



# Análisis del uso de software pirata sobre el software libre o de código abierto.

Autores:

- Marotta, Alejandro Adrián.
- Soria Gava, Lucas Damián.

# Índice:

<b>1. Introducción</b>	<b>2</b>
<b>2. Desarrollo</b>	<b>3</b>
2.1. Tipos de software	3
2.1.1. Software propietario	3
2.1.2. Software libre y código abierto	3
2.1.3. Software pirata	4
2.2. Peligros del software pirata	4
2.3. ¿Por qué se da la piratería?	5
2.4. Software comúnmente pirateados	6
2.5. Beneficios del FOSS	8
2.5.1. Calidad en el FOSS	9
2.5.1.1. ¿Qué es la calidad de Software?	9
2.5.1.2. Mejoras de la comunidad al FOSS	9
2.5.1.3. ¿Qué es la ISO 27001?	10
2.6. ¿Por qué eligen entonces lo pirateado al FOSS?	11
2.7. Relación del FOSS con la economía naranja	12
<b>3. Conclusión</b>	<b>13</b>
<b>4. Bibliografía</b>	<b>14</b>

# 1. Introducción

El propósito de esta monografía es poder reflexionar sobre la piratería de software, analizando algunas de las razones por las que este fenómeno se suele presentar incluso frente a alternativas de uso libre.

Partiremos por clarificar la diferencia entre los distintos tipos de software, para posteriormente describir qué es y qué problemas o consecuencias puede traer el uso de software pirata. Compararemos la percepción que tienen los usuarios sobre el software propietario frente al software de uso libre y por qué estos pueden preferir la piratería ante su alternativa. Adicionalmente, planteamos algunas de las ventajas que presenta el software tanto el software libre, como también las ventajas del mismo software propietario.

Por último, analizaremos el cumplimiento de estándares de calidad o excelencia en la producción de software libre o de código abierto y el rol que tiene la comunidad en su mejora continua.

## 2. Desarrollo

### 2.1. Tipos de software

Para poder discutir y analizar las diferencias entre los cuatro tipos de software planteados (propietario, libre y de código abierto, y pirata) primero debemos poder describir a cada uno y más concretamente, definir qué software es propenso a ser pirateado, ya que de estos partirá el análisis planteado anteriormente. La forma en la que comúnmente se denomina al software distribuido listo para usar es “software enlatado”.

#### 2.1.1. Software propietario

Es un modelo de desarrollo de software donde toda pieza de código queda en manos de la entidad que la produce, ya sea una persona, empresa u organización. Los derechos de uso, copia, modificación o distribución del software están reservados a dicha entidad, la cual puede distribuir el mismo de la forma que prefiera, pudiendo cobrar por su uso o no, pero el código fuente estará siempre y exclusivamente en su dominio.

#### 2.1.2. Software libre y código abierto

El término software libre y de código abierto (FOSS, por sus siglas en inglés) es un término neutro utilizado para describir tanto al software libre como al software de código abierto.

Software libre es un término acuñado por Richard Stallman en 1990 y consiste en software regido por una filosofía de libertad, respetando la de los usuarios y comunidades. Esta filosofía da la libertad a los usuarios de usar, copiar, estudiar, cambiar, mejorar y distribuir el software a gusto, siempre y cuando el desarrollo se rija por sus cuatro libertades esenciales. Es importante destacar que libre no es sinónimo de gratis, ya que sus creadores pueden cobrar por este tipo de programas, si así lo desearan.

Por otra parte, el software de código abierto no solo es aquel que como su nombre lo indica, permite a los usuarios acceder al código fuente, sino que sus términos de distribución deben cumplir con diez principios o criterios. Dichos criterios restringen sobre todo las licencias de redistribución del software, las cuales deben atenerse a las limitaciones o permisos que presentaba el desarrollo original.

Tanto el software libre como el software de código abierto hacen referencia a un mismo tipo de licencias, pero se diferencian en los valores que representan, en palabras textuales de Stallman:

*“The two now describe almost the same category of software, but they stand for views based on fundamentally different values. For the free software movement, free software is an ethical imperative, essential respect for the users' freedom. By contrast, the philosophy of open source considers issues in terms of how to make software “better”—in a practical sense only.”*

*“Ambos describen casi la misma categoría de software, pero representan puntos de vista basados en valores fundamentalmente diferentes. Para el movimiento del software libre, el software libre es un imperativo ético, un respeto esencial por la libertad de los usuarios. Por el contrario, la filosofía del código abierto considera los problemas en términos de cómo hacer que el software sea “mejor” — solo en un sentido práctico.”*

La principal diferencia entre ambos es que el software de código abierto es más restrictivo y permite a su creador decidir sobre su uso, copia, modificación y distribución, mientras que “libre” significa (casi) total libertad.

### 2.1.3. Software pirata

Por último, software ilegal o pirata describe el uso, copia, falsificación y/o distribución no autorizada de software con copyright. Esto puede incluir, copia no autorizada de un producto pago, ganar acceso ilegal a software protegido o redistribuir software sin el permiso del dueño del software, entre otras.

Siguiendo las definiciones dadas anteriormente, el software propenso a ser pirateado es: el software propietario, el software libre y el software de código abierto. En definitiva, todo el software es susceptible de piratería, pero el principal afectado es el software enlatado, que se distribuye masivamente listo para su uso, especialmente los de aplicación general (por ejemplo: procesamiento de texto, hojas de cálculo, paquetes de gráficos básicos, etc.).

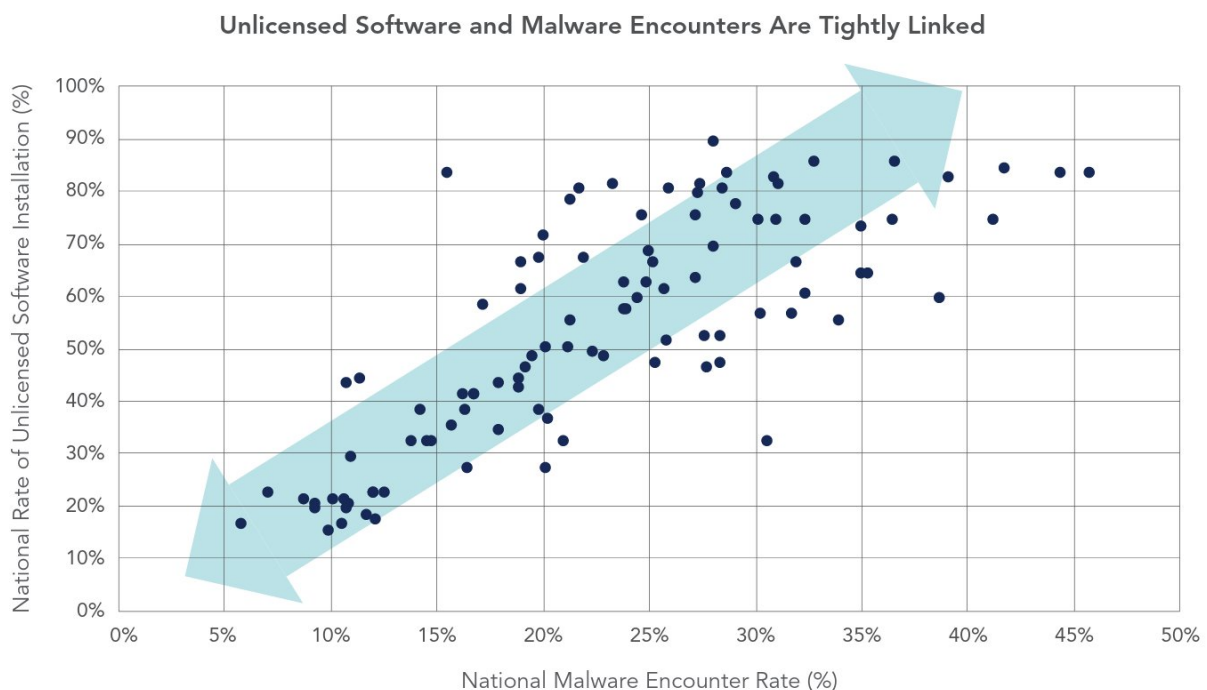
## 2.2. Peligros del software pirata

Los riesgos que se pueden enfrentar al descargar este tipo de software, ya sea en las computadoras personales o del trabajo, son:

- **Mal funcionamiento:** Estos programas aumentan las posibilidades de que sus herramientas no funcionen correctamente.
- **Seguridad:** Muchos de estos programas informáticos están infectados por virus, troyanos o spybots, los cuales podrían provocar consecuencias irreparables en la computadora donde esté instalado.
- **Servicio técnico:** Al trabajar con un software pirata se pierde el acceso a su servicio técnico en línea, en tienda o vía telefónica, fundamental cuando el programa informático da cualquier tipo de problema funcional.
- **Mayores gastos:** A la larga, arreglar los problemas que un software pirata puede causar en el equipo, llegan a significar un mayor costo monetario que el que se hubiera invertido en el programa original.
- **Multas por incumplimiento de obligaciones legales:** El uso de softwares ilegales infringe la ley y puede causar sanciones legales contra empresas y personas. Se violan los derechos de la propiedad intelectual de la empresa que ha creado el programa, se pueden recibir denuncias y hasta penas de prisión.
- **Perder la certificación de calidad ISO:** Si la empresa no se ajusta a las prácticas de calidad necesarias por el mal uso de la tecnología, puede correr el riesgo de perder la certificación de calidad ISO, cuando el organismo de certificación, tras la auditoría, compruebe el uso de softwares ilegales.

- **Desconfianza de clientes y proveedores:** Si para contratar a la empresa un cliente decide realizar una auditoría de segundo orden, descubrirá que trabaja con programas ilegales y desconfiará de la seguridad informática de la empresa y dañará la imagen de esta.
- **Inconsistencia y desactualización:** Los programas sin licencia suelen tener fallos y generar un funcionamiento inadecuado en la aplicación. El software ilegal generalmente no tiene actualizaciones, por lo que en poco tiempo el programa se quedará obsoleto. Además, las posibilidades de infectar un equipo con virus son mayores, ya que muchas actualizaciones incluyen mejoras que te protegen de los ataques de virus recientes aprovechando alguna vulnerabilidad del software.

Estos peligros, especialmente la presencia de malware (comúnmente denominados virus informático), tienen una fuerte relación con el uso e instalación de software sin licencia. Dicha tendencia se puede visualizar en el gráfico N°1.



Source: IDC

Gráfico N°1.

Fuente: BSA Global Software Survey, June 2018.

### 2.3. ¿Por qué se da la piratería?

Dentro de las principales razones se encuentra la resistencia al pago, piratería instaurada culturalmente y el desafío o reto de crackers que piratean el software para demostrar que son capaces y que tienen las habilidades suficientes.

Otra de las causas más comunes del consumo de piratería es la imposibilidad de comprar el producto por vías legales. Este hecho se puede observar con más frecuencia en países en vías de desarrollo, ya que los precios suelen ser altos para sus niveles económicos.

Como ese software les permite desarrollarse y competir en el mercado actual que es altamente dinámico, muchos deciden descargar el software pirata sin prestarle atención a los riesgos que conlleva o directamente ignoran que existen. Partiendo de esta premisa, tomar un control estricto contra la piratería podría ser contraproducente para la producción de un país en vías de desarrollo, ya que podrían perder el acceso a ciertas tecnologías si no se les presenta una alternativa.

Las alternativas que surgen para poder brindarle acceso a todos los usuarios pueden venir por dos vías: diferenciación de precios o software libre. En el primer caso una empresa decide poner su producto a precios acordes a la economía del lugar, por ejemplo, compañías como *Valve Corporation* (empresa detrás de *Steam*) pone el precio de los videojuegos más baratos en Latinoamérica que en Europa o América del Norte, de esa forma no se corta la distribución de software y se controla la piratería. Pero esto representa un reto para las empresas ya que tienen que generar un mecanismo por el cual los usuarios no se aprovechen de esta funcionalidad y compren los productos más baratos estando en cualquier parte del mundo.

Por otro lado, se encuentra el software libre, que no es comúnmente utilizado ya que los usuarios presentan una fuerte resistencia hacia su uso o lo desconocen totalmente. Este tema lo abordaremos con más detalle en secciones posteriores.

## 2.4. Softwares comúnmente pirateados

Según la *Business Software Alliance* (BSA), los softwares más pirateados en el mundo son:

- Microsoft Windows.
- Microsoft Office.
- Adobe Photoshop and Creative Cloud.
- Norton y McAfee anti-virus.
- Autodesk CAD software.

Además, la compañía británica *Computer Disposal Limited* realizó un estudio sobre los programas más pirateados en base a información de búsquedas de Google con los términos “torrent” y “download free”, la empresa encontró cuáles son las consultas “piratas” más populares en lo que a software se refiere, llegando a una coincidencia con el estudio anterior. *Adobe* se lleva el podio con sus programas para creativos. *Adobe Photoshop*, *Adobe Illustrator* y *Adobe Premiere Pro* lideran las búsquedas ilegales. Seguido por programas como *Malwarebytes Premium* (antivirus) y los productos ofrecidos por *Microsoft*.

Se puede observar que hay una tendencia a descargar ilegalmente software de productividad y de producción de contenido creativo, lo cual indica que los usuarios utilizan estos programas para sus trabajos.

Llama fuertemente la atención que entre los softwares principalmente pirateados se encuentre el rubro del antivirus, ya que justamente se busca una alta fiabilidad en estos

productos, lo cual al descargarlos por esta vía pierden toda garantía de seguridad y confiabilidad.

Estas preferencias de consumo se ven reflejadas en el estudio de mercado global que realizó la empresa BSA. En el gráfico N°2 podemos observar que el uso del software sin licencia supera en la mayoría de las regiones el 50%.

#### Average Rate of Unlicensed Software Use

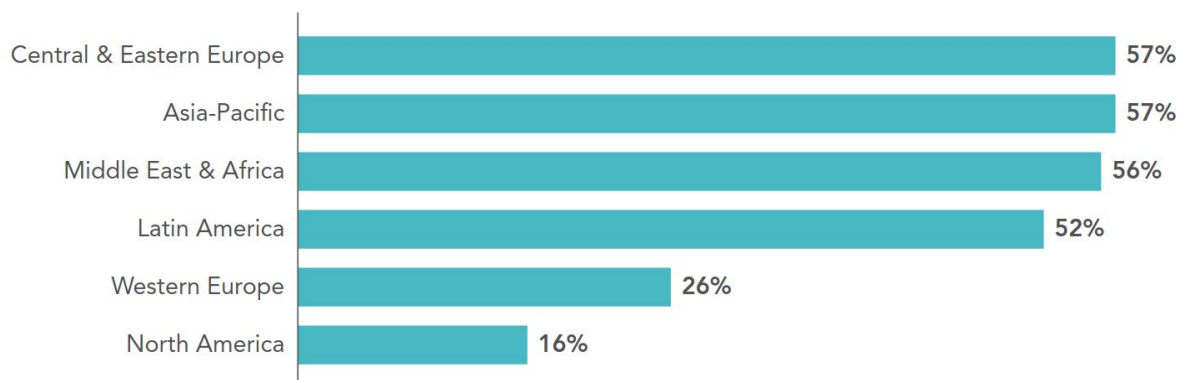


Gráfico N°2.

Fuente: BSA Global Software Survey, June 2018.

Al ser la producción de software una de las industrias más grandes de la actualidad, la piratería produce una gran pérdida monetaria para las empresas. Como lo indica el gráfico N°3, llega a superar los miles de millones de dólares, representando un grave problema ya que se generan pérdidas de posibilidades de inversión, crecimiento y desarrollo.

#### Commercial Value of Unlicensed Software Use (in Billions)

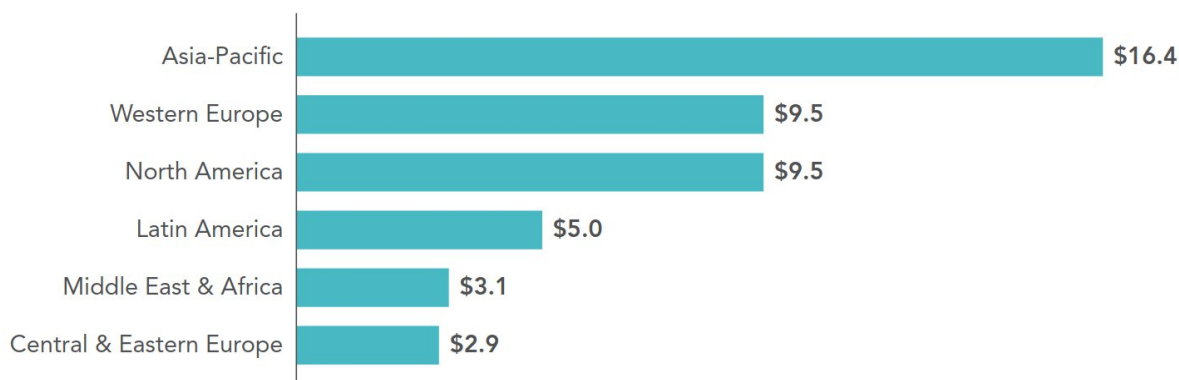


Gráfico N°3.

Fuente: BSA Global Software Survey, June 2018.

## 2.5. Beneficios del FOSS

Los beneficios principales del software libre son las cuatro libertades fundamentales de las que se habló en su definición. Estas son:

- Libertad de usar el programa, con cualquier propósito.



- Libertad de acceder y analizar el código, y cambiarlo a placer.
- Libertad de copiado y redistribución para ayudar a otros.
- Libertad de distribuir copias modificadas y mejorar el programa, para que la comunidad se beneficie de los cambios.

Por lo tanto, nos brinda libertades para poder ejercer apropiación del conocimiento y por ello tiene una gran cantidad de beneficios comparados con el software privativo, entre ellas se encuentran:

1. Crea personas libres, independientes y críticas. El uso de programas con licencia gratuita elimina la dependencia y ofrece libertad para elegir el programa que más nos agrade y que mejor se adapte a nuestras necesidades.
2. Reduce costos de equipos. La ventaja del software libre en relación al hardware es que no requiere de equipos tan potentes y es posible que funcione en la mayoría de las computadoras que existen en el mercado.
3. Reduce costos de licencias.
4. Podemos trabajar con las mismas herramientas que en un software privativo y de forma legal.
5. Puede ofrecer mayor seguridad, ya que los principales ataques se realizan a software privativos, por ser más populares. Además, al poder revisar el código, la comunidad puede informar y arreglar los problemas que encuentre.
6. Existe menos riesgo de que el software utilizado tenga “backdoors” (puertas traseras) que filtren la información, ya que, al tener acceso al código, este puede ser revisado por especialistas.
7. Crea comunidades para compartir conocimiento. Al compartir material educativo libre estamos compartiendo conocimiento y formando comunidades de aprendizaje en las que se valoran las experiencias de éxito.
8. Permite profundizar en la informática. El software libre permite conocer cómo trabajan los programas y fomenta el interés, la investigación y el aprendizaje de nuevos conceptos en este campo.
9. Permite el trabajo en cualquier entorno. Trabajando en contextos de software libre, más que enseñar a utilizar un producto se enseña a utilizar una tecnología, por lo que el usuario estará preparado para utilizar otros entornos cuando sea necesario.
10. Permite ver más allá. El software privativo crea una burbuja alrededor del usuario que no le permite vislumbrar lo que hay más allá. El uso del Software libre permite romper esa burbuja y conocer las alternativas disponibles que nos brindan posibilidades infinitas para realizar aquello que deseamos.

## 2.5.1. Calidad en el FOSS

### 2.5.1.1. ¿Qué es la calidad de Software?

La calidad de software es *“La capacidad de una aplicación Software de ajustarse a los requerimientos funcionales y no funcionales”*, además de solucionar cabalmente los requerimientos de los usuarios, un software de calidad debe ser fácilmente extensible, debe cumplir con ciertos lineamientos de seguridad, debe ser fácil de utilizar para los usuarios y debe tener un funcionamiento confiable y estable.

El proceso de aseguramiento de la calidad del software, según Pressman, en su libro “Ingeniería de Software” involucra:

1. Un enfoque de Gestión de Calidad.
2. Métodos y Herramientas de ingeniería de software efectiva.
3. Revisiones técnicas formales que se aplican durante todo el ciclo de vida del Software (Análisis, Diseño, Codificación, etc.).
4. Modelos y una estrategia de pruebas multiescala.
5. El control de la documentación del software y de los cambios realizados.
6. Un procedimiento que asegure un ajuste a los estándares de desarrollo del software.
7. Mecanismos de medición y de generación de informes.

Por último, como bien es sabido, “*Lo que no se puede medir, no se puede controlar*” algo imprescindible para controlar y evaluar la calidad del software son las métricas, las más reconocidas están incluidas en los estándares de calidad de Software como la norma ISO-9126.

#### 2.5.1.2. Mejoras de la comunidad al FOSS

Las comunidades de software contribuyen a la formación de profesionales en tecnología y estimulan que cualquiera pueda beneficiarse de los avances y las innovaciones introducidas.

La manera clásica de desarrollar proyectos de Software Libre es que un grupo de desarrolladores, al inicio de un número moderado de miembros, genera un producto de software que considera que será de utilidad para la comunidad. Posteriormente pueden subir su código fuente a un repositorio público, e invitar a la comunidad de programadores con determinado perfil a revisar y extender dicho código. Los proyectos de Software Libre más exitosos llegan a tener una comunidad activa de cientos o miles de personas alrededor de todo el mundo colaborando en las revisiones y las mejoras al software. Esta característica en particular hace que sea difícil el aplicar procesos y métricas que permitan asegurar la calidad del software resultante de la mayoría de los proyectos.

Por eso mismo en el 2011 la consultora *Coverity*, junto al Departamento de Seguridad Nacional de los Estados Unidos, publicaron un informe llamado “*Coverity Scan 2011 Open Source Integrity*” en donde se trató este tema.

El estudio realizado por la administración y por la firma Coverity se inició en 2006 y es uno de los más completos que se encuentran a la fecha, donde los investigadores analizaron 300 millones de líneas de código propietario y 37 millones de líneas de código libre. El resultado: el código del software libre y abierto tiene al menos la misma calidad que el código propietario, si es que esta no es mayor en ciertos aspectos. Una de las conclusiones de por qué sucede esto es que la gente que participa en estos proyectos suele tener niveles de conocimiento avanzado y, por lo tanto, generan código de calidad. A diferencia de algunas empresas privadas, en donde se pueden encontrar novatos programando.

### 2.5.1.3. ¿Qué es la ISO 27001?

La norma ISO 27001 es una norma internacional creada por la Organización Internacional de Normalización, ISO, que ayuda a reducir riesgos de carácter informático en las organizaciones. Permite el aseguramiento, la confidencialidad e integridad de los datos y de la información, así como de los sistemas que la procesan.

El estándar ISO 27001 para los Sistemas Gestión de la Seguridad de la Información permite a las organizaciones la evaluación del riesgo y la aplicación de los controles necesarios para mitigarlos o eliminarlos.

Con la implantación del sistema de gestión de la seguridad de la información se trata de proteger la información que tiene carácter confidencial tanto de ataques de piratería, como de violación de confidencialidad o pérdida de datos informáticos.

La norma ISO 27001 proporciona un modelo para la creación, documentación e implantación de un sistema de gestión de la seguridad de la información (SGSI) y se integra perfectamente con otros sistemas de gestión como son los sistemas de gestión de la calidad, bajo la norma ISO 9001 y los sistemas de gestión ambiental bajo la norma ISO 14001.

#### Ventajas:

- Reducir el riesgo de pérdida, robo o corrupción de información.
- Permite establecer una metodología de gestión de la seguridad clara.
- Mejora la imagen de la empresa, diferenciándose con respecto a la competencia.
- Permite a la empresa ganar cuota de mercado gracias a la confianza que genera entre los clientes y socios estratégicos.
- Es posible continuar con las operaciones necesarias del negocio tras incidentes de gravedad.
- Da la posibilidad a la empresa de medir la eficacia de su sistema de gestión de acuerdo con normas nacionales e internacionales a través de la certificación de terceros.
- Establece los cimientos para poder mejorar continuamente los procesos internos.
- Refuerza la habilidad de la organización para alcanzar los objetivos estratégicos, en este caso la seguridad de la información.

Cuando se realiza una auditoría para certificar una norma ISO o si ya cuentas con una, tener un software pirata genera una no conformidad grave la cual además de anular tu certificado, le quitara a tu empresa credibilidad y pudiera afectar tu reputación.

Eso sin contar el impacto económico de tu empresa al tener alguna demanda y cargos legales o incluso perder información, gracias a todos los huecos de seguridad que existen en software pirata.

## 2.6. ¿Por qué eligen entonces lo pirateado al FOSS?

A pesar de la existencia de las alternativas FOSS, la mayoría de los usuarios finales prefiere utilizar software propietario, incluso si este es pirata a pesar de todos los riesgos que conlleva.

Una de las principales razones es el marketing. Aunque el software libre es fácil de conseguir, las personas no lo conocen debido a las agresivas campañas promocionales realizadas por las empresas creadoras de sistemas privativos, que se encargan de publicitar sus productos para que poder adoptar la mayor parte del mercado posible.

Otra desventaja de los sistemas libres es el miedo, incertidumbre y duda que se crea alrededor de ellos (fenómeno que en inglés se denomina FUD). La mayoría de las personas que han escuchado, por ejemplo, algo sobre *Linux*, piensan que la dificultad de uso es elevada y al estar acostumbradas a las soluciones que conocen tienen gran resistencia a utilizarlo. Por lo tanto, se debería educar a las personas sobre este tema para que los usuarios no generen dependencia hacia los softwares privativos, ya que normalmente en las escuelas o en las universidades se enseñan a utilizar este tipo de software. Por ejemplo, en las escuelas secundarias se enseña a utilizar el paquete de *Microsoft Office* y en carreras universitarias como en Diseño Gráfico se enseña a usar el paquete de *Adobe*, sin explicar que existen otras alternativas.

Por lo general el software libre es más utilizado por especialistas o usuarios con conocimientos más técnicos, los cuales no tienen miedo de cambiar de plataformas y aprovechan al máximo los beneficios del FOSS. Los usuarios comunes solo quieren utilizar una computadora que le resuelva o facilite sus tareas, por lo tanto, utilizan el software preinstalado que viene en su ordenador y buscan herramientas fáciles de usar, instalar y conseguir.

Desde el punto de vista empresarial existe un sesgo conservador, en el cual prefieren utilizar software que todos conocen y saben cómo funciona. Esto es óptimo en el caso de pagar una licencia ya que además tendrán soporte y garantía de que el programa se adapta a sus necesidades. En caso de no tener la posibilidad de pagar la licencia, el uso de software pirata, no debería ser una opción a considerar ya que, en el largo plazo, esta decisión resultará más costosa, por todas las razones expuestas anteriormente.

## 2.7. Relación del FOSS con la economía naranja

Las industrias creativas y culturales, lo que se conoce como Economía Naranja, es el conjunto de actividades que, de manera encadenada, permiten que las ideas se transformen en bienes y servicios culturales, cuyo valor está basado en la propiedad intelectual. La Economía Naranja tiene al talento y la creatividad como insumos principales para generar un desarrollo urbano sostenible. Tiene una gran capacidad de generar empleo, riquezas e impacto en la calidad de vida de las personas, tanto es así que en la clasificación que realizó el Centro Internacional de Comercio en 2012, la Economía Naranja representa la quinta mercancía más transada del planeta y en la actualidad debe ser una de las actividades más importantes, debido a su crecimiento sostenido.

Unas de las actividades que forman parte de este modelo son los medios digitales, dentro de los cuales se encuentran los Videojuegos, contenidos interactivos, audiovisuales, creación de software, creación de aplicaciones y animación. La creación de software es parte de la Economía Naranja al ser una actividad centrada en la generación de propiedad intelectual. El FOSS podría ser un gran benefactor dentro de este modelo ya que aspira a que el conocimiento se convierta en un producto que pueda estar al alcance de todos y alienta a sus usuarios a la creación e innovación a partir de las creaciones anteriores.

Por lo tanto, la adopción de la filosofía del FOSS no es solo una forma de combatir la piratería, sino que también es impulsora de la Economía Creativa o, dicho de otro modo, es catalizador de industrias creativas que mejoran la calidad de vida de las personas y permiten a países sin recursos naturales poder exportar software, diversificar su economía y competir a nivel internacional. Japón, Corea del Sur e India son evidencias de que la economía cultural es capaz de desarrollar económicamente un país hasta un nivel competitivo internacionalmente.

### 3. Conclusión

En conclusión, la piratería del software es un problema que llega a todas partes del mundo, produciendo grandes pérdidas de dinero a las compañías que lo producen. Además, conlleva muchos peligros a sus usuarios ya que podrían estar sufriendo robo de información, infringiendo leyes, entre otros.

La piratería surge principalmente por la resistencia que tienen los usuarios a pagar por las licencias de los productos, ya sea porque consideran que son caros o porque las economías de sus países no les permiten acceder a estos niveles de precios. Como los usuarios necesitan de estos programas para trabajar y crecer profesionalmente, optan por adquirirlos por la vía ilegal. Esto se ve reflejado en los principales tipos de software que son pirateados (Office, Adobe, etc.).

La mejor alternativa que se encuentra hoy en día es la utilización de software libre que permite realizar las mismas tareas con costos menores y productos de similar calidad. El problema del software libre es que no es conocido o los usuarios piensan que son difíciles de utilizar. Por lo tanto, se debería educar a la población, para dar a conocer las ventajas y el potencial que este ofrece.

## 4. Bibliografía

1. Bautista, C. E. B., Fernandez, E. L., Lopez, J. A. G., & Vargas, V. M. B. (s/f). CALIDAD EN EL SOFTWARE LIBRE TÍTULO. 3.
2. Boulanger, A. (2005). Open-source versus proprietary software: Is one more reliable and secure than the other? IBM Systems Journal, 44(2), 239–248. <https://doi.org/10.1147/sj.442.0239>
3. Cosovanu, C. (2006). Open Source Software in Eastern Europe and Other Emerging Markets: The Moral Alternative to Piracy? 12.
4. Dalle, J.-M., & Jullien, N. (s/f). OPEN-SOURCE vs. PROPRIETARY SOFTWARE. 16.
5. ECONOMÍA NARANJA | Regional Naranja. (s/f). Recuperado el 3 de junio de 2022, de <https://www.regionnaranja.org/economiaNaranja>
6. Gramstad, A. R. (2016). Piracy in Commercial vs. Open-Source Software Competition. SSRN Electronic Journal. <https://doi.org/10.2139/ssrn.2743290>
7. Home. (s/f). SIIA. Recuperado el 3 de junio de 2022, de <https://www.siiia.net/>
8. Home—2018 BSA Global Software Survey. (s/f). BSA | 2018 Global Software Survey. Recuperado el 3 de junio de 2022, de <https://gss.bsa.org>
9. IDC - Search Results. (s/f). IDC: The premier global market intelligence company. Recuperado el 3 de junio de 2022, de [https://www.idc.com/search/simple/perform\\_.do?query=piracy&page=1&hitsPerPage=25&sortBy=RELEVANCY&lang=English&srchIn=ALLRESEARCH&src=&athrT=10&cmpT=10&pgT=10&trid=121075086&siteContext=IDC](https://www.idc.com/search/simple/perform_.do?query=piracy&page=1&hitsPerPage=25&sortBy=RELEVANCY&lang=English&srchIn=ALLRESEARCH&src=&athrT=10&cmpT=10&pgT=10&trid=121075086&siteContext=IDC)
10. Machado, F. S., Sainam, P., Raghu, T. S., & Sinha, R. (s/f). Software Piracy in the Presence of Open Source Alternatives. Journal of the Association for Information Systems, 18(1), 21.
11. Peterson, S. K. (s/f). What's the difference between open source software and free software? | Opensource.com. Recuperado el 29 de mayo de 2022, de <https://opensource.com/article/17/11/open-source-or-free-software>
12. Platzi. (2018, octubre 26). Qué es la Economía Naranja (y qué no) | PlatziLive. <https://www.youtube.com/watch?v=fs31Woz8f80>
13. Pressman, R. S. (2013). Ingeniería del software: Un enfoque práctico. [http://www.ingebook.com/ib/NPcd/IB\\_BooksVis?cod\\_primaria=1000187&codigo\\_libro=4272](http://www.ingebook.com/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=4272)
14. Sen, R. (2007). A Strategic Analysis of Competition Between Open Source and Proprietary Software. Journal of Management Information Systems, 24(1), 233–257. <https://doi.org/10.2753/MIS0742-1222240107>
15. The Open Source Definition | Open Source Initiative. (s/f). Recuperado el 29 de mayo de 2022, de <https://opensource.org/osd>
16. What is Free Software? - GNU Project—Free Software Foundation. (s/f). Recuperado el 29 de mayo de 2022, de <https://www.gnu.org/philosophy/free-sw.html>
17. Why Open Source Misses the Point of Free Software—GNU Project—Free Software Foundation. (s/f). Recuperado el 29 de mayo de 2022, de <https://www.gnu.org/philosophy/open-source-misses-the-point.html.en>