



UNIVERSIDADE CATÓLICA PORTUGUESA

O Direito de Autor e as API
Novos rumos da propriedade intelectual
sobre o *software*

Daniel Leopoldino Resende Duarte Gomes

Mestrado em Direito

Faculdade de Direito | Escola do Porto

2021



UNIVERSIDADE CATÓLICA PORTUGUESA

O Direito de Autor e as API
Novos rumos da propriedade intelectual
sobre o *software*

Daniel Leopoldino Resende Duarte Gomes

Orientadora: Maria Victória Rocha

Mestrado em Direito

Faculdade de Direito | Escola do Porto

2021

Aos meus pais e avós, pelas oportunidades

À minha esposa, pelo carinho e suporte

Agradecimentos

À minha orientadora, Prof.^a Dr.^a Maria Victória Rocha, pelo cuidado e dedicação que empregou à orientação desta dissertação de Mestrado.

Aos programadores Igor Tupini e Tetsuo Matsumura, pelos longos debates e valiosas contribuições à compreensão das API.

Não sou nada.

Nunca serei nada.

Não posso querer ser nada.

À parte isso, tenho em mim todos os sonhos do mundo.

Fernando Pessoa

Abstract: The legal case Google LLC v. Oracle America, Inc. brought to light several questions regarding the definition of computer programs in the context of copyright law. The subject matter that originated this uncertainty is the copyrightability of application programming interfaces (API), which is a software intermediary that allows two applications to communicate and is present in most computer programs. The complex nature of this tool makes it difficult to classify it within the copyright law but also gives us the opportunity to refine our understanding regarding computer program's protection. The Supreme Court of the United States wasted this opportunity when it chose to confine the analysis to the application of *fair use*, but the European Court of Justice (ECJ) has yet to define its position. Aside from unfounded apocalyptic theories, there is no reason for the ECJ to define API as something separate from computer programs or to withdraw its protection *a priori*.

Keywords: API; Software; computer programs; copyright; Google LLC v. Oracle America, Inc.; fair use; merger doctrine; scène à faire.

Resumo: O caso Google LLC v. Oracle America, Inc. trouxe à tona diversas questões relacionadas com o conceito de programas de computador no contexto do direito de autor. A incerteza que motivou tais questionamentos é a tutela pelos direitos autorais das interfaces de programação de aplicativos (API), cuja função é atuar como intermediárias que permitem a comunicação entre dois *softwares*, estando presentes na maioria dos programas de computador. A natureza complexa desta ferramenta torna difícil a sua classificação perante as disposições do direito autoral, porém traz consigo a oportunidade de refinar o entendimento sobre a proteção dos programas de computador. O Supremo Tribunal Americano ignorou esta oportunidade quando optou por limitar a análise à aplicação do *fair use*, mas o Tribunal de Justiça da União Europeia (TJUE) ainda não definiu sua posição. Descartadas teorias apocalípticas infundadas, não há razão para o TJUE definir as API de forma apartada dos programas de computador ou retirar sua proteção *a priori*.

Palavras-chave: API; *Software*; programa de computador; direito de autor; Google LLC v. Oracle America, Inc.; *fair use*; *merger doctrine*; *scène à faire*.

Sumário

1. Introdução	8
2. A API.....	11
2.1. Definindo as API	11
2.2. Natureza híbrida das API	12
2.3. O funcionamento das API	13
3. As API e o Direito da União Europeia	18
3.1. Questões introdutórias	18
3.2. A qualificação jurídica das API	20
3.3. A originalidade do código declarativo	23
3.4. Efeitos concorrenciais de uma possível proteção das API	26
3.5. A API e os Tribunais Europeus	28
3.6. Conclusões quanto à proteção das API na União Europeia	30
4. Análise do caso Google LLC v. Oracle America, Inc.	32
4.1. As circunstâncias gerais	32
4.2. A análise do Supremo Tribunal Americano	33
5. O caso Google LLC v. Oracle America, Inc. sob a perspetiva da União Europeia	40
6. Conclusão	44

1. Introdução

Os direitos de autor são uma criação recente da humanidade, mas a sua existência rapidamente se tornou imprescindível. Esta afirmação tem especial relevância na atualidade, visto que em nenhum momento na história estes monopólios sobre bens imateriais proporcionaram tão alto grau de controlo aos seus proprietários quanto na era dos meios digitais. E se em algum momento a transferência de um direito de utilização perpétuo por meio do *download* representou um entrave à ingerência dos titulares, a popularização do acesso por assinatura pôs fim a quaisquer receios. No entanto, nem tudo são flores – a digitalização do mercado também acarreta transtornos para as empresas que usufruem deste processo.

Na mesma proporção que produtos e serviços deixaram de ser analógicos, estabelecimentos¹ e meios de produção perderam a sua materialidade. Já não é necessário um investimento substancial com maquinaria ou patentes de invenção para vencer neste mercado. Basta que as empresas consigam reconhecer tendências e atuar em função de oportunidades de mercado por meio da formulação de *softwares* ou incorporação de *start-ups* unicórnio. Com a atual proteção dos programas de computador pelas normas de direito de autor europeias, o conteúdo que é criado pelos programadores encontra-se protegido mesmo antes de estar pronto, reduzindo imensamente os riscos deste investimento. Há apenas uma condição à obtenção desta prerrogativa: o código presente no programa de computador deve ser uma criação original dos seus programadores.

Percebe-se com clareza uma incongruência entre este requisito e os objetivos das empresas ao investirem na criação de um programa de computador. Haja em vista o atual dinamismo do mercado tecnológico, a originalidade do código não se encontra entre os atributos mais valorizados em um *software*, quais sejam a acessibilidade, a consistência, a facilidade de aprendizagem, a perspicácia e a racionalidade humana do seu funcionamento². Todas estas características fazem menção ao campo das ideias e possuem valor estritamente funcional e, portanto, são irrelevantes à obtenção de qualquer proteção por meio da tutela oferecida pelo direito de autor.

¹ Barbosa, P. M. N. (2017) – *E-stabelecimento*, Rio de Janeiro: Quarter Latin, p. 27 e ss.

² Bostock, M. (2016) – “What Makes Software Good?”, 10/Mar/2016. <https://medium.com/@mbostock/what-makes-software-good-943557f8a488>, consult. em 10/Jul/2021.

É a partir deste ponto que a proteção se torna um obstáculo. Para a frustração das empresas que trabalham neste mercado, mesmo um produto inovador e transformador pode estar contaminado com um código-fonte alheio. Desta forma, ainda que se formule um novo programa de computador por meio de uma utilização inovadora e transformativa de um código-fonte pré-existente, este poderá depender da autorização do titular do referido código para ser explorado comercialmente.

É importante destacar que muitas destas empresas não estão preocupadas em proteger os seus programas por meio do direito de autor. A Google faz fortunas através de anúncios, razão pela qual tornou o seu sistema operacional Android *open source* desde a sua conceção, e a Apple integrou o sistema IOS ao seu hardware de tal maneira que a simples reprodução do código-fonte é irrelevante para utilizações em produtos que não carregam a maçã. Além disso, a maioria dos programas de computador atuais atuam como um serviço e, portanto, mesmo que se copie o código-fonte, dificilmente será possível assimilar o ecossistema de *softwares* necessários ao seu funcionamento, bem as constantes alterações que este sofrerá por meio de atualizações. Há que questionar se o programa de computador receberia esta mesma proteção se a discussão tivesse surgido após a transição de *software as product* para *software as a service*.

No contexto desta mudança de paradigma, surge a discussão quanto à possibilidade de proteger as API pelo direito de autor. As API permitem a comunicação entre *softwares* e estão na base do funcionamento de quase todos os programas de computador atuais. Em especial, as Web API, que permitem a comunicação entre computadores por meio da Internet, são o mecanismo por meio do qual todos os serviços *on demand*, aplicações e redes sociais operam. Tendo em vista a discussão anterior, é fácil concluir que na maioria das ocasiões o mercado tecnológico atual pouco se importa com a proteção do código-fonte destas API, afinal elas são apenas um mecanismo para entregar um produto, não o produto em si.

É fácil imaginar a surpresa destas empresas quando se noticiou que uma decisão do Supremo Tribunal Americano no caso Google LLC v. Oracle America, Inc. poderia colocar em risco este ecossistema tão eficiente. Afinal, imaginou-se que com a proteção das API pelo direito de autor seria criado o cenário perfeito para o surgimento de uma multiplicidade de litígios, assim como um novo entrave à inovação e ao desenvolvimento de *softwares* – seria o fim da indústria de

tecnologia da informação³. Porém, com o anúncio da referida decisão, suspirou-se de alívio ao se perceber que a doutrina legal do *fair use* provavelmente seria a solução padrão para este tipo de litígio⁴. O Supremo Tribunal Americano, todavia, não ofereceu uma resposta à questão principal.

O *fair use* não existe na União Europeia e, como veremos, não existe nenhum mecanismo legal que permita uma solução similar. Desta forma, o remédio americano não pode ser aplicado, e a questão principal deverá ser respondida: Serão as API objetos de proteção pelo direito de autor? E, em caso de uma resposta positiva, deve indagar-se: Quais os efeitos da proteção das API pelo direito de autor?

A primeira questão é complexa e ainda que este trabalho procure delinear uma resposta, ficará claro que não é factível extrair da jurisprudência atual qualquer padrão que permita prever o posicionamento do Tribunal de Justiça da União Europeia (doravante TJUE). Verifica-se, no entanto, o inverso no tocante ao segundo questionamento, razão pela qual se adianta desde já que o reconhecimento desta proteção não seria o apocalipse que muitos proclamam.

³ Quanto à referida atmosfera de incerteza e análise de fundo: Hurst, A. (2018) – “The Report of Api Copyright’s Death is Greatly Exaggerated”. *Harvard Journal of Law & Technology, Volume 31, Special Issue Spring 2018*.

⁴ Sarid, U. (2014) - “A Non-Apocalypse: APIs, Copyright, and Fair Use”, s.d. [2014]. <https://www.wired.com/insights/2014/05/non-apocalypse-apis-copyright-fair-use/>, consult. em 11/Set/2021.

2. A API

2.1. Definindo as API

A utilização no acrónimo API esconde uma nomenclatura simples e autoexplicativa, *Application Programming Interface* (Interface de Programação de Aplicações)⁵. Desta forma, já se sabe que o objeto em estudo não se encaixa perfeitamente na definição clássica de um programa de computador. Enquanto um programa de computador é constituído por um conjunto de instruções que são interpretadas por um *hardware* para a realização de uma determinada tarefa, a API possui natureza híbrida que lhe permite exercer a sua função como interface entre dois programas.

Ainda que a definição e o funcionamento das API não sejam nenhum mistério para aqueles que há tempos as utilizam como uma ferramenta, as suas características singulares tornam-nas num objeto de estudo complexo no mundo jurídico, fator que muito contribui para analogias funestas e simplificações inconvenientes. Neste sentido a separação das API em duas parcelas de qualificação jurídica distinta (“declaração” e “implementação”), método este que foi trazido à luz durante o caso *Google LLC v. Oracle America, Inc.*⁶, será vital para a formação de um estudo coerente e aprofundado sobre este tema. Na sua parcela declarativa, as API são compostas por um conjunto de signos, porém o conteúdo deste código (código declarativo) não tem a finalidade de permitir a comunicação com um *hardware*, mas com um programa de computador. É apenas por meio do código de implementação que o *software* com o qual se pretende interagir será capaz de compreender qual instrução deverá ser realizada e, após a sua realização, poderá dar uma resposta ao programa que fez uso do código de declaração.

Desta forma, verifica-se que as partes declarativa e de implementação se complementam. Sem o código de implementação, o de declaração é apenas um conjunto de palavras/signos sem utilidade ou significado. Ao mesmo tempo, o código de implementação precisa do declarativo para se orientar, pois, caso o contrário, não consegue fazer qualquer interação entre os programas de computador.

⁵ Artigo “Interface de programação de aplicações”, Wikipédia, a enciclopédia livre., s.d.. https://pt.wikipedia.org/wiki/Interface_de_programação_de_aplicações, consult. em 20/Jan/2021.

⁶ *Google LLC v. Oracle Am., Inc.* 141 S. Ct. 593 U. S. 21, 22 (2021).

A utilização das API é uma prática muito anterior à própria criação do termo, porém foi no início deste milénio que a sua capacidade de permitir a existência de um mundo altamente conectado começou a ser explorada⁷. No entanto, para compreender o potencial e a natureza desta ferramenta, é necessário possuir uma compreensão mínima da sua utilização e do seu funcionamento.

2.2. Natureza híbrida das API

Ao formular um programa de computador com o qual outros programas podem interagir, os desenvolvedores deste produto atribuem signos às instruções que determinam a realização das mais variadas funções presentes no código-fonte. Estes signos são disponibilizados, frequentemente de forma pública, juntamente com uma documentação explicando o funcionamento de cada uma destas instruções.⁸ Este conjunto de signos, bem como a sua estrutura organizativa, dá origem a parte declarativa de uma API. É importante frisar que certos exemplos utilizados neste trabalho podem transmitir uma ideia equivocada quanto à complexidade do conteúdo presente nas API. Para que os signos possam ser invocados por um *software* terceiro (*method call*) e associados pelo código de implementação, as instruções não são identificadas simplesmente pelo termo que indica a sua função, mas pela classe (ou subclasse), pelo pacote ao qual pertence e, por fim, por uma nomenclatura que precisa de condensar a função da instrução que pretende identificar (algo que pode exigir certo esforço criativo, dependendo da complexidade da instrução).⁹

Ainda, a utilização de qualquer instrução dentro de um pacote requer a apresentação das informações relevantes à sua realização. Por exemplo, uma instrução cujo nome, resumido a título exemplificativo, é “*realizarSoma*” evidentemente exige, no mínimo, que lhe sejam apresentados dois números (p. ex. “*realizarSoma(2, 3)*”, cujo *feedback* será “5”), enquanto uma função com o nome “*valorAtualEuro*” poderá fornecer o valor atual do Euro sem que qualquer dado imperativo lhe seja fornecido. Este componente é chamado de argumento e a forma como é utilizado define a

⁷ Hawkins, M. - “The History And Rise Of APIs”, 23/Jun/2020. <https://www.forbes.com/sites/forbestechcouncil/2020/06/23/the-history-and-rise-of-apis/>, consult. em 25/Jan/2021.

⁸ “What Is API Documentation?”, API Lifecycle Management, 31/Ago/2020. <https://www.akana.com/blog/what-is-api-documentation.html>, consult. em 25/Jan/2021.

⁹ “Java™ Platform, Standard Edition 7 API Specification”, s.d.. <https://docs.oracle.com/javase/7/docs/api/>, consult. em 25/Jan/2021.

instrução que será realizada. Desta forma, um mesmo significante pode possuir diferentes significados e, portanto, constituir uma multiplicidade de signos.

Um exemplo de como todos estes fatores criam uma grande camada de complexidade e variabilidade é a subclasse de instruções “*java.security.cert.CertPathBuilderException*”, pertencente a classe “*GeneralSecurityException*”, referente ao pacote “*java.security*” da API Sun Java, capaz de realizar quatro instruções diferentes de acordo com o tipo de argumento que for utilizado.¹⁰ Mesmo com toda a criatividade e engenhosidade empenhada na criação dos significantes, para compreender o significado de determinado signo, um programador normalmente será obrigado a consultar uma documentação explicando o funcionamento de cada instrução (fator este que também constitui um investimento necessário à criação de uma API).

Uma vez que o código declarativo apenas identifica os signos, é necessária a existência de um código-fonte que saiba associá-los às parcelas de código apropriadas e que realize as funções esperadas, isto é, a parcela de implementação da API. Enquanto os programas de computador que fazem uso de uma determinada API requisitam instruções por meio de um *method call*, os programas de computador que são alvo desta instrução necessitam de código de implementação. Esta parcela da API prevê um comportamento diferente para cada signo presente no código declarativo e em nada difere de um programa de computador no seu funcionamento.

2.3. O funcionamento das API

Como consequência desta natureza híbrida, uma mesma função de um *software* pode estar associada a diferentes nomenclaturas e, portanto, diferentes API, assim como uma mesma API pode ser adaptada para ser utilizada por outros programas com funções similares. Um exemplo desta situação são as API que permitem a comunicação entre o *software* de aplicações e o *driver* relacionado ao *hardware* de computadores. API como Vulkan, OpenGL e WebGPU possuem funções similares, permitir a comunicação de programas de computador com os *drivers* de Unidades de Processamento Gráfico, e todas as três são integradas na maioria dos *drivers* e sistemas operacionais atuais. Isto significa que o desenvolvedor de um programa de computador poderá optar por diferentes interfaces de comunicação com o *hardware* por uma questão de

¹⁰ Ibid., <https://docs.oracle.com/javase/7/docs/api/java/security/cert/CertPathBuilderException.html>, consult. em 25/Jan/2021.

preferência ou disponibilidade¹¹. Diferentes API podem permitir diferentes níveis de interação com um determinado *software*, em uma tentativa de balancear interatividade e intuitividade.

Há de concluir-se que existem diversas maneiras de formular e implementar uma API e obter o mesmo resultado, bem como que esta diferença influencia a experiência dos programadores a ponto de obrigar o mercado a assegurar esta variedade. Porém, dificilmente esta individualidade provém de uma atividade inventiva substancial por parte dos seus criadores. A diferença está na criatividade por meio da qual os seus criadores realizam o referido balanço (interatividade e intuitividade). *Verbi gratia*, para um mesmo programa de computador, cujo objetivo é criar *sites*, poder-se-ia criar duas API diversas com a mesma função, porém com diferentes níveis de complexidade, conforme se verificaria por uma hipotética instrução referente à escolha de cores: enquanto na primeira, mais complexa, se optava por “atribuirRGB” para permitir a escolha com precisão de cores por meio do sistema de cores RGB (ex.: o código (169,169,169) para representar a cor cinza escuro), a segunda, mais intuitiva, definia apenas “atribuirCor”, na qual o desenvolvedor poderia escolher a cor pelos nomes habitualmente utilizados (ex.: o texto “Cinza Escuro”).

Entre os programadores, ouve-se corriqueiramente o termo “nível de abstração” para descrever a quantidade de detalhes contida numa determinada informação. Este conceito pode ser exemplificado pela diferença entre uma manchete e a leitura da notícia na íntegra: enquanto a manchete representa um alto nível de abstração, a notícia na íntegra, uma vez que contém uma extensa gama de detalhes, representa um baixo nível de abstração. Desta forma, a abstração permite excluir aspetos irrelevantes ou secundários de uma determinada informação.¹² Para o mundo jurídico, isto significa que quanto maior o nível de abstração, menor a sobreposição entre o significativo e o significado. Desta forma, um código de declaração representa um nível superior de abstração em comparação com o código de implementação, enquanto este, por sua vez, é mais abstrato do que um código-objeto.

¹¹ “Unity Documentation”, 02/Jun/2018. <https://docs.unity3d.com/Manual/GraphicsAPIs.html>, consult. em 5/Fev/2021.

¹² Santana, G. - “O que é abstração em programação?”, 05/Mai/2020. <https://medium.com/@gabrielsantana765/o-que-é-abstração-em-programação-bf8ac2fcd96f>, consult. em 20/Fev/2021.

É importante destacar que existem diversos tipos de API, assim como diversas formas de as classificar. Para este estudo, é importante compreender que existem diferenças fundamentais entre as, já referidas, Web API, que utilizam o protocolo HTTP para permitir a comunicação entre diferentes *softwares*¹³, e as API utilizadas restritamente para facilitar a criação de novos *softwares*. Enquanto as API como a Youtube Data API, que permite a utilização de recursos do Youtube em diferentes *sites* e aplicações¹⁴, são primeiramente ferramentas de comunicação entre *softwares*, API como a API Sun Java foram criadas com o intuito de tornar mais eficiente a geração de novos *softwares* por meio de pacotes de *software* pré-programados.

A utilização de API tem duas funções principais, tornar desnecessária a disponibilização do código-fonte, permitindo a sua proteção através do segredo, bem como criar uma camada de abstração, permitindo a utilização de funções sem que seja necessária a compreensão da implementação do programa. Além disso, as API costumam possuir sistemas de autenticação e autorização, permitindo um maior controlo em relação a quem utiliza e como se utiliza um determinado serviço. Sendo tamanha a utilidade desta ferramenta, o seu uso é extremamente comum tanto para a criação de *workflows* dentro das empresas, quanto para a disponibilização de *softwares* para terceiros.¹⁵

Vale destacar que as API não são apresentadas diretamente ao consumidor final. Entretanto, em muitos casos, as GUI (*graphic user interfaces*) fazem a ligação entre API e consumidor final, que, sem perceber, dá instruções vitais (os referidos argumentos) ao funcionamento desta API. Neste sentido, é comum um determinado grau de interação entre ambos, ainda que por meio de uma barreira de abstração. É o caso dos formulários e dos botões presentes na maioria dos *sites*, que disfarçam o facto de que estão a ser recolhidos argumentos e solicitados os signos presentes nas API, respetivamente.

Entretanto, não se podem permitir comparações diretas entre as GUI e as API. Ainda que ambas constituam dois tipos de interface utilizados no ambiente digital, as suas similaridades

¹³ “Web APIs – Introduction”, W3Schools, https://www.w3schools.com/js/js_api_intro.asp, consult. em 25/Fev/2021.

¹⁴ Ibid.

¹⁵ Bettilyon, T. E. - “What Is an API and Why Should I Use One?”, 11/Jan/2018. <https://medium.com/@TebbaVonMathenstien/what-is-an-api-and-why-should-i-use-one-863c3365726b>, consult. em 13/Mar/2021.

acabam aí. Enquanto as GUI são um resultado do código-fonte e, portanto, uma funcionalidade do programa de computador, as API são uma parte integrante dos programas de computador pois permitem direta ou indiretamente a execução das suas funções. Esta diferença é fundamental para compreender a razão pela qual análises jurídicas relativas a uma dificilmente se aplicarão à outra.

Poder-se-ia ainda buscar similaridades entre o código de declaração e as linguagens de computador. Neste sentido, deve esclarecer-se que o objetivo mínimo de toda a linguagem de programação é alcançar a *Turing completeness*¹⁶, isto é, criar uma linguagem que permita computar qualquer sequência computável. Desta forma, os signos que formam as linguagens de programação não descrevem instruções possivelmente complexas, como os códigos de declaração, sendo formados frequentemente pelos próprios termos que descrevem uma determinada ação (“*if*”, “*print*”, “*for*”) ou por simples abreviações. Uma vez que estes signos geralmente pertencem a determinado dialeto (frequentemente o inglês) e se repetem na maioria das línguas de programação, apenas se poderia procurar alguma originalidade no método e nas regras que estabelecem, dois objetos que fogem da proteção do direito autoral.

Ainda, enquanto as linguagens de programação utilizam um conjunto de regras, símbolos e palavras especiais para construir um programa de computador¹⁷, os códigos declarativos não permitem a mesma funcionalidade pois os seus signos não estabelecem relações lógicas entre si quando expressos no código-fonte (i.e., não há sintaxe), servindo unicamente para representar um código-fonte já existente. Entretanto, como ainda veremos neste trabalho, existem situações em que esta divisão se mostra com tamanha clareza – ao comparar linguagens de programação com altos níveis de abstração e API com baixos níveis de abstração, as disparidades podem diminuir.

Com estas informações já é possível extrair três factos muito relevantes para qualquer jurista e que permitirão compreender as questões jurídicas apresentadas no presente estudo. Em primeiro lugar, as API são interfaces integradas no código de diversos *softwares*, porém a sua caracterização como programa de computador é incerta, visto que a sua parcela declarativa não possui função operacional direta, apenas indireta. Em segundo lugar, API com funções similares

¹⁶ Artigo “Turing completeness”, Wikipédia, a enciclopédia livre. https://pt.wikipedia.org/wiki/Turing_completeness, , consult. em 13/Mar/2021.

¹⁷ “Lenguajes de programación”. Informática IV. Universidad Nacional Autónoma de México. http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/4/informatica_4.pdf. consult. em 13/Ago/2021.

coexistem devido a diferenças criativas e funcionais. E, em terceiro lugar, devido à sua natureza, as API permitem diferentes níveis de criatividade na sua formulação, porém dificilmente representam uma novidade ou exigem uma atividade inventiva relevante.

3. As API e o Direito da União Europeia

3.1. Questões introdutórias

Os últimos anos do século XX foram marcados por uma extensa discussão acerca da proteção jurídica dos programas de computador.¹⁸ Ainda que ao fim de todo o debate o direito autoral se tenha consagrado como o mecanismo de proteção, tal interpretação só se deu por uma qualificação *ex lege* dos programas de computador como obras literárias. Esta conclusão não surgiu de forma axiomática, mas no contexto de uma variedade de posicionamentos divergentes que englobavam desde a defesa por meio de patentes¹⁹ até a criação de um direito *suis generis*²⁰. Foi apenas após a adoção da equiparação dos programas de computador às obras literárias pelo Congresso norte-americano em 1980, permitindo-se assim a proteção pelo *copyright*, que um modelo similar foi adotado na União Europeia por meio da Diretiva 91/250/CEE, de 14 de maio de 1991. Entretanto, a diretiva europeia não se limitou a impor o sistema de direito autoral vigente e, reconhecendo as especificidades dos programas de computador, acrescentou diversas particularidades ao seu regime normativo. Desta forma, compreendeu-se desde o início as exigências específicas da proteção do programa de computador, razão pela qual era necessária uma norma específica.

A versão codificada desta norma veio, posteriormente, por meio da Diretiva 2009/24/CE do Parlamento Europeu e do Conselho, de 23 de abril de 2009. Ao abordar, no seu artigo 1.º e inciso segundo, o objeto da proteção por ela abrangido, vêm determinar que “As ideias e princípios subjacentes a qualquer elemento de um programa de computador, incluindo os que estão na base das respectivas interfaces, não são protegidos pelos direitos de autor ao abrigo da presente directiva.”. Desta forma, ao representar a única fonte do Direito da União Europeia que aborda diretamente as interfaces de *softwares*, e ao fazê-lo em sentido negativo, esta norma dá margem a

¹⁸ Vieira, J. A. (2020) – *Direito de Autor: Dogmática Básica*, Coimbra: Editora Edições Almedina S.A., pp. 509 – 510.

¹⁹ É importante destacar que a proteção pelo direito autoral não impede a admissibilidade de patentes codificados nos programas de computador, conforme: Pereira, A. L. D. (2019) – *Direito da Propriedade Intelectual & Novas Tecnologias, Estudos – Volume I*, Coimbra: Gestlegal, pp. 451 e ss.

²⁰ Saavedra, R. (1998) – *A Proteção Jurídica do Software e a Internet*, Lisboa: Editora Sociedade Portuguesa de Autores Publicações Dom quixote, p. 73.

interpretações precipitadas no sentido de negar qualquer possibilidade de proteção das API pelo direito de autor.

Preliminarmente, há dois fatores a serem destacados neste artigo. Em primeiro lugar, a ausência de proteção de ideias e princípios pelo direito de autor não representa qualquer inovação jurídica, de forma que esta previsão vem apenas destacar um dos postulados do direito autoral²¹ (conforme se pode verificar no item 11 do preâmbulo desta diretiva, que contém uma informação análoga ao artigo, porém iniciando com a frase “De forma a evitar qualquer dúvida, tem de se deixar claro que [...]”).

Em segundo lugar, a própria norma esclarece que a limitação está ligada a ideias e princípios presentes “na base das respectivas interfaces”, deixando claro que não é atribuída esta natureza às próprias interfaces. Extrai-se que as ideias e os princípios aplicados ao funcionamento da interface não são passíveis de proteção pelo direito autoral, porém quanto à proteção da expressão da interface (i.e., o código-fonte) não se encontra nesta norma qualquer restrição. Quanto à problemática, José Alberto Viera defende que “enquanto for possível diferenciar a expressão das interfaces das ideias, regras e métodos de operação que estruturam o seu funcionamento num programa de computador, a tradicional contraposição entre expressão – protegida – e ideias – não protegidas – serve perfeitamente para definir o âmbito de proteção das interfaces, assegurando uma perfeita identificação daquilo que pode ser objeto do direito de autor”.²²

Desta forma, percebe-se que esta diretiva não contém qualquer restrição direta à proteção das API. Uma leitura atenciosa revela justamente o contrário, visto que a própria menção das interfaces permite depreender a sua condição de elemento integrante do escopo de expressões agrupadas sob a designação de programa de computador. Entretanto, seria leviano ignorar a razão pela qual as API levantam dúvidas quanto à sua definição como programa de computador, enquanto outras formas de código-fonte são implicitamente aceites. Verifica-se que na natureza

²¹ Barbosa, D. B. (2010) – *Uma Introdução à Propriedade Intelectual*, Rio de Janeiro: Editora Lumen Juris, p. 182.

²² Viera, J. A. (2005) – *Proteção dos Programas de Computador pelo Direito de Autor*, Dissertação de Doutoramento em Ciências Jurídicas, Lisboa: Editora LEX, p. 382.

incerta da sua parcela declarativa e na sua errónea associação com as interfaces gráficas está a origem destes receios, conforme a análise que se segue.

3.2. A qualificação jurídica das API

Como vimos, o código de implementação da API é expresso em código-fonte e, além da sua função de interpretar os signos do código de declaração, em nada difere de um programa de computador habitual. Esta parcela nada mais é do que um código-fonte capaz de comunicar diretamente com o *hardware*, ainda que precise ser invocado por meio dos *method calls*, para que realize as suas funcionalidades. A necessidade de uma interação externa, no entanto, não altera a sua natureza, uma vez que esta circunstância se mostra uma exigência para qualquer tipo de programa de computador. Desta forma, desde que se demonstre a existência de um espaço de realização expressiva do programador, não há razões para excluir da proteção do referido artigo a parcela da API referente à sua implementação²³.

Quanto à sua parcela declarativa, esta difere parcialmente dos conceitos previamente abordados. Conforme se delineou, a forma como o código de declaração se expressa difere de um programa de computador habitual pois os seus signos não comunicam diretamente com o *hardware*, exigindo a atuação do código de implementação para que as suas instruções sejam traduzidas e executadas. Há de verificar-se, portanto, se tal fator representa um obstáculo à sua proteção por meio da Diretiva 2009/24/CE ou pelo direito de autor geral.

Ainda que o artigo 10.º, n.º 1, do acordo ADPIC restrinja os programas de computador a expressões em código-fonte ou em código objeto, o Direito da União Europeia não define com clareza os requisitos que configuram estes objetos de proteção. A WIPO Pearl (portal terminológico multilingue da OMPI que apresenta as definições de termos técnicos e científicos por meio de documentos de patentes)²⁴ define o código-fonte como “(...) *textual code written in a programming language (such as C, C++, Pascal, Java™, or Python)*”. Sob estes parâmetros, qualquer texto, independentemente da sua utilidade prática, escrito numa linguagem de programação deveria ser protegido como um programa de computador. Verdade seja dita, os

²³ Vieira, J. A. (2020), ob. cit., p. 520.

²⁴ WIPO Pearl – WIPO’s Multilingual Terminology Portal - <https://www.wipo.int/reference/en/wipopearl/>, consult. em 13/Abr/2021.

signos presentes no código declarativo não são formados pelos termos que compõem uma linguagem de programação, porém uma interpretação tão restritiva também levaria à conclusão de que os nomes das funções, as *strings* e as variáveis presentes no código-fonte não seriam componentes relevantes de um programa de computador.

Trata-se de um exemplo nítido do Paradoxo do Navio de Teseu. É impossível estabelecer quantos e quais atributos habitualmente relacionados ao conceito de um programa de computador são necessários para que um código-fonte assim possa ser identificado. Na ausência de uma definição precisa por parte do Direito comunitário é difícil compreender os limites deste conceito. A equiparação do material de conceção preliminar do programa de computador ao conceito de programa de computador é um exemplo claro desta ausência de parâmetros claros.²⁵ Ainda assim, há que empenhar esforços para compreender a atual perceção dos tribunais europeus quanto à proteção do código de declaração, visto que é nesta parcela que se identifica a real identidade da API.

Poder-se-ia questionar se a proteção do código declarativo como base de dados não representaria uma solução alternativa para este embaraço. Verifica-se que a Diretiva 96/6/CE atribui direito *suis generis* para que fabricantes de bases de dados que tenham empenhado um investimento relevante na sua criação. Ainda que seja possível encontrar o código declarativo expresso em forma de coletânea nas documentações de API, não é esta a sua expressão original, assim como não representa a sua utilização como interface entre programas de computador, tornando equívoca tal equiparação.

Ainda que se considere, por qualquer hipótese que seja, que a parcela declarativa das API escape do escopo normativo estabelecido de proteção pela Diretiva 2009/24/CE, este facto isolado não é suficiente para lhe retirar a proteção do direito autoral. Tal circunstância pode ser claramente observada no acórdão *Bezpečnostní softwarová asociace* (doravante BSA)²⁶, no qual se verificou que, apesar de não constituir um programa de computador na aceção da referida diretiva, as interfaces gráficas ainda assim poderiam estar protegidas pelo direito autoral “genérico”. Entretanto, por mais que existam poucas dúvidas quanto à possibilidade de proteção autoral da

²⁵ Vieira, J. A. (2020), ob. cit., p. 512.

²⁶ Acórdão do Tribunal de Justiça (Terceira Secção) de 22 de Dezembro de 2010. *Bezpečnostní softwarová asociace* - Svaz softwarové ochrany contra Ministerstvo kultury. Processo C-393/09.

interface gráfica de forma desagregada da proteção oferecida aos programas de computador, admite-se, desde já, que o enfrentamento deste tema quanto à parcela declarativa das API levanta diversos obstáculos. Desta forma, também será útil esclarecer se o direito autoral geral também se mostra viável à proteção das interfaces não gráficas.

Independentemente do caminho pelo qual se desejar optar, seja a proteção pela Diretiva 2009/24/CE ou o direito autoral geral, devem-se destacar as dificuldades relacionadas com a proteção da parcela declarativa. Mesmo que se leve em consideração a primeira hipótese, atribuindo a ambas parcelas a natureza de programa de computador, de forma isolada, os signos constantes no código declarativo das API jamais seriam passíveis de proteção isoladamente, por ausência de um espaço de realização expressiva relevante. A função do código de declaração como uma parcela integrante da API apenas se dá uma vez que o conjunto de signos, organizados de forma sistematizada, permitam a utilização indireta do código de implementação. Entretanto, o valor desta sistematização de informações perante o direito de autor não é claro.

A questão de compreender se a organização de um programa de computador deve ser suscetível de proteção autoral não é recente, nem está limitada à discussão acerca das API.²⁷ Neste sentido, José Alberto Vieira apresenta dois argumentos que, segundo o autor, “permitem, sem dificuldade, considerar englobada a estrutura do programa de computador na expressão protegida de um programa de computador”. O primeiro está relacionado com o art. 2, n.º 5 da Convenção de Berna, acompanhado por todos os sistemas jurídicos modernos, que faz menção específica à “seleção ou disposição das matérias” como forma de expressão a ser protegida. O autor ainda destaca que “o Memorando Explanatório com que a Comissão fez acompanhar a primeira proposta de diretiva admitia (...) para o caso específico dos programas de computador, que estes revelassem originalidade na compilação de rotinas e algoritmos e, nessa medida, pudessem ser protegidos como compilações.” Complementa ainda que, conforme é aceite usualmente na doutrina, a equiparação dos programas de computador às obras literárias permite a tutela da composição expressiva (forma interna).

²⁷ Barbosa (2001) – *A Proteção do Software*, disponível em: <https://www.dbba.com.br/wp-content/uploads/a-proteco-do-software.pdf>. p. 12, consult. em 15/Abr/2021.

O segundo argumento está relacionado com o art 1.º, n.º 1 da Diretiva 91/250/CEE (e que se mantém na Diretiva 2009/24/CE). Ao incluir o material de conceção na expressão “programas de computador”, ter-se-ia optado por abarcar a “seleção, arranjo e combinação dos módulos e subrotinas do programa de computador”, de forma que não se poderia abandonar a proteção deste trabalho expressivo quanto à proteção do programa de computador em si.²⁸ Não se está por meio deste cotejo a defender a proteção das ideias e dos princípios de um código declarativo. Os métodos de operação, os processos, e até mesmo a sua natureza de interface devem ser rejeitados nesta avaliação, haja visto que a sua funcionalidade como elemento integrativo de programas de computador representa um elemento não expressivo. Nestes termos, o valor criativo do código declarativo não pode surgir da utilização que lhe é atribuída (i.e., função como parcela de uma interface), mas da forma como foi escrito e estruturado.

Vale ainda destacar que, negada a natureza do código declarativo como um programa de computador, não se vislumbra a sua proteção nem mesmo pela disciplina legal da concorrência desleal, visto que, aos olhos do Direito, este consistirá apenas informações (ou dados) cuja proteção escapa até mesmo à proteção oferecida pelo mecanismo da concorrência desleal.²⁹ Esta recusa não encontra fundamentos teóricos e, mesmo em termos empíricos, não justifica a negação total da proteção *a priori*, i.e., antes de qualquer avaliação quanto aos pressupostos básicos do direito autoral. Nestes termos, deve avaliar-se se, ainda que privado de todas os seus elementos não expressivos, o código declarativo dá margem à expressão criativa do seu autor. Porém, antes de realizar esta avaliação, importa destacar que a dicotomia clássica entre ideia e expressão sofreu certos abalos quando passou a ser aplicada aos programas de computador e as doutrinas clássicas não se aplicam de forma tão espontânea.³⁰ Desta forma, deve manter-se em mente que, apesar da equiparação a obras literárias, a proteção autoral atribuída aos programas de computador pelas referidas diretivas é de natureza *suis generis*.

3.3. A originalidade do código declarativo

²⁸ Vieira, J. A. (2005), ob. cit., pp. 48 – 49.

²⁹ Silva, N. S. (2020), ob. cit., pp. 447, 448.

³⁰ Rahmatian, A. (2011) – *Copyright and Creativity: The making of Property Rights in Creative Works*, Cheltenham: Editora Edward Elgar, pp. 136 – 137.

A adoção da proteção da parcela declarativa por meio da referida diretiva, atribuindo-lhe o *status* de programa de computador, retira a necessidade de avaliação estética³¹, porém impõe a demonstração da existência de um espaço de realização expressiva, bem como a utilização deste espaço. Compreende-se que em qualquer API que lide com instruções complexas e um número elevado de signos haverá a existência deste espaço, porém isto não importa ao afirmar-se que haverá uma utilização desta oportunidade criativa. Desta forma, um código declarativo com poucos signos, mas com indícios de criatividade, dificilmente fará jus à proteção do Direito, enquanto um código declarativo com muitos signos, mas alheio de qualquer expressão criativa, jamais poderá ser considerado como uma obra perante o direito autoral.

A proteção individual de signos linguísticos não se mostra verossímil fora do campo da propriedade industrial, diferentemente do que se constata quanto aos signos gráficos. Há uma clareza maior quanto à expressão criativa nos signos presentes interfaces gráficas quando comparadas aos signos presentes em uma interface não gráfica: existem inúmeras formas de satisfatoriamente uma criar representação imagética, mas apenas algumas formas de escrever uma informação com um reduzido número de caracteres. Enquanto os signos gráficos conseguem desempenhar o seu papel como obras audiovisuais satisfatoriamente, o signo linguístico tem muita dificuldade em se adequar aos requisitos de uma obra literária. Não se pode negar que descrever atos complexos de forma sucinta, bem como organizar tais definições de forma intuitiva, pode exigir alguma carga criativa, porém as hipóteses de uma única nomenclatura simplesmente representar uma *scènes à faire* é substancial.

Segundo Alberto de Sá e Mello, “É na organização dos elementos que compõem o programa que se encontra a originalidade da sua expressão (...) Assim, no que respeita ao programa de computador, merece tutela jusautoral não apenas a linguagem que lhe dá expressão, não a sua (eventual) aplicação, não o código, mas o conjunto destes reunido num objecto novo com expressão criativa.”³² Verifica-se, portanto, que a visão conjunta dos signos que formam uma

³¹ Quanto à dispensabilidade da avaliação estética para determinadas obras: Barbosa, D. B. (2014-2015) – *Ensaio e estudos de Propriedade Intelectual, Volume III*, Rio de Janeiro: Edição do Instituto Brasileiro da Propriedade Intelectual, pp. 724 e ss.

³² Mello, A. S. (2020) – “Manual de Direito de Autor e Direitos Conexos”, Coimbra: Edições Almedina S.A., p. 125.

API não apenas é essencial para entender a sua qualificação jurídica mas também é necessária para que se possa dela extrair qualquer sinal de originalidade.

Desta forma, é importante destacar que, na hipótese em que o código declarativo seja considerado um objeto de estudo desligado do conceito de programa de computador, torna-se muito difícil considerar esta parcela da API um objeto de proteção do direito autoral. Conforme observamos, a expressão do código declarativo ocorre em dois momentos, nos *method calls* e na integração realizada pelo código de implementação. Desta forma, não lhe sendo atribuído o reconhecimento de componente integral do código-fonte que dá origem ao programa de computador, seria impossível extrair qualquer significado da estrutura deste código declarativo e a sua condição de interface única e coesa estaria prejudicada.

Manuel Oehen Mendes destaca que “os programas informáticos minimamente complexos – que gozam de uma presunção de *originalidade* (...) utilizam e integram habitualmente diversas ferramentas de programação e compilação, bem como inúmeros componentes pré-existentes (...), que incluem rotinas, sub-rotinas, «scripts», «interfaces» e outros módulos da mais variada natureza, que podem ser facilmente licenciados para ajudar a construir um novo programa, que satisfaça uma ou mais finalidades concretas” – a originalidade surgiria principalmente da organização destes objetos. Porém destaca que, não obstante a utilização destes componentes na criação de um programa transformado seja suficiente à aquisição de direitos de autor, “a sua utilização/exploração depende da autorização do titular dos direitos sobre o programa originário (...)”.³³

Não é factível considerar que o código declarativo é um programa de computador que goza de uma presunção de *originalidade*, afinal a complexidade à qual este autor se refere está na pluralidade de componentes. Há de conferir-se em que grau a sua expressão permite a verificação de criatividade.³⁴ Segundo o Tribunal de Justiça Europeu, esta apuração exige que se apresentem cumulativamente dois requisitos: “(...) é necessário que o objeto em causa seja original, no sentido

³³ Mendes, M. O. (2021) – *Revista de Direito Intelectual Nº1 – 2021*, “A obra enquanto objeto do direito de autor”, Lisboa: Editora Almedina, pp. 69 e 70

³⁴ Mello, A. S. (2020) - ob. cit., pp. 53 – 54.

de constituir uma criação intelectual do próprio autor” e que “sejam a expressão dessa criação intelectual”³⁵.

Desta forma, um código declarativo deve ser capaz de demonstrar o traço pessoal do autor antes de se tornar uma obra protegida. Caso se verifique que a sua criação foi baseada em escolhas que seriam tomadas por qualquer pessoa em circunstâncias similares, será impossível arrancar-lhe qualquer traço de individualidade e, portanto, conceder-lhe a tutela do direito autoral. Sendo a facilidade de compreensão, por meio da qual se incentiva uma utilização espontânea da API, um dos principais critérios para avaliar a qualidade de um código declarativo, é fácil perceber como este requisito pode ir em sentido contrário à sua proteção.

Porém defende-se aqui que se deve observar a criatividade do autor justamente no seu empenho para condensar milhares de ações complexas em signos simples e bem organizados. Desta forma, quanto maior a complexidade e o número de instruções que um código declarativo pretenda abstrair, maior a sua individualidade e menores as hipóteses de que surjam, involuntariamente, criações similares. Não obstante, quanto mais simples e menos numerosas as instruções que pretenda abstrair, menores serão as opções criativas pelas quais o seu autor poderá optar. Conclui-se que a originalidade de um código declarativo anda de mãos dadas com o valor criativo daquilo que pretenda expressar. Caso venha a descrever ações simples, agindo de forma similar a uma linguagem de programação, será difícil delinear com clareza a dicotomia ideia/expressão³⁶.

3.4. Efeitos concorrenciais de uma possível proteção das API

Admitida a existência de originalidade num código declarativo, a integralidade da API estará protegida pelo direito autoral, concedendo ao seu proprietário um direito de exclusivo cujos efeitos serão possivelmente nocivos à concorrência e aos consumidores. Torna-se necessário avaliar a plausibilidade deste resultado, bem como a aptidão dos mecanismos legais existentes na União Europeia para conter estas sequelas e para dirimir receios quanto a esta possível proteção.

³⁵ Acórdão do Tribunal de Justiça (Grande Secção) de 13 de novembro de 2018. *Levola Hengelo BV contra Smilde Foods BV*. Processo C-310/17, n.º 36 e 37.

³⁶ Quanto à imprecisão deste conceito: Hebl, A. B. (2007-2008) - *A Heavy Burden: Proper Application of Copyright's Merger and Scenes a Faire Doctrines*, *Wake Forest Intellectual Property Law Journal*.

Nestes termos, é interessante observar como esta temática interage com a doutrina das *essential facilities*. A recusa ao licenciamento de um direito de propriedade intelectual é evidentemente capaz de “limitar a produção, a distribuição ou o desenvolvimento técnico em prejuízo dos consumidores”. Neste sentido, verifica-se com facilidade que empresas dominantes, caso se encontrem na condição de proprietárias de direito de autor sobre API, sejam capazes de gerar distorções no mercado, impedindo a concorrência efetiva ou o surgimento de novos produtos.

Mesmo que o ponto de uma contenda de natureza concorrencial seja o serviço oferecido pela API e não a sua expressão *per se*, um impedimento à utilização do código declarativo que permite o acesso a este serviço acarretaria o mesmo resultado. Nesta hipótese, não se vislumbra o direito à reprodução como empecilho à utilização, visto que, para a realização de *method calls*, copiar-se-iam apenas os signos de uma API, não a estrutura por meio da qual se verifica a sua originalidade. Como é possível avaliar pelo caso *Microsoft Corp. contra Comissão*³⁷, é no direito de não disponibilizar a obra que surgem os subterfúgios para a utilização nociva da proteção autor sobre a API. Conforme o posicionamento da Comissão Europeia neste caso, que será aprofundado no subcapítulo a seguir, nestes casos é possível vislumbrar a possibilidade de licenciamento compulsório destas API com base no artigo 102.º do TFUE, em detrimento aos direitos de autor dos seus titulares.

Por outro lado, admite-se que o Regulamento (CE) n.º 772/2004³⁸, ao abranger especificamente as licenças de direito de autor sobre programas de computador e introduzir isenções à aplicação do artigo 101.º do TFUE, diminui as ferramentas legais que poderiam impedir o uso abusivo da propriedade sobre o código declarativo.³⁹ No entanto, também é verdade que trouxe como objetivo principal facilitar e incentivar acordos de transferência tecnológica, tornando mais simples o licenciamento de API.⁴⁰ O enquadramento destas licenças no escopo desta norma

³⁷ Acórdão do Tribunal de Primeira Instância (Grande Secção) de 17 de Setembro de 2007. *Microsoft Corp. contra Comissão das Comunidades Europeias*. Processo T-201/04.

³⁸ Regulamento este que já caducou graças ao Regulamento (UE) N.º 316/2014, em vigor desde 1 de maio de 2014. Esta alteração legislativa não modifica, no entanto, o cenário definido pelo antecessor.

³⁹ Pais, S. O. (2011) – *Entre Inovação e Concorrência: Em Defesa de um Modelo Europeu*, Lisboa: Editora Universidade Católica, pp. 346 e ss.

⁴⁰ Pais, S. O. (2011) ob. cit., pp. 311 e ss.

veio justamente como uma tentativa das instituições europeias de dirimir possíveis obstáculos à inovação causados pela proteção dos programas de computador pelo direito de autor.⁴¹

Conclui-se que no ordenamento da União Europeia já existem mecanismos jurídicos suficientemente avançados para lidar com quaisquer efeitos nocivos que possam surgir em função da proteção das API. Também não se encontram razões que justifiquem uma maior precaução quanto aos possíveis efeitos anticoncorrenciais causados pela proteção das API em comparação com aqueles já verificados pela proteção dos programas de computador em geral.

3.5. A API e os Tribunais Europeus

O acórdão BSA, de 2010,⁴² aparenta resolver a questão ao descartar a aplicação da Diretiva 2009/24/CE às interfaces gráficas com base no argumento de que estas seriam funcionalidade do programa de computador. Conforme anteriormente observado, o direito autoral protege o programa de computador na sua expressão como código-fonte ou objeto, o resultado da interação deste código com o computador não integra esta definição. As interfaces gráficas permitem a interação do utilizador com o código-fonte, porém a sua existência é consequência da interpretação do código-fonte por uma máquina. Como já se destacou no capítulo anterior, esta característica não pode ser atrelada ao código de declaração da API uma vez que este não surge como resultado da interação do *software* com o *hardware*, mas como parcela integrante da expressão (código-fonte) do programa de computador.

No processo C-263/18⁴³, a Tom Kabinet tentou classificar os *e-books* como programas de computador com o objetivo de ver aplicada a disposição do artigo 4.º, n.º 2, da Diretiva 2009/24, referente ao esgotamento do direito de distribuição, conforme ocorrera previamente no caso

⁴¹ Pais, S. O. (2011) ob. cit. , pp. 327 e 328.

⁴² Acórdão do Tribunal de Justiça (Terceira Secção) de 22 de Dezembro de 2010. Bezpečnostní softwarová asociace - Svaz softwarové ochrany contra Ministerstvo kultury. Processo C-393/09.

⁴³ Acórdão do Tribunal de Justiça (Grande Secção) de 19 de dezembro de 2019. Nederlands Uitgeversverbond e Groep Algemene Uitgevers contra Tom Kabinet Internet BV e o., Processo C-263/18.

UsedSoft⁴⁴. Entretanto, os programas de computador possuem apenas carácter acessório relativamente às obras contida nos livros, razão pela qual o TJUE entendeu não ser possível atribuir este rótulo a livros eletrónicos. Descartada a aplicação desta *lex specialis*, entendeu-se que não haveria direito ao esgotamento devido a aplicação do art. 4.º, n.º 2 da Diretiva InfoSoc (2001/29/CE) – rigidez interpretativa que foi alvo de críticas da professora Maria Victória Rocha por questões relativas à liberdade de circulação de bens e livre concorrência, bem como às políticas culturais e ao respeito dos direitos fundamentais⁴⁵. Porém, não obstante a qualidade desta decisão quanto ao tema do esgotamento no ambiente virtual, nela pôde verificar-se mais um passo importante em prol da delimitação de um conceito mais preciso para os programas de computador.

À vista disso, é pertinente destacar a opinião do Advogado-Geral Maciej Szpunar, que se mostrou determinante à posição adotada pelo TJUE. Segundo ele, um programa de computador seria “mais uma ferramenta do que uma obra no verdadeiro sentido da palavra” e que se expressaria como “um conjunto de instruções para que o computador realize determinadas operações”⁴⁶. É evidente que um *e-book* não preenche esta definição, porém é difícil negar a similaridade entre esta definição e aquela que anteriormente vinculamos às API.

Outrossim, conforme foi demonstrado, a API não constitui uma funcionalidade de um programa de computador, uma linguagem de programação ou um formato de ficheiro, de forma que o cotejo realizado no processo C-406/10, referente ao caso SAS Institute Inc. contra World Programming Ltd.⁴⁷, também não nos fornece amparo jurisprudencial. Ainda que fossem traçadas semelhanças entre o código de declaração e a linguagem de programação, o tribunal determinou, *in casu*, que “nem a funcionalidade de um programa nem a linguagem de programação e o formato de ficheiros de dados utilizados no âmbito de um programa de computador para explorar algumas das suas funções constituem uma forma de expressão desse programa na aceção do artigo 1.º, n.º 2, da Diretiva 91/250” porém, por fim, salientou que tal observação “não afeta a possibilidade de

⁴⁴ Acórdão do Tribunal de Justiça (Grande Secção) de 3 de julho de 2012. UsedSoft GmbH contra Oracle International Corp. Processo C-128/11.

⁴⁵ Rocha, M. V. (2021) – *Decisão Tom Kabinet: Um Forte Retrocesso do Esgotamento Digital na União Europeia?*. Comunicação na Universidade de Vigo, III Congreso Internacional C. Fernández Nóvoa: Nuevas Tendencias en el Derecho de la Competencia y de la Propiedad Industrial e Intelectual, 28 de setembro de 2021, publicação em preparação.

⁴⁶ Conclusões do advogado-geral M. Szpunar apresentadas em 10 de setembro de 2019, Processo C-263/18.

⁴⁷ Acórdão do Tribunal de Justiça (Grande Secção) de 2 de maio de 2012. SAS Institute Inc. contra World Programming Ltd., Processo C-406/10.

a linguagem SAS e de o formato de ficheiros de dados da SAS Institute beneficiarem, enquanto obras, da proteção do direito de autor, por força da Diretiva 2001/29, se forem uma criação intelectual própria do seu autor”. Esta última observação apenas contribui para levantar mais questionamentos quanto aos critérios de proteção exigidos pelos Tribunais Europeus.

Foi apenas numa decisão da Comissão Europeia em 2006⁴⁸ que se verificou o enfrentamento da temática das API. A Comissão verificou que a Microsoft infringira o artigo 82.º do Tratado CE (atual 102.º do TFUE) ao se recusar a fornecer “(...) informações à Sun que lhe permitissem conceber sistemas operativos de servidores de grupos de trabalho que possam integrar, ininterruptamente, a arquitetura do domínio «Active Directory» (...)”, ressaltando ainda que “a Microsoft apenas teria de fornecer as especificações dos protocolos relevantes, ou seja, documentação técnica, e não dar acesso aos códigos do *software* do Windows e muito menos permitir a sua reprodução pela Sun.” Em outras palavras, verificou-se que o código declarativo, bem como a documentação que descreve a sua utilização, poderiam representar uma *essencial facility* capaz de impedir o desenvolvimento de novos produtos. Outro ponto interessante vem ao fim desta decisão, quando se enuncia: “Visto que a decisão pode exigir que a Microsoft se abstenha da plena aplicação dos seus direitos de propriedade intelectual, a mesma justifica-se pela necessidade de pôr termo ao abuso.”

3.6. Conclusões quanto à proteção das API na União Europeia

Desta forma, verifica-se que há um surpreendente vácuo normativo e jurisprudencial quanto a este tema. Ainda assim, é possível compreender que a simples circunstância de atuar de forma indireta não é suficiente para deturpar a sua natureza com um programa de computador, afinal, verdade seja dita, qualquer código-fonte é fundamentalmente indireto – para que um programa de computador escrito em código-fonte comunique com um computador, este deve ser transformado, por meio um de um compilador específico à linguagem de programação utilizada, em um código objeto.

Apenas se vislumbra excluir o código declarativo da proteção da Diretiva 2009/24/CE por um receio exacerbado em relação aos potenciais efeitos negativos que uma proteção de código

⁴⁸ Decisão da Comissão, de 24 de maio de 2006, relativa a um processo nos termos do artigo 82.º do Tratado CE e do artigo 54.º do Acordo EEE contra a Microsoft Corporation. Processo COMP/C-3/37.792 — Microsoft.

poderia causar no mercado de tecnologia. Entretanto, basta lembrar-se que na ausência de um espaço de realização expressiva não haverá proteção para compreender que tais receios são despropositados. Também se destaca que o código declarativo, ao demonstrar sua criatividade na forma concatenada em que seus signos são formulados e organizados, apenas pode ser protegido de forma global, sendo necessária uma reprodução substancial de seu conteúdo para que se possa considerar a reprodução do seu todo.

Neste sentido, é difícil vislumbrar uma situação na qual um programador de boa-fé poderá produzir um código declarativo que reproduz centenas de linhas de código de outra API, com uma organização em grupos e subgrupos idêntica, acidentalmente. A única circunstância que permitiria tal desdobramento seria na ausência de formas alternativas de gerar a mesma interface com um código declarativo diferente, hipótese esta que retira qualquer proteção da API “original” pois esta, na sua concepção, já estaria alheia de qualquer criatividade, sendo totalmente funcional. Jamais se pode proteger a única forma de expressão de um determinado objeto. Desta forma, defende-se que os únicos argumentos que devem influenciar a decisão acerca da proteção são aqueles ligados à própria natureza do objeto, devendo ser deixadas de lado as teorias catastróficas que sugerem o fim do mercado de tecnologia em função de um simples reconhecimento de autoria.

A decisão da Comissão Europeia no caso Microsoft veio a demonstrar como os direitos intelectuais não representam uma barreira à repressão de utilizações abusivas mecanismos de interoperabilidade, tornando desnecessário um posicionamento antagónico a este fundamento. Portanto, na exceção de uma determinada API representar uma barreira à inovação, verifica-se que a União Europeia já formulou mecanismos legais atentos a tais circunstâncias.

Conforme se tentou esclarecer, a proteção do código declarativo jaz na sua inclusão no escopo de proteção presente no artigo 1.º da Diretiva 2009/24/CE, uma vez que dificilmente a sua proteção autoral subsistirá sem o reconhecimento da natureza de código-fonte. Apesar destas observações, não é possível oferecer uma resposta definitiva quanto à atual proteção desta parcela da API, bem como é difícil prever a forma como se dará a construção jurisprudencial deste tema com base nas decisões anteriores dos Tribunais Europeus.

Nada obstante, a União Europeia não existe em um vácuo jurídico e influências externas já trouxeram alterações substanciais às “regras do jogo”. A realidade é que a sociedade da informação

tornou inverosímil a existência de contrastes relevantes na forma como os ordenamentos jurídicos abordam determinados objetos que se mostram imprescindíveis à existência do mercado globalizado, como os programas de computador e, mais recentemente se verificou, as API. Neste sentido, é fácil entender pertinência do case americano Google LLC v. Oracle America, Inc. para a compreensão da atual conjuntura deste tema fora da Europa.

4. Análise do caso Google LLC v. Oracle America, Inc.

4.1. As circunstâncias gerais

Em 5 Abril de 2021, após grande anseio do mundo jurídico e tecnológico, chegou ao fim a batalha de mais de uma década entre as gigantes Google e Oracle. O caso teve origem quando, em 2005, foram utilizadas aproximadamente 11 500 linhas da parcela declarativa da API Sun Java, pertencente à Oracle, na criação do sistema Android, o notório sistema operacional de *smartphones* criado pela Google.

A utilização da referida API era excepcionalmente comum no ambiente digital, razão pela qual a Google, visando atrair programadores já familiarizados com este sistema, considerou conveniente adotá-la no seu sistema Android. Ao realizar esta integração, apenas a parcela declarativa da API fora copiada e um código-fonte diferente foi criado para a sua implementação no novo sistema. A decisão da Google mostrou-se acertada e o seu sistema foi amplamente adotado por utilizadores, tornando-se a proprietária do principal sistema operacional de *smartphones* do mundo.

Por outro lado, a Oracle, dada a livre disponibilidade do sistema Android, sofreu diversos prejuízos com o fracasso do modelo de negócios baseado no licenciamento de API, como se pôde observar com o desconto de 97,5 % num contrato com a Amazon e a redução do valor do contrato de 40 milhões de dólares para apenas 1 milhão num contrato com a Samsung. Basta destacar o facto de que a Google por quatro vezes tentou obter a licença para a utilização da API Sun Java para compreender o valor que esta possuía no mercado de telemóveis.

Nos Estados Unidos, a norma mais relevante está presente no Copyright Act, o qual determina que um programa de computador é um conjunto de declarações ou instruções, utilizadas diretamente ou indiretamente, num computador com o objetivo de alcançar um determinado resultado. Neste sentido, a norma norte-americana não parece fazer distinções entre as interfaces e o código que diretamente dá origem ao programa, pois inclui na sua definição declarações que indiretamente são utilizadas pelo computador. Há de destacar-se também que a forma como as obras são protegidas pelo sistema *copyright* americano difere dos sistemas jurídicos derivados do

droit d'auteur e, em especial em relação aos “usos livres”, será necessária uma análise mais aprofundada destas diferenças.

4.2. A análise do Supremo Tribunal Americano

Ao longo do processo, as diferentes instâncias viram-se compelidas a responder a dois questionamentos: Estaria a API protegida pelo *copyright*? Na existência de tal proteção, estaria a utilização realizada pelo Google abrangida pelo *fair use*, tornando-a permitida? Num primeiro momento, o Tribunal de Recurso Federal concluiu que as linhas de código declarativo copiadas pela Google seriam passíveis de proteção pelo sistema *copyright* americano e o júri compreendeu que, *in casu*, se tratava de um *fair use*. Ainda que a conclusão do júri tenha sido revertida por este tribunal, o próprio terminou por reverter novamente a decisão, chegando-se à conclusão inicial de que se tratava de um caso de *copyright* e de *fair use*.

O Supremo Tribunal Americano assumiu uma posição diversa. Sob o argumento de que as circunstâncias tecnológicas, económicas e negociais estariam em constante mudança, decidiu não analisar o primeiro questionamento, assumindo-se a sua veracidade em prol da discussão quanto ao *fair use*. Retira-se da decisão do Tribunal que a identificação de um *fair use* tornaria desnecessária a discussão quanto à proteção da API Sun Java. Há que concordar com a declaração do juiz Clarence Thomas: “*The Court wrongly sidesteps the principal question that we were asked to answer: Is declaring code protected by copyright?*”. É também do seu voto que se retira a única conclusão quanto a este questionamento: “*I would hold that it is.*”⁴⁹. Infelizmente, não se pode basear um estudo apenas em uma opinião dissidente da daquela que se consagrou vitoriosa. Nada obstante, a decisão fornece-nos pistas do seu posicionamento dentro da sua extensa análise quanto aos critérios do *fair use*, assim como dos seus contrapontos ao voto do juiz Clarence Thomas, que foi acompanhado pelo juiz Samuel Alito. Neste sentido, apesar deste posicionamento inusitado, a decisão do Tribunal fornece-nos uma grande quantidade de informações quanto ao cenário legal e jurídico americano no que se refere à proteção das API.

Os argumentos apresentados pelo juiz Stephen Breyer, representantes da opinião emitida por seis membros do Tribunal, apesar de se concentrarem na questão do *fair use*, fornecem diversas

⁴⁹ Thomas, J., Voto Dissidente (2021), *Google LLC v. Oracle Am., Inc.* 141 S. Ct. 593 U. S., p. 4.

pistas sobre o posicionamento do tribunal. A sua apreciação quanto à natureza das API deu origem à divisão em “*implementing code*” e “*declaring code*”, cujas nomenclaturas e definições certamente formarão o alicerce de futuras discussões quanto ao tema, como o presente trabalho acadêmico. Além disso, conforme dito, a sua apreciação dos limites negativos da proteção das API permite uma melhor compreensão dos critérios para uma potencial proteção, razão pela qual será realizado um cotejo específico quanto às razões que levaram o Tribunal a concluir em favor do *fair use*.

Segundo o Tribunal, os criadores da API Sun Java, ao formularem o código de declaração, optaram por nomes que seriam de fácil recordação dando origem a um código focado no utilizador. Desta forma, o uso de um código de declaração estaria atrelado a ideias que não são passíveis de proteção pelo *copyright*, assim como a criações expressivas, como o código de implementação. Ainda, o valor do código declarativo estaria mais associado àqueles que não são os proprietários do *copyright*, i.e., os programadores, que, segundo o magistrado: “*invest of their own time and effort to learn the API’s system*”⁵⁰. O Tribunal conclui esta análise inicial evidenciando que: “*the declaring code is, if copyrightable at all, further than are most computer programs (such as the implementing code) from the core of copyright.*”⁵¹ Verifica-se claramente que o Tribunal, ainda que não assuma uma posição definitiva, também não está convicto da proteção do código declarativo pelo *copyright*. Neste sentido, os magistrados não parecem identificar níveis relevantes de criatividade na formulação dos códigos declarativos, apenas na sua utilização, constatação que lhes direciona à aplicação do *fair use*.

Em sentido contrário, o voto dissidente concorda que o código de declaração estava atrelado a objetos que fogem da proteção por *copyright*, porém questiona que tipo de obra literária não estaria sujeita a esta mesma circunstância. Segundo o juiz Clarence, “*Books are inherently bound with uncopyrightable ideas—the use of chapters, having a plot, or including dialogue or footnotes. This does not place books far «from the core of copyright».*” E utiliza ainda outra analogia para rejeitar a posição do Tribunal: “*A Broadway musical script needs actors and singers to invest time learning and rehearsing it. But a theater cannot copy a script—the rights to which are held by a smaller theater—simply because it wants to entice actors to switch theaters and*

⁵⁰ Breyer, J., Opinião do Tribunal (2021), Google LLC v. Oracle Am., Inc. 141 S. Ct. 593 U. S., p. 24.

⁵¹ Ibid.

because copying the script is more efficient than requiring the actors to learn a new one.” Por fim, conclui a sua argumentação quanto a este primeiro ponto lembrando que o próprio Tribunal declarara que os códigos de implementação e de declaração seriam intrinsecamente dependentes, porém neste cotejo ignorou que o valor atribuído pelos programadores ao código de implementação é diretamente proporcional àquele do código de declaração.⁵²

Definida, ainda que de forma inconclusiva, a natureza do objeto em análise, o Tribunal analisou o propósito e o tipo de uso praticado pela Google. Tratou-se então de se verificar se a utilização do código declarativo da API Sun Java no sistema Android se mostrava transformativa, ou seja, se agregava algo relevante ou original à obra. Considerou-se que o simples facto de introduzir o código de declaração a um novo sistema e, portanto, a um novo mercado, já tornaria evidente o efeito transformativo desta utilização. Porém admite-se que o facto de este uso ser estritamente de natureza comercial e possivelmente de má-fé, pesa em sentido contrário à aplicação do *fair use*.⁵³ O juiz Thomas, por sua vez, ironiza a decisão do Tribunal, argumentando que esta atribui à definição de uso transformativo um sentido efémero. Neste sentido, mesmo que um uso se mostre idêntico, bastaria que este permitisse a criação de novos produtos para que fosse considerado transformativo. Segundo ele, esta definição seria o fim do *copyright*, uma vez que estariam protegidas obras que possuem os mesmos propósitos e tipos de uso, ou seja, obras derivadas.⁵⁴

Quanto à proporção e substancialidade do código declarativo copiado pela Google, o Tribunal compreende que se devem observar as 11 500 linhas no contexto geral da API Sun Java. Neste sentido, somadas as linhas referentes ao código de implementação e de declaração da totalidade da API Sun Java, estas linhas representariam apenas um total de 0,4 por cento de um total de 2,86 milhões. Ainda que estas linhas constituíssem o “coração” da obra (facto este reconhecido pela Google) e, portanto, representassem uma parcela substancial da obra, o Tribunal compreende que a utilização deste código era necessária para a concretização do referido uso transformativo.⁵⁵ O voto dissidente critica a utilização do código de implementação na contagem

⁵² Thomas, J., Voto Dissidente (2021), ob.cit., pp. 9 - 11.

⁵³ Breyer, J., Opinião do Tribunal (2021), ob.cit., pp. 24 -28.

⁵⁴ Thomas, J., Voto Dissidente (2021), ob.cit., pp. 15 - 17.

⁵⁵ Breyer, J., Opinião do Tribunal (2021), ob.cit., pp. 28 – 30.

total de linhas, ressaltando que foi o código de declaração que atraiu os programadores e não a implementação realizada pela Google.⁵⁶

Segundo o Tribunal, a Sun (empresa criadora da API Sun Java, adquirida pela Oracle) não possuía capacidade de entrar no mercado de telemóveis e, nas ocasiões em que a API Sun Java foi utilizada em dispositivos portáteis, chegou-se a resultados muito distintos. Além disso, o investimento que os programadores haviam colocado em aprender a API Sun Java seria perdido caso não se pudesse transpor este conhecimento ao mercado de *smartphones*, o que restringiria a criatividade no ambiente tecnológico. Desta forma, não apenas teriam as API das duas empresas sido utilizadas em mercados diversos, dever-se-ia levar em consideração os efeitos positivos ao mercado acarretados pela reprodução realizada pela Google.⁵⁷

O juiz Thomas, por outro lado, destacou que as diferenças nas estratégias de mercado da Google e da Oracle impediam questionamentos quanto às razões pelas quais a segunda deixou de entrar no mercado de *smartphones*. Enquanto a Google estava interessada em distribuir o seu *software* gratuitamente, obtendo lucro por meio de anúncios que geravam coleta de dados dos utilizadores no meio Android, a Oracle dependia do licenciamento das suas API. Ao tornar gratuito um código que anteriormente exigiria licenciamento, é evidente que a API Android teria uma adoção maior e seria mais benéfica para os programadores. Além disso, tanto a Apple quanto a Microsoft foram capazes de criar suas próprias API para o mercado *mobile*, tornando fraco o argumento de que o espaço criativo dos programadores seria limitado pela necessidade de licenciamento. Desta forma, não apenas a Google deixou de realizar o licenciamento do código declarativo da Oracle, tornou dispensável a necessidade de qualquer outro *player* do mercado de *smartphones* fazê-lo (conforme se verifica pelo desconto de 97,5 % na licença da API Sun Java para a Amazon com base na livre disponibilidade do sistema Android).⁵⁸

Tendo em conta que qualquer utilização *fair use* está intrinsecamente associada à existência de um direito de *copyright* passível de violação, seria possível assumir o entendimento de que a decisão do Tribunal tende a reconhecer implicitamente a proteção da parcela declarativa da API. Desta forma, ao evitar o cotejo do primeiro questionamento devido a uma suposta obsolescência,

⁵⁶ Thomas, J., Voto Dissidente (2021), ob.cit., pp. 17, 18.

⁵⁷ Breyer, J., Opinião do Tribunal (2021), ob.cit., pp. 30 – 35.

⁵⁸ Thomas, J., Voto Dissidente (2021), ob.cit., pp. 11 – 14.

seria possível concluir que este óbice estaria associado apenas à própria análise. Neste sentido, ainda que tendesse a concordar com o posicionamento do Tribunal de Recurso Federal no sentido de reconhecer a proteção do código declarativo, o Tribunal não se considerava preparado para uma análise profunda quanto ao tema.

Ou a conclusão poderia ser justamente em sentido contrário. Evitar o principal questionamento de um processo com tamanha relevância também pode significar que o Tribunal não está confiante na existência desta proteção e tende a negá-la, visão esta que, conforme-me já se observou, parece muito provável. O juiz Clarence Thomas deixou claro o seu receio de que esta se torne a interpretação predominante, haja em conta que, na sua visão, tal posicionamento seria contrário ao objetivo traçado pelo Congresso Americano por meio do Copyright Act.

Uma outra interpretação possível, com a qual mais me identifico, é que o Supremo Tribunal Americano sabia a importância desta decisão e ainda não se sentia confortável em assumir uma posição sem antes ter a oportunidade de observar a relevância das API na sociedade e no mercado de tecnologia. Neste sentido, assumiu uma postura mais pragmática, abrindo mão da segurança jurídica em prol de uma decisão mais madura em um momento futuro. Ainda, considerando a relevância do tema e o interesse de certos *players* muito relevantes, seria possível transferir esta tarefa para o Congresso Americano, que em algum momento seria impelido a legislar sobre esta questão.

Porém, ainda que exista uma incerteza quanto à posição de sete dos nove membros do Supremo Tribunal Americano, é possível afirmar que – apesar de toda a pressão exercida pela maior companhia de tecnologia do mundo – não estão convictos quanto à inexistência deste direito, razão pela qual a possibilidade de compreender a visão daqueles que assumiram uma posição tem grande relevância acadêmica. Nada obstante, o voto realizado pelo juiz Clarence Thomas é divergente apenas no que se refere à necessidade de existir um julgamento quanto à proteção da parcela declarativa da API, não sendo possível presumir que a posição dos outros membros do Tribunal seria contrária. Para o proveito deste trabalho, o juiz não se limitou a criticar a ausência de uma análise e de uma decisão, mas ofereceu-nos um cotejo minucioso das razões pelas quais considera necessário o reconhecimento do direito de *copyright* para o objeto em estudo.

Conforme se destacou anteriormente, o Copyright Act atribui ao conceito de programa de computador uma amplitude que vai muito além do código-fonte que permite diretamente a execução de funções em *hardware*. A previsão do §101 da referida norma apresenta a seguinte definição: “A «*computer program*» is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.” Neste sentido, o voto sublinha que o código que declara as instruções da API estaria claramente incluído na definição de *software*, uma vez que indiretamente realiza funções computacionais ao acionar o código-fonte já implementado.⁵⁹

Porém, segundo a opinião do magistrado, não seria necessário ater-se a especificidade da referida previsão visto que o código declarativo já satisfaria o teste genérico de proteção por *copyright*. Retira-se do §102(a) do referido ato que “*Copyright protection subsists (. . .) in original works of authorship fixed in any tangible medium of expression.*”, assim como da convergência dos §§101, 102(a) que: “*Works of authorship include (. . .) literary works,*” quer sejam “*works (. . .) expressed in words, numbers, or other verbal or numerical symbols.*” Portanto, no seu entendimento, haveria um critério extremamente baixo para determinar se uma expressão seria um objeto passível de proteção pelo *copyright*, garantindo assim a inclusão do código declarativo no seu rol de proteção.⁶⁰

Segundo a jurisprudência americana, representada pelo caso *Feist Publications, Inc. v. Rural Telephone Service Co.*, (1991), bastaria que uma obra fosse criada independentemente por um autor e possuísse uma quantidade módica de criatividade para que pudesse ser considerada original. Desta forma, *in casu*, a originalidade surgiria da simples premissa de que a Oracle poderia ter criado o código declarativo de diversas formas. A própria Google reconheceu este facto, razão pela qual optou por contestar a própria existência de uma expressão a ser protegida, acusando o código declarativo de constituir apenas uma ideia.

Segundo a Google, o código declarativo representaria um método, além de possuir uma natureza estritamente funcional. Neste sentido, apenas o código de implementação estaria protegido pelo *copyright*, parcela esta que não foi copiada com a criação do sistema Android.

⁵⁹ Thomas, J., Voto Dissidente (2021), ob.cit., pp. 4, 5.

⁶⁰ Ibid.

Porém o voto destaca que, mesmo sem analisar a proteção da API, a decisão proferida pelo Supremo Tribunal Americano reconheceu que as duas parcelas da API estavam “*inextricably bound*”, e que, assim como a parcela declarativa não possuiria função sem o código de implementação, o mesmo ocorreria em sentido contrário. Isto é, uma vez que o código de implementação se encontraria inaproveitável sem o código de declaração, não se poderia proteger um sem proteger o outro. Argumenta ainda que esta proteção seria justamente a razão pela qual o Congresso teria optado por permitir a proteção do código que seria utilizado apenas indiretamente pelo computador.

Por fim, avaliou-se a incidência da “*merger doctrine*”, doutrina americana segundo a qual não deveriam ser protegidas pelo *copyright* as expressões de ideias que representem a única, ou uma das únicas, formas de representar estas ideias. Segundo o magistrado, esta doutrina apenas seria um desdobramento do requisito de originalidade previsto no referido §102(a) do Copyright Act, anteriormente discutido, destacando ainda a forma como a Apple e a Microsoft foram capazes de criar os seus próprios códigos declarativos, inexistindo qualquer razão para que a Google não pudesse fazer o mesmo.

5. O caso Google LLC v. Oracle America, Inc. sob a perspectiva da União Europeia

Haja visto o interesse associado à garantia de um elevado nível de harmonização entre as diferentes formas de proteção jurídica sobre bens virtuais ao redor do globo, as diferenças fundamentais entre os sistemas *copyright* e direito de autor são muitas vezes ignoradas em prol de uma lógica normativa similar. Desta forma, assim como a alteração dos §§ 101 a 107 do U.S. Code influenciou a forma como se deu a proteção jurídica de programas de computador na Europa, há de verificar-se se as recentes evoluções jurisprudenciais quanto às API poderiam causar um efeito similar.

Não obstante, conforme se pôde verificar por meio da análise do acórdão do Supremo Tribunal Americano no caso Google LLC v. Oracle America, Inc., ainda não é possível extrair uma doutrina americana quanto à proteção de API. Por outro lado, ainda que a ausência de uma conclusão concreta torne improvável qualquer influência relevante desta decisão em relação ao conjunto normativo e jurisprudencial da União Europeia, faz-se possível avaliar em que medida a análise e a conclusão assumidas pelo Tribunal Americano se coadunam com este ordenamento. Nestes termos, é importante, brevemente e de forma introdutória, destacar algumas diferenças entre os sistemas de proteção por *copyright*, de origem inglesa, e aqueles provenientes do *droit d'auteur*, de origem francesa, como o sistema de direito autoral da União Europeia.

A origem desta heterogeneidade está nos fundamentos que justificam a existência de cada um destes sistemas. Enquanto o *copyright* foi formulado para promover “o progresso da ciência e das artes úteis”, no *droit d'auteur* há um maior foco na figura do autor, afigurando-se a obra como uma extensão da personalidade do seu criador. Esta visão mais idílica e menos pragmática do modelo francês, além atribuir uma maior significância aos direitos morais (pouco desenvolvidos no sistema anglo-saxónico), também acarreta diversas diferenças em relação aos requisitos e limites da proteção.⁶¹

No que se refere ao requisito de originalidade, é interessante verificar que em ambos sistemas não basta que a obra não seja uma cópia, visto que a atividade criativa também é exigida

⁶¹ Silva, N. S. (2014) – “Uma Introdução ao Direito de Autor Europeu”, Portal da Ordem dos Advogados, <https://portal.oa.pt/upl/%7Bdca5e510-7e6b-403c-9e2c-fd091a1cb5dc%7D.pdf>, p. 1336, consult. em 18/Ago/2021.

(com exceção do Reino-Unido, que funciona com base na máxima “*What is worth copying is prima facie worth protecting*”⁶²). Entretanto, os níveis de exigência variam enormemente entre os tribunais americanos e os tribunais da UE. Enquanto no direito de autor continental se busca nas obras elevados graus de autoria e atividade intelectual, o professor Nuno Sousa e Silva esclarece que os sistemas de *copyright* são menos exigentes pois estão focados na proteção do investimento.⁶³ Entretanto, o próprio autor reconhece que existe uma certa incoerência quanto ao critério europeu de originalidade e aplicação dos tribunais: “De facto, ao afirmar que 11 palavras (Infopaq) ou que um manual de utilização de um programa de computador (SAS Institute) podem ser protegidos por direitos de autor, o TJUE parece ter acolhido um critério pouco exigente em termos de originalidade.”⁶⁴

Conforme abordado nos capítulos anteriores, a constatação de originalidade é primordial à análise quanto à proteção das API e, ainda que seja possível a existência de um espaço criativo no código de declaração, a verificação da sua existência e da sua utilização não é garantida. Se mesmo o Supremo Tribunal Americano, com o reduzido rigor do sistema *copyright*, deixou claro não vislumbrar elevados níveis de originalidade na API Sun Java, há de avaliar-se que num Tribunal Europeu não se verificaria cumprido tal requisito.

No entanto, a API Sun Java é um objeto de estudo muito particular uma vez que foi concebida com o intuito de permitir uma utilização mais eficiente da linguagem Java, chegando a ser denominada “*language-level API*”⁶⁵. Este fator por si só não retira qualquer oportunidade de expressão criativa, mas evidencia que o objetivo principal dos seus autores não era formular algo único, mas simplesmente intuitivo. Conforme o cotejo já realizado neste trabalho, isto não aproxima a API Sun Java de uma linguagem de programação, porém traz à tona as mesmas questões que geralmente as afastam da proteção autoral. Neste sentido, ainda que milhares de linhas de código permitam amplo espaço criativo, uma API cuja função seja apenas resumir

⁶² “Aquilo que vale a pena copiar, *prima facie* vale a pena proteger” - University of London Press v University Tutorial Press (1916).

⁶³ Silva, N. S. (2014), ob. cit., p. 1341.

⁶⁴ Silva, N. S. (2014), ob. cit., p. 1372.

⁶⁵ Sarid, Uri, s.d., ob. cit.

conjuntos de código de forma frugal com muita dificuldade permitirá ao autor “imprimir o seu «toque pessoal» à obra criada”⁶⁶.

Como é de costume no direito autoral europeu, não há uma saída fácil para a complexidade dos problemas trazidos pelas rápidas mudanças tecnológicas e culturais do novo milénio - o *fair use*, a válvula de escape do sistema *copyright* americano, possui uma complacência que não é encontrada em nenhum dos Estados-Membros da UE. O *numerus clausus* de limitações e exceções previstos nas legislações destes países, fruto da Convenção de Berna e do Acordo Trips, bem complementado pelo artigo 5.º na Diretiva 2001/29/CE, relativa à sociedade da informação, não permitiriam uma solução similar à verificada no caso Google LLC v. Oracle America, Inc.⁶⁷

Mesmo o *fair use* teve a sua flexibilidade levada ao limite para permitir a decisão do Supremo Tribunal Americano, conforme é possível extrair pelo voto divergente do juiz Clarence Thomas. Não há dúvidas de que o argumento referente ao uso transformativo do código declarativo encontraria obstáculos ainda maiores na União Europeia. Isto advém do objetivo principal do sistema *copyright*, previsto na Constituição Americana, qual seja, “*to promote the Progress of Science and useful Arts (...)*”⁶⁸ ainda que, possivelmente, em detrimento dos autores – visão contrária àquela formulada no sistema de direito de autor continental. Outro fator que explica esta dicotomia entre os sistemas está no contraste entre a *common law* e os ordenamentos romanísticos, tendo em vista que o primeiro privilegia os precedentes jurisprudenciais frente a rigidez da lei, enquanto o segundo busca na codificação uma maior segurança jurídica.

Desta forma, com o inevitável advento de um caso similar dentro da jurisdição da UE, o TJUE dificilmente encontrará na regra dos três passos uma saída para este nó górdio. É importante destacar que a regra dos três passos exige que a “(...) reprodução não afete a exploração normal da obra nem cause prejuízo injustificado aos interesses legítimos do autor”, enquanto o *fair use* verifica o prejuízo da reprodução perante um determinado mercado. O benefício acarretado pela cópia exercida pela Google foi fundamental para a decisão do tribunal americano, apesar de representar um fator irrelevante à aplicação da referida regra. Isto não significa, no entanto, afirmar

⁶⁶ Acórdão do Tribunal de Justiça (Terceira Secção) de 1 de dezembro de 2011. Eva-Maria Painer contra Standard VerlagsGmbH e outros. § 92. Processo C-145/10.

⁶⁷ Hugenholtz, P. B. (2017) – *Flexible Copyright: Can the EU Author’s Rights Accommodate Fair Use?*, col. “Copyright Law in an Age of Limitations and Exceptions”, Cambridge: Cambridge University Press, p. 276.

⁶⁸ Artigo 1.º, Seção 6 da Constituição dos Estados Unidos.

que o TJUE será obrigado a assumir uma decisão definitiva quanto à proteção das API. A análise feita anteriormente demonstra que este tribunal está inclinado a permitir que este tipo de matéria seja tratado casuisticamente.⁶⁹

Porém, conforme se tentou demonstrar, serão raras as situações nas quais haverá uma coincidência relevante entre dois códigos declarativos criados em ambientes separados. Os casos mais comuns serão similares ao exemplo americano – isto é, uma empresa reproduzindo de forma proposital a parcela declarativa de outra. Assim se verifica a utilização do argumento “limitação do espaço de criatividade” como uma forma de camuflar o real objetivo deste debate que é dispensar a necessidade de licenciamento do código declarativo. Entretanto, o sistema de proteção autoral não deve ser limitado por argumentos provenientes da inércia. É evidente que a exigência de licenciamento pode acarretar diversos atrasos tecnológicos graças à necessidade de reimplementar soluções preexistentes; porém também é verdade que esta obrigação serve de incentivo à inovação pois evita comportamentos parasitários.

Conforme já foi demonstrado, a União Europeia tem empenhado esforços para facilitar e incentivar os acordos de transferência tecnológica, bem como para impedir o uso abusivo das propriedades sobre bens intelectuais. Constata-se nesta postura a possibilidade de fomentar a inovação por meio do incentivo à colaboração entre as empresas, em contraposição a alternativa mais simples, porém precária, de negar ou limitar a proteção autoral a criações que resultam do tempo e esforço despendido por diversos programadores.

⁶⁹ Recomenda-se a leitura de um artigo de Nuno Sousa e Silva para uma visão mais aprofundada sobre a hipótese enunciada neste capítulo: Silva, N. S (2022) - *Ligações Perigosas? – Reflexões Sobre Apis e Direito de Autor a Partir do Acórdão Google V. Oracle do Supremo Tribunal dos EUA*, Revista de Direito Intelectual – N.º 01—2022: 217-231.

6. Conclusão

Os direitos de autor podem ser vistos como um obstáculo inconveniente ou uma ferramenta em prol da inovação, a depender de quem (e quando) se questione. Por sorte, a sua existência não está vinculada ao posicionamento flutuante dos grandes *players* do mercado tecnológico internacional, mas a numerosas décadas de desenvolvimento jurídico-normativo. Ainda que se admita que as bases desta disciplina legal sofreram certos abalos com a introdução da proteção dos programas de computador, não se justifica o abandono do debate académico em prol da conveniência do mercado. A propriedade intelectual é historicamente uma área de estudo jurídico que nunca se contentou com simplificações excessivas e sempre admitiu a necessidade de compreender as peculiaridades de todos os seus objetos de estudo. Justifica-se assim um mergulho em questões mais técnicas da área informática, admitindo-se o valor de um ensaio interdisciplinar para a solução de novos desafios.

O estudo da API representa mais uma oportunidade para que os juristas possam solidificar estes novos caminhos traçados para o direito autoral, afinal são as hipóteses mais excêntricas que permitem delimitar os contornos de qualquer teoria. O Supremo Tribunal Americano decidiu não oferecer respostas quanto à natureza e proteção das API sob o risco de estabelecer um precedente que poderia não se ajustar às circunstâncias tecnológicas, económicas e mercadológicas de um futuro próximo – este trabalho não carrega a mesma responsabilidade. Foram justamente estas mudanças rápidas na nossa sociedade que justificaram esta investigação.

Apesar da inconsistência normativa e de um cenário jurisprudencial nebuloso, pode afirmar-se com certo grau de confiança que as API devem ser consideradas, na sua integralidade, como uma espécie de programa de computador. Ao introduzir a proteção dos programas de computador pelo direito autoral, tanto a União Europeia quanto os EUA optaram por criar definições amplas, ainda que incertas, quanto ao objeto da tutela. Uma vez que ambas as parcelas da API atuam conjuntamente para permitir o funcionamento da grande maioria dos *softwares* atualmente, não existem pressupostos jurídicos para, *a priori*, negar a tutela do código declarativo.

Também não parece razoável imaginar que doutrinas jurídicas como a *scène à faire* ou a *merger doctrine* possam ser aplicadas de forma automática ao código declarativo. Verificou-se que a tarefa de descrever um elevado número de instruções complexas proporciona um relevante

espaço de expressão criativa dentro do qual um programador pode imprimir a sua personalidade, fugindo do óbvio e banal. Ainda que seja difícil dizer se a utilização deste espaço criativo será a regra ou a exceção no mercado da tecnologia, não se pode pressupor a sua ausência.

Os receios quanto a prejuízos relevantes à inovação e à concorrência causados pela proteção das API mostram-se, no mínimo, desmedidos. Há uma razão clara para a falta de jurisprudência sobre a questão: são incomuns as situações que justificam restrições à utilização de uma API e ainda mais inusitadas as situações em que há interesse substancial em evitar a sua reprodução. A grande maioria das API não são criadas como um produto, mas como uma porta de acesso a um determinado serviço – se esta porta estiver fechada, não será em função da propriedade intelectual sobre esta API, mas por uma escolha empresarial da fornecedora deste serviço. Nada obstante, a Comissão já deixou claro não considerar a existência da proteção de interfaces informáticas pelo direito autoral um obstáculo para a aplicação de normas obstativas de práticas anticoncorrenciais.

Ainda que seja impossível negar o pioneirismo do caso *Google LLC v. Oracle America, Inc.* e a sua relevância como objeto de estudo, não se pode deixar de notar a forma como este evidenciou algumas deficiências do sistema americano. Da mesma forma que o *fair use* é uma valiosa válvula de escape para garantir a inovação, também permite decisões omissas, atrasando o desenvolvimento do conhecimento jurídico. Conforme destacou o juiz Clarence Thomas, as razões pelas quais o Supremo Tribunal Americano evitou analisar a proteção da API são uma constante no ambiente digital⁷⁰, de forma que um momento ideal para a realização desta análise nunca chegará.

Por mais intimidante que seja a rápida evolução tecnológica e os desafios jurídicos que a acompanham, faz parte da obrigação de todo o jurista harmonizar a realidade e o Direito. Felizmente, no caso das API, o desafio parece ser apenas confiar nos mecanismos jurídicos já existentes.

⁷⁰ Thomas, J., Voto Dissidente (2021), ob.cit., p. 7.

Referências Bibliográficas:

Barbosa, D. B. (2001) – *A Proteção do Software*, disponível em: <https://www.dbba.com.br/wp-content/uploads/a-proteo-do-software.pdf>. p. 12, consult. em 15/Abr/2021.

Barbosa, D. B. (2010) – *Uma Introdução à Propriedade Intelectual*, Rio de Janeiro: Editora Lumen Juris.

Barbosa, D. B. (2015) – *Ensaio e estudos de Propriedade Intelectual, Volume III*, Rio de Janeiro: Edição do Instituto Brasileiro da Propriedade Intelectual.

Barbosa, P. M. N. (2017) – *E-stabelecimento*, Rio de Janeiro: Quarter Latin

Bettilyon, T. E. (2018) – *What Is an API and Why Should I Use One?*, 11/Jan/2018. <https://medium.com/@TebbaVonMathenstien/what-is-an-api-and-why-should-i-use-one-863c3365726b>, consult. em 13/Mar/2021.

Bostock, M. (2016) – *What Makes Software Good?*, 10/Mar/2016. <https://medium.com/@mbostock/what-makes-software-good-943557f8a488>, consult. em 10/Jul/2021.

Hawkins, M. (2020) – *The History And Rise Of APIs*, 23/Jun/2020. <https://www.forbes.com/sites/forbestechcouncil/2020/06/23/the-history-and-rise-of-apis/>, consult. em 25/Jan/2021.

Hebl, A. B. (2007-2008) – *A Heavy Burden: Proper Application of Copyright's Merger and Scenes a Faire Doctrines*, Carolina do Norte: Wake Forest Intellectual Property Law Journal.

Hugenholtz, P. B. (2017) – *Flexible Copyright: Can the EU Author's Rights Accommodate Fair Use?*, col. "Copyright Law in an Age of Limitations and Exceptions", Cambridge: Cambridge University Press.

Hurst, A. (2018) – *The Report of Api Copyright's Death is Greatly Exaggerated*. Harvard Journal of Law & Technology, Volume 31, Special Issue Spring 2018.

Mello, A. S. (2020) – *Manual de Direito de Autor e Direitos Conexos*, Coimbra: Edições Almedina S.A.

Mendes, M. O. (2021) – *A obra enquanto objeto do direito de autor*, Revista de Direito Intelectual Nº1 – 2021, Lisboa: Editora Almedina.

Pais, S. O. (2011) – *Entre Inovação e Concorrência: Em Defesa de um Modelo Europeu*, Lisboa: Editora Universidade Católica.

Pereira, A. L. D. (2019) – *Direito da Propriedade Intelectual & Novas Tecnologias, Estudos - Volume I*, Coimbra: Gestlegal.

Rahmatian, A. (2011) – *Copyright and Creativity: The making of Property Rights in Creative Works*, Cheltenham: Editora Edward Elgar.

Rocha, M. V. (2021) – *Decisão Tom Kabinet: Um Forte Retrocesso do Esgotamento Digital na União Europeia?*, III Congreso Internacional Carlos Fernández Nóvoa: Nuevas Tendencias en el Derecho de la Competencia y de la Propiedad Industrial e Intelectual, 28 de setembro, publicação no prelo.

Saavedra, R. (1998) – *A Proteção Jurídica do Software e a Internet*, Lisboa: Editora Sociedade Portuguesa de Autores Publicações Dom quixote.

Santana, G. (2020) – *O que é abstração em programação?*, 05/Mai/2020. <https://medium.com/@gabrielsantana765/o-que-é-abstração-em-programação-bf8ac2fcd96f>, consult. em 20/Fev/2021.

Sarid, U. (2014) – *A Non-Apocalypse: APIs, Copyright, and Fair Use*, s.d., <https://www.wired.com/insights/2014/05/non-apocalypse-apis-copyright-fair-use/>, consult. em 11/Set/2021.

Silva, N. S. (2014) – *Uma Introdução ao Direito de Autor Europeu*, Portal da Ordem dos Advogados, <https://portal.oa.pt/upl/%7Bdca5e510-7e6b-403c-9e2c-fd091a1cb5dc%7D.pdf>, p. 1336, consult. em 18/Ago/2021.

Silva, N. S (2022) - Ligações Perigosas? – Reflexões Sobre Apis e Direito de Autor a Partir do Acórdão Google V. Oracle do Supremo Tribunal dos EUA, Revista de Direito Intelectual – Nº01— 2022: 217-231.

Vieira, J. A. (2005) – *Proteção dos Programas de Computador pelo Direito de Autor*, Dissertação de Doutoramento em Ciências Jurídicas, Lisboa: Editora LEX.

Vieira, J. A. (2020) – *Direito de Autor: Dogmática Básica*, Coimbra: Editora Edições Almedina S.A.

WIPO Pearl – WIPO’s Multilingual Terminology Portal - <https://www.wipo.int/reference/en/wipopearl/>, consult. em 13/Abr/2021.

“Java™ Platform, Standard Edition 7 API Specification”, s.d.. <https://docs.oracle.com/javase/7/docs/api/>, consult. em 25/Jan/2021.

“Lenguajes de programación”. Informática IV. Universidad Nacional Autónoma de México. http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/4/informatica_4.pdf. consult. em 13/Ago/2021.

“Web APIs – Introduction”, W3Schools, https://www.w3schools.com/js/js_api_intro.asp, consult. em 25/Fev/2021.

“What Is API Documentation?”, API Lifecycle Management, 31/Ago/2020. <https://www.akana.com/blog/what-is-api-documentation>, consult. em 25/Jan/2021.

“Interface de programação de aplicações”, Wikipédia, a enciclopédia livre., s.d. https://pt.wikipedia.org/wiki/Interface_de_programação_de_aplicações, consult. em 20/Jan/2021.

“Turing completude”, Wikipédia, a enciclopédia livre. https://pt.wikipedia.org/wiki/Turing_completude, consult. em 13/Mar/2021.

“Unity Documentation”, 02/Jun/2018. <https://docs.unity3d.com/Manual/GraphicsAPIs.html>, consult. em 5/Fev/2021.