instruções para o processamento, produção e interpretação e transferência de textos, manuais, codificações, dentre outros” (WACHOWICZ, 2004, p.71). Esta definição do citado autor, apesar de antiga (datada do ano de 2004), ainda é relevante do ponto de vista da abrangência dos aspectos utilitários do *software*.

Segundo Frederick P. Brooks Jr. (1986, p.3), todo *software* possui quatro elementos essenciais: complexidade, conformidade, mutabilidade e invisibilidade. É complexo porque nenhuma outra construção humana se assemelha ao programa de computador (BROOKS JR., 1986, p.3). Tem conformidade porque é o suporte operacional que deve se adaptar ao *software*, uma vez estando este conforme os requisitos do ambiente em que é desenvolvido² (BROOKS JR., 1986, p.3). Possui mutabilidade porque o *software* está constantemente sujeito a pressões por mudança e pode ser fácil e infinitamente maleável, podendo ser modificado para atender novas necessidades ou especificidades (BROOKS JR., 1986, p.3-4). É invisível porque a realidade do *software* não está inteiramente enquadrada no espaço e mesmo estruturado, permanecerá não visualizado, porque o código-fonte não é de conhecimento dos comuns, mas somente de especialistas em programação de computadores (BROOKS JR., 1986, p.4-5).

² Entende-se como ambiente de desenvolvimento a linguagem de programação, o compilador e demais elementos técnicos (BROOKS JR, 1986, p. 3).

Já o código-fonte (*source code*) pode ser conceituado como sendo: “Source code (also referred to as source or code) is the version of software as it is originally written (i.e., typed into a computer) by a human in plain text (i.e., human readable alphanumeric characters)” (LINUX, 2006, p.1)³. Portanto, sendo o código-fonte a versão originalmente escrita do programa de computador, ambos serão tratados no presente artigo como sinônimos.

Feitas estas breves conceituações e caracterizações, importa dizer que os programas de computador, porque são frutos do intelecto humano, podem ser considerados como bens imateriais. Neste sentido, já decidiu o STJ:

BRASIL. Superior Tribunal de Justiça (1. Turma). Recurso Especial 39457/SP. Tributário - ISS - programas de computador - lei 7.649, art. 27 - decreto-lei 406/68, lista de serviços - item 24. 1. Os sistemas de computação, constituídos de programas, exprimem o resultado de atividade intelectual, de sorte que configuram bem imaterial e não mercadoria, a afastar a hipótese de incidência do ICMS. 2. A exploração econômica de programa de computador, mediante contratos de licença ou de cessão, sujeita-se a cobrança do ISS (item 24, da lista de serviços, anexa ao decreto-lei 406/68). 3. Recurso desprovido. Relator: Ministro Humberto Gomes de Barros, 13/08/1994: DJ 05.09.1994 p. 23040. Disponível em: https://ww2.stj.jus.br/processo/ita/listarAcordaos?classe=&num_processo=&num_registro=199300278037&dt_publicacao=05/09/1994. Acesso em: 20 fev. 2020.

Como tal, acaba sendo objeto de tutela no direito brasileiro comparado às demais obras artísticas, literárias e científicas. Neste sentido, a Lei 9.609/1998, em seu artigo 2º, especifica que “O regime de proteção à propriedade intelectual de programa de computador é o conferido às obras literárias pela legislação de direitos autorais e conexos vigentes no País, observado o disposto nesta Lei”(BRASIL, 1998, p.1).

A mesma Lei também trata da proteção do direito de autor de *software* e em seu artigo 7º, inciso XII, preconiza que: “São obras intelectuais protegidas as criações do espírito, expressas por qualquer meio ou fixadas em qualquer suporte, tangível ou intangível, conhecido ou que se invente no futuro, tais como: XII - os programas de computador” (BRASIL, 1998, p.1). Esta é, em resumo, a expressa disposição de lei que fundamenta especificamente o regime jurídico de proteção ao programa de computador na legislação brasileira.

É importante dizer que esta proteção é devida, segundo João Marcelo de Lima Assafim (2007, p.4) por causa da inovação:

A inovação, por demandar determinados níveis de investimento, implica em um regime de proteção com fundamento em direitos exclusivos, como por exemplo, aquele da patente para as invenções (criações técnicas de fundo com aplicação industrial) e aquele do direito de autor sobre programa de computador para o *software*[...].

³ Tradução livre: “código-fonte (também referido como fonte ou código) é a versão do *software* que é originalmente escrita (ex: feito em um computador) por um humano em texto simples (ex: ou em caracteres legíveis ao ser humano)”

Já em relação ao direito comparado, cumpre trazer para a discussão, ainda que brevemente, o tratamento dado ao programa de computador nos Estados Unidos. Neste país, a patente de *software* é plenamente possível, porque depois de décadas de discussões, o Congresso americano formou a Comissão de Novas Utilizações Tecnológicas de Obras Protegidas pelo Direito Autoral (CONTU).

Esta Comissão, por sua vez e em seu relatório final, sugeriu que a seção 117 do título 17 do *United States Code* (que é a codificação por assunto das leis gerais e permanentes dos Estados Unidos) fosse alterada a fim de estender ao programa de computador as mesmas proteções de direito de autor constantes na seção 106, além de estabelecer requisitos para a criação e comercialização de cópias de programa de computador, tornando explícito, ainda, que os programas de computador estão sob a proteção do direito autoral, o que ocorreu com a Lei pública nº 96-517 de 1980 da seguinte forma:

Limitations on exclusive rights: Computer programs

Notwithstanding the provisions of section 106, it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:

(1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or

(2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.

Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner (GOVINFO, 1980, p.14-15)⁴.

Veja-se que por consequência lógica das alterações promovidas pelo relatório que integrou o *United States Code* restou possibilitada a patente para o programa de computador, já que se estendeu ao *software* os mesmos direitos autorais e de invenção anteriormente previstos no mesmo Código e também ficou consignada expressamente a necessidade de autorização prévia do proprietário do programa de computador para qualquer transferência onerosa que decorra de adaptações de sua obra original:

⁴ Tradução livre: Limitações de direitos exclusivos: Programas de computador "Não obstante as disposições da seção 106, não é uma violação para o proprietário de uma cópia de um programa de computador ou autorizar a realização de outra cópia ou adaptação dessa programa de computador fornecido: "(1) que uma nova cópia ou adaptação seja criada como um passo essencial na utilização do programa de computador em conjunto com uma máquina e que ela não é usada em nenhuma outra maneira, ou "(2) que essa nova cópia ou adaptação seja para fins de arquivo somente e que todas as cópias arquivadas sejam destruídas no caso de a posse continuada do programa de computador deve deixar de ser legítimo. Quaisquer cópias exatas preparadas de acordo com as disposições desta seção pode ser alugada, vendida ou transferida de outra forma, juntamente com a cópia a partir da qual essas cópias foram preparadas, apenas como parte do arrendamento, venda ou outra transferência de todos os direitos do programa. As adaptações assim preparadas só podem ser transferidas com a autorização do proprietário dos direitos autorais.

Federal courts, and to a lesser extent the U.S. Patent and Trademark Office (USPTO), dramatically changed standards for patenting software-related inventions over the last three decades. During the 1970s, federal court decisions typically described computer programs as mathematical algorithms, which are unpatentable subject matter under U.S. law. Systems using software could be patented, but only if the novel aspects of the invention did not reside entirely in the software. At this time, the U.S. Congress considered the question of patenting software and instead opted to protect computer programs under copyright law (BESSEN; HUNT, 2004, p.3)⁵.

A razão de ser da proteção pela via da patente ao *software* nos Estados Unidos pode ser explicada pelo fato de que a cultura empreendedora sempre foi a tônica neste país. Conforme explicita Hilton Ricardo Rocha (2005, p.1):

A busca incessante por tecnologias, no final dos anos 70, levou ao crescimento das indústrias de Hardwares e Softwares, fazendo assim surgir um novo conceito de softwares chamado “Softwares Proprietários”. Os Softwares Proprietários eram, e ainda são, modalidade de comercialização do software onde o comprador não mais tem direito sobre os códigos fontes. Isto fez com que a margem de lucro entre os fabricantes de softwares aumentasse, fazendo com que esta prática fosse adotada por todos os outros fabricantes. A adoção, pelos fabricantes de softwares, do conceito de software proprietário levou ao aumento da concorrência entre eles, motivo este que instigou o setor a buscar mecanismos de proteção de propriedade intelectual, como direitos autorais e patentes. Neste momento, o código-fonte passou a ser protegido, pois ele é o próprio conhecimento do software e a própria vantagem competitiva entre os fabricantes.

Não à toa, esta é a disposição da seção 200 do capítulo 38 do *United States Code*, alterado pela Lei pública nº 96-517 de 1980, que passou a vigorar com a seguinte redação:

It is the policy and objective of the Congress to use the patent system to promote the utilization of inventions arising from federally supported research or development; to encourage maximum participation of small business firms in federally supported research and development efforts; to promote collaboration between commercial concerns and nonprofit organizations, including universities; to ensure that inventions made by nonprofit organizations and small business firms are used in a manner to promote free competition and enterprise; to promote the commercialization and public availability of inventions made in the United States by United States industry and labor; to ensure that the Government obtains sufficient rights in federally supported inventions to meet the needs of the Government and protect the public against nonuse or unreasonable use of inventions; and to minimize the costs of administering policies in this area (GOVINFO, 1980, p.5)⁶.

⁵ Tradução livre: Tribunais federais e, em menor grau, o Instituto de Marcas e Patentes dos EUA (USPTO), alterou drasticamente os padrões para patentear invenções relacionadas a software nas últimas três décadas. Nos anos 70, as decisões dos tribunais federais geralmente descreviam programas de computador como algoritmos matemáticos, que são um assunto não patenteável perante a lei dos Estados Unidos. Sistemas usando software podem ser patenteados, mas somente se os novos aspectos da invenção não residir inteiramente no software. Atualmente, o Congresso dos EUA considerou a questão de patentear software e optou por proteger o programa de computador sob a lei de direitos autorais.

⁶ Tradução livre: É política e objetivo do Congresso usar o sistema de patentes para promover a utilização de invenções resultantes de pesquisas ou desenvolvimento apoiados pelo governo federal; incentivar a participação máxima de pequenas empresas em esforços de pesquisa e desenvolvimento apoiados pelo governo federal; promover a colaboração entre interesses comerciais e organizações sem fins lucrativos, incluindo universidades; garantir que as invenções feitas por organizações sem fins lucrativos e pequenas empresas sejam usadas de maneira a promover a livre concorrência e o empreendedorismo; promover a comercialização e a disponibilidade pública de invenções feitas nos Estados Unidos pela indústria e trabalho dos Estados Unidos; garantir que o governo obtenha direitos suficientes em invenções apoiadas pelo governo federal para atender às necessidades do governo e proteger o público contra o uso não utilizado ou irracional de invenções; e minimizar os custos de administração de políticas nessa área.

Assim, vislumbrar a fundamental importância do código-fonte para a existência do programa de computador foi a peça-chave para a percepção de que a competição que tanto estimula a concorrência e tem o potencial de beneficiar a economia e a sociedade só poderia ocorrer se ao *software* fosse concedida a adequada e individuada proteção, garantida não só pela via do direito de autor mas também pelo direito de propriedade industrial, do qual a patente garante a exclusividade e primazia pela obra imaterial realizada, que é, *prima facie*, o código-fonte.

O resultado da permissão normativa americana se traduz em patentes de importantes programas de computador que são aplicáveis no mundo inteiro, tais como a do protocolo de comunicação pela Internet, denominado de TCP/IP, que segundo Adam Mossoff (2013, p.2), foi a razão precursora que permitiu ao *Google* receber fundo de investimento em capital e assim, fazer nascer a empresa, que inclusive possui várias patentes de algoritmos. Entre eles, o mais famoso e que constitui o mecanismo principal (*engine*) do referido buscador:

In fact, few people realize the vast numbers of valid and valuable patents on computer programs. The entire Internet rests on patented innovation in computer programs: the packet-switching technology used to transmit information over the Internet was patented by Donald Watts Davies (Patent No. 4,799,258). Robert Kahn and Vinton Cerf, the inventors of the TCP/IP packet-switching protocol, later patented their follow-on invention of a packet-switching version of a knowbot. (Patent No. 6,574,628). Larry Page and Sergey Brin patented their famous search algorithm when they were graduate students at Stanford, and such patented innovation was a reason why Page and Brin received venture-capital funding for their start-up company, Google (there are several patents, but Patent No. 6,285,999 is one of the core ones)⁷.

No Brasil, contudo, o cenário é diferente. Como já trazido na introdução, não há empreendedorismo que resista ao desestímulo da falta de uma efetiva proteção ao *software* do ponto de vista da possibilidade de sua patente, sendo que solução semelhante ao sistema americano já poderia ter sido adotada pelo legislador brasileiro até mesmo para se evitar problemas relacionados à configuração e constatação da ocorrência de plágio e de contrafação de programa de computador, práticas estas que acabam sendo mal combatidas também por conta do inadequado método usado para suas detecções, uma vez que são aplicadas técnicas voltadas às obras literárias.

⁷ Tradução livre: De fato, poucas pessoas percebem o grande número de patentes válidas e valiosas em programas de computador. Toda a Internet se apoia na inovação patenteada em programas de computador: a tecnologia de comutação de pacotes usada para transmitir informações pela Internet foi patenteada por Donald Watts Davies (Patente nº 4.799.258). Robert Kahn e Vinton Cerf, os inventores do protocolo de comutação de pacotes TCP / IP, mais tarde patentearam sua invenção subsequente de uma versão de comutação de pacotes de um *knowbot*. (Patente No. 6.574.628). Larry Page e Sergey Brin patentearam seu famoso algoritmo de pesquisa quando eram estudantes de graduação em Stanford, e essa inovação patenteada foi uma das razões pelas quais Page e Brin receberam financiamento de capital de risco para sua empresa iniciante, o *Google* (existem várias patentes, mas a patente 6.285.999 é uma das principais).

3 Problemas relacionados à impossibilidade de patente sobre softwares no Brasil

A comparação que a legislação brasileira faz ao *software* com as demais obras artísticas e literárias (inclusive quanto à impossibilidade de patente), pode levar à conclusão de que estaria o programa de computador suficientemente protegido e assegurado. Contudo, tal dedução é bastante equivocada, na medida em que a previsão normativa é claramente insuficiente para dar segurança jurídica àquele que reivindica a “paternidade” e primazia sobre o programa de computador.

As características e o conceito deste, já abordadas no capítulo anterior, por si só deveriam servir a alertar o legislador para o fato de que, ao adotar a solução simplista e minimalista da concessão de uma proteção comum do direito de autor e da propriedade sobre o *software*, a lei está, na verdade, tornando vulnerável a pessoa do seu originador (física ou jurídica) e permitindo a ocorrência de plágio e contrafação a partir do código-fonte. Ocorrências essas que por vezes não encontram solução jurídica adequada ou tem uma solução carente de eficácia prática.

O grande problema da comparação normativa entre *software* e demais obras artísticas e literárias é o de que, por lei (artigo 7^a da Lei 9.610/98), todas passam a ser consideradas “criações do espírito”. Ocorre que o legislador parece ter se esquecido de que programa de computador não tem nada de obra “moral” ou de “criação da alma” ou da sensibilidade do seu originador.

O *software* é pura linguagem binária e, portanto, impossível de carregar ou expressar emoção, sendo que o mesmo não pode ser dito sobre as obras literárias e artísticas. Trata-se, portanto, de pura aplicação da Ciência de Computação, ou seja, emprego de técnicas de programação, as quais, por sua vez, são disciplinadas pela Matemática e pelas Ciências Exatas; logo, lógicas e objetivas. Cabe explicar que linguagens de programação “não existem no vazio; são ferramentas para escrever sistemas de programação” (GHEZZI, 1985, p.18). E, “uma linguagem de programação é uma notação formal para a descrição de algoritmos que serão executados por um computador” (GHEZZI, 1985, p.49).

É importante também afirmar que, em sendo uma notação formal, toda linguagem de programação possui 02 (dois) componentes, a saber: sintaxe e semântica. Em poucas palavras, a sintaxe “é o conjunto de regras formais que especificam a composição de programas a partir de letras, dígitos e outros símbolos” (GHEZZI, 1985, p.49). Já as regras de semântica “especificam o ‘significado’ de qualquer programa, sintaticamente válido, escrito na linguagem” (GHEZZI, 1985, p.49) de programação. Por isso, não basta escrever o programa de

computador com uma sintaxe válida, o programa deve ter “significado” na linguagem aplicada para poder realizar o que se solicita que a máquina realize.

E ainda, que todo sistema computacional ou *software* é implementado de modo a satisfazer o projeto e as especificações do sistema, sendo o projeto deste por sua vez iniciado na fase de análise e especificação de requisitos que deverão ser atendidos pelo sistema. Do projeto resulta a documentação de especificação do sistema propriamente dito, a qual identifica todos os módulos que compõem o sistema, bem como suas interfaces.

A metodologia de projeto tem impacto sobre a escolha da linguagem de programação a ser utilizada na implementação do sistema. Portanto, entende-se que o programa de computador é resultado de um alto nível de abstração (análise, especificação de requisitos e projeto), antes de se tornar o objeto “programa de computador”. Na verdade, “um programa de computador é uma abstração da realidade” (GHEZZI, 1985, p.31).

A abstração, por sua vez, é um processo pelo qual são identificadas as qualidades e/ou propriedades relevantes do fenômeno que está sendo modelado, projetado. Ela permeia todas as fases do processo de desenvolvimento de um *software*, sendo a área de Engenharia de *Software*, na Ciência da Computação, a responsável por estudar e desenvolver métodos e técnicas voltados ao projeto de sistemas computacionais. A linguagem de programação é que permite tornar modelos abstratos em programas de computador.

Cabe destacar que para um mesmo problema, diferentes equipes de análise e programação desenvolverão códigos-fonte distintos. E, além disto, para um mesmo projeto de sistema, diferentes programadores aplicarão a linguagem de programação, suas funcionalidades, procedimentos e classes de maneira diferenciada, dando àquele programa de computador uma identidade. Portanto, a “paternidade” de um programa de computador não se refere somente à autoria (quem fez), mas as características intrínsecas do código-fonte (como fez).

Logo, é premissa básica da pesquisa realizada no presente artigo que o programa de computador está, portanto, muito mais ligado a necessidade de adoção e respeito a uma rigidez científica do que propriamente a alguma manifestação do “espírito humano” que lhe confira emoção, sentimento e vivacidade. O *software*, apesar de flexível em seu processo de criação, não pode ser entendido como um jogo de palavras sem sintaxe e semântica específicas e rigorosas, portanto, deveria ter sido tratado como tal pela legislação.

Errou também o legislador - ao comparar o programa de computador e obras literárias e artísticas - quando não percebeu que a ausência de elementos subjetivos para a linguagem de programação do *software* faz com que este possa ser facilmente alterado sem que seja possível

se detectar (ao menos pelos meios tradicionais atualmente usados) adequadamente a ocorrência de cópias ou modificações indevidas (FREITAS; BARBOSA, 2015, p.131).

De fato, é perfeitamente compreensível a *mens legis* de deixar de fora da possibilidade de patente os bens autenticamente fruto do “espírito humano”, tais como as obras literárias e artísticas. Ademais, a subjetividade deste tipo de criação facilita em muito a detecção e configuração do plágio e da contrafação por métodos de comparação textuais e uso de critérios também subjetivos, tais como a perquirição sobre se o autor teve ou não acesso à obra original, se é notável a forte semelhança de ideias ou ainda se a criação é independente e encontrada casualmente (MAXWELL; BOLGER; ZEGANE, 2009, p.3).

Mas quando a criação é eminentemente lógica, matemática e exata, o tratamento dado pelo legislador brasileiro acaba sendo comodista e simplista, portanto, equivocado. Afinal, não há como tratar no mesmo nível aquilo que é objetivo e o que é subjetivo, ainda mais quando os métodos de investigação de “paternidade” de *software* não podem ser os mesmos usados nas obras artísticas e literárias.

Desta forma, dúvidas que surgem no processo de investigação da origem do programa de computador ou serão solucionados de forma equivocada ou remanescerão insolúveis. Afinal, como atestar, pelos métodos subjetivos empregados em obras literárias e artísticas, qual é o programa de computador original e o que é cópia? Como concluir quem plagiou quem? Pode o investigador, sem o método adequado e correto, constatar a ocorrência de contrafação para fins de subsunção ao elemento-tipo do Código Penal ou, ainda, se são obras distintas que não se confundem e, portanto, não haveria qualquer ocorrência de crime?

O fato é que o trabalho é árduo e existem, embora não sejam objeto de análise do presente artigo, propostas de soluções visando aplicar métodos objetivos que pontuam a similaridade entre códigos-fonte por meio de um valor entre 0 e 100%, tais como os métodos e técnicas baseados em ferramentas automáticas de mineração de texto relacionada à área de aprendizagem de máquina (*machine learning*) (FREITAS; BARBOSA, 2015, p.135).

Embora alguns doutrinadores digam que a intenção do legislador brasileiro, ao não prever a possibilidade de patente de *softwares*, seja a de evitar a concentração de mercado (TIGRE; MARQUES, 2009, p.557), prevenir a ocorrência de monopólios e também a de dar ao programa de computador uma função social que estimule a concorrência (BARBOSA, 2010, p.121 e 122), é de se reafirmar que a ausência normativa de patenteabilidade vai justamente estimular a concorrência desleal, além de submeter a pessoa do originador do programa de computador aos riscos da impossibilidade de provar a “paternidade” e primazia do código-

fonte, vilipendiando assim, inclusive, a teleologia da norma existente, que é a especial proteção dos direitos de autor.

Alertando sobre os efeitos de uma inadequada proteção da propriedade intelectual, escreve Marcelo Krokosz (2015, p.39): “A completa banalização daquilo que se considera propriedade intelectual, artística, industrial etc., caracterizando assim uma situação generalizada de anarquia, fraude, pirataria, um vale-tudo decorrente da ausência de leis ou ausência de sua aplicação”.

Importante destacar também a afirmação da consultoria *Pricewaterhouse Coopers* (1999, p.14), que comentando a Lei 9.609/98, assevera: “noticie-se que, durante a fase de tramitação e discussão do projeto de lei de *software* no Senado Federal, alguns congressistas agasalharam a tese de que se deveria buscar a proteção desses programas na seara do direito da propriedade industrial”. E o mesmo comentário cita a fala do Senador Rubens Requião, à época:

Assim, a razão da crítica ao enquadramento irrestrito do esforço inventivo nesta área como direito autoral de obra literária, consideramos que certos aspectos do instituto jurídico de proteção ao programa de computador seriam mais bem regulados em analogia com o direito patentário, principalmente no que diz respeito ao prazo de proteção (PRICEWATERHOUSE COOPERS, 1999, p.14).

A analogia feita pelo legislador não poderia ser mais inoportuna. Isso porque sendo o registro do *software* junto ao INPI, algo facultativo e o código-fonte não estar adequadamente tutelado ou protegido por não se enquadrar nas hipóteses de patente de invenção (PI) ou como modelo de utilidade (MU) previstos pela Lei nº 9.279/96, abre-se inúmeras possibilidades para que alterações causadas por terceiros ocorram no programa de computador sem que sejam configurados como um ilícito do ponto de vista penal ou mesmo civil, infringindo a propriedade industrial.

Inclusive é de se criticar o inciso V do artigo 10º da referida lei, que assevera que o programa de computador em si não é considerado invenção ou modelo de utilidade. Principalmente pelo fato de que a própria lei sequer traz o conceito de invenção, apenas se limitando a trazer seus requisitos no artigo 8º, bem como sequer explicita o que seria modelo de utilidade, limitando-se a afirmar, no artigo 9º, que ele é patenteável.

Aqui cabe um parêntese para explicar o que é plágio e contrafação. Segundo o Departamento de Saúde e Serviços Humanos do Governo dos Estados Unidos, ele pode ser definido como: “Plagiarism is the appropriation of another person's ideas, processes, results, or

words without giving appropriate credit”⁸ (U.S DEPARTMENT OF HEALTH AND HUMAN SERVICES, 2019, p.1). Ainda, segundo o CNPQ (2011, p.4), plágio:

Consiste na apresentação, como se fosse de sua autoria, de resultados ou conclusões anteriormente obtidos por outro autor, bem como de textos integrais ou de parte substancial de textos alheios sem os cuidados detalhados nas Diretrizes. Comete igualmente plágio quem se utiliza de ideias ou dados obtidos em análises de projetos ou manuscritos não publicados aos quais teve acesso como consultor, revisor, editor, ou assemelhado.

Já o conceito normativo de contrafação está previsto no inciso VII do artigo 5º da Lei nº 9.610/98: “Para os efeitos desta Lei, considera-se: VII - contrafação - a reprodução não autorizada” (BRASIL, 1998, p.1). Portanto, para que haja contrafação e esta seja punida, é necessário haver dois requisitos: a reprodução e a não autorização (JESUS, 2015, p.42).

Considerando que o vocábulo reprodução é definido como sendo “imitação fiel/cópia” e fiel “aquilo que guarda fidelidade” e fidelidade como “exatidão” (PRIBERAM, 2019, p.1), fica evidenciado que o programa de computador resultante de código-fonte plagiado, no fundo poderá estar reproduzindo tudo com aparência de nada.

Explica-se: devido ao fato de não haver possibilidade de patente para o *software* em si no Brasil, tal ausência normativa pode servir a ensejar a interpretação e alegação - inclusive pelo Judiciário - de que não há exclusividade ou primazia a se respeitar, e mais, que não há imitação fiel ou cópia porque as alterações aplicadas no código-fonte original faz surgir um código-fonte diferente em existência, mas não em essência.

Ademais, a constatação pelo Judiciário brasileiro, de plágio e contrafação pode ser extremamente difícil, pois não obstante a expressa disposição de lei não ensejar ao *software* o direito de patente, existem inúmeras linguagens de computador que são postas à disposição e que, cada uma a sua maneira, possuem “pontos fortes e limitações” (FREITAS; BARBOSA, 2015, p.129).

Por esta razão é que os mesmos métodos de investigação de autoria em obras literárias e artísticas não podem ser empregados aos programas de computador. Repise-se que, segundo a doutrina, o método para detecção de plágio e contrafação em *softwares* necessita ser diferente diante da natureza distinta de tais bens. Neste sentido:

Como já mencionado, o plágio em software não pode ser tratado tal qual o plágio em trabalhos literários. Inicialmente, tem-se por premissa metodológica que o plágio em programas de computador necessita ser avaliado no código-fonte não podendo ser verificado por meio de comparações diretas entre telas e funcionalidades (FREITAS; BARBOSA, 2015, p.131).

⁸ Tradução livre: Plágio é a apropriação das ideias, processos, resultados ou palavras de outra pessoa sem dar o devido crédito.

Em resumo, o que se pode aferir de todas as afirmações anteriores é que a forma de tratamento normativo e também o método para detecção de ilícitos relacionados ao *software* é (e também por causa do tratamento errôneo dado pela lei) o mesmo para obras literárias e artísticas (FREITAS; BARBOSA, 2015, p.125-126).

E também que é necessário considerar que o programa de computador é algo iminentemente imaterial (WACHOWICZ, 2004, p.81), carecendo, portanto, de substância no mundo físico que lhe dê concretude, o que torna difícil, sem a técnica adequada, a análise dos elementos configuradores da infração ao direito de autor e à propriedade intelectual ou industrial.

Veja-se que, se de um lado há como se verificar a prática do plágio ou contrafação quando uma pessoa utiliza a mesma linguagem de programação ou reproduz *ipsis literis* o programa de computador original⁹, fica muito difícil a mesma imputação quando há alterações de código-fonte baseadas na utilização de diferentes linguagens de programação, variáveis, estruturas e outros elementos binários ou matemáticos. Neste sentido: “detecção e constatação de plágio em *software* não são tarefas simples, exigindo dos profissionais a realização de análises em diferentes níveis de abstração, bem como de diferentes elementos caracterizadores do *software*” (FREITAS; BARBOSA, 2015, p.133).

É necessário, portanto, analisar os seguintes elementos dos códigos-fonte (tanto o original quanto aquele que foi questionado): cadeias (*strings*), *tokens*, análise de estruturas em árvores (*parse trees*), gráficos de dependência do programa (PDG - *program dependency graphs*) e, ainda, as métricas sobre os segmentos do código-fonte de acordo com certos critérios, como por exemplo, o número de estruturas de repetição (*loops*) e estruturas condicionais, ou o número de variáveis utilizadas (FREITAS; BARBOSA, 2015, p.133).

Como se denota, a análise é altamente objetiva e especializada. E justamente da sua dificuldade é que deriva a possibilidade de alguém, utilizando-se de um código-fonte original, lançar no mercado um programa de computador (independentemente da linguagem de programação aplicada e com funcionalidades similares) contendo em seu núcleo (*core*) um código-fonte idêntico ou similar ao original, obtendo desta forma lucro indevido e à custa do esforço de planejamento, análise, testes e validações de outrem.

Logo, é inadequado qualquer forma simples de detecção de ilícitos quando o assunto é plágio e contrafação de *software*, pois a adoção de métodos não subsumidos às especificidades da tecnologia pode servir para inadequadamente concluir que um determinado caso prático não

⁹ Neste sentido, vide o julgado do STF sobre pirataria e falsificação de fonografia - HC: 120994 SP. Relator: Min. Luiz Fux. Data de julgamento: 29/04/2014, primeira turma. Data de publicação: 16-05-2014.

se adequada, para fins de sanção criminal, ao artigo 184º e seus parágrafos do Decreto-Lei 2.848/1940 (Código Penal) ou ainda, que não haveria a ocorrência de ilícito civil apto a ensejar indenização, pois existem, até mesmo na Lei nº 9.609/98, situações que não são consideradas ofensas aos direitos do titular de programa de computador:

Art. 6º Não constituem ofensa aos direitos do titular de programa de computador:
[...]

III - 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;

IV - a integração de um programa, mantendo-se suas características essenciais, a um sistema aplicativo ou operacional, tecnicamente indispensável às necessidades do usuário, desde que para o uso exclusivo de quem a promoveu (BRASIL, 1998, p.1).

Ao mesmo tempo, o viés comparativo simplista da legislação brasileira existente permite concluir que, independentemente da linguagem de programação aplicada e de funcionalidades similares, o código-fonte original e o código-fonte questionado podem tratar-se de obras imateriais completamente distintas e não copiadas entre si.

Desta forma, também pela falta de especialização dos peritos e *experts* e do desconhecimento do Poder Judiciário acerca dos métodos diferenciados de detecção de plágio e contrafação em programas de computador, bem como pela falta de proteção específica e possibilidade de patente, claramente se percebe que a lacuna normativa torna difícil ao originador do *software* provar que teve seus direitos de propriedade intelectual e de autor violados e, ainda, sem direito à exclusividade e primazia.

Se a leitura do parágrafo primeiro do artigo 2º da Lei nº 9.609/98 não deixa margem para dúvida sobre ser direito do autor de *software*, reivindicar a paternidade sobre o mesmo, e sendo o programa de computador, código-fonte que, como visto anteriormente, envolve pura atividade humana intelectual racional, lógica, exata e inventiva, porque não assegurar, então, ao *software* em si, a sua patente, dando-lhe especial e específica proteção no que concerne a prioridade e aos prazos de segurança para exercício da exclusividade sobre referido bem imaterial?

O problema, repise-se, advém da completa inadequação do legislador às necessidades da sociedade contemporânea. Faltou-lhe, também, *expertise* técnica para reconhecer no *software*, um invento, pois é proposição do presente artigo afirmar que todo código-fonte é atividade intelectual inventiva, podendo ser ou não, novidade, mas que traz inerente em si, um potencial para aplicação industrial e, portanto, pode ser patenteadado.

Apenas para se citar exemplos do potencial de aplicação industrial de *softwares*, há a ocorrência cada vez maior na sociedade global, da chamada era da Internet das Coisas (*Internet*

of Things – IoT). Ainda, da maciça aplicação de algoritmos computacionais, da inteligência artificial e de técnicas de *machine learning* (aprendizado de máquina), que não serão abordados no presente artigo, mas que vem revolucionando o modo como empresas, pessoas e Governos atribuem eficiência, eficácia e desenvolvimento econômico aos seus negócios.

Há se de salientar ainda que, em sendo original, o *software* – e para ser novidade - deve propor solução a um problema ainda não enfrentado computacionalmente. Portanto, desta forma, estaria apto a atender todos os critérios da lei para ser considerada invenção, e assim, e por este *viés*, ser passível de patente.

A falta de proteção específica ao programa de computador redundando no fato de que este acabará tendo de ser regido e tutelado pela norma comum civil de propriedade de bens móveis no Brasil, que se transferem com a tradição. Neste sentido, é o que dispõe o artigo 1.226 da Lei 10.406/2002: “Os direitos reais sobre coisas móveis, quando constituídos, ou transmitidos por atos entre vivos, só se adquirem com a tradição” e o artigo 1.267: “A propriedade das coisas não se transfere pelos negócios jurídicos antes da tradição” (BRASIL, 2002, p.1).

Também entende a jurisprudência brasileira, em sua maioria - e em interpretação do artigo 1.196 do Código Civil, que a propriedade de bem móvel pode ser provada também pela demonstração da posse direta: “Considera-se possuidor todo aquele que tem de fato o exercício, pleno ou não, de algum dos poderes inerentes à propriedade” (BRASIL, 2002, p.1).

Diante deste cenário normativo e jurisprudencial, grandes inquietações surgem quando o assunto é a adequada proteção do direito de autor e propriedade sobre *software*: como é que se fará tradição ou prova da posse ou propriedade de algo absolutamente imaterial? Como demonstrar qual código-fonte foi elaborado primeiro, e qual é cópia ou plágio? Como provar que o originador do programa de computador é o possuidor primário – ou imediato - e que detém a posse plena do código-fonte e que aquele a que se imputa o plágio e contrafação é, na verdade, possuidor secundário – ou mediato - e por ter “copiado” o *software* original, sua posse é ilegal, precária ou ainda, indevida?

Aprofunda a gravidade de tais indagações o fato de o Poder Judiciário brasileiro, e como já mencionado anteriormente, estar despreparado tecnicamente para julgar casos complexos envolvendo tecnologia, bem como não conseguir contar, muitas vezes, com a salutar expertise técnica de peritos e engenheiros especializados e que servirão a efetivamente auxiliar o desfecho das lides apresentadas.

Fato é que, pela ausência de patente de *software* em si (e até que haja alguma solução final da controvérsia gerada) a vulnerabilidade e os riscos da pessoa que possui e pretende provar a “paternidade” do programa de computador passam a ser potencializados na mesma

velocidade da Internet, que diariamente dá exemplos de programas de computador que são devassados por meio de plágio e contrafação:

Nesta senda, é evidente a grande insegurança jurídica que vivem os titulares dos softwares registrados, pois nada impede que, após toda a etapa criativa e executiva da invenção, um terceiro copie o produto protegido quase que integralmente e registre-o como seu. Além disso, com a permissão legal de semelhanças entre programas de computador, a tendência é que surja no Brasil uma espécie de padronização dos softwares registrados, pois assim que um novo programa emerge no mercado e soluciona um problema existente ou propõe uma inovação em alguma área econômica, outros criadores, visando também explorar economicamente aquele conceito inventado, poderão copiar o programa registrado quase que na íntegra e “criar” os seus próprios, sem precisar pagar os direitos de exploração ao criador originário. Com isso, os inventores legítimos, quais sejam, os primeiros a pensar em um novo conceito para os softwares, sem a merecida tutela legal, serão cada vez mais desestimulados a produzir novos programas, o que prejudicará demasiadamente a inovação nessa área, a qual é responsável por movimentar parte considerável da economia brasileira (LIMA; LIMA; GUIMARÃES, 2019, p.167-168).

Logo, muitas cópias ilegais terão sido criadas e extintas sem que se possa analisá-las a contento, na medida em que são rapidamente retiradas do mercado ou substituídas por outras com inúmeras modificações que, por sua vez, vão originar verdadeiros “*franksteins*”, ou seja, arremedos ou “retalhos” da programação de computador original, em um completo vilipêndio ao direito daquele que deveria, por lei, ter o direito de patente do *software*, assegurando-se assim a primazia e exclusividade do invento. Desta forma, e visando melhor elucidar tal situação, apresenta-se a seguir o célebre caso do jogo para celulares denominado *Flappy Bird*.

3.1 A Patenteabilidade do Programa de Computador e o Plágio

Um célebre exemplo da violação dos direitos do autor e propriedade de programas de computador está no caso do jogo denominado *Flappy Bird*. Totalmente inspirado no jogo *Mario Brothers* da empresa *Nintendo*, o *Flappy Bird* usa do mesmo cenário (canos verdes e fundo azul e verde) e da mesma sistemática de jogo (rolamento lateral da tela) que fez do precursor um sucesso mundialmente conhecido.

Além disto, o controle do personagem, em ambos os jogos, é feito da mesma forma: basta pressionar uma vez o botão do controle ou a tela do dispositivo para fazer com que o personagem pule (no caso do *Mario Brothers*) ou voe (no caso do *Flappy Bird*), como mostrado na Figura 1:

Figura 1 - Comparação entre os programas de computador Flappy Bird (a esquerda) e Mario Bros (no meio e a direita)



Adaptado de: MOORE (2014, p. 1).

Também o *Flappy Bird* se vale de um personagem (pássaro) que já existia no jogo *Mario Brothers*. Como resultado, o jogo *Flappy Bird* foi retirado, com interferência da *Nintendo*, de comercialização pela própria *Apple*, proprietária da loja online *App Store* (ELISE, 2014, p.1). Infelizmente, sem antes deixar uma sequência de inúmeros “clones” que, embarcando no sucesso da aplicação, aproveitaram o momento de euforia para gerar lucros (ELISE, 2014, p.1).

Sobre tal questão e falando sobre Dong Nguyen, - o criador do jogo *Flappy Bird* - o repórter July Clover (2014, p.1) disse, em matéria jornalística:

At the height of its popularity, *Flappy Bird* was earning Nguyen more than \$50,000 per day, and even removed from the App Store, the game has continued to earn money. In its absence, hundreds of *Flappy Bird* clone apps have been released on the App Store, each one attempting to earn some of the revenue Nguyen was able to pull in.¹⁰

O originador do jogo *Flappy Bird*, por sua vez, alegou que criou sozinho o *software* e não copiou o jogo da *Nintendo* (MACK, 2014, p.1). Tendo este caso prático em mente, é necessário indagar se a simples análise de características estéticas ou de funcionalidade de ambos os *softwares* (como atualmente é feito com obras artísticas e literárias) é suficiente a ensejar a declaração de que há plágio ou contrafação?

A toda evidência, não, pois como visto no presente artigo, é necessário ir muito mais além para uma correta análise do problema: é preciso, como já mencionado anteriormente, aplicar métodos e técnicas voltados à detecção de plágio em minuciosa análise de código-fonte (FREITAS; BARBOSA, 2015, p.131).

¹⁰ Tradução livre: No auge da sua popularidade, Flappy Bird remunerava Nguyen em mais de US\$ 50.000 por dia, e mesmo removido da App Store, o jogo continuou a ganhar dinheiro. Na sua essência, centenas de aplicativos clones de Flappy Bird foram lançados na App Store, cada um com o intento de ganhar a renda que Nguyen era capaz de atrair.

Supondo que tal caso fosse analisado perante o Judiciário brasileiro, muito provavelmente o método de detecção de ilícitos ao direito de autor e propriedade de *software* da *Nintendo* iria se basear, por conta da comparação dada pela lei, nos mesmos métodos empregados em outros tipos de obras artísticas e literárias, o que claramente traz uma inadequação.

Agora, supondo também que uma adequada análise do código-fonte detectasse a ocorrência de plágio ou contrafação, diante da ausência de possibilidade de patente do *software* em si no Brasil, poder-se-ia falar em perfeita subsunção do fato à norma para fins de caracterização de alguma infração penal ao direito de propriedade industrial?

A impossibilidade normativa de se obter a patente de *software* no Brasil deixa clara que a resposta a tal pergunta é negativa. Como já mencionado anteriormente, isto dá margem à ocorrência de nefastos efeitos relacionados à injustiça, concorrência desleal, prejuízos morais e materiais decorrentes da violação da criatividade e capacidade intelectual humana, sem falar na inexistência de primazia e exclusividade de uso sobre o programa de computador.

4 Conclusão

Urge uma mudança de paradigma normativo quando o assunto é a possibilidade de patente de *software* no Brasil. Um cenário ideal e adequado às particularidades do programa de computador precisa levar em consideração tanto a ótica do direito de autor e da propriedade intelectual, mas também o direito de propriedade industrial, do qual o *software* necessita fazer parte.

Ou seja, tanto as vias comuns de proteção às obras intelectuais imateriais quanto a garantia de primazia e exclusividade que a patente proporciona são formas encorajadoras ao empreendedorismo sustentável. Ambas devem coexistir no ordenamento jurídico brasileiro e não serem excludentes em relação ao *software*, como atualmente acontece.

Isso porque toda a inventividade e criatividade que advém de uma ideia nova podem rapidamente se esvaír em meio a falta da efetiva proteção legal que conceda ao proprietário do *software* o direito de patentear este por si só e, assim, estar realmente seguro juridicamente de que sua obra será respeitada e estará especificamente tutelada quanto a ocorrência de plágios e contrafações.

A atual forma de tratamento das leis existentes não coaduna com os contemporâneos anseios da sociedade tecnológica e faz com que a *mens legis* (espírito da lei) de proteção à propriedade industrial, intelectual e ao direito de autor no Brasil se esvaia em meio a uma

verdadeira cadeia de perpetuação de práticas ilícitas que, a depender do tamanho e do espraiamento serão impossíveis de serem coibidas e reparadas, gerando inúmeros prejuízos das mais diversas ordens àquele que reclama a “paternidade” e exclusividade com primazia sobre o programa de computador.

Enquanto nos Estados Unidos, US\$ 7.5 bilhões de valor de mercado da *Apple* no ano de 2014 foram gerados apenas pela reação do mercado à publicação das patentes da empresa (VISHNUBHAKAT, 2015, p.5) e o total de patentes originais (segundo o último relatório disponível do Escritório de Patentes e Marcas Comerciais dos Estados Unidos - USPTO) de desenvolvimento, instalação e gestão de *software* até o ano de 2015 foi de 17.336 (USPTO, 2015, p.1)¹¹, no Brasil não há qualquer patente de *software* por si só, haja vista a explícita vedação legal.

Desta forma não há economia que gere riquezas ao país em meio a insegurança jurídica, diante da ausência de possibilidade de patente sobre o *software* no Brasil. Nenhum investidor ou *startup* de tecnologia se sentirá incentivada a produzir seus negócios em território nacional. Portanto, o Estado brasileiro deve atualizar a legislação o mais breve possível para se adequar a diversos países desenvolvidos que protegem o programa de computador ao lhe conceder vários direitos; dentre eles, o de uma patente exclusiva e individualizada. Somente desta forma será possível o desenvolvimento sustentável que também se almeja no campo tecnológico brasileiro.

Referências

ASSAFIM, João Marcelo de Lima. **Controles sociais extrínsecos do exercício de direitos de propriedade intelectual: antitruste como tutela de direitos fundamentais**. Encontro Preparatório do Conpedi, 2007, Campos dos Goytacazes. v. 1. p.1-26. Disponível em: http://www.publicadireito.com.br/conpedi/manaus/arquivos/anais/campos/joao_marcelo_de_1ima_assafim.pdf. Acesso em: 20 fev. 2020.

BARBOSA, Denis Borges. **Uma introdução à propriedade intelectual**. 2ª ed. São Paulo: Lumen Juris, 2010.

BESSEN, James; HUNT, Robert M. **An empirical look at software patents**. Federal Reserve Bank of Philadelphia. Working paper nº 03-17/R. March, 2004.

BRASIL, **Lei 9.279 de 14 de maio de 1996 – regula direitos e obrigações relativos a propriedade industrial**. 1996. Disponível em: http://www.planalto.gov.br/ccivil_03/leis/L9279.htm. Acesso em: 20 fev. 2020.

¹¹ O relatório não especifica o ano em que se iniciou a contagem e acompanhamento das patentes de *softwares*.

BRASIL. **Lei 9.609/1998. Dispõe sobre a proteção da propriedade intelectual de programa de computador, sua comercialização no País, e dá outras providências.** 1998. Disponível em: http://www.planalto.gov.br/ccivil_03/leis/L9609.htm Acesso em: 20 fev. 2020.

BRASIL. **Lei 9.610/1998. Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências.** 1998. Disponível em: https://www.planalto.gov.br/ccivil_03/leis/L9610.htm. Acesso em: 20 fev. 2020.

BRASIL. **Lei 10.406 de 10 de janeiro de 2002. Institui o Código Civil.** 2002. Disponível em: http://www.planalto.gov.br/ccivil_03/leis/2002/L10406.htm. Acesso em: 20 fev. 2020.

BRASIL. **Tribunal Regional Federal (3ª. Região).** Apelação cível nº 00333201203503008 0000333-18.2012.5.03.0035. Relator: Oswaldo Tadeu B.Guedes. Turma Recursal de Juiz de Fora. Minas Gerais, 11 de outubro de 2012. Jusbrasil: jurisprudência. Disponível em: <http://trt-3.jusbrasil.com.br/jurisprudencia/124291017/agravo-de-peticao-ap-333201203503008-0000333-1820125030035>. Acesso em: 20 fev. 2020.

BROOKS JR, Frederick P. **No silver bullet – essence and accident in software engineering.** University of North Carolina, 1986. Disponível em: <http://worrydream.com/refs/Brooks-NoSilverBullet.pdf>. Acesso em: 20 fev. 2020.

CLOVER, July. **Flappy Bird to Return to the App Store.** Portal Macrumors. 2014. Disponível em: <http://www.macrumors.com/2014/03/19/flappy-bird-return/>. Acesso em: 20 fev. 2020.

CNPQ. **Relatório da Comissão de Integridade de Pesquisa.** 2011, p.4. Disponível em: <http://www.cnpq.br/documents/10157/a8927840-2b8f-43b9-8962-5a2ccfa74dda>. Acesso em: 20 fev. 2020.

PRIBERAM. Dicionário Priberam de Língua Portuguesa. Disponível em: <http://www.priberam.pt/dlpo/reprodu%C3%A7%C3%A3o>. Acesso em: 20 fev. 2020.

ELISE, Abigail. **'Flappy Bird' Game Deleted Over Nintendo Copyright? Creator Receiving Death Threats.** Disponível em: <http://www.ibtimes.com/flappy-bird-game-deleted-over-nintendo-copyright-creator-receiving-death-threats-1554401>. Acesso em: 20 fev. 2020.

FREITAS, Cinthia Obladen de Almendra; BARBOSA, Claudia Maria. **As novas tecnologias à frente da detecção de plágio em software e do direito de autor.** In: Direitos e novas Tecnologias. ROVER, Aires José; NASCIMENTO, Valéris Ribas do; CELLA, José Renato Gaziero (Orgs.). Conpedi/UFS. Florianópolis: CONPEDI, p. 123-143, 2015.

GHEZZI, Carlo; JAZAYERI, Mehdi. **Conceitos de linguagens de programação.** Trad. A. S. Veloso. Rio de Janeiro: Campus, 1985.

GOVINFO.ORG. **Public Law 96-517/80.** 1980. Disponível em: <https://www.govinfo.gov/content/pkg/STATUTE-94/pdf/STATUTE-94-Pg3015.pdf>. Acesso em: 20 fev. 2020.

JESUS, Damásio de. **Direito Penal, 3º volume. Parte especial: crimes contra a propriedade imaterial a crimes contra a paz pública.** 23ª ed. São Paulo: Saraiva, 2015.

KROKOSZ, Marcelo. **Outras palavras sobre autoria e plágio.** São Paulo: Atlas, 2015.

LIMA, Gabriel Maciel de; LIMA, Thaisi Leal Mesquita de; GUIMARÃES, Patrícia Borba Vilar. **A proteção jurídica de softwares e sua contribuição para o desenvolvimento brasileiro.** Revista Cadernos de Dereito Actual. Vigo, Espanha. nº 11. p.161-172, 2019.

LINUX. Linux Informatio Project. 2006. Disponível em:
http://www.linfo.org/source_code.html. Acesso em: 20 fev. 2020.

MACK, Eric. **Flappy Bird Creator May Have Been Scared By Legal Threats After All.** Disponível em: <http://www.forbes.com/sites/ericmack/2014/02/10/flappy-bird-creator-may-have-been-scared-by-legal-threats-after-all/#1ddf03014bb4>. Acesso em: 20 fev. 2020.

MAXWELL, Winston; BOLGER, Katherine; ZEGGANE, Thomas. **A comparative French and U.S. law approach to scènes à faire and other non-protectable elements in copyright law.** Propriétés Intellectuelles. France, nº 30, January 2009. Disponível em:
<https://www.hoganlovells.com/~/.media/2b2d2cde1d424a9aa4c07013ecfcb61d.ashx>. Acesso em: 20 fev. 2020.

MOORE, Todd. **Flappy Bird is flapping about Copyright Infringement:** Disponível em:
<http://toddmooore.com/2014/06/06/flappy-bird-is-flapping-about-copyright-infringement/>. Acesso em: 20 fev. 2020.

MOSSOFF, Adam. **A brief history of software patents (and why they're valid).** George Mason University School of Law. Center for the protection of intellectual property, Arlington, VA: 2013.

PRICEWATERHOUSE COOPERS. **Lei do Software e seu regulamento.** São Paulo: Atlas, 1999.

ROCHA, Hilton Ricardo. **Software & Direito - Definição, Criação e Propriedade.** Portal Âmbito Jurídico. Disponível em: http://www.ambito-juridico.com.br/site/index.php?n_link=revista_artigos_leitura&artigo_id=149. Acesso em: 20 fev. 2020.

SALES, Diego Felis. **A patenteabilidade do software no direito brasileiro.** Viajus. Disponível em:
<http://www.viajus.com.br/viajus.php?pagina=artigos&id=1840&idAreaSel=12&seeArt=yes>. Acesso em: 20 fev. 2020.

TIGRE, Paulo Bastos; MARQUES, Felipe Silveira. **Apropriação tecnológica na economia do conhecimento: inovação e propriedade intelectual de software na América Latina.** Revista Economia e Sociedades. Campinas, v.18, nº3, p.547-566, dez. 2009.

USPTO. Unite States Patent and Trademark Office. **Patents counts by class by year – Report.** 2020. Disponível em: <https://www.uspto.gov/web/offices/ac/ido/oeip/taf/cbcby.htm>. Acesso em: 20 fev. 2020.



U.S DEPARTMENT OS HEALTH AND HUMAN SERVICES. **Definition os Research Misconduct.** Disponível em: <http://ori.hhs.gov/definition-misconduct>. Acesso em: 28 fev. 2019.

VISHNUBHAKAT, Saurabh. **The commercial value of software patents in the high-tech industry.** Center for the protection of intellectual property of George Mason University. School of Law. May 2015.

WACHOWICZ, Marcos. **Propriedade intelectual do software e revolução da tecnologia da informação.** Curitiba: Juruá, 2004.