

SOBRE LA POSIBILIDAD DE DESCOMPILAR PROGRAMAS DE ORDENADOR PARA CORREGIR ERRORES

[Comentario de la STJUE (Sala Quinta), de 6 de octubre de 2021,
asunto C-13/20, *Top System vs. Estado Belga*]

ABOUT THE POSSIBILITY OF DECOMPILING COMPUTER PROGRAMS TO CORRECT ERRORS

[Comment on the sentence of the CJEU (Fifth Chamber),
of October 6, 2021, case C-13/20, *Top System vs. Belgian State*]

JOSÉ LUIS GONZÁLEZ SAN JUAN*

RESUMEN

La descompilación es un proceso de ingeniería inversa que permite obtener un código cuasi fuente a partir del código objeto de un programa. En los artículos 6 de las Directivas 91/250/CEE (ya derogada) y 2009/24/CE (actualmente vigente) se incluye expresamente la posibilidad de descompilar un programa para garantizar su interoperabilidad, pero no existe una referencia expresa en dichas directivas sobre la posibilidad de realizar la descompilación cuando el fin sea la corrección de los errores del programa que afecten a su funcionamiento. El TJUE considera que los actos necesarios para llevar a cabo la descompilación (reproducción y traducción) estarán autorizados en virtud de los artículos 5, de ambas directivas, cuando sean necesarios para utilizar el programa con arreglo a su finalidad propuesta, incluida la corrección de errores.

Palabras clave: descompilación, programas de ordenador, código fuente, código objeto.

ABSTRACT

Decompilation is a reverse engineering process that allows a quasi-source code to be obtained from the object code of a program. In articles 6 of Directives 91/250/CEE (already repealed) and 2009/24/CE (currently in force) expressly includes the possibility of decompiling a program to guarantee its interoperability, but there is no express reference in said directives about the possibility of carry out decompilation when the purpose is to correct program errors that affect its operation. The CJEU considers that the acts necessary to carry out the decompilation (reproduction and translation) will be authorized under articles 5 of both directives, when they are necessary to use the program in accordance with its intended purpose, including error correction.

Keywords: decompilation, computer programs, source code, object code.

* Doctor en Derecho Privado por la Universidad de Salamanca e Ingeniero de telecomunicación por la Universidad Politécnica de Madrid. Dirección de correo electrónico: gonzalezsanjuan@icasal.com.

SUMARIO: I. INTRODUCCIÓN: ANTECEDENTES Y RESUMEN DE LA SENTENCIA.—1. Origen de la controversia y resolución en primera instancia.—2. Recurso de apelación y cuestión prejudicial.—3. Conclusiones del Abogado General.—4. Respuesta del TJUE a las cuestiones planteadas.—II. CONSIDERACIONES DE ORDEN CONCEPTUAL.—1. Características específicas de los programas de ordenador.—2. Código fuente y código objeto.—3. Concepto de descompilación.—4. La descompilación como actividad limitada por los derechos de autor.—5. Descompilación y obra derivada.—6. Concepto de error y necesidad de descompilar para su corrección.—III. ANÁLISIS DE LA SENTENCIA DEL TJUE.—1. Primera cuestión prejudicial.—2. Segunda cuestión prejudicial.—3. El papel de la autonomía de la voluntad.—IV. RESUMEN Y CONCLUSIONES.—V. BIBLIOGRAFÍA.

CONTENTS: I. INTRODUCTION: BACKGROUND AND SUMMARY OF THE JUDGMENT.—1. Origin of the controversy and resolution in first instance.—2. Appeal and preliminary ruling.—3. Opinion of Advocate General.—4. Response of the CJEU to the preliminary ruling.—II. CONCEPTUAL ORDER CONSIDERATIONS.—1. Specific characteristics of computer programs.—2. Source code and object code.—3. Decompiling concept.—4. Decompilation as an activity limited by copyright.—5. Decompilation and derivative works.—6. Concept of error and need to decompile for its correction.—III. ANALYSIS OF THE CJEU JUDGMENT.—1. The first question.—2. The second question.—3. The role of the autonomy of the will.—IV. SUMMARY AND CONCLUSIONS.—V. BIBLIOGRAPHY.

I. INTRODUCCIÓN: ANTECEDENTES Y RESUMEN DE LA SENTENCIA

La Sentencia de la Sala Quinta del TJUE, de 6 de octubre de 2021, asunto *Top System* (C-13/20), resuelve una cuestión prejudicial elevada por el Tribunal de Apelación de Bruselas (Cour d'Appel de Bruxelles) sobre la interpretación del artículo 5.1 de las Directiva 91/250/CEE¹, en el marco de un litigio entre la mercantil Top System SA² y el Estado belga, en relación con la descompilación llevada a cabo por Selor (Oficina de la Administración Federal de Bélgica³) del programa de ordenador TSF (Top System Frameworks) desarrollado por Top System, que formaba parte de diversas aplicaciones informáticas sobre las que dicha oficina tenía licencias de uso⁴.

1. Origen de la controversia y resolución en primera instancia

La controversia tiene su origen en un recurso interpuesto el 6 de julio de 2009 por Top System contra Selor y el Estado Belga, ante el Tribunal de lo Mercantil de Bruselas (Tribunal de Commerce de Bruxelles), en el que se solicitaba, por una parte, que se declarara que Selor había procedido a la descompilación no autorizada de las aplicaciones informáticas desarrolladas por Top Sytem vulnerando sus derechos exclusivos, y por otra, que se condenara al Estado Belga

¹ Directiva 91/250/CEE del Consejo, de 14 de mayo, sobre la protección jurídica de programas de ordenador. Esta Directiva fue derogada por la Directiva 2009/24/CE del Parlamento Europeo y del Consejo de 23 de abril, sobre la protección jurídica de programas de ordenador, que es la actualmente vigente. La Directiva 91/250/CEE es aplicable, *ratione temporis*, a los hechos objeto de la controversia, aunque como el tenor de los artículos 5.1 y 6 de ambas Directivas es esencialmente idéntico, la sentencia resulta plenamente aplicable también para la Directiva 2009/24/CE.

² Sociedad Belga dedicada al desarrollo de programas de ordenador y a la prestación de servicios informáticos.

³ Selor es un organismo público belga encargado de la selección y orientación de los futuros colaboradores de los servicios públicos de la Administración. A raíz de la integración de Selor en el servicio federal «Estrategia y Apoyo», el Estado belga sustituyó a este como parte demandada en el litigio principal. *Vid.* apartado 14 de la STJUE.

⁴ Entre ellas, las aplicaciones «SWA» (Selor Web Access) y «eRecruiting» desarrolladas por Selor, que incorporaban funcionalidades procedentes del programa TSF y también otras funcionalidades creadas *ad hoc* para cubrir las necesidades específicas de Selor. *Vid.* apartado 17 de las conclusiones del Abogado General.

al pago de una indemnización de daños y perjuicios por haber realizado dicha descompilación, así como por la copia no autorizada de los códigos fuente del *software*, que también se había producido.

El 26 de noviembre de 2009 el asunto fue remitido al Tribunal de Primera Instancia de Bruselas (Tribunal de Première Instance de Bruxelles), que desestimó en lo esencial la demanda de Top System, en Sentencia de 19 de marzo de 2013.

2. Recurso de apelación y cuestión prejudicial

La sentencia de primera instancia fue recurrida en apelación por la actora ante el Tribunal de Apelación de Bruselas (Cour d'Appel de Bruxelles).

Top System mantenía que Selor había procedido a descompilar ilícitamente su programa TSF, puesto que consideraba que la normativa interna aplicable solo permitía la descompilación cuando existiera autorización del titular de los derechos o se realizara con fines de interoperabilidad, quedando por tanto excluida la descompilación realizada a los efectos de corregir errores del programa⁵.

Por otra parte, Selor mantenía que tanto los términos del contrato (algo que fue rechazado por el Tribunal), como la normativa interna belga, le facultaban para realizar la descompilación cuando el fin era corregir errores del programa que hacían imposible su utilización con arreglo a su finalidad⁶.

En este momento procesal, el Tribunal de Apelación decidió suspender el procedimiento y plantear las siguientes cuestiones prejudiciales al TJUE:

«1) ¿Debe interpretarse el artículo 5, apartado 1, de la [Directiva 91/250] en el sentido de que permite al adquirente legítimo de un programa de ordenador descompilar total o parcialmente dicho programa cuando tal descompilación sea necesaria para permitirle corregir errores que afectan a su funcionamiento, incluido cuando la corrección consiste en desactivar una función que afecta al buen funcionamiento de la aplicación de la que forma parte ese programa?

2) En caso de respuesta afirmativa, ¿deben cumplirse además los requisitos previstos en el artículo 6 de la Directiva u otros requisitos?».

3. Conclusiones del Abogado General

El Abogado General⁷, Sr. Maciej Szpunar, considera que para llevar a cabo la descompilación se requiere la realización de actos de reproducción y de transformación, que quedan dentro del ámbito de los derechos exclusivos del autor previstos en el artículo 4, letras *a*) y *b*) de la Directiva 91/250/CEE, y que por ello, y salvo que exista una reserva en el contrato⁸, el artículo 5.1 de la Direc-

⁵ En concreto, resultan aplicables los artículos 5, 6 y 7 de la Ley belga de 30 de junio de 1994, por la que se adaptó el Derecho belga a la Directiva 91/250/CEE.

⁶ Selor reconoció haber llevado a cabo la descompilación de una parte del programa TSF para desactivar una función que provocaba errores. Y entendemos que esto implica, necesariamente, que además habría realizado una compilación posterior del cuasi código fuente, una vez modificado para eliminar el error, con el fin de crear una versión ejecutable del programa TSF con la función problemática desactivada.

⁷ Las conclusiones del Abogado General fueron publicadas el 10 de marzo de 2021.

⁸ No obstante, en base al tenor del considerando 17 de la Directiva 91/250/CEE, que es esencialmente idéntico al considerando 13 de la Directiva 2009/24/CE actualmente vigente, el TJUE defiende la imposibilidad

tiva 91/250/CEE debe interpretarse en el sentido de que permite al adquirente legítimo de un programa de ordenador descompilarlo, cuando ello sea preciso para corregir errores que afecten a su funcionamiento⁹.

Y en cuanto a la segunda cuestión prejudicial, considera el Sr. Szpunar que los requisitos del artículo 6 de la Directiva 91/250/CEE «no pueden aplicarse, ni directamente ni por analogía, a la excepción prevista en el artículo 5, apartado 1, de dicha Directiva», pero ello no significa que la descompilación realizada para corregir errores no esté sujeta a ningún requisito¹⁰, puesto que solamente podrá llevarla a cabo el adquirente legítimo y debe ser necesaria para la corrección de errores, siendo por ello preciso delimitar correctamente los conceptos de error y necesidad¹¹.

4. Respuesta del TJUE a las cuestiones planteadas

El TJUE, siguiendo en lo esencial las tesis del Abogado General, respondió a la primera cuestión planteada indicando que el artículo 5.1 de la Directiva 91/250/CEE «debe interpretarse en el sentido de que el adquirente legítimo de un programa de ordenador tiene derecho a descompilar total o parcialmente dicho programa con el fin de corregir errores que afecten a su funcionamiento, incluso cuando la corrección consiste en desactivar una función que afecta al buen funcionamiento de la aplicación de la que forma parte ese programa».

Y respecto a la segunda cuestión prejudicial, considera el TJUE que el adquirente legítimo «no está obligado a cumplir los requisitos previstos en el artículo 6 de dicha Directiva. Sin embargo, ese adquirente solo tiene derecho a proceder a tal descompilación en la medida necesaria para dicha corrección y respetando, en su caso, las condiciones establecidas contractualmente con el titular de los derechos de autor sobre ese programa».

II. CONSIDERACIONES DE ORDEN CONCEPTUAL

A continuación, para centrar el problema y antes de entrar a analizar los detalles de la sentencia del TJUE, comentaremos ciertas características específicas de los programas de ordenador, e intentaremos definir algunos conceptos fundamentales para entender esta controversia.

1. Características específicas de los programas de ordenador

Como bien nos recuerda el Abogado General en la introducción de sus conclusiones, los programas de ordenador están protegidos por derechos de autor como obras literarias, siempre que sean una creación humana original¹², si bien

de prohibir en el contrato completamente dicha descompilación, pero el Abogado General considera que tal interpretación no es posible, como veremos con más detalle en el epígrafe III.3.

⁹ Vid. apartados 49 y 50 de las conclusiones del Abogado General.

¹⁰ Vid. apartados 68 y 69 de las conclusiones del Abogado General.

¹¹ Conceptos que intentaremos delimitar en el epígrafe II.6.

¹² Artículos 1 y 3 de la Directiva 91/250/CEE (ya derogada), y artículos 1 y 3 de la Directiva 2009/24/CE (actualmente vigente), cuyo tenor es exactamente idéntico en ambas normas y que se corresponden esencialmente con el contenido del artículo 96 en relación con el 10.1.i de la LPI española —hubiera sido más correcto

tienen ciertas peculiaridades que hacen que algunos autores hablen de la existencia, *de facto*, de un régimen de protección *sui generis*¹³.

Además, los programas de ordenador son el único tipo de obras protegidas por los derechos de autor que no están destinadas a ser utilizadas directamente a través de la percepción humana, pues es el ordenador el que adquiere conocimiento de ellos¹⁴, y esto hace que su expresión sea tal que resulte comprensible para el ordenador (lenguaje máquina¹⁵), pero que no es directamente cognoscible para un ser humano, aunque sea un programador experto, a menos que se trate de un programa extremadamente sencillo.

Por último, destacamos una característica peculiar de los programas de ordenador, y es que para su uso o disfrute es preciso realizar una reproducción de los mismos, algo que con carácter general no ocurre con otras obras protegidas por los derechos de autor¹⁶.

incluirlo en el artículo 10.1.a), pero no se hizo posiblemente por un error de técnica legislativa. *Vid.* APARICIO VAQUERO (2017A), pág. 195—. Por tanto, el único requisito para que un programa de ordenador quede protegido por los derechos de autor es que sea «una creación propia e individual de su autor», esto es, que no sea copia de otro, sin necesidad de que exista un mínimo de altura creativa. *Vid.* SAIZ GARCÍA (2000), págs. 115, 116 y 118. Y, como bien indica el profesor Fernando Carbajo, «Se trata de un criterio de originalidad subjetiva débil (es original la creación no copiada de otra anterior de un autor diferente)», *vid.* CARBAJO CASCÓN (2018), pág. 45, criterio que si bien en principio solo estaría armonizado en el UE para los programas de ordenador, las bases de datos y las fotografías, en la STJUE de 16 de julio de 2009, asunto *Infopaq* (C-5/08), se defiende que resultaría aplicable a todo tipo de obras, aunque no se trata de un tema pacífico. *Vid.* GONZÁLEZ SAN JUAN (2020B), pág. 424. Para Juan Pablo Aparicio, «el legislador europeo opta por un requisito de requisito de naturaleza subjetiva en su forma más estricta o absoluta, descartando la concurrencia de cualquier otro elemento de valoración técnico o cualitativo». *Vid.* APARICIO VAQUERO (2017B), pág. 1405. Finalmente, es preciso que, al igual que ocurre con el resto de obras protegidas por los derechos de autor, se trate de una creación humana, no quedando protegidos aquellos programas que hayan sido creados de forma automática sin intervención humana o cuando dicha intervención se limite a la mera selección de los parámetros o funciones de un *software* generador de programas. *Vid.* APARICIO VAQUERO (2017B), pág. 1410.

¹³ *Vid.* CARBAJO CASCÓN (2002), pág. 56 y el apartado 1 de las conclusiones del Abogado General. De hecho, existen importantes peculiaridades, por ejemplo, en relación con la originalidad, la titularidad de los derechos, las limitaciones previstas sobre los derechos morales o la eliminación del límite de copia privada. *Vid.* FERNÁNDEZ MASIÁ (2017), pág. 1196. En la norma española es especialmente conflictiva la consideración de «autor» que el art. 97.2 LPI otorga a las personas jurídicas, más propia del ámbito anglosajón y ajena a nuestro ordenamiento jurídico. *Vid.* APARICIO VAQUERO (2017B), págs. 1420-1428. También es muy cuestionable el mantenimiento del plazo general de duración de los derechos de exclusividad, plazo considerado excesivo de forma prácticamente unánime por la doctrina, opinión con la que coincidimos, pues la mayor parte del *software* tiene una vida comercial útil inferior a los diez años. *Vid.* LÓPEZ-TARRUELLA MARTÍNEZ (2017), págs. 1219-1220. Otro elemento específico de los programas de ordenador es la copia de seguridad. *Vid.* APARICIO VAQUERO (2016), pág. 33.

¹⁴ Afirma el Abogado General que la utilidad de un programa reside esencialmente en su funcionalidad, y que por ello se asemejan más a las invenciones protegidas por patentes que a las obras clásicas cubiertas por los derechos de autor. *Vid.* apartado 3 de las conclusiones del Abogado General. De hecho, la protección mediante patentes fue una de las opciones planteadas inicialmente para la protección de los programas de ordenador, aunque sería descartada al no ajustarse las patentes plenamente a las características del *software* (*v. gr.*, las ideas que subyacen en un programa no siempre son nuevas), además de que existían también poderosas razones económicas, burocráticas y políticas. *Vid.* FERNÁNDEZ MASIÁ (2017), págs. 1188-1190, y APARICIO VAQUERO (2017B), págs. 1362-1369.

¹⁵ El lenguaje máquina o código máquina está formado por un conjunto de instrucciones que son directamente ejecutables por el microprocesador del ordenador, representadas directamente en código binario. El código binario es un medio de expresión que utiliza únicamente dos dígitos para codificar la información, el «1» y el «0», dígitos que realmente son representaciones en lenguaje humano de los estados lógicos «activo» e «inactivo», respectivamente, de un circuito electrónico. No obstante, debemos tener presente que cualquier información que maneje un ordenador debe estar, en última instancia, codificada en código binario (por ejemplo, un fichero Word estará también codificado en código binario).

¹⁶ *Vid.* APARICIO VAQUERO (2017B), pág. 1464.

2. Código fuente y código objeto

Pocas veces se utiliza directamente el lenguaje máquina¹⁷ para crear los programas de ordenador, salvo que se trate de programas extremadamente sencillos. De hecho, la inmensa mayoría de los programas actuales se desarrollan utilizando lenguajes de programación de alto nivel, que son aquellos que permiten expresar los algoritmos, las operaciones y las variables de una manera adecuada a la capacidad cognitiva humana.

El código fuente de un programa¹⁸, que puede estar formado por un solo fichero informático o por un conjunto de ficheros (lo más habitual en los programas complejos), es el conjunto de instrucciones y otros elementos que lo definen, escritos en un lenguaje de programación de alto nivel¹⁹.

Pero como el ordenador no es capaz de interpretar directamente el código fuente, es necesario «traducirlo» para que pueda ser ejecutado por el microprocesador de la máquina. Este proceso de traducción se denomina compilación²⁰, y el resultado de la misma es un fichero o conjunto de ficheros, denominado código objeto, comprensible y ejecutable directamente por el ordenador.

Es importante tener presente que la protección del programa de ordenador que otorgan los derechos de autor comprende todas las formas de expresión del mismo (art. 96.3 LPI), junto con la documentación preparatoria, la documentación técnica y los manuales de uso, de forma que tanto el código fuente como el código objeto estarán igualmente protegidos. No obstante, no se protegen las ideas o principios que en él subyacen, ni tampoco su funcionalidad²¹.

¹⁷ O el lenguaje ensamblador, que es un lenguaje de programación de bajo nivel, esencialmente equivalente al lenguaje máquina, ya que está formado por un conjunto de nemónicos (palabras o conjuntos de letras que sustituyen los códigos binarios de las instrucciones) que representan instrucciones directamente ejecutables por el microprocesador. Por ejemplo, si en el lenguaje máquina de un determinado microprocesador la instrucción para mover el contenido de una posición de memoria a otra es «10111001 1111010», en lenguaje ensamblador la podemos representar mediante el nemónico «MOV», expresión mucho más fácil de recordar y procesar para el ser humano que su equivalente binario.

¹⁸ Debemos aclarar que el código fuente no es exclusivo de los programas de ordenador, puesto que existen otros tipos de *software* que no son programas y también tienen código fuente, por ejemplo, las páginas Web o los ficheros de un editor de texto (v. gr., Word), o de un editor de planos (p. ej. Autocad). Vid. GONZÁLEZ SAN JUAN (2017), pág. 94, nota 13 (pág. 3, nota 13 versión *online*).

¹⁹ El TJUE y una parte relevante de la doctrina consideran que un lenguaje informático (y en particular un lenguaje de programación) puede ser a su vez objeto de propiedad intelectual, opinión con la que coincidimos. Vid. APARICIO VAQUERO (2017B), págs. 1394-1395 y STJUE de 2 de mayo de 2012, asunto C-406/10, *SAS Institute vs. World Programming Ltd.* Es conveniente recordar que el concepto de lenguaje informático es más amplio que el de lenguaje de programación, esto es, existen lenguajes informáticos que no son lenguajes de programación, como por ejemplo el HTML (lenguaje con el que se escriben las páginas Web). Vid. GONZÁLEZ SAN JUAN (2020A), pág. 40.

²⁰ La compilación da lugar a una versión ejecutable del programa (código objeto), pero también existen lenguajes interpretados, en los que no se realiza una traducción previa para generar ficheros ejecutables, sino que la traducción se va haciendo línea a línea a partir del código fuente, en tiempo real mientras se ejecuta el programa. Por ello, no todos los programas de ordenador tienen código objeto. Por otra parte, cuando el programa ha sido escrito en lenguaje ensamblador el proceso de compilación se denomina ensamblado (este proceso comprende la traducción de los nemónicos al código binario y otra serie de operaciones).

²¹ Vid. APARICIO VAQUERO (2017B), págs. 1398-1404.

3. Concepto de descompilación

Cuando tenemos acceso al código fuente de un programa es relativamente sencillo conocer las ideas que subyacen en él²², pero lo habitual es disponer únicamente de la versión compilada del mismo (el código objeto), de forma que en este caso el conocimiento de la arquitectura interna del programa y de las ideas que en él subyacen queda tremendamente limitado²³.

En estos casos, cabe la posibilidad de realizar un proceso de ingeniería inversa, denominado descompilación, para obtener una versión fuente a partir del código objeto del programa, versión que se denomina cuasi código fuente²⁴.

El cuasi código fuente, aunque será diferente (o incluso muy diferente) del código fuente original²⁵, permite conocer la estructura del programa y las ideas que subyacen en el mismo, de forma que es posible aplicarlas o utilizarlas en otros programas diferentes.

En todo caso, debemos tener presente, como bien nos recuerda el Abogado General, que la descompilación es un proceso laborioso, costoso y muchas veces con resultados impredecibles (pues no siempre se obtiene una versión del código fuente adecuada para los fines buscados), de forma que los usuarios solo recurrirán a esta técnica como último recurso²⁶.

²² Especialmente los algoritmos utilizados y la arquitectura del programa, así como las estructuras de datos implantadas, y siempre, claro está, para un programador experto en el concreto lenguaje de programación en el que está escrito el código fuente.

²³ Pues como indica el Abogado General, los programas de ordenador «constituyen la única categoría de obras protegidas que no permite acceder a las ideas subyacentes mediante un mero análisis sensorial que no implique la realización de actos sujetos al monopolio del autor». *Vid.* apartado 7 de las conclusiones del Abogado General.

²⁴ Como bien afirma el profesor Juan Pablo Aparicio, con quien coincidimos: «En todo caso, obsérvese que, aun cuando las ideas y principios no quedan tutelados, sí se encuentran protegidos por un obstáculo físico que recibe cierto reconocimiento jurídico: en muchas ocasiones, cuando el programa se encuentra suministrado (distribuido o comunicado públicamente) solo mediante su código objeto (programas ejecutables) llegar a aquellos puede ser muy complicado, al no poder acceder a nivel comprensible a su código fuente sin realizar determinadas y muy complicadas operaciones técnicas, que reciben el nombre de “ingeniería inversa” o “descompilación”». *Vid.* APARICIO VAQUERO (2017B), pág. 1401.

²⁵ Esto también ocurriría en cierta medida si, partiendo de una traducción de una obra literaria, realizáramos una traducción retrógrada al idioma original, pues nunca obtendríamos un resultado completamente idéntico a la obra inicial. Pues bien, en el caso de los programas de ordenador descompilados esta disparidad suele ser muchísimo mayor, por múltiples razones: en primer lugar, un mismo código fuente puede generar diferentes códigos objeto, dependiendo de cómo se realice la compilación, esto es, la compilación no es un proceso unívoco, al igual que tampoco lo es la descompilación; en segundo lugar, al compilar desaparece cierta información del código fuente que no es necesaria para la ejecución del programa, pero que puede resultar fundamental para entender su arquitectura (*v. gr.*, los nombres de las variables o los comentarios incluidos por el programador), y por ello no es posible recuperarla al descompilar; en tercer lugar, el compilador realiza algunas optimizaciones del código, como por ejemplo la corrección de redundancias incluidas durante la programación (debidas a la metodología utilizada o las propias limitaciones humanas), redundancias que al haberse eliminado tampoco podrán ser recuperadas descompilando; en cuarto lugar, el proceso de descompilación no siempre es completamente automático, siendo necesario en ocasiones realizar actuaciones manuales o tomar ciertas decisiones que influyen en el resultado final de la descompilación; por último, en ocasiones el lenguaje de descompilación (esto es, el lenguaje de programación en el que está escrito el cuasi código fuente obtenido tras la descompilación) no es el mismo que aquel con el que se creó el programa, ya que para descompilar suele elegirse un lenguaje de programación de alto nivel lo más simple y estructurado posible, para facilitar el proceso de ingeniería inversa, siendo C uno de los lenguajes más utilizados.

²⁶ *Vid.* apartado 83 de las conclusiones del Abogado General. Además, es preciso disponer de *software* especializado (descompilador) y tener ciertos conocimientos técnicos, algo que no está normalmente al alcance de la mayor parte de los usuarios.

El cuasi código fuente obtenido tras la descompilación puede ser, a su vez, compilado nuevamente para generar una nueva versión ejecutable del programa inicial (un nuevo código objeto)²⁷.

4. La descompilación como actividad limitada por los derechos de autor

Descompilar un programa es una transformación del mismo que implica, como mínimo, llevar a cabo una reproducción y una traducción, actos ambos que obviamente requieren autorización del titular de los derechos²⁸, de forma que solo será posible realizarla cuando exista dicho consentimiento o esté autorizada *ex lege*.

Por ello, no cabe descompilar libremente cualquier programa, aunque tengamos una licencia de uso del mismo, cuando el objetivo de la descompilación sea simplemente conocer las ideas o los principios subyacentes²⁹, y ello a pesar de que como sabemos las ideas no quedan protegidas por los derechos de autor³⁰.

No obstante, los artículos 6 de la Directiva 91/250/CEE (ya derogada) y de la 2009/24/CE (actualmente vigente) autorizan la posibilidad de descompilar un programa cuando el objetivo sea garantizar su interoperabilidad, exigiendo en ese caso que se cumplan una serie de requisitos, pero las citadas directivas no hacen una referencia expresa a si la descompilación pueda realizarse o no para corregir errores de funcionamiento del programa (situación planteada en la controversia que originó la cuestión prejudicial que estamos analizando).

5. Descompilación y obra derivada

Dado que la descompilación es esencialmente un acto de transformación que produce una versión diferente, o en ocasiones muy diferente, del programa original (el cuasi código fuente), cabe plantearse la posibilidad de que exista una obra derivada.

Si partimos del hecho de que para descompilar es preciso realizar una traducción del programa (y a veces a un lenguaje de programación diferente de aquel en que fue originalmente escrito)³¹, parece lógico afirmar que, cuando

²⁷ Y esta nueva compilación puede realizarse bien después de realizar cambios en el código fuente (por ejemplo, para corregir errores), o también sin cambiar nada.

²⁸ Vid. apartados 38 y 39 de la STJUE.

²⁹ Vid. LÓPEZ-TARRUELLA MARTÍNEZ (2017), pág. 1253.

³⁰ La protección de los derechos de autor solo alcanza a las expresiones concretas de las ideas, esto es, la forma con la que estas se expresan. En el caso de los programas de ordenador, así lo indican expresamente el artículo 2 de la Directiva 91/250/CEE (ya derogada), y también el art. 2 de la Directiva 2009/24/CE (actualmente vigente), cuyo tenor es exactamente idéntico. En este sentido, *vid.* APARICIO VAQUERO (2017B), pág. 1398.

³¹ Coincidimos con el Abogado General (*vid.* apartado 45 de las conclusiones del Abogado General) y con el TJUE (*vid.* apartados 38 y 39 de la STJUE), que afirman que la descompilación es una traducción del programa, aunque existen posiciones doctrinales no totalmente coincidentes, como la de Aurelio López-Tarruella que mantiene que la descompilación no es una propiamente traducción: «La descompilación, operación que supone bastante gasto en tiempo y dinero, no conlleva la creación de una obra derivada, ni siquiera supone propiamente la traducción del *software* original — pues sigue siendo el mismo *software* —». Vid. LÓPEZ-TARRUELLA MARTÍNEZ (2017), págs. 1255-1256. También Juan Pablo Aparicio afirma, respecto de las operaciones de compilación y descompilación, que «siendo transformación en sentido amplio, no es propiamente un acto de “traducción”, como si podría serlo su escritura en un lenguaje diferente». Vid. APARICIO VAQUERO (2017B), pág. 1469. Pero discrepamos con la tesis de que la compilación o la descompilación no sean auténticas traducciones pues el código fuente y el código objeto están escritos en lenguajes diferentes (el primero en un lenguaje de programación

exista una intervención humana original, el resultado debería ser una obra derivada.

No obstante, entendemos que esto no ocurrirá práctica nunca, en primer lugar porque el proceso de descompilación es esencialmente automático, siendo por ello la intervención humana inexistente en la mayor parte de los casos o limitada a elegir algunos parámetros en el programa de descompilación y a ejecutarlo, y en segundo lugar porque cuando la intervención humana sea relevante, será difícil que se cumpla el requisito de originalidad, aunque no puede descartarse completamente.

Por todo ello, aunque cabe la posibilidad de que el proceso de descompilación produzca una obra derivada, entendemos que esto normalmente no va a ocurrir.

6. Concepto de error y necesidad de descompilar para su corrección

A la hora de analizar la problemática objeto de la sentencia, es esencial definir correctamente el concepto de error, y también en qué medida es necesario realizar una descompilación del programa para su corrección³².

En primer lugar, podemos definir error informático como todo funcionamiento incorrecto que impida utilizar el programa con arreglo a la finalidad definida por su autor o acordada por el proveedor y el adquirente³³.

Se trata de una concepción amplia del concepto de error, pero que no incluye la obsolescencia técnica del programa, de forma que no será posible descompilar un programa cuando el objetivo sea únicamente corregir su obsolescencia (v. gr., hacer que funcione en un Sistema Operativo más moderno que aquel para el que fue diseñado, o para que pueda ejecutarse en un *hardware* de nueva generación).

Además, entendemos que tampoco cabría realizar la descompilación cuando se trate de un error que no afecte a la funcionalidad del programa, v. gr., un mero error tipográfico u ortográfico en alguno de los menús o pantallas del programa³⁴.

Y en cuanto a la necesidad de realizar la descompilación para corregir errores, debemos partir del hecho de que la corrección de los errores debe hacer-

de alto nivel y el segundo en lenguaje máquina), y el hecho de que la transformación de uno a otro sea más o menos automática (especialmente en el caso de la compilación), no desvirtúa que sea una auténtica traducción.

³² De hecho, como bien pone de manifiesto el Abogado General, en muchas ocasiones la misma existencia del error es un punto de conflicto entre el autor y el usuario del programa: «En efecto, la existencia misma de un error en un programa de ordenador puede constituir un punto de discordia entre el autor y el usuario de dicho programa. Lo que desde el punto de vista de este último puede constituir un error, para el autor del programa puede ser una funcionalidad o una característica deliberadamente incluida». *Vid.* apartado 73 de las conclusiones del Abogado General. De hecho, esto es algo que ocurrió en el asunto objeto de la controversia, puesto que Top System negaba la existencia del error a pesar de los informes periciales en sentido contrario, como se pone de manifiesto en la nota 30 de las conclusiones del Abogado General.

³³ Coincidimos plenamente con el Abogado General, que tras aclarar que: «La finalidad de un programa de ordenador en la definida por su autor o, en su caso, la pactada por el proveedor y el adquirente del programa en el momento de su adquisición», define error informático como «un funcionamiento defectuoso que impide la utilización del programa con arreglo a dicha finalidad». *Vid.* apartado 75 de las conclusiones del Abogado General.

³⁴ Así, si por ejemplo, en el menú del programa aparecieran las palabras «Generar resultados» en lugar de «Generar resultados» sin que ello afectara a su funcionalidad, entendemos que este mero error ortográfico no justificaría la necesidad de realizar una descompilación del programa, de forma que esta estaría prohibida.

se, salvo muy raras excepciones, modificando el código fuente (o cuasi código fuente), puesto que realizarla directamente en el código objeto suele ser absolutamente inviable.

Además, si bien es cierto que normalmente la corrección del error implica modificar una sola línea (o como mucho unas pocas líneas) del código fuente, también lo es que para localizar la línea o líneas a cambiar es preciso conocer con bastante detalle las ideas que subyacen en el programa y su arquitectura, algo para lo que normalmente es necesario realizar la descompilación de una parte relevante o incluso de la totalidad del programa (salvo que se tenga acceso al código fuente original).

Por ello, la corrección de errores hace que resulte necesario, cuando no se disponga del código fuente, realizar una descompilación de la totalidad o de una parte sustancial del programa.

III. ANÁLISIS DE LA SENTENCIA DEL TJUE

A continuación, vamos a resumir el razonamiento que lleva a cabo el Alto Tribunal para responder a las dos cuestiones planteadas, que coincide en lo esencial con el expuesto por el Abogado General en sus conclusiones.

1. Primera cuestión prejudicial

Como indica el TJUE, en la primera cuestión prejudicial se pregunta, en esencia, si «el adquirente legítimo de un programa de ordenador tiene derecho a descompilar total o parcialmente dicho programa con el fin de corregir errores que afecten a su funcionamiento, incluso cuando la corrección consiste en desactivar una función que afecta al buen funcionamiento de la aplicación de la que forma parte ese programa»³⁵, esto es, si la finalidad de corregir los errores existentes en un programa justifica su descompilación para poder acceder a las ideas que subyacen en el mismo.

El TJUE nos recuerda que el titular de los derechos de autor sobre un programa de ordenador tiene, entre otros, los derechos exclusivos de reproducción y de transformación sobre el mismo, si bien el adquirente legítimo puede realizar dichos actos cuando sean necesarios para utilizar el programa con arreglo a su finalidad propuesta, incluida la corrección de errores, y salvo que existan disposiciones contractuales específicas (art. 5.1 Directiva 91/250/CEE).

Adicionalmente, el adquirente legítimo también puede realizar actos de reproducción y de transformación para descompilarlo, cuando resulte indispensable para obtener información con el fin de hacer posible la interoperabilidad del programa, y siempre que se cumplan ciertos requisitos (art. 6.1 Directiva 91/250/CEE)³⁶.

No obstante, el TJUE pone de manifiesto que, si bien la descompilación no se encuentra incluida entre los actos enumerados en el artículo 4, letras *a*) y *b*)

³⁵ *Vid.* apartado 28 de la STJUE.

³⁶ El TJUE rechaza, coincidiendo con la opinión del Abogado General, la tesis mantenida por Top System de que el artículo 6 de la Directiva 91/205/CEE sería una especie de *lex specialis* y por ello, la única disposición relativa a la descompilación en dicha directiva. *Vid.* apartados 40 de la STJUE y 52 de las conclusiones del Abogado General.

de la Directiva 91/250/CEE, constituye una transformación del código de un programa que lleva aparejada una reproducción (al menos parcial y provisional) y una traducción de la forma de este³⁷, que estarán autorizadas en virtud del artículo 5.1 Directiva 91/250/CEE cuando sean necesarias para utilizar el programa con arreglo a su finalidad propuesta, incluida la corrección de errores, y no existan disposiciones contractuales que lo impidan.

Por ello, el TJUE concluye que el usuario legítimo de un programa de ordenador puede descompilarlo con el fin de corregir errores que afecten a su funcionamiento, salvo pacto en sentido contrario.

2. Segunda cuestión prejudicial

En la segunda cuestión prejudicial se pregunta, en esencia, «si el artículo 5, apartado 1, de la Directiva 91/250 debe interpretarse en el sentido de que el adquirente legítimo de un programa de ordenador que desee descompilar ese programa con el fin de corregir los errores que afecten a su funcionamiento debe cumplir los requisitos establecidos en el artículo 6 de dicha Directiva u otros requisitos»³⁸.

El TJUE comienza indicando que al ser el ámbito de aplicación y las finalidades del artículo 6 de la Directiva 91/250/CEE distintos de los del artículo 5.1 de dicha Directiva, los requisitos establecidos en el artículo 6 no resultan aplicables a la excepción prevista en el artículo 5.1, pero esto no significa que dicha excepción no esté sujeta a ningún requisito.

En primer lugar, los actos deben ser necesarios para que el adquirente legítimo pueda utilizar el programa con arreglo a su finalidad propuesta, incluida la corrección de errores, entendiendo error como un defecto que afecta al programa y que origina un fallo de este³⁹.

En segundo lugar, la descompilación debe ser necesaria para corregir el error. A este respecto señala el TJUE que la corrección de errores implicará en la mayoría de los casos una modificación del código del programa, de forma que será necesario el acceso al código fuente del mismo. Por ello, cuando el código fuente esté legal o contractualmente accesible, no podrá considerarse que la descompilación es necesaria para corregir el error⁴⁰.

En tercer lugar, el artículo 5.1 de la Directiva 91/250/CEE permite la corrección de errores sin perjuicio de las disposiciones contractuales específicas, si bien no es posible excluir contractualmente toda posibilidad de subsanar los errores del programa⁴¹. No obstante, el titular y el adquirente tienen libertad para regular contractualmente el ejercicio de esta facultad (*v. gr.*, limitar o impedir la descompilación, garantizando el mantenimiento correctivo u ofreciendo acceso al código fuente).

³⁷ Como indica el TJUE, esta interpretación se ve corroborada por el tenor del artículo 6.1 Directiva 91/250/CEE. *Vid.* apartado 40 de la STJUE.

³⁸ *Vid.* apartado 54 de la STJUE.

³⁹ Como afirma el TJUE, al no incluir la Directiva 91/250/CEE una definición de «error», ni existir una referencia al Derecho de los Estados miembros, el concepto de error debe interpretarse en el sentido habitual en el lenguaje corriente, teniendo en cuenta el contexto en el que se utiliza. *Vid.* apartado 58 de la STJUE.

⁴⁰ *Vid.* apartado 63 de la STJUE.

⁴¹ *Vid.* apartado 66 de la STJUE. En este punto el TJUE se aparta de la posición del Abogado General, como veremos con más detalle en el epígrafe siguiente.

En cuarto lugar, el adquirente legítimo del ordenador que haya procedido a descompilar el programa para corregir los errores no podrá utilizar el resultado de dicha descompilación para fines diferentes de la corrección de errores.

Por ello, el TJUE concluye que el adquirente legítimo que descompile un programa para corregir errores no está obligado a cumplir los requisitos del artículo 6 de la Directiva 91/250/CEE, aunque solo podrá realizar la descompilación cuando esta sea necesaria para corregir los errores, deberá respetar la condiciones contractuales establecidas y no podrá utilizar el resultado de la descompilación para fines distintos a la corrección de los errores.

3. El papel de la autonomía de la voluntad

Como ya hemos indicado, en esta sentencia el TJUE sigue esencialmente las tesis del Abogado General en la práctica totalidad de los argumentos, si bien existe un punto en el que el Alto Tribunal discrepa con el Abogado General, y es en el papel que juegan las disposiciones contractuales en relación con la descompilación realizada para corregir los errores del programa.

Para analizar este punto debemos partir del considerando 17 de la Directiva 91/250/CEE, que es idéntico al considerando 13 de la Directiva 2009/24/CE actualmente vigente, y en el que se afirma lo siguiente: «los actos de carga y de desarrollo necesarios a la utilización de una copia de un programa legalmente adquirido, y el acto de corrección de sus errores, no pueden ser prohibidos por contrato».

El TJUE considera que esta afirmación es plenamente aplicable, y por ello, el titular de los derechos no puede excluir totalmente toda posibilidad de realizar los actos necesarios para corregir los errores (entre los que se encontraría la descompilación cuando no se tenga acceso al código fuente). No obstante, el titular podrá prohibir la descompilación cuando permita que el adquirente legítimo acceda al código fuente o le proporcione un mantenimiento correctivo para solventar tales errores⁴².

Sin embargo, el Abogado General mantiene una posición diferente, pues entiende que dicho considerando no debe ser tenido en cuenta en relación con la descompilación llevada a cabo para corregir errores, ya que los considerandos carecen de efectos normativos y la parte dispositiva de la Directiva conduce a la conclusión contraria, esto es, admite que la posibilidad de realizar la descompilación con la finalidad de corregir errores podría ser totalmente prohibida contractualmente⁴³.

En nuestra opinión, resulta más adecuada la posición del TJUE, con la que coincidimos, puesto que establecer convencionalmente una prohibición total de llevar a cabo la descompilación con el fin de corregir los errores podría impedir al usuario legítimo la utilización del programa de ordenador de acuerdo a su finalidad, algo que puede limitar enormemente la utilidad del programa o incluso hacerlo completamente inoperativo.

⁴² No obstante, en el contrato del asunto objeto de la sentencia que estamos comentando no existía ninguna reserva en este sentido, de forma que la posibilidad de realizar la descompilación del programa para corregir errores no estaba limitada convencionalmente.

⁴³ *Vid.* apartado 31 de las conclusiones del Abogado General.

Es por ello esencial que las eventuales limitaciones contractuales relativas a la descompilación, cuando sea llevada a cabo para corregir errores, vayan acompañadas de medidas adicionales que permitan al usuario legítimo la utilización del programa de acuerdo a su finalidad (*v. gr.*, permitirle el acceso al código fuente cuando fuera necesario o bien que el titular de los derechos se comprometa contractualmente a corregir los errores del programa que afecten a su finalidad).

IV. RESUMEN Y CONCLUSIONES

Los programas de ordenador están protegidos como obras literarias, pero tienen ciertas peculiaridades que los hacen únicos en el ámbito de los derechos de autor. Probablemente, la más relevante sea que son las únicas obras protegidas que no permiten que el usuario acceda directamente a las ideas que subyacen en ellas, pues normalmente este no va a tener acceso al código fuente.

Descompilar (obtener el código cuasi fuente a partir del código objeto) implica realizar actos de transformación y de reproducción, y por ello es una actividad que queda dentro del ámbito de protección de los derechos de autor y está sujeta a la autorización de los titulares de los derechos, pero que difícilmente dará lugar a una obra derivada, ya que o apenas existe intervención humana en el proceso o bien faltará el requisito de la originalidad.

En virtud del artículo 5.1 de la Directiva 91/250/CEE (y del art. 5.1 de la Directiva 2009/24/CE actualmente vigente) el usuario legítimo de un programa de ordenador está autorizado a realizar una descompilación de este, cuando tal actuación sea necesaria para corregir errores de funcionamiento que impidan su utilización de acuerdo a su finalidad, y siempre que no exista una prohibición expresa en el contrato, si bien no cabe una interdicción total, salvo que el titular de los derechosarezca un mantenimiento del programa que incluya la corrección de dichos errores o permita al usuario legítimo el acceso al código fuente del programa.

La descompilación puede ser de la totalidad del programa o de un parte de este, pero en cualquier caso ha de ser necesaria para llevar a cabo la corrección de los errores, algo que no ocurrirá cuando el código fuente del programa ya esté disponible.

Por otra parte, deben tratarse de auténticos errores, esto es, de defectos del programa que den lugar a fallos en su funcionamiento que impidan su uso de acuerdo a la finalidad del mismo, quedando excluida la obsolescencia del programa en el concepto de error (por ejemplo, que el programa no funcione en un sistema operativo más moderno para el que fue diseñado, o que lo haga de forma inapropiada).

Finalmente, al ser la descompilación una actividad costosa que requiere disponer de ciertos conocimientos técnicos y de un *software* especializado (descompilador), y cuyos resultados no están garantizados, será utilizada únicamente como último recurso, y normalmente solo por empresas o usuarios profesionales, ya que será muy complicado que un usuario doméstico pueda llegar a realizarla.

V. BIBLIOGRAFÍA

- APARICIO VAQUERO, Juan Pablo (2016), «Propiedad intelectual y suministro de contenidos digitales», *InDret. Revista para el análisis del Derecho* 3, págs. 1-58. <http://www.indret.com/pdf/1246.pdf> (última consulta 26 de marzo de 2022).
- (2017A), «Artículo 10.1.i», en BERCOVITZ RODRÍGUEZ-CANO, R. (coord.), *Comentarios a la ley de propiedad intelectual*, 4.^a ed., Tecnos, Madrid, págs. 193-195.
- (2017B), «Título VII. Programas de ordenador. Artículos 95-104», en BERCOVITZ RODRÍGUEZ-CANO, R. (coord.), *Comentarios a la ley de propiedad intelectual*, 4.^a ed., Tecnos, Madrid, págs. 1353-1529.
- CARBAJO CASCÓN, Fernando (2018), «La originalidad de las obras publicitarias y de las páginas web», en AAVV, *XXXIII Jornadas de Estudio sobre Propiedad Industrial e Intelectual*, AIPPI, Madrid 8 y 9 de febrero de 2018, págs. 39-60.
- (2002), *Publicaciones electrónicas y propiedad intelectual*, Colex, Madrid.
- FERNÁNDEZ MASIÁ, Enrique (2017), «Artículos 95-97», en PALAU RAMÍREZ, F., y PALAO ROMERO, G. (dirs.), *Comentarios a la ley de propiedad intelectual*, Tirant lo Blanch, Valencia, págs. 1183-1218.
- GONZÁLEZ SAN JUAN, José Luis (2017), «Protección jurídica de las páginas Web», en AAVV, *Reflexiones sobre Derecho Privado Patrimonial* (vol. VI), *Ratio Legis*, Salamanca, págs. 91-110. Disponible en https://diarium.usal.es/joselgon2006/files/2020/03/Proteccion_juridica_paginas_Web.pdf (última consulta 10 de marzo de 2022).
- (2020A), *Enlaces en la Web y derechos de autor y conexos*, Tirant lo Blanch, Valencia.
- (2020B), «Cuando lo que se pretende es proteger una idea o un modelo de negocio: comentario de la SAP de Barcelona (sección 15.^a) núm. 280/2017, de 29 de junio, asunto *Asco de Vida*», *ADI* 40 (2019-2020), págs. 419-434.
- LÓPEZ-TARRUELLA MARTÍNEZ, Aurelio (2017), «Artículos 98-104», en PALAU RAMÍREZ, F., y PALAO ROMERO, G. (dirs.), *Comentarios a la ley de propiedad intelectual*, Tirant lo Blanch, Valencia, págs. 1218-1271.
- SAIZ GARCÍA, Concepción (2000), *Objeto y sujeto del derecho de autor*, Tirant lo Blanch, Valencia.