

# TEMA 069. SOFTWARE DE CÓDIGO ABIERTO. SOFTWARE LIBRE. CONCEPTOS BASE. APLICACIONES EN ENTORNO OFIMÁTICO Y SERVIDORES WEB.

Actualizado a 07/05/2023

## 1. SOFTWARE PROPIETARIO, SOFTWARE LIBRE Y SOFTWARE ABIERTO

- Derechos de autor ⇒ sistema de protección de las creaciones humanas originales, expresadas por cualquier medio o soporte, conocido o futuro, entre las que se incluye el software (**SW**).
- Licencia ⇒ contrato contraído libremente entre las partes y que se atiene a unas normas básicas.
- Licencia de uso ⇒ contrato que se emplea por el titular de un SW para otorgar permisos de uso del SW a terceros, a cambio o no de una remuneración económica.
- **Software propietario** o privativo ⇒ por contraposición a las licencias libres, SW cuyo uso, redistribución o modificación está prohibido o requiere permiso expreso del titular del SW.
- **Software libre** ⇒ SW cuya licencia respeta las **cuatro libertades** definidas por la Free Software Foundation (**FSF**).
- **Software copyleft** ⇒ SW cuya licencia incluye unos términos de distribución que impiden añadir restricciones a la licencia o al SW original u obras derivadas.
- **Software abierto** ⇒ SW cuya licencia cumple las **diez directrices** de la definición de código fuente abierto u Open Source Definition (**OSD**) establecida por la Open Source Initiative (**OSI**).

### SW LIBRE (*FREE SOFTWARE*)

El primer argumento utilizado por los defensores del SW libre es filosófico e ideológico, apoyado en la teoría de la génesis social del conocimiento. Su principio básico es que el conocimiento como tal no pertenece a nadie, pues cualquier noción proviene de unos conocimientos anteriores. Así, si su génesis es social, su uso debe seguir siendo social. Desde ese punto de vista, la función principal de la generación de conocimiento es mejorar la sociedad y, por lo tanto, llegar al máximo número posible de personas.

Aunque hay un precedente en la licencia Berkeley Software Distribution (**BSD**), los iniciadores del movimiento del SW libre son Richard Stallman y la FSF. En 1984 Stallman inició el proyecto GNU No es Unix (**GNU**) y sentó los fundamentos éticos del SW libre, poniendo el foco en que el SW es conocimiento y debe tener esa función social.

El término “*free*” no implica gratuidad, sino que se refiere a que el SW se distribuye bajo una licencia que otorga unas libertades al usuario, incluso si este es una empresa. En sentido estricto es la licencia la que es libre, no el SW como tal. Estas cuatro libertades básicas son:

- ✓ Libertad 0: libertad de ejecutar y usar el SW para cualquier propósito.
- ✓ Libertad 1: libertad de estudiar el programa y adaptarlo a tus necesidades.
- ✓ Libertad 2: libertad de distribuir copias.
- ✓ Libertad 3: libertad de modificar el programa y liberar esas modificaciones al público.

Que el usuario disponga del código fuente no es una libertad *per se*, pero es imprescindible para que existan la libertad 1 y la libertad 3. Así, el objetivo de Stallman y la FSF es que se respeten las libertades fundamentales de los usuarios y que estas se mantengan en relación con obras derivadas de SW originalmente libre. Para ello crean la General Public License (**GPL**), una licencia con copyleft, es decir, que establece la imposibilidad legal de capturar SW libre, modificarlo y volverlo privativo.

Para la FSF una aplicación solo es libre si se distribuye con GPL, que es una licencia persistente porque cualquier código añadido a un programa bajo licencia GPL conserva la licencia GPL. Aun así, no todo el SW

libre se distribuye con copyleft, que es lo que diferencia al SW libre de la FSF del resto de SW libre con licencias de código abierto.

#### SW DE CÓDIGO ABIERTO (OPEN SOURCE SOFTWARE)

En 1998 la FSF se escindió con la creación de la iniciativa OSI [NOTA: nada que ver con “Open Systems Interconnection”, la norma ISO 7498]. Esta estableció la definición de OSD con la intención de ofrecer una perspectiva del SW libre más pragmática y orientada al mundo empresarial. Para ello la OSI toma como punto de partida las directrices Debian Free Software Guidelines porque son lo suficientemente amplias como para incluir desde la licencia GPL a la Apache.

Es importante tener en cuenta que la OSD no es una licencia, ni un modelo de licencia, sino 10 directrices para clasificar licencias de aplicaciones. La OSD de facto es libre, porque respeta las cuatro libertades básicas del usuario, pero no incluye el concepto de copyleft para flexibilizar la distribución posterior de obras modificadas. El OSI Certified indica que una licencia está alineada con la OSD.

En el material adicional se puede leer el desarrollo de las directrices que ha de cumplir una licencia acorde con la OSD:

1. Permitir redistribución libre.
2. Dar acceso al código fuente.
3. Permitir realizar obras derivadas y distribuirlas en los mismos términos.
4. Asegurar la integridad del código fuente del autor.
5. No discriminar a personas o grupos.
6. No discriminar a sectores de actividad.
7. Distribuir la licencia asociada al programa sin necesidad de firmar una licencia adicional.
8. No ser específica de un producto o de una versión.
9. No limitar a otro SW que se distribuya conjuntamente.
10. Ser neutra respecto de la tecnología.

#### SW LIBRE VS SW DE CÓDIGO ABIERTO

No son dos movimientos opuestos entre sí, sino que persiguen objetivos diferentes (aunque no contrapuestos). Mientras que la FSF prima la libertad de los usuarios, la OSI busca dar mayor libertad a los desarrolladores.

La FSF defiende la libertad de uso y distribución a cualquier precio para beneficio de la comunidad, aun con el riesgo de perder el apoyo y las contribuciones de desarrolladores que quieren mantener un control económico sobre sus obras derivadas. La OSI busca que se cree SW libre con su certificación y que grandes compañías se sumen al proyecto.

Por su parte, los defensores del sistema GNU consideran que el SW libre que no esté bajo el paraguas de la licencia GPL no es verdaderamente libre, porque permite que en el futuro sea distribuido como SW privativo que no garantice las cuatro libertades básicas. Por otro lado, los defensores de las licencias de código abierto argumentan que distribuir el SW libre sin copyleft garantiza el acceso a un mayor número de usuarios y la mejor comercialización del SW.

## 2. TIPOS DE LICENCIAS LIBRES

Una licencia de uso de SW establece lo que el usuario puede hacer o no con el SW. La diferencia entre las licencias de uso de SW libre y las de SW propietario reside en los derechos y obligaciones, que suelen ser directamente opuestos entre sí. Así, las licencias libres no pretenden proteger los derechos exclusivos de explotación del SW, sino proteger los derechos morales del autor del SW, proteger las libertades de uso, modificación y distribución de los usuarios e impedir la privatización del SW libre. Aun así, no todas las licencias de SW libre son iguales, puesto que algunas como la BSD otorgan más libertad a los desarrolladores, mientras que la GPL transmite más libertad al usuario final.

### LICENCIAS LIBRES CON COPYLEFT ROBUSTO

Estas licencias no solo pretenden asegurar las cuatro libertades del SW libre, sino que mediante el copyleft buscan que sus modificaciones y obras derivadas sean distribuidas en las mismas condiciones.

#### GNU GPLv2

La Licencia General Pública de la FSF es la principal exponente de las licencias con copyleft robusto. Se creó en 1986 y en su preámbulo incluye una enunciación de la filosofía del SW libre y un resumen sencillo de la licencia. Las cuatro libertades fundamentales del SW libre garantizan:

- a) Derecho de copia, modificación y distribución del código fuente original.
- b) Derecho a distribuir las modificaciones, siempre bajo GPL y sin cobrar por usar la licencia (cláusula copyleft).
- c) Derecho de copia y distribución del programa en forma de código objeto o ejecutable, siempre que el código fuente se ponga a disposición de terceros y sin poder cobrar por nada más que su coste de entrega.

Por lo tanto, la GPL permite modificar un programa para uso personal (privado o comercial) y distribuir copias de estas modificaciones sin necesidad de avisar de los cambios a los autores originales, pero sí a los destinatarios. Asimismo, permite cobrar por la distribución de una copia y ofrecer garantías a cambio de un pago. Si un programa se distribuye bajo GPL en formato binario se debe acompañar del código fuente, u ofrecerse a proveerlo durante un plazo mínimo de 3 años, para lo que basta con dar acceso al código desde Internet.

Una distribución de un programa bajo licencia GPL debe incluir:

- a) Un aviso de autoría.
- b) Una indicación de las modificaciones realizadas y su fecha.
- c) Un aviso de ausencia de garantías.
- d) La copia de la licencia, para lo que basta con indicar la URL en [www.fsf.org](http://www.fsf.org) de la licencia.

El usuario no tiene que aceptar la licencia GPL para usar el programa, pero sí ha de aceptarla para modificar o distribuir el SW. Por lo tanto, si un usuario modifica o distribuye el SW implícitamente acepta la GPL y sus términos y condiciones. Eso incluye “no agregar más restricciones” sobre el ejercicio de derechos de autor y las libertades propuestas por la FSF, ni sobre el derecho aplicable, la manera de distribuir obras derivadas, el alcance de las licencias de patentes, obligaciones en cuanto a documentación, publicidad, obligaciones adicionales para la provisión del código fuente, etc.

#### GNU GPLv3

Las principales diferencias respecto a la GPLv2 son:

- a) Se impide que el HW que implemente SW libre empotrado imponga restricciones al usuario vía Digital Rights Management (**DRM**), de modo que el usuario no pueda instalar, modificar o ejecutar versiones de dicho SW. Esto, conocido como “cláusula anti TiVO” (un grabador de vídeo digital), no impide que los fabricantes incorporen dispositivos DRM con SW libre, pero sí que se limiten las libertades básicas de los usuarios.
- b) Si el titular de una patente cede los derechos de licencia sobre esa patente a un SW GPL, se establece que esos derechos sobre la patente se extienden a todo el programa y derivados, como una licencia ilimitada de patente.
- c) Es más fácil utilizar código GPLv3 combinado con SW publicado bajo otras licencias.

Para flexibilizar y aumentar la compatibilidad con otras licencias se permite integrar SW con restricciones adicionales a las de GPLv3 en aspectos como marcas, indemnizaciones, garantías... En cualquier caso, las garantías y responsabilidades de GPLv3 quedan excluidas en la medida que permita la ley de cada país. Respecto a los programas distribuidos con GPLv2, los programas GPLv3 son incompatibles si la licencia dice que “se distribuye únicamente con GPLv2”. Sí son compatibles si la licencia indica “GPLv2 o posteriores”.

#### GNU LGPLv3

Diseñada para aplicarse a bibliotecas informáticas, la FSF la denominó “lesser” porque garantiza menos libertades a los usuarios que la GPL. Eso es así porque la Lesser LGPL o Library GPL (**LGPL**) se creó para permitir que se enlacen algunos componentes de SW específico (las bibliotecas o librerías) con programas no libres, de modo que no afecte al programa resultante. Por tanto, una biblioteca bajo LGPL ofrece comodidad a un desarrollador de aplicaciones propietarias que quiere vincular (usar, pero no integrar en el código privativo) sus programas con componentes bajo licencias libres, sin entrar en el efecto copyleft de la GPL.

#### GNU AGPLv3

La Affero GPL (**AGPL**) Es la licencia oficial de la FSF para SW en red. Sus términos son en la práctica los mismos que la GPLv3, con un párrafo adicional que permite a los usuarios que interactúan con SW bajo esta licencia recibir la fuente de tal SW. Esto resuelve el “ASP loophole”, que permitía a un Application Service Provider (ASP) no activar el copyleft al utilizar código GPL como aplicación en páginas web, sin que hubiera distribución por hacerse uso del SW por la red.

AGPL es incompatible con GPLv2. Tampoco es técnicamente compatible con GPLv3, ya que no puede tomarse código publicado bajo AGPL, copiarlo o modificarlo bajo los términos de GPLv3, y viceversa. Sin embargo, sí se pueden combinar módulos separados o archivos fuente publicados bajo ambas licencias y obtener un único proyecto.

#### EUPLv1.2

El Esquema Nacional de Interoperabilidad (**ENI**) recomienda que se use la European Union Public License (**EUPL**) si se quiere liberar el SW privativo de la Administración. Esta licencia de ámbito mundial, a título gratuito y no exclusiva, autoriza al licenciatarario (quien recibe la licencia) a:

- a) Dar cualquier uso a la obra y sus copias.
- b) Reproducirla, modificarla y realizar obras derivadas.

- c) Comunicarla al público.
- d) Distribuir, prestarla y alquilarla.
- e) Subcontratar los derechos de la obra o de las copias de la misma.

Las obligaciones o restricciones que ha de respetar el licenciario son:

- a) Derecho de atribución: debe citar íntegras las advertencias y menciones a los derechos de autor, patentes, marcas registradas, licencia o responsabilidades. Las modificaciones u obras derivadas del licenciario incluirán una advertencia declarando que se ha modificado la obra y la fecha de la modificación.
- b) Cláusula copyleft: el licenciario no puede ofrecer ni imponer condiciones adicionales sobre la obra u obras derivadas que modifiquen o limiten las condiciones de la licencia.
- c) Cláusula de compatibilidad: si se mezcla código EUPL y código bajo una licencia compatible, el resultado puede distribuirse con esa licencia compatible (la GPL es más restrictiva).
- d) Al distribuir una copia de la obra, el licenciario debe suministrar el código fuente.
- e) Salvaguarda de otros derechos: no se permite el uso de nombres comerciales o marcas, excepto cuando sea necesario para indicar origen de la obra y mencionar los derechos de autor.
- f) Con solo usar el SW ya se acepta la licencia (en GPL se acepta al modificar o distribuir el código).

Por su parte, el licenciante (quien otorga la licencia) debe adjuntar el código fuente, o al menos indicar un repositorio para acceder a él gratuitamente mientras siga distribuyendo la obra. Además, el licenciante suministra la obra en su condición de inacabada y sin garantías (exclusión de garantía), y no se hace responsable de daños y perjuicios salvo dolo o daños directos a personas físicas (exclusión de responsabilidad).

En los países que contemplen los derechos morales el licenciante renuncia a su ejercicio, si bien en España eso no es posible porque son legalmente irrenunciables. Se cede al licenciario el derecho de uso no exclusivo sobre patentes en tanto sea necesario para utilizar los derechos sobre la obra con licencia EUPL. También se permite cobrar por ofrecer mantenimiento o proporcionar garantías a los usuarios, pero nunca en nombre del licenciante original ni de ningún otro colaborador.

Son licencias compatibles con la EUPL:

- GNU GPLv2.0 y GPLv3.0
- GNU AGPL v. 3.0
- Open Software License OSLv2.1 y OSLv3.0
- Common Public License CPLv1.0
- Eclipse Public License EPLv1.0
- CEA CNRS INRIA Logiciel Libre CeCILLv2.0 y CeCILLv3.0
- Mozilla Public License MPLv2.0
- LGPLv2.1 LGPLv3.0
- Licence Libre du Québec - Réciprocité LiLIQ-R y Réciprocité forte LiLIQ-R+

#### LICENCIAS LIBRES SIN COPYLEFT ROBUSTO O PERMISIVAS

Las licencias libres permisivas posibilitan que una obra pueda ser redistribuida como libre o como privativa. En ese sentido, hay que aclarar que el copyleft robusto no tiene nada que ver con la compatibilidad (que se mira entre dos licencias comparando sus cláusulas), sino con el análisis del respeto

de una licencia por las cuatro libertades del SW libre. Así, aunque la BSDv1.0 y la BSDv1.1 no son de copyleft robusto, la segunda es compatible con la GPL, mientras que la primera no lo es.

#### BSD

Fue la primera licencia libre y quizás la más simple de todas. Nace en los 70 de la distribución de versiones de UNIX de la Universidad de California, Berkeley. Como el código es fruto de las investigaciones financiadas por los EEUU y los impuestos de sus ciudadanos, es de acceso libre, obligando solo a mantener los avisos de autoría ("*copyright notice*"). La BSD transfiere al usuario todos los derechos que permite la propiedad intelectual:

- a) Permite el uso, modificación, copia y redistribución sin restricción del SW, en código objeto o fuente.
- b) Obliga a distribuir los códigos incluyendo el aviso de *copyright*, las condiciones y la negación de cualquier garantía y responsabilidad.
- c) Impide usar sin permiso el nombre de los autores para promocionar obras derivadas.

Se puede hacer cualquier cosa con código BSD, respetando siempre la mención de los autores originales del programa inicial. Esta cláusula hace que BSDv1.0 sea incompatible con GLPv2, pero el código BSDv1.1 es compatible con GPLv2 y GPLv3 (aunque no al revés).

Como se permite modificar el SW e imponer otra licencia, la BSD es una licencia no persistente. Por ello es muy favorable para desarrollos comerciales y propietarios y, al igual que LGPL, posibilita una gran difusión del código y su uso como estándar para protocolos o servicios. También permite la bifurcación de código (*forking*), porque cualquiera puede crear una versión "similar, pero suficientemente diferente", lo que se traduce en la proliferación de sistemas operativos de tipo BSD.

La BSD sirve de base para la licencia del Massachusetts Institute of Technology (**MIT**), una de las más populares en el mundo del SW libre. Así, en enero de 2022 se posicionaba como la segunda más empleada en la plataforma de desarrollo colaborativo GitHub, con un 26% de cuota.

La BSD también ha sido el modelo de las licencias OpenBSD, FreeBSD y NetBSD, de la familia X (X, XFree86, XOpen, X11) y de otras como Apache, Python, W3C Software Notice and License, OpenLDAP public license, Phorum o Zope Public License (ZPL). Por su parte, aunque las licencias OpenSSL y Sleepycat siguen el modelo simplificado de BSD, sí que incluyen cláusulas de copyleft.

#### ASLv1.0 y ASLv1.1

El servidor web Apache nace en el National Center for Supercomputing Applications de la universidad de Illinois, y mantiene una licencia tipo BSD con algunos extras. Por un lado, está la obligación de mantener la publicidad sobre los autores originales en la documentación o en el SW de redistribuciones, lo que hace que la Apache Software License (**ASL**) ASLv1.0 y la ASLv1.1 no sean compatibles con la GPLv2 ni con la GPLv3. Por otro lado, las obras derivadas no pueden usar el nombre Apache sin autorización de la Apache Software Foundation.

Al igual que cualquier programa bajo la BSD, se puede combinar código Apache con otro SW, no se puede usar el nombre para fines publicitarios, no se incluyen garantías y los licenciantes no se hacen responsables por daños y perjuicios.

#### ASLv2

La ASLv2 intenta facilitar su uso en proyectos que no son de la Apache Software Foundation, mejora la compatibilidad con SW GPL y requiere una licencia de patente sobre las contribuciones que necesariamente infrinjan patentes propias de un contribuyente. Obliga a mantener una copia de la licencia, el aviso de titularidad y las exclusiones de garantías y responsabilidades en el código fuente. Además, si se redistribuye el SW en forma de código objeto esta información debe estar en la documentación. También obliga a indicar cualquier modificación realizada y a mantener el fichero “notice.txt” (con comentarios jurídicos) con el código o la documentación de SW redistribuido. Por último, prohíbe usar el nombre y las marcas de licenciante para promocionar SW derivado.

ASLv2 es compatible con la mayoría de las licencias de código abierto, incluida la GPLv3, pero es incompatible con la GPLv2 porque la licencia de patente es una restricción adicional. Aun así, precisamente por esta restricción de patente, la FSF recomienda la ASLv2 frente a cualquier otra licencia sin copyleft.

#### OTRAS LICENCIAS

En 1998 Netscape Navigator se hizo de código abierto. Ante la negativa de la empresa a usar GPL y la necesidad de respetar la propiedad intelectual de terceros, desarrollaron la Netscape Public License (**NPL**), una licencia que cumple la definición de la OSI. A su vez se desarrolló la Mozilla Public License (**MPL**), con características copyleft para asegurar que las contribuciones seguían siendo de código abierto, y que ahora se emplea para Mozilla Firefox y LibreOffice, entre otros programas.

Ambas son muy similares, pero la NPL se reserva derechos por Netscape para su código inicial, por lo que no es compatible con GPL. La MPLv1.1 tampoco lo es, pero permite utilizar una licencia simultánea que sí es compatible. Por su parte, la MPLv2.0 hace la licencia más fácil de aplicar y logra la compatibilidad con las licencias GNU GPL y Apache ASL. Aun así, la MPLv2.0 permite añadir una nota que diga que la licencia es incompatible con licencias secundarias.

Por último, las licencias pseudolibres son creadas por empresas que intentan beneficiarse del modelo de desarrollo libre y cumplen solo algunas libertades básicas. Algunos ejemplos serían la Sun Community Source License y la Microsoft Shared Source Initiative.

#### LICENCIAS CREATIVE COMMONS

Creative Commons es una organización estadounidense que permite utilizar una serie de licencias gratuitas que ofrecen al autor de una obra una manera simple y estandarizada de elegir los términos y condiciones bajo los que otorga permiso al público en general de compartir y usar su trabajo creativo. Estas licencias no reemplazan a los derechos de autor, sino que se apoyan en estos para pasar de tener “todos los derechos reservados” sobre una obra a “algunos derechos reservados”, de la manera que mejor satisfaga las necesidades del autor original.

Las licencias Creative Commons están compuestas por cuatro módulos de condiciones:

1. BY - Reconocimiento (*attribution*): requiere la referencia al autor original.
2. SA - Compartir igual (*share alike*): permite obras derivadas bajo la misma licencia o similar.
3. NC - No comercial (*non-commercial*): impide que la obra sea empleada con fines comerciales.
4. ND - Sin obra derivada (*no derivative works*) (ND): impide la modificación de la obra.



Estos módulos se combinan para dar lugar a las seis licencias Creative Commons:

- a) CC BY: Reconocimiento - Sin obra derivada.
- b) CC BY-SA: Reconocimiento - Compartir igual.
- c) CC BY-ND: Reconocimiento - Sin obra derivada.
- d) CC BY-NC: Reconocimiento - No comercial.
- e) CC BY-NC-SA: Reconocimiento - No comercial - Compartir igual.
- f) CC BY-NC-ND: Reconocimiento - No comercial - Sin obra derivada.

Una licencia contractual especial es la opción CC0 o “sin derechos reservados”, que cede la obra al dominio público a escala global, sin limitarse a algunas jurisdicciones. Todas las licencias Creative Commons permiten el derecho fundamental de redistribuir la obra con fines no comerciales y sin modificaciones. Las opciones NC y ND son restrictivas, por lo que hacen que la obra no sea libre. En el material adicional se puede ver una infografía que clarifica su funcionamiento.

### 3. PRINCIPALES ACTORES E INTERESES

#### GENERADORES Y DISTRIBUIDORES DE SOFTWARE LIBRE

Para las empresas de SW propietario el modelo más habitual se basa en el desarrollo interno por programadores asalariados. En cambio, el SW libre o de código abierto se suele generar cooperativamente por programadores, muchas veces voluntarios, que trabajan de forma coordinada a través de Internet.

Los proyectos de generación de SW libre se pueden clasificar en:

- Vinculados a empresas: el pago por licencia pierde su sentido, con lo que las fuentes de ingresos se encuentran en el valor añadido que dan al producto, especialmente el soporte técnico, la formación y la personalización. Las empresas más representativas son Red Hat, Mozilla o IBM, que ha colaborado en las comunidades de Java, OpenStack, Docker, Kubernetes...
- Desarrollados en universidades: reciben financiación de empresas o gobiernos, como el desarrollo de la familia de sistemas operativos BSD.
- Liderados por grupos de voluntarios: como la propia FSF, la OSI, Debian o Abiword.

#### SW LIBRE EN LA ADMINISTRACIÓN

A lo largo de los años, diferentes Comunidades Autónomas y universidades han desarrollado sus propias distribuciones de Linux. Una de las pioneras fue gnuLinEx, impulsada por la Consejería de Educación, Ciencia y Tecnología de Extremadura, si bien ha terminado por ser discontinuada. La dificultad del mantenimiento de muchos de estos proyectos ha hecho que la mayoría hayan sido abandonados, por lo que, aún en el caso de que se puedan descargar, no disponen de parches de actualización desde hace años. Es el caso de Asturix (Asturias), Galinux (Galicia), Guadalinux (Andalucía), Melinux (Melilla) o Molinux (Castilla La Mancha), entre otros muchos.

Aun así, existen iniciativas con el respaldo suficiente que se mantienen vivas a día de hoy:

- Linkat (Cataluña)
- Lliurex (Valencia)

- MAX Linux (Madrid)
- Trisquel (Galicia)
- Vitalinux (Aragón)

Desde un punto de vista legislativo, el Real Decreto 4/2010 que regula el ENI en el ámbito de la Administración Pública, señala sobre estándares aplicables en su artículo 11.1 que las Administraciones Públicas (**AAPP**) usarán preferentemente estándares abiertos en los documentos y servicios de administración electrónica que se pongan a disposición de la ciudadanía y de otras AAPP. Se pretende de este modo satisfacer el principio de neutralidad tecnológica y la adaptabilidad al progreso de la tecnología, así como evitar la discriminación a los ciudadanos por razón de su elección tecnológica. A su vez, el artículo 11.2 indica que las AAPP promoverán las actividades de normalización con el fin de facilitar la disponibilidad de los estándares abiertos relevantes para sus necesidades. En cuanto al formato de los documentos, el artículo 23 del ENI indica que estos deberán conservarse preferentemente empleando un estándar abierto que preserve a lo largo del tiempo la integridad del contenido del documento, de la firma electrónica y de los metadatos que lo acompañan.

Inciendo en la importancia de los formatos, la disposición adicional primera del ENI contempla el desarrollo de la Norma Técnica de Interoperabilidad de Catálogo de estándares. Aquí se recogen un buen número de estándares abiertos para emplear en las AAPP. Un ejemplo es la norma ISO/IEC 26300:2006, que incluye OpenDocument Format (**ODF**), un formato de documento abierto y basado en XML recomendado en la Unión Europea para intercambiar texto, hojas de cálculo, gráficos y presentaciones.

#### USUARIOS DE SOFTWARE LIBRE

Tradicionalmente el principal usuario de SW libre era la propia comunidad de desarrolladores y distribuidores, si bien se ha multiplicado su uso en la sociedad en general, gracias a la Administración Pública, los centros de enseñanza y las organizaciones de cooperación al desarrollo. Las motivaciones de estas iniciativas son variadas:

- Ahorrar costes por no pagar licencias y poder usar tantas copias como se necesite.
- Evitar la dependencia tecnológica de grandes multinacionales.
- Tener la posibilidad de personalizar el SW a medida.
- Luchar por reducir la “división digital”, el acceso a tecnología en regiones menos desarrolladas, dado que el SW propietario ha sido históricamente poco sensible a economías desfavorecidas y lenguas minoritarias.

#### 4. PROYECTOS DE SW LIBRE

##### KDE

El proyecto KDE Desktop Environment fue iniciado en 1996 por el programador alemán Matthias Ettrich, que pretendía crear una interfaz gráfica unificada para sistemas Unix en la que todos sus componentes pudieran ser configurados por el usuario. Bajo licencia GPL y LGPL, estaba basado en Qt, un framework multiplataforma orientado a objetos ampliamente usado para desarrollar SW con interfaz gráfica.

En 2014 KDE Desktop Environment comienza a llamarse sencillamente KDE, refiriéndose así tanto a su comunidad de usuarios y desarrolladores como a las tecnologías que estos crean. El nombre KDE deja de emplearse para abarcar todos sus proyectos y se crean varias ramas, cada una con su propio calendario de publicaciones:

- El entorno de escritorio pasa a denominarse KDE Plasma 5.
- KDE Frameworks es la colección de librerías en Qt que permiten construir aplicaciones.
- El conjunto de aplicaciones como tal se queda como KDE Applications y termina como KDE Gear.

## GNOME

GNOME, con licencia GPL para sus aplicaciones y LGPL para sus librerías, es el entorno de escritorio por defecto de muchas de las distribuciones más importantes de Linux, como Debian, Fedora, OpenSUSE, Ubuntu o Red Hat. Esta última es la mayor de las empresas que colaboran junto a personas voluntarias en el GNOME Project, grupo que lleva a cabo diversos proyectos de desarrollo de frameworks y de aplicaciones de usuario.

GNOME fue creado en 1997 bajo el nombre GNU Network Object Model Environment por los programadores mexicanos Miguel de Icaza y Federico de Mena. Se presentaba como una alternativa a KDE completamente libre, ya que Qt en origen era SW propietario. En su lugar se empleó Gnome Toolkit, también conocido como GIMP Toolkit, y actualmente como GTK. Los principales componentes de GTK en su versión 44 (marzo de 2023) son:

- Glib: conjunto de librerías de bajo nivel de propósito general.
- Cairo: biblioteca gráfica que proporciona imágenes basadas en gráficos vectoriales.
- GDK: actúa como envoltorio entre la biblioteca de componentes gráficos GTK y la API del sistema gráfico nativo y ofrece primitivas gráficas de dibujo, cursores, fuentes, eventos de ventanas, etc.
- Pango: librería de renderizado de texto.
- ATK: API de desarrollo de aplicaciones accesibles.
- GIO: biblioteca de interfaz de Sistema de ficheros virtual.

## APACHE

Apache es un servidor web HTTP de código abierto y multiplataforma con licencia Apache 2.0. Nacido en 1995 y desarrollado y mantenido bajo el auspicio de la Apache Software Foundation, se encuentra en su versión 2.4.57 (abril de 2023). Se usa sobre todo para enviar páginas web estáticas y dinámicas por la World Wide Web, así como para poner contenido a disposición de terceros de forma segura y confiable. Destaca por su modularidad, que permite un alto grado de personalización de las funcionalidades, así como por su robustez.

La combinación de Apache con otras tecnologías ha dado lugar a la pila tecnológica denominada LAMP, una de las más comunes para crear aplicaciones web en Internet. Esta emplea Linux como sistema operativo, Apache como servidor web, MySQL o MariaDB como base de datos y PHP, Perl o Python como lenguaje de programación.

## LIBREOFFICE Y APACHE OPENOFFICE

StarOffice era una suite ofimática de código cerrado comprada por Sun Microsystems en 1999. Un año después liberaron el código fuente y formaron la comunidad OpenOffice. En 2011 Oracle adquirió Sun Microsystems y una buena parte de la comunidad, vista la actitud de Oracle ante otros de sus proyectos de código abierto, creó una bifurcación del código que se denominó LibreOffice.

LibreOffice se licenció bajo la MPLv2.0 y quedó bajo el paraguas de la organización sin ánimo de lucro The Document Foundation. Es compatible con Microsoft Office y soporta el estándar OpenDocument (ISO 26300) para intercambio de datos en formato ODF. Sus herramientas principales son:

- Writer: procesador de texto.
- Calc: SW de hojas de cálculo.
- Impress: programa de presentación de transparencias.
- Draw: editor de gráficos vectoriales.
- Base: gestor de bases de datos.
- Math: aplicación de creación y edición de fórmulas matemáticas.

Por su parte, Oracle cedió la marca OpenOffice y el código a la Apache Software Foundation. Esta licenció Apache OpenOffice bajo la Apache License 2.0, con lo que su bifurcación del código es de código abierto, pero no es software libre según la definición de la FSF. En cualquier caso, esta suite tiene los mismos componentes que LibreOffice, pero con una comunidad de desarrolladores mucho más pequeña.

## MONO

Con el respaldo de Microsoft y liderado por Xamarin, Mono es una implantación del framework .NET de código abierto. Creada en 2004, tiene licencia MIT y se basa en los estándares ECMA para C# y Common Language Runtime. La versión 6.12 (junio de 2022) ofrece paquetes para macOS, Linux, Windows y Docker, gracias a su pila de compatibilidad con Microsoft y a su pila de desarrollo Mono/Linux/GNOME.

## OTRAS HERRAMIENTAS DE SW LIBRE

En el mercado hay centenares de aplicaciones con licencias libres orientadas a usuarios finales. Algunas de las más populares son:

- Navegador web: Mozilla Firefox, Google Chrome (aunque esta tiene componentes propietarios).
- Correo electrónico: Mozilla Thunderbird, GNOME Evolution.
- Grabación de CD y DVD: Brasero, K3b.
- Reproductores multimedia: VLC Media Player, Rhythmbox.
- Editor gráfico: GIMP.
- IDE: Eclipse, Netbeans.

## ACRÓNIMOS

- **AGPL:** Affero GPL
- **AAPP:** Administraciones Públicas
- **ASL:** Apache Software License
- **BSD:** Berkeley Software Distribution
- **DRM:** Digital Rights Management
- **ENI:** Esquema Nacional de Interoperabilidad
- **EUPL:** European Union Public License
- **FSF:** Free Software Foundation
- **GNU:** GNU No es Unix
- **GPL:** General Public Licence
- **LGPL:** Lesser LGPL o Library GPL
- **MIT:** Massachusetts Institute of Technology
- **MPL:** Mozilla Public License
- **NPL:** Netscape Public License
- **ODF:** OpenDocument Format
- **OSD:** Open Source Definition
- **OSI:** Open Source Initiative
- **SW:** Software