Software Libre y Propietario



Antonio Carrillo Ledesma
Karla Ivonne González Rosas

Software Libre y Propietario

Antonio Carrillo Ledesma y Karla Ivonne González Rosas
Facultad de Ciencias, UNAM
http://academicos.fciencias.unam.mx/antoniocarrillo

La última versión de este trabajo se puede descargar de la página:
https://sites.google.com/ciencias.unam.mx/acl/en-desarrollo
http://132.248.181.216/acl/EnDesarrollo.html

2025, Versión 1.0α[1]

[1]El presente trabajo está licenciado bajo un esquema Creative Commons Atribución CompartirIgual (CC-BY-SA) 4.0 Internacional. Los textos que componen el presente trabajo se publican bajo formas de licenciamiento que permiten la copia, la redistribución y la realización de obras derivadas siempre y cuando éstas se distribuyan bajo las mismas licencias libres y se cite la fuente. ¡Copia este libro! ... Compartir no es delito.

Índice

1 Introducción **3**
1.1 Sistemas Operativos . . . 3
1.2 Software Propietario y Libre . . . 5
1.2.1 Software Propietario . . . 5
1.2.2 Software Libre . . . 6
1.3 ¿Qué tan Seguro es el Software Libre? . . . 12
1.4 Agradecimientos . . . 16

2 Software Libre y Propietario **17**
2.1 Software Propietario . . . 20
2.2 Software Libre . . . 22
2.3 Seguridad del Software . . . 29
2.4 Tipos de Licencias . . . 32
2.4.1 Licencias Creative Commons . . . 38
2.4.2 Nuevas Licencias para Responder a Nuevas Necesidades 40
2.5 Implicaciones Económico-Políticas del Software Libre . . . 43
2.5.1 Software Libre y la Piratería . . . 43
2.5.2 ¿Cuánto Cuesta el Software Libre? . . . 44
2.5.3 La Nube y el Código Abierto . . . 47
2.5.4 El Código Abierto como Base de la Competitividad . . 50
2.5.5 Software Libre en Empresas y Corporaciones . . . 51
2.6 Código Abierto y las Organizaciones Internacionales . . . 59
2.6.1 Las Naciones Unidas y el Código Abierto . . . 60
2.6.2 La Comisión Europea se Compromete a Liberar Todo el Software que Pueda Beneficiar a la Sociedad . . . 61
2.6.3 El Software Codigo Abierto, un sector de 7.700 millones de dólares anuales en 2024 . . . 64

3 Seguridad y Privacidad en el Software **67**
3.1 Consideraciones de Seguridad . . . 68
3.2 Consideraciones Sobre la Privacidad . . . 76
3.3 Software Libre e Infraestructura Crítica . . . 82

4 Consideraciones y Comentarios Finales **89**
4.1 El Cómputo en Instituciones Educativas . . . 92
4.2 Integración del Cómputo en Ciencias e Ingenierías . . . 96

4.3 Ventajas, Desventajas y Carencias del Software Libre 97
4.4 Comentarios Finales 98

5 Bibliografía **101**

1 Introducción

Ante los retos que el vertiginoso y dinámico cambio informático que enfrenta el mundo globalizado y ante las exigencias de la sociedad de la información, se requiere por parte de los usuarios un manejo mínimo de las Tecnologías de la Información y de la Comunicación (TIC). Las TIC son todos aquellos recursos, herramientas y programas que se utilizan para procesar, administrar y compartir la información mediante diversos soportes tecnológicos, tales como: computadoras, teléfonos móviles, televisores, reproductores portátiles de audio y video o consolas de juego, entre otros.

El papel de las TIC en la sociedad es muy importante porque ofrecen muchos servicios como: correo electrónico, búsqueda de información, banca en línea, descarga de música y video, comercio electrónico, etc. Por esta razón las TIC han incursionado fácilmente en diversos ámbitos de la vida, entre ellos: el de la educación, finanzas, mercadotecnia, medicina, ingeniería, ciencias y un sinfín de áreas.

1.1 Sistemas Operativos

Actualmente tenemos 3 grandes sistemas operativos en el mercado[1]:

- Windows
- Unix
- GNU[2]/Linux

De los cuales, sus dignos representantes son: Windows, macOS, iOS, Android, Chrome OS y GNU/Linux con todas sus diferentes distribuciones[3]. Y sin temor a equivocarnos aseguramos que Android es la distribución de GNU/Linux más popular e iOS es el más popular de los UNIX.

[1] Cuotas de mercado de diferentes sistemas operativos:

https://gs.statcounter.com/os-market-share/desktop/worldwide
https://netmarketshare.com

[2] GNU -es un acrónimo recursivo de «GNU no es UNIX»- es un sistema operativo de Software libre, es decir, respeta la libertad de los usuarios. El sistema operativo GNU consiste en paquetes de GNU además de Software libre publicado por terceras partes con distintas licencias que conforman una distribución.

[3] Una distribución de Linux es un sistema operativo compuesto por el Kernel de Linux, herramientas GNU, Software adicional y un administrador de paquetes. También puede

¿Qué Sistema Operativo Usar? ¿Apple o Microsoft? ¿Windows o Linux? ¿Android o iOS? Son preguntas frecuentes que todos nos hemos hecho alguna vez y es que elegir un sistema operativo, una computadora o un dispositivo móvil no es tan simple. Al menos no lo era años atrás. En la actualidad las diferencias entre sistemas operativos de escritorio son cada vez menos, hasta el punto que prácticamente cualquier servicio Online es compatible con Windows, Mac, GNU/Linux y las principales firmas de Software crean aplicaciones para las tres plataformas principales, salvo excepciones.

Poco tendremos que decir del sistema operativo de Apple, Mac o iOS, ya que son los sistemas operativos más bonitos y que mejores resultados han dado a todos los usuarios que los han probado. Mac es un sistema pensado para los profesionales de los sectores que necesitan de un equipo de cómputo que sea capaz de proveer programas específicos para: desarrolladores, programadores, diseñadores, periodistas, fotógrafos, músicos, DJ's y muchos más empleos que se benefician de este sistema operativo.

Después tenemos a Windows, un sistema operativo versátil pensado principalmente para un uso doméstico, aunque eso no quita que muchas empresas utilicen Windows en sus equipos de cómputo ya que es un sistema operativo que puede dar muy buenos resultados en este aspecto.

Sin embargo, llegamos a Linux, el gran desconocido por muchos. Un sistema operativo mucho más versátil que Windows y que puede ser igual o más profesional que Mac. Sin embargo, la ventaja que tienen estos dos sistemas operativos, es que vienen ya preparados y configurados para el tipo de mercado al que van dirigidos, pero GNU/Linux no. Esto es una ventaja y una desventaja al mismo tiempo, ya que si tenemos práctica, podemos hacer que el sistema operativo se adapte a nuestras necesidades sin problemas, además es Software libre.

---

incluir un servidor de pantalla y un entorno de escritorio que se utilizarán como sistema operativo de escritorio normal. El término es distribución de Linux (o distribución en forma abreviada) porque una entidad como Debian o Ubuntu 'distribuye' el Kernel de Linux junto con todo el Software y las utilidades consideradas por cada entidad como necesarias (como administrador de red, administrador de paquetes, entornos de escritorio, etc.) para que pueda ser utilizado como sistema operativo. Sus distribuciones también asumen la responsabilidad de proporcionar actualizaciones para mantener el Kernel y otras utilidades.

Entonces, Linux es el Kernel, mientras que la distribución de Linux es el sistema operativo. Esta es la razón por la que también se les conoce como sistemas operativos basados en Linux (hay otros Kernels como son FreeBSD, NetBSD y Hurd).

1.2 Software Propietario y Libre

Con el constante aumento de la comercialización de las computadoras y su relativo bajo costo, las computadoras se han convertido en un objeto omnipresente, ya que estas se encuentran en las actividades cotidianas de millones de usuarios, en formas tan diversas como teléfonos celulares, tabletas, computadoras portátiles y de escritorio, etc.

Las computadoras por sí solas tienen poca utilidad, pero su uso en conjunción con el Software adecuado forman un dúo que nos ha permitido tener los avances de los que actualmente disfrutamos. El Software -sistema operativo y los programas de aplicaciones- son los que realmente generan las soluciones al interactuar uno o más paquetes informáticos con los datos del usuario. También, es común que al comprar una computadora, en el costo total, se integre el del sistema operativo, aplicaciones ofimáticas y de antivirus, sean estos usados por el usuario o no; y en la mayoría de los casos no es posible solicitar que no sean incluidos en el costo de la computadora.

Por otro lado, el Software comercial suele quedar obsoleto muy rápido, ya que constantemente se le agregan nuevas funcionalidades al mismo y estas en general son vendidas como versiones independientes de la adquirida originalmente. Esto obliga al usuario -si quiere hacer uso de ellas- a comprar las nuevas versiones del Software para satisfacer sus crecientes necesidades informáticas y la obsolescencia programada. Por lo anterior y dada la creciente complejidad de los paquetes de cómputo y el alto costo de desarrollo de aplicaciones innovadoras, en muchos casos, el costo total del Software que comúnmente los usuarios instalan -y que no necesariamente usan las capacidades avanzadas del programa, por las cuales el Software tiene un alto costo comercial- en su computadora, suele ser más caro que el propio equipo en el que se ejecutan.

1.2.1 Software Propietario

En entornos comerciales, es posible por parte de la empresa, adquirir y mantener actualizado el Software necesario para sus actividades comerciales, pues el costo del mismo se traslada al consumidor final del bien o servicio que la empresa proporcione. En entornos educativos, de instituciones sin fines lucrativos e incluso, en sector gubernamental, no se cuenta con los recursos necesarios para adquirir y mantener actualizado el Software requerido para todas y cada una de las aplicaciones usadas en las computadoras, ya que

en general, las licencias de uso del Software propietario son asignadas en forma individual a cada computadora y no es fácilmente transferible a otra computadora.

Dado que existe una gran demanda de programas de cómputo tanto de uso común como especializado por nuestras crecientes necesidades informáticas, y por la gran cantidad de recursos económicos involucrados, existe una gran cantidad de empresas que tratan de satisfacer dichas necesidades, para generar y comercializar, además de proveer la adecuada documentación y opciones de capacitación que permita a las empresas contratar recursos humanos capacitados.

Por otro lado, generalmente se deja la investigación y desarrollo de productos computacionales nuevos o innovadores a grandes empresas o Universidades -que cuenten con la infraestructura y el capital humano- con la capacidad de analizar, diseñar y programar las herramientas que requieran para sus procesos de investigación, enseñanza o desarrollo.

Existen hoy en día, una gran cantidad de paquetes y sistemas operativos comerciales de Software propietario que mediante un pago oneroso, permiten a los usuarios de los mismos ser productivos en todas y cada una de las ramas comerciales que involucra nuestra vida globalizada, pero el licenciamiento del uso de los programas comerciales es en extremo restrictivo en su uso y más en su distribución.

1.2.2 Software Libre

El Software libre son programas de cómputo -sistema operativo, paquetes de uso común y especializados-, desarrollados por usuarios y para usuarios que, entre otras cosas, comparten el código fuente y el programa ejecutable otorgando de esa forma la libertad para estudiar, adaptar y redistribuir a quien así lo requiera, el programa y todos sus derivados.

El Software libre es desarrollado por una creciente y pujante comunidad de programadores, usuarios y empresas desarrolladoras de Software y Hardware que tratan de poner la mayor cantidad de programas a disposición de todos los interesados, de forma que permiten al usuario promedio sacar el mayor provecho del equipo de cómputo que use, sin importar el sistema operativo subyacente (pero es mejor que todo el Software de un equipo incluyendo el sistema operativo sea Software libre).

¿Qué es el Software Libre? La definición exacta y sus diversas variantes se verán en la siguiente sección, pero podemos entender su esencia a través de los documentos de la fundación para el Software libre. El Software libre concierne a la libertad de los usuarios para ejecutar, copiar, distribuir, cambiar y mejorar el Software:

0. La libertad de usar el programa, con cualquier propósito.
1. La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades.
2. La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo.
3. La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.

La lista de proyectos de este tipo es realmente impresionante (véase [9], [8] y [6]). Algunos han conseguido un uso y alta calidad, por ejemplo el compilador GCC (véase [11]), el Kernel de Linux (véase [12]) y el sistema operativo Debian GNU/Linux y Android. Mientras que otros proyectos han caído en el olvido, pero de la gran mayoría se tiene copia del código fuente que permitiría a quienes estén interesados en dicho proyecto poder reusarlo y en su caso ampliarlo.

La característica más importante que aparece típicamente en un proyecto de este tipo, es que un conjunto de personas separadas a gran distancia, sean capaces, a través de la Web, de los E-mail y de foros de aunar sus esfuerzos para crear, mejorar y distribuir un producto, de forma que todos ellos se benefician unos de otros. Evidentemente, la mayor parte del peso recae en los desarrolladores, pero también es necesaria una difusión para que los usuarios documenten, encuentren errores, hagan foros de discusión, etc.

Si bien, el Software libre no es más seguro (en el sentido de invulnerable) que el propietario, la diferencia estriba en que el código fuente en el Software libre está disponible para todos y cualquiera puede aportar una solución, y por lo general al poco tiempo de detectarse una vulnerabilidad (a veces en cuestión de horas) se puede disponer de una solución para la misma. Además, al tener acceso al código fuente se puede detectar fácilmente si alguien introdujo código malicioso a una determinada aplicación.

¿Por qué se Interesan los Autores, Alumnos y Profesores Universitarios en el Software Libre? La ventaja principal es porque bajo el Software libre subyace la idea de compartir conocimiento y favorecer la existencia de nuevas ideas[4]; y ¿qué es investigar y enseñar?, sino crear conocimiento y procurar que los alumnos aprendan e incluso vayan más allá de lo aprendido. Se comparte la idea, que el espíritu del Software libre es similar al que debería reinar en las instituciones educativas:

- Porque así no se condiciona a los estudiantes a usar siempre lo mismo.
- No se fomenta la piratería en los estudiantes y se evita pagar licencias que no son necesarias al existir alternativas gratuitas.
- Es mucho más seguro ya que el Software libre es público y se puede ver qué hace exactamente sin recelos.
- Se ofrece libertad de elección a los estudiantes y profesores al no limitarlos a usar una solución determinada, ampliando sus opciones y permitiendo un mayor aprendizaje.

Concretando estas ideas, profesores e investigadores necesitan herramientas para la investigación y docencia y estas deben tener una calidad mínima y ser fácilmente distribuibles entre los alumnos. En muchos casos las compañías desarrolladoras y distribuidoras de programas de cómputo no han sabido ofrecer sus productos con la flexibilidad adecuada para las labores docentes o, en otros casos, los productos desarrollados no tienen la calidad esperada.

El Software libre es aún joven, pese a las decenas de miles de proyectos actuales (véase [6] y [7]) -en los que se trabaja constantemente en mejorar la parte computacional de los algoritmos involucrados en el proyecto, haciendo y puliendo interfaces gráficas, generando ayuda en línea así como la documentación necesaria para que usuarios noveles y avanzados usen la mayor cantidad de opciones programadas- existen muchas otras necesidades profesionales y de investigación que requieren el desarrollo innovador de programas de cómputo para automatizarlas y hacerlas eficientes. Esto queda plasmado

[4] ¿Por qué el Software creado con dinero de los impuestos no se publica como Software Libre?

¡El código pagado por los ciudadanos debería estar disponible para los ciudadanos y el mismo gobierno!

en las decenas de proyectos que a diario son registrados en las páginas especializadas en busca de difusión y apoyo para su proyecto (véase [6] y [7]).

En los últimos años, muchos proyectos han pasado de ser simples programas en línea de comandos a complejas aplicaciones multiplataforma -se ejecutan en distintos sistemas operativos como son Windows, Linux, Unix, Mac OS, Android- con ambientes gráficos multimedia que en muchos casos han superado a sus contrapartes comerciales -por ejemplo los navegadores Web-. Para muestra de este maravilloso avance, tomemos el proyecto del sistema operativo Android, que actualmente se ejecuta en millones de equipos -como celulares, tabletas, electrodomésticos, etc.- y en los cuales se pueden descargar miles de aplicaciones y está soportado por una gran cantidad de usuarios y empresas comerciales como Google, IBM y últimamente Microsoft -que años atrás era acérrima enemiga del Software libre-.

El Software libre ha logrado desplazar a muchos de sus competidores por sus múltiples bondades y bajo costo de desarrollo -es el caso de Windows Phone que fue reemplazado por Android de Google-, al reusar miles de aplicaciones ya existentes que usan Software libre y permitir desarrollar otro tanto de aplicaciones bajo una plataforma que se ejecuta en los más diversos procesadores. Además, el uso de Software libre y su posibilidad de ampliarlo y/o especializarlo según sea necesario, ha permitido crear de forma cada vez más rápida y confiable; para poner a disposición de un gran público programas de uso común, así como especializado que satisfagan las nuevas necesidades de los usuarios.

Software Libre en Ciencia y Educación Algunos puntos y reflexiones sobre porqué se considera que es interesante el Software libre en Ciencia y Educación son:

- Accesible a todo el mundo aunque no sea rentable su desarrollo: El Software libre entre sus libertades permite que se pueda ejecutar por terceros, copiarlo, distribuirlo y estudiarlo/modificarlo. Eso hace que si en ciencia se usa Software libre el acceso a esos programas no suponga una barrera (se puede distribuir y ejecutar).

- Muchos de los desarrollos en el campo de la accesibilidad se realizan en universidades y son distribuidos como Software libre: Aunque no sea rentable muchas veces el desarrollo de herramientas de accesibilidad (no sea algo monetizable) en la universidad se consigue escapar a esa la

lógica capitalista de solamente invertir en lo que pueda ofrecer beneficio económico.

- Transparencia: En ciencia es importante ver las costuras para comprobar si es verdad lo que se afirma, tener acceso al código fuente del Software empleado permite poder estudiarlo por si realiza algún cálculo mal.

- Propicia el espíritu crítico: Si no tienes acceso a las revistas o el acceso es privativo para los bolsillos de mucha gente no puedes comprobar la información. Se nos pide que seamos críticos con la información que se nos da del mundo científico pero no podemos entrenar el espíritu crítico sin acceso al conocimiento libre.

- Caramelos con droga en la puerta del colegio: Muchas empresas buscan introducir en los colegios, institutos y universidad su Software. Ofrecer un programa que permita trabajar y genere una dependencia quedando los datos muchas veces en las nubes (ordenadores de otras personas). Un ejemplo es Microsoft con Office 365. Dando cuentas gratuitas durante un tiempo para que se use su Software. Otro ejemplo podría ser Unity3D en vez de por ejemplo Godot.

- Especificaciones de protocolos abiertos VS cerrados: Gracias a que los protocolos TCP/IP, HTTP, POP, SNMP, DHCP, etc. son abiertos es posible construir herramientas por cualquier con conocimientos de programación. Con protocolos cerrados solamente quienes tuvieran acceso a las especificaciones podrían desarrollar y conocer cómo funcionan.

- Uso de estándares: Existiendo un estándar para documentos ofimáticos (procesador de textos, hoja de cálculo, presentaciones, etc.) algunas empresas como Microsoft se empeñan en ir con su propio formato y estándar en vez de sumarse a que sea más sencillo ir a una y que el usuario pueda optar por que herramientas usar para editar o trabajar con documentos ofimáticos.

- Software libre para la Ciencia ciudadana: Un ejemplo en el que es importante la colaboración ciudadana es el cambio climático y la defensa medioambiental del territorio. Desde imvec.tech usan herramientas de Software libre para medición y monitorización de contaminación.

Software Libre: Beneficios Más Allá de la Informática El uso de las tecnologías de código abierto supone cultivar el conocimiento y la puesta en valor de la libertad individual, lo personal y lo privado, todo ello sin menospreciar lo público y la construcción de una sociedad. El movimiento del Software libre, con Linux a la cabeza, ha capitaneado durante décadas planteamientos para un cambio en el modo de producción: el abaratamiento de costes empresariales para grandes y pequeños, el trabajo en línea, el desarrollo de Software y Hardware a pequeña escala, el replanteamiento del negocio informático, el sistema de normas éticas que rigen los grupos, la documentación abierta, etc.

Todo ello derivado de una simple idea: la libertad. Ha sido la clave que ha llevado a todo este movimiento hacia una autonomía y motivación que pocas veces se ve en otros sectores. El Open Source está lleno de alternativas con la libertad como pilar y consecuencia filosófica, de hecho muchos Forks (derivaciones) de proyectos aparecen cuando la disputa sobre la misma toma relevancia. Esta manera de hacer las cosas debería trasladarse directamente a la sociedad promoviendo esos valores para evitar caer en un mundo despótico que toma fuerza a pasos agigantados.

No estaría mal promover la comprensión de las licencias libres. Leer y entender una licencia GPL, MIT o BSD es infinitamente más sencillo y rápido que hacerlo con otras, siendo unas normas fáciles de cumplir porque encajan con un modelo ético y práctico comprensible por muchos.

Podríamos decir que GPL, BSD y MIT son las Constituciones que vertebran todo el movimiento del Software Libre, cosechando derechos de uso y logros como el ahorro o la legítima copia privada. Lo mismo podría ocurrir con las licencias Creative Commons para cierto contenido, un arma poderosa en el sector divulgativo que está poco extendida por desconocimiento y el Status Quo de la propiedad intelectual.

La cooperación libre y voluntaria supuso un éxito hasta ahora pero está siendo amenazada por la dinámica actual de la Web, donde la centralización de las principales plataformas supone estar bajo el yugo de normas que rastrean con lupa y cambian sus términos con demasiada frecuencia. Ya ni digamos cuando eso se junta con el ansia de monetización: si quieres dinero más te vale no cabrear a los anunciantes o no tener un Strike por uso de contenido que podría ser reclamado.

Los claroscuros de este sistema hace que los creadores cada vez hagan menos de forma libre y altruista, y ello podría verse potenciado por las búsquedas con inteligencia artificial, lo que apunta a un empobrecimiento

del contenido cultural fresco y renovador. Las polémicas con GitHub Copilot o ChatGPT sobre el entrenamiento de las IAs hace sobrevolar una vez más la cuestión del Copyright y el uso legítimo de los resultados brindados por éstas, lo que también condiciona la creación.

La libertad de expresión, de uso, de creación, de modificación ... esas cuestiones llevan irremediablemente a una libertad de pensamiento y acción, a una adaptación creativa que puede ser la motivación para romper moldes en todos los aspectos. El código abierto y sus licencias son, por tanto, un beneficio personal y social mucho más grande que el simple hecho de usar Linux o Software libre. El contenido despreocupado, que no prioriza el dinero y el posicionamiento/visualizaciones, se vuelve esencial para el inconformismo.

1.3 ¿Qué tan Seguro es el Software Libre?

No es ningún secreto que el sistema operativo que elijas es un determinante clave de su seguridad (no sólo en internet, también la privacidad de tus datos en el equipo que usas). Después de todo, el sistema operativo es el Software más crítico que se ejecuta en nuestra computadora o dispositivo inteligente: administra su memoria y procesos, así como todo su Software y Hardware. El consenso general entre los expertos es que Linux es un sistema operativo altamente seguro, posiblemente el sistema operativo más seguro por diseño. Examinaremos aquí algunos factores clave que contribuyen a la sólida seguridad de Linux y veremos el nivel de protección contra vulnerabilidades y ataques que Linux ofrece a los administradores y usuarios.

Seguro por Diseño cuando se trata de seguridad, los usuarios de Linux tienen una clara ventaja sobre sus contrapartes que usan Windows o MacOs. A diferencia de los sistemas operativos propietarios, Linux, en muchos sentidos, tiene seguridad integrada en su diseño central. El sistema operativo de código abierto cada vez más popular es de alta flexibilidad, configurable y diverso. También implementa un modelo estricto de privilegios de usuario y ofrece una selección de defensas de seguridad de Kernel integradas para protegerse contra vulnerabilidades y ataques. La transparencia del código fuente de Linux significa que las vulnerabilidades en él, que son inevitables hasta cierto punto en cualquier sistema operativo, casi siempre son de corta duración. Echemos un vistazo más de cerca a cada uno de estos factores y cómo contribuye a la seguridad anunciada de Linux.

La Ventaja de la Seguridad de Código Abierto el código fuente de Linux se somete a una revisión exhaustiva y constante por parte de los miembros de la vibrante comunidad global de código abierto y, como resultado de este escrutinio, las vulnerabilidades de seguridad de Linux generalmente se identifican y eliminan muy rápidamente. Por el contrario, los proveedores propietarios como Microsoft y Apple emplean un método conocido como "seguridad por oscuridad", donde el código fuente se oculta a los extraños en un intento de ocultar las vulnerabilidades de los actores de amenazas. Sin embargo, este enfoque generalmente es ineficaz para prevenir las vulnerabilidades modernas y, en realidad, socava la seguridad del código fuente "oculto" al evitar que personas ajenas identifiquen y reporten fallas antes de que sean descubiertas por actores malintencionados. Seamos realistas: cuando se trata de descubrir errores de seguridad, un pequeño equipo de desarrolladores propietarios no es rival para la comunidad mundial de usuarios-desarrolladores de Linux que están profundamente involucrados en su trabajo tanto para su propio beneficio como para el beneficio de la comunidad.

Un Modelo Superior de Privilegios de Usuario a diferencia de Windows, donde "todo el mundo es administrador", Linux restringe en gran medida el acceso a la raíz a través de un modelo estricto de privilegios de usuario. En Linux, el superusuario posee todos los privilegios, y a los usuarios comunes solo se les otorgan suficientes permisos para realizar tareas comunes. Debido a que los usuarios de Linux tienen pocos derechos de acceso automático y requieren permisos adicionales para abrir archivos adjuntos, acceder a archivos o ajustar las opciones del Kernel, es más difícil propagar Malware y Rootkits en un sistema Linux. Por lo tanto, estas restricciones inherentes sirven como una defensa clave contra los ataques y el compromiso del sistema.

Defensas de Seguridad de Kernel Incorporadas el Kernel de Linux cuenta con una variedad de defensas de seguridad integradas que incluyen Firewalls que utilizan filtros de paquetes en el Kernel, el mecanismo de verificación de Firmware UEFI Secure Boot, la opción de configuración Linux Kernel Lockdown y los sistemas de mejora de seguridad SELinux o AppArmor Mandatory Access Control (MAC). . Al habilitar estas funciones y configurarlas para brindar el más alto nivel de seguridad en una práctica conocida como autoprotección del Kernel de Linux, los administradores pueden agre-

gar una capa adicional de seguridad a sus sistemas.

Seguridad a Través de la Diversidad existe un alto nivel de diversidad posible dentro de los entornos de Linux como resultado de las muchas distribuciones de Linux disponibles y las diferentes arquitecturas de sistema y componentes que presentan. Esta diversidad no solo ayuda a satisfacer los requisitos individuales de los usuarios, sino que también ayuda a protegerse contra los ataques al dificultar que los actores maliciosos elaboren de manera eficiente Exploits que puedan usarse contra una amplia gama de sistemas Linux/Linux. Por el contrario, la "monocultura" homogénea de Windows convierte a Windows en un objetivo de ataque relativamente fácil y eficiente (algo parecido también les pasa a las Mac).

Además de la diversidad de diseño que se ve en Linux, ciertas distribuciones seguras de Linux se diferencian en formas que abordan específicamente las preocupaciones de seguridad y privacidad avanzadas compartidas entre los Pentesters, los ingenieros inversos y los investigadores de seguridad.

Altamente Flexible y Configurable hay muchas más opciones de configuración y control disponibles para los administradores de Linux que para los usuarios de Windows y MacOs, muchas de las cuales se pueden usar para mejorar la seguridad. Por ejemplo, los administradores de sistemas de Linux tienen la capacidad de usar SELinux o AppArmor para bloquear su sistema con políticas de seguridad que ofrecen controles de acceso granulares, proporcionando una capa adicional crítica de seguridad en todo el sistema. Los administradores también pueden usar la opción de configuración Linux Kernel Lockdown para fortalecer la división entre los procesos de la zona de usuario y el código del Kernel, y pueden fortalecer el archivo sysctl.conf, el principal punto de configuración de parámetros del Kernel para un sistema Linux, para darle a su sistema una base más segura.

Linux: Un Objetivo Cada Vez Más Popular Entre los Ciberdelincuentes Linux alimenta la mayoría de los dispositivos y supercomputadoras de alto valor del mundo y la base de usuarios del sistema operativo está creciendo constantemente, y los ciberdelincuentes han tomado nota de estas tendencias. Los autores y operadores de Malware apuntan cada vez más a los sistemas Linux en sus campañas maliciosas. Por ejemplo, en los últimos años han estado plagados de cepas emergentes de Malware para Linux: Cloud

Snooper, EvilGnome, HiddenWasp, QNAPCrypt, GonnaCry, FBOT y Tycoon se encuentran entre las más notorias. Dicho esto, Linux sigue siendo un objetivo relativamente pequeño, con el 96 % del nuevo Malware dirigido a Windows en 2022. Además, el reciente aumento de los ataques de Malware de Linux no es un reflejo de la seguridad de Linux. La mayoría de los ataques a los sistemas Linux se pueden atribuir a configuraciones incorrectas y una administración deficiente, lo que destaca una falla generalizada entre los administradores de sistemas Linux para priorizar la seguridad.

Afortunadamente, a medida que el Malware de Linux continúa siendo cada vez más frecuente y problemático, Linux cuenta con protección integrada contra ataques de Malware a través de su estricto modelo de privilegios de usuario y diversidad de diseño, y hay una selección de excelentes herramientas, Kits de herramientas y utilidades de análisis de Malware e ingeniería inversa que incluyen REMnux, Chkrootkit, Rkhunter, Lynis y Linux Malware Detect (LMD) disponibles para ayudar a los administradores a detectar y analizar Malware en sus sistemas.

Para Tomar en Cuenta la seguridad del sistema operativo que implementa es un determinante clave de su seguridad en internet, pero de ninguna manera es una protección segura contra Malware, Rootkits y otros ataques. La seguridad efectiva depende de la defensa en profundidad, y otros factores, incluida la implementación de las mejores prácticas de seguridad y el comportamiento inteligente internet, juegan un papel central en su postura de seguridad digital. Dicho esto, elegir un sistema operativo seguro es de suma importancia, ya que el sistema operativo es la pieza de Software más crítica que se ejecuta en nuestros dispositivos computacionales, y Linux es una excelente opción ya que tiene el potencial de ser altamente seguro, posiblemente más que su contraparte propietaria, debido a su código de fuente abierta, modelo estricto de privilegios de usuario, diversidad y base de usuarios relativamente pequeña.

Sin embargo, Linux no es una "bala de plata" cuando se trata de seguridad digital: el sistema operativo debe configurarse de manera adecuada y segura, y los administradores de sistemas deben practicar una administración responsable y segura para evitar ataques. Además, es fundamental tener en cuenta que la seguridad tiene que ver con las compensaciones, tanto entre seguridad y facilidad de uso como entre seguridad y facilidad de uso. Los administradores deben configurar sus sistemas para que sean tan seguros

como sea práctico dentro de su entorno. En lo que respecta a la conveniencia, Linux tiene una pequeña curva de aprendizaje, pero ofrece importantes ventajas de seguridad sobre Windows o MacOs.

1.4 Agradecimientos

Este texto es una recopilación de múltiples fuentes, nuestra aportación -si es que podemos llamarla así- es plasmarlo en este documento, en el que tratamos de dar coherencia a nuestra visión de los temas desarrollados.

En la realización de este texto se han revisado -en la mayoría de los casos indicamos la referencia, pero pudimos omitir varias de ellas, por lo cual pedimos una disculpa- múltiples páginas Web, artículos técnicos, libros, entre otros materiales bibliográficos, los más representativos y de libre acceso los ponemos a su disposición en la siguiente liga:

Herramientas
http://132.248.181.216/Herramientas/

Además, la documentación y los diferentes ejemplos que se presentan en este trabajo, se encuentran disponibles en dicha liga, para que puedan ser copiados desde el navegador y ser usados. En aras de que el interesado pueda correr dichos ejemplos y afianzar sus conocimientos, además de que puedan ser usados en diferentes ámbitos a los presentados aquí.

2 Software Libre y Propietario

Con el constante aumento de la comercialización de equipos de cómputo y/o comunicación (teléfonos inteligentes, tabletas, computadoras portátiles y de escritorio, etc.) y su relativo bajo costo, estos equipos se han convertido en objetos omnipresentes en nuestra vida diaria, ya que estos permiten realizar un creciente número de actividades cotidianas de miles de millones de usuarios.

Dichos equipos de cómputo y/o comunicación por sí solos tienen poca utilidad, pero su uso en conjunción con el Software adecuado forman un dúo que nos ha permitido tener los avances de los que actualmente disfrutamos. El Software -sistema operativo y los programas de aplicaciones- son los que realmente generan las soluciones al interactuar uno o más paquetes informáticos con los datos del usuario. También, es común que al comprar un equipo de cómputo y/o comunicación, en el costo total, se integre el del sistema operativo, aplicaciones ofimáticas y de antivirus, sean estos usados por el usuario o no y en la mayoría de los casos no es posible solicitar que no sean incluidos en el costo del equipo.

Por otro lado, el Software comercial suele quedar obsoleto muy rápido, ya que constantemente se le agregan nuevas funcionalidades al mismo y estas en general son vendidas como versiones independientes de la adquirida originalmente. Esto obliga al usuario -si quiere hacer uso de ellas- a comprar las nuevas versiones del Software para satisfacer sus crecientes necesidades informáticas y la obsolescencia programada.

Por lo anterior y dada la creciente complejidad de los paquetes de cómputo y el alto costo de desarrollo de aplicaciones innovadoras, en muchos casos, el costo total del Software que comúnmente los usuarios instalan -y que no necesariamente usan las capacidades avanzadas del programa, por las cuales el Software tiene un alto costo comercial- en sus equipos, suele ser más caro que el propio equipo en el que se ejecutan.

Hoy en día los usuarios disponemos de dos grandes opciones para adquirir el Software necesario para que nuestros equipos funcionen, a saber:

- Por un lado, podemos emplear programas comerciales (Software propietario), de los cuales no somos dueños del Software, sólo concesionarios al adquirir una licencia de uso del Software y nos proporcionan un instalable del programa adquirido. La licencia respectiva es en la gran mayoría de los casos muy restrictiva, ya que restringe su uso a un solo

equipo y/o usuario simultáneamente.

- Por otro lado, existe el Software libre[5], desarrollado por usuarios y para usuarios que, entre otras cosas, comparten los códigos fuente, el programa ejecutable y dan libertades para estudiar, adaptar y redistribuir a quien así lo requiera el programa y todos sus derivados.

Sobre la Obsolescencia Programada Es un conjunto de estrategias deliberadas destinadas a asegurarse que la versión actual de un determinado producto quedará desfasada o inservible en un plazo de tiempo predeterminado. De esta manera, los fabricantes se aseguran que los consumidores se verán obligados a reemplazarlo aunque funcione adecuadamente.

La obsolescencia puede lograrse mediante la introducción de un modelo con características superiores o diseñando intencionadamente un producto para que deje de funcionar correctamente en un plazo determinado. En cualquiera de los dos casos, se espera que los consumidores opten por el nuevo producto de la misma marca. Muchas veces la obsolescencia no es sobre el propio producto sino aplicando restricciones al producto de un competidor con la ayuda de una tercera empresa.

Tipos de Obsolescencia Programada Podemos dividir la obsolescencia programada en 4 tipos:

1- Establecimiento artificial del plazo de duración: Los productos se fabrican con piezas cuya duración tienen una vida útil limitada cuando, si se usaran otras de calidad superior ese plazo se extendería.

2- Actualizaciones de Software: Los desarrolladores de Software sacan nuevas versiones de sus aplicaciones que en un momento determinado dejan de ser compatibles con dispositivos antiguos. En muchos casos se ha podido comprobar que esa incompatibilidad es absolutamente artificial ya que al «engañar» al Software este funcionaba sin problemas.

[5] A veces también se han usado términos como FOSS y FLOSS. Ambas cosas son similares, ya que FOSS (Free and Open Source Software) traducido como "Software de código abierto" y FLOSS (Free/Libre and Open Source Software) "Software libre y de código abierto". Según quienes adoptan estos términos, lo hacen por tener una imparcialidad entre la carga filosófica del Software libre y el aspecto técnico y/o las ventajas que brinda este modelo de desarrollo. Richard Stallman nos invita a no usarlas y no se trata de un ad hóminem. Stallman y el proyecto GNU nos aconsejan que hablemos siempre de Software libre y aquí no cabe imparcialidad.

3- Obsolescencia percibida: Esta es una táctica psicológica, se trata de convencer al consumidor mediante publicidad y el uso de influenciadores de que el producto que se tiene actualmente está viejo y que se necesita uno nuevo. Como por ejemplo: ¿cuantos megapíxeles necesitas en tu teléfono para sacar una buena foto de tu mascota?

4- Trabas a la reparación: En el caso de los teléfonos por ejemplo, lo de impedir sacar la batería (con la excusa de hacer los teléfonos más delgados) es una forma de obligar a los consumidores a recurrir a los servicios oficiales y a disuadirlos de reemplazarlas por sustitutos más económicos. Otras tácticas son la utilización de piezas no estándar o que necesitan herramientas específicas para la reparación. Muchas veces se suele restringir el acceso a estas piezas o hacer una reducida producción de las mismas para aumentar artificialmente el costo.

Ejemplos de Obsolescencia Programada

- iPhone cada vez más lentos: La Justicia francesa comprobó que actualizaciones de Software hacían cada vez más lento el rendimiento de los modelos más viejos. La empresa le echó la culpa a las baterías, pero pagó una compensación de decenas de millones de dólares. Además rebajó los precios de sus baterías de repuesto para que los teléfonos fueran más rápidos con el nuevo Software y se comprometió a hacer más en el futuro para garantizar que los teléfonos no volvieran a ser más lentos. Con la salida de un nuevo modelo de teléfono cada año, seguro que hay algo de obsolescencia planificada en alguna parte.

- Impresoras: Esto es algo que todos conocemos. Muchas veces nos encontramos con impresoras a precio rebajado, pero al momento de tener que comprar un cartucho de tinta nos encontramos con que este tiene un precio igual o superior a comprar una nueva. Además, se ponen restricciones a la recarga o al uso de cartuchos alternativos. Hubo denuncias de que algunos modelos dejaban de funcionar a partir de cierta cantidad de páginas impresas o cierto tiempo desde la primera impresión.

- Certificados de seguridad: Por ejemplo, el pasado 30 de septiembre de 2021 caduco otro certificado de autentificación (CA de DST Root CA X3 de Let's Encrypt) que ayudaba a validar la conexión en internet a

los dispositivos que no fueron actualizados a otro certificado más actual -en la mayoría de los casos por no ser del interés económico de sus creadores-. Esto ocasionó que millones de dispositivos (teléfonos inteligentes, Smart TV, tabletas, computadoras portátiles y de escritorio, etc.) con algunos años de ser creados y perfectamente funcionales dejarán de conectarse a internet de un día para otro, forzando a sus dueños a desechar el dispositivo por carecer del servicio de internet en las aplicaciones instaladas.

- Cambio de la versión del sistema operativo: En el caso del sistema operativo Windows 10 a 11, la solicitud de requisitos mínimos de Hardware es para muchos equipos excesivo, ya que se estima que dejará fuera en su actualización a casi todos los equipos con más de 4 años de antigüedad por no contar por ejemplo con el Chip TPM 2.0 o GPU compatible con DirectX 12, siendo perfectamente funcionales con la versión actual del sistema operativo. Si bien Windows 10 seguirá con soporte hasta 2025, los usuarios que deseen tener las nuevas características del sistema operativo tendrán que cambiar de equipo.

2.1 Software Propietario

No existe consenso sobre el término a utilizar para referirse al opuesto del Software libre. La expresión «Software propietario (Proprietary Software)» (véase [4]), en la lengua anglosajona, "Proprietary" significa «poseído o controlado privadamente (Privately Owned and Controlled)», que destaca la manutención de la reserva de derechos sobre el uso, modificación o redistribución del Software. Inicialmente utilizado, pero con el inconveniente de que la acepción proviene de una traducción literal del inglés, no correspondiendo su uso como adjetivo en el español, de manera que puede ser considerado como un barbarismo.

El término "propietario" en español resultaría inadecuado, pues significa que «tiene derecho de propiedad sobre una cosa», por lo que no podría calificarse de "propietario" al Software, porque éste no tiene propiedad sobre nada (es decir, no es dueño de nada) y además, no podría serlo (porque es una cosa y no una persona). Así mismo, la expresión "Software propietario" podría ser interpretada como: "Software sujeto a propiedad" (derechos o titularidad) y su opuesto, el Software libre, también está sujeto al derecho de autor. Otra interpretación es que contrariamente al uso popular del término, se puede

afirmar que "todo Software es propietario", por lo que la forma correcta de referirse al Software con restricciones de uso, estudio, copia o mejora es la de Software privativo, según esta interpretación el término "propietario" podría aplicarse tanto para Software libre como Software privativo, ya que la diferencia entre uno y otro está en que el dueño del Software privativo lo licencia como propiedad privada y el de Software libre como propiedad social.

Con la intención de corregir el defecto de la expresión "Software propietario" aparece el llamado "Software con propietario", sin embargo se argumenta contra el término "con propietario" y justamente su similitud con Proprietary en inglés, que sólo haría referencia a un aspecto del Software que no es libre, manteniendo una de las principales críticas a éste (de "Software sujeto a derechos" o "propiedad"). Adicionalmente, si "propietario" se refiere al titular de los derechos de autor -y está claro que no se puede referir al usuario, en tanto éste es simplemente un cesionario-, no resuelve la contradicción: todo el Software libre tiene también titulares de derechos de autor.

La expresión Software no libre (en inglés Non-Free Software) es usado por la FSF para agrupar todo el Software que no es libre, es decir, incluye al llamado en inglés "Semi-Free Software" (Software semilibre) y al "Propietary Software". Asimismo, es frecuentemente utilizado para referirse al Software que no cumple con las Directrices de Software libre de Debian GNU/Linux, las cuales siguen la misma idea básica de libertad en el Software, propugnada por la FSF y sobre las cuales está basada la definición de código abierto de la Open Source Initiative.

Adicionalmente el Software de código cerrado nace como antónimo de Software de código abierto y por lo tanto se centra más en el aspecto de ausencia de acceso al código que en los derechos sobre el mismo, éste se refiere sólo a la ausencia de una sola libertad por lo que su uso debe enfocarse sólo a este tipo de Software y aunque siempre signifique que es un Software que no es libre, no tiene que ser Software de código cerrado.

La expresión Software privado es usada por la relación entre los conceptos de tener y ser privado. Este término sería inadecuado debido a que, en una de sus acepciones, la palabra "privado" se entiende como antónimo de "público", es decir, que «no es de propiedad pública o estatal, sino que pertenece a particulares», provocando que esta categoría se interpretará como no referente al Estado, lo que produciría la exclusión del Software no libre generado por el aparato estatal. Además, el "Software público" se asocia generalmente con Software de dominio público.

2.2 Software Libre

La definición de Software libre (véase [9], [10], [2], [3], [1] y [5]) estipula los criterios que se tienen que cumplir para que un programa sea considerado libre. De vez en cuando se modifica esta definición para clarificarla o para resolver problemas sobre cuestiones delicadas. «Software libre» significa que el Software respeta la libertad de los usuarios y la comunidad. En términos generales, los usuarios tienen la libertad de copiar, distribuir, estudiar, modificar y mejorar el Software. Con estas libertades, los usuarios -tanto individualmente como en forma colectiva- controlan el programa y lo que hace.

Cuando los usuarios no controlan el programa, el programa controla a los usuarios. Los programadores controlan el programa y a través del programa, controlan a los usuarios. Un programa que no es libre, llamado «privativo o propietario», es considerado por muchos como un instrumento de poder injusto.

El Software libre es la denominación del Software que respeta la libertad de todos los usuarios que adquirieron el producto y por tanto, una vez obtenido el mismo puede ser usado, copiado, estudiado, modificado y redistribuido libremente de varias formas. Según la Free Software Foundation (véase [9]), el Software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir y estudiar el mismo, e incluso modificar el Software y distribuirlo modificado.

Un programa es Software libre si los usuarios tienen las cuatro libertades esenciales:

0. La libertad de usar el programa, con cualquier propósito.

1. La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades.

2. La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo.

3. La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.

Un programa es Software libre si los usuarios tienen todas esas libertades. Por tanto, el usuario debe ser libre de redistribuir copias, tanto con o sin modificaciones, ya sea gratuitamente o cobrando una tarifa por la distribución,

a cualquiera en cualquier parte. El ser libre de hacer estas cosas significa, entre otras cosas, que no tiene que pedir ni pagar el permiso.

También debe tener la libertad de hacer modificaciones y usarlas en privado para su propio trabajo o pasatiempo, sin siquiera mencionar que existen. Si publica sus cambios, no debe estar obligado a notificarlo a nadie en particular, ni de ninguna manera.

La libertad de ejecutar el programa significa que cualquier tipo de persona u organización es libre de usarlo en cualquier tipo de sistema de computación, para cualquier tipo de trabajo y finalidad, sin que exista obligación alguna de comunicarlo al programador ni a ninguna otra entidad específica. En esta libertad, lo que importa es el propósito de los usuarios, no el de los programadores. El usuario es libre de ejecutar el programa para alcanzar sus propósitos y si lo distribuye a otra persona, también esa persona será libre de ejecutarlo para lo que necesite; nadie tiene derecho a imponer sus propios objetivos.

La libertad de redistribuir copias debe incluir las formas binarias o ejecutables del programa, así como el código fuente, tanto para las versiones modificadas como para las que no lo estén. Distribuir programas en forma de ejecutables es necesario para que los sistemas operativos libres se puedan instalar fácilmente. Resulta aceptable si no existe un modo de producir un formato binario o ejecutable para un programa específico, dado que algunos lenguajes no incorporan esa característica, pero debe tener la libertad de redistribuir dichos formatos si encontrara o programara una forma de hacerlo.

Para que se de la libertad que se menciona en los puntos 1 y 3 de realizar cambios y publicar las versiones modificadas tenga sentido, el usuario debe tener acceso al código fuente del programa. Por consiguiente, el acceso al código fuente es una condición necesaria para el Software libre. El «código fuente» compilado no es código fuente real y no cuenta como código fuente.

La libertad 1 incluye la libertad de usar su versión modificada en lugar de la original. Si el programa se entrega con un producto diseñado para ejecutar versiones modificadas de terceros, pero rechaza ejecutar las suyas, una práctica conocida como «tivoización» o «arranque seguro» [«Lockdown»] la libertad 1 se convierte más en una ficción teórica que en una libertad práctica, esto no es suficiente, en otras palabras, estos binarios no son Software libre, incluso si se compilaron desde un código fuente que es libre.

Una manera importante de modificar el programa es agregándole subrutinas y módulos libres ya disponibles. Si la licencia del programa especifica que no se pueden añadir módulos que ya existen y que están bajo una licencia

apropiada, por ejemplo si requiere que usted sea el titular de los derechos de autor del código que desea añadir, entonces se trata de una licencia demasiado restrictiva como para considerarla libre.

La libertad 3 incluye la libertad de publicar sus versiones modificadas como Software libre. Una licencia libre también puede permitir otras formas de publicarlas; en otras palabras, no tiene que ser una licencia de Copyleft. No obstante, una licencia que requiera que las versiones modificadas no sean libres, no se puede considerar libre.

«Software libre» no significa que «no es comercial». Un programa libre debe estar disponible para el uso comercial, la programación comercial y la distribución comercial. La programación comercial de Software libre ya no es inusual; el Software libre comercial es muy importante, ejemplo de ello es la empresa RedHat (ahora propiedad de IBM). Puede haber pagado dinero para obtener copias de Software libre, o puede haber obtenido copias sin costo. Pero sin tener en cuenta cómo obtuvo sus copias, siempre tiene la libertad de copiar y modificar el Software, incluso de vender copias.

El término Software no libre se emplea para referirse al Software distribuido bajo una licencia de Software más restrictiva que no garantiza estas cuatro libertades. Las leyes de la propiedad intelectual reservan la mayoría de los derechos de modificación, duplicación y redistribución para el dueño del Copyright; el Software dispuesto bajo una licencia de Software libre rescinde específicamente la mayoría de estos derechos reservados.

Los manuales de Software deben ser libres por las mismas razones que el Software debe ser libre y porque de hecho los manuales son parte del Software. También tiene sentido aplicar los mismos argumentos a otros tipos de obras de uso práctico, es decir, obras que incorporen conocimiento útil, tal como publicaciones educativas y de referencia. Wikipedia es el ejemplo más conocido.

La lista de proyectos de este tipo es realmente impresionante, algunos han conseguido un uso y alta calidad, por ejemplo el compilador GCC, el Kernel de Linux y el sistema operativo Debian GNU/Linux y Android. Mientras que otros proyectos han caído en el olvido, pero de la gran mayoría se tiene copia del código fuente que permitiría a quienes estén interesados en dicho proyecto poder reusarlo y en su caso ampliarlo.

La característica más importante que aparece típicamente en un proyecto de este tipo, es que un conjunto de personas separadas a gran distancia, sean capaces, a través de la Web, de los E-mail y de foros de aunar sus esfuerzos para crear, mejorar y distribuir un producto, de forma que todos

ellos se benefician unos de otros. Evidentemente, la mayor parte del peso recae en los desarrolladores, pero también es necesaria una difusión para que los usuarios documenten, encuentren errores, hagan foros de discusión, etc.

Si bien, el Software libre no es más seguro (en el sentido de invulnerable) que el propietario, la diferencia estriba en que el código fuente en el Software libre está disponible para todos y cualquiera puede aportar una solución, y por lo general al poco tiempo de detectarse una vulnerabilidad (a veces en cuestión de horas) se puede disponer de una solución para la misma. Además, al tener acceso al código fuente se puede detectar fácilmente si alguien introdujo código malicioso a una determinada aplicación.

¿Por qué se Interesan los Autores, Alumnos y Profesores Universitarios en el Software Libre? La ventaja principal es porque bajo el Software libre subyace la idea de compartir conocimiento y favorecer la existencia de nuevas ideas[6]; y ¿qué es investigar y enseñar?, sino crear conocimiento y procurar que los alumnos aprendan e incluso vayan más allá de lo aprendido. Se comparte la idea, que el espíritu del Software libre es similar al que debería reinar en las instituciones educativas:

- Porque así no se condiciona a los estudiantes a usar siempre lo mismo.
- No se fomenta la piratería en los estudiantes y se evita pagar licencias que no son necesarias al existir alternativas gratuitas.
- Es mucho más seguro ya que el Software libre es público y se puede ver qué hace exactamente sin recelos.
- Se ofrece libertad de elección a los estudiantes y profesores al no limitarlos a usar una solución determinada, ampliando sus opciones y permitiendo un mayor aprendizaje.

Concretando estas ideas, profesores e investigadores necesitan herramientas para la investigación y docencia y estas deben tener una calidad mínima y ser fácilmente distribuibles entre los alumnos. En muchos casos las compañías desarrolladoras y distribuidoras de programas de cómputo no han

[6] ¿Por qué el Software creado con dinero de los impuestos no se publica como Software Libre?

¡El código pagado por los ciudadanos debería estar disponible para los ciudadanos y el mismo gobierno!

sabido ofrecer sus productos con la flexibilidad adecuada para las labores docentes o, en otros casos, los productos desarrollados no tienen la calidad esperada.

El Software libre es aún joven, pese a las decenas de miles de proyectos actuales -en los que se trabaja constantemente en mejorar la parte computacional de los algoritmos involucrados en el proyecto, haciendo y puliendo interfaces gráficas, generando ayuda en línea así como la documentación necesaria para que usuarios noveles y avanzados usen la mayor cantidad de opciones programadas- existen muchas otras necesidades profesionales y de investigación que requieren el desarrollo innovador de programas de cómputo para automatizarlas y hacerlas eficientes. Esto queda plasmado en las decenas de proyectos que a diario son registrados en las páginas especializadas en busca de difusión y apoyo para su proyecto.

En los últimos años, muchos proyectos han pasado de ser simples programas en línea de comandos a complejas aplicaciones multiplataforma -se ejecutan en distintos sistemas operativos como son Windows, Linux, Unix, Mac OS, Android- con ambientes gráficos multimedia que en muchos casos han superado a sus contrapartes comerciales -por ejemplo los navegadores Web-. Para muestra de este maravilloso avance, tomemos el proyecto del sistema operativo Android, que actualmente se ejecuta en millones de equipos -como celulares, tabletas, electrodomésticos, etc.- y en los cuales se pueden descargar miles de aplicaciones y está soportado por una gran cantidad de usuarios y empresas comerciales como Google, IBM y últimamente Microsoft -que años atrás era acérrima enemiga del Software libre-.

El Software libre ha logrado desplazar a muchos de sus competidores por sus múltiples bondades y bajo costo de desarrollo -es el caso de Windows Phone que fue reemplazado por Android de Google-, al reusar miles de aplicaciones ya existentes que usan Software libre y permitir desarrollar otro tanto de aplicaciones bajo una plataforma que se ejecuta en los más diversos procesadores. Además, el uso de Software libre y su posibilidad de ampliarlo y/o especializarlo según sea necesario, ha permitido crear de forma cada vez más rápida y confiable; para poner a disposición de un gran público programas de uso común, así como especializado que satisfagan las nuevas necesidades de los usuarios.

Software Libre en Ciencia y Educación Algunos puntos y reflexiones sobre porqué se considera que es interesante el Software libre en Ciencia y

Educación son:

- Accesible a todo el mundo aunque no sea rentable su desarrollo: El Software libre entre sus libertades permite que se pueda ejecutar por terceros, copiarlo, distribuirlo y estudiarlo/modificarlo. Eso hace que si en ciencia se usa Software libre el acceso a esos programas no suponga una barrera (se puede distribuir y ejecutar).

- Muchos de los desarrollos en el campo de la accesibilidad se realizan en universidades y son distribuidos como Software libre: Aunque no sea rentable muchas veces el desarrollo de herramientas de accesibilidad (no sea algo monetizable) en la universidad se consigue escapar a esa la lógica capitalista de solamente invertir en lo que pueda ofrecer beneficio económico.

- Transparencia: En ciencia es importante ver las costuras para comprobar si es verdad lo que se afirma, tener acceso al código fuente del Software empleado permite poder estudiarlo por si realiza algún cálculo mal.

- Propicia el espíritu crítico: Si no tienes acceso a las revistas o el acceso es privativo para los bolsillos de mucha gente no puedes comprobar la información. Se nos pide que seamos críticos con la información que se nos da del mundo científico pero no podemos entrenar el espíritu crítico sin acceso al conocimiento libre.

- Caramelos con droga en la puerta del colegio: Muchas empresas buscan introducir en los colegios, institutos y universidad su Software. Ofrecer un programa que permita trabajar y genere una dependencia quedando los datos muchas veces en las nubes (ordenadores de otras personas). Un ejemplo es Microsoft con Office 365. Dando cuentas gratuitas durante un tiempo para que se use su Software. Otro ejemplo podría ser Unity3D en vez de por ejemplo Godot.

- Especificaciones de protocolos abiertas VS cerradas: Gracias a que los protocolos TCP/IP, HTTP, POP, SNMP, DHCP, etc. son abiertos es posible construir herramientas por cualquier con conocimientos de programación. Con protocolos cerrados solamente quienes tuvieran acceso a las especificaciones podrían desarrollar y conocer cómo funcionan.

- Uso de estándares: Existiendo un estándar para documentos ofimáticos (procesador de textos, hoja de cálculo, presentaciones, etc.) algunas empresas como Microsoft se empeñan en ir con su propio formato y estándar en vez de sumarse a que sea más sencillo ir a una y que el usuario pueda optar por que herramientas usar para editar o trabajar con documentos ofimáticos.

- Software libre para la Ciencia ciudadana: Un ejemplo en el que es importante la colaboración ciudadana es el cambio climático y la defensa medioambiental del territorio. Desde imvec.tech usan herramientas de Software libre para medición y monitorización de contaminación.

Software Libre: Beneficios Más Allá de la Informática El uso de las tecnologías de código abierto supone cultivar el conocimiento y la puesta en valor de la libertad individual, lo personal y lo privado, todo ello sin menospreciar lo público y la construcción de una sociedad. El movimiento del Software libre, con Linux a la cabeza, ha capitaneado durante décadas planteamientos para un cambio en el modo de producción: el abaratamiento de costes empresariales para grandes y pequeños, el trabajo en línea, el desarrollo de Software y Hardware a pequeña escala, el replanteamiento del negocio informático, el sistema de normas éticas que rigen los grupos, la documentación abierta, etc.

Todo ello derivado de una simple idea: la libertad. Ha sido la clave que ha llevado a todo este movimiento hacia una autonomía y motivación que pocas veces se ve en otros sectores. El Open Source está lleno de alternativas con la libertad como pilar y consecuencia filosófica, de hecho muchos Forks (derivaciones) de proyectos aparecen cuando la disputa sobre la misma toma relevancia. Esta manera de hacer las cosas debería trasladarse directamente a la sociedad promoviendo esos valores para evitar caer en un mundo despótico que toma fuerza a pasos agigantados.

No estaría mal promover la comprensión de las licencias libres. Leer y entender una licencia GPL, MIT o BSD es infinitamente más sencillo y rápido que hacerlo con otras, siendo unas normas fáciles de cumplir porque encajan con un modelo ético y práctico comprensible por muchos.

Podríamos decir que GPL, BSD y MIT son las Constituciones que vertebran todo el movimiento del Software Libre, cosechando derechos de uso y logros como el ahorro o la legítima copia privada. Lo mismo podría ocurrir con las licencias Creative Commons para cierto contenido, un arma poderosa

en el sector divulgativo que está poco extendida por desconocimiento y el Status Quo de la propiedad intelectual.

La cooperación libre y voluntaria supuso un éxito hasta ahora pero está siendo amenazada por la dinámica actual de la Web, donde la centralización de las principales plataformas supone estar bajo el yugo de normas que rastrean con lupa y cambian sus términos con demasiada frecuencia. Ya ni digamos cuando eso se junta con el ansia de monetización: si quieres dinero más te vale no cabrear a los anunciantes o no tener un Strike por uso de contenido que podría ser reclamado.

Los claroscuros de este sistema hace que los creadores cada vez hagan menos de forma libre y altruista, y ello podría verse potenciado por las búsquedas con inteligencia artificial, lo que apunta a un empobrecimiento del contenido cultural fresco y renovador. Las polémicas con GitHub Copilot o ChatGPT sobre el entrenamiento de las IAs hace sobrevolar una vez más la cuestión del Copyright y el uso legítimo de los resultados brindados por éstas, lo que también condiciona la creación.

La libertad de expresión, de uso, de creación, de modificación ... esas cuestiones llevan irremediablemente a una libertad de pensamiento y acción, a una adaptación creativa que puede ser la motivación para romper moldes en todos los aspectos. El código abierto y sus licencias son, por tanto, un beneficio personal y social mucho más grande que el simple hecho de usar Linux o Software libre. El contenido despreocupado, que no prioriza el dinero y el posicionamiento/visualizaciones, se vuelve esencial para el inconformismo.

2.3 Seguridad del Software

Si bien, el Software Libre no es más seguro (en el sentido de invulnerable) que el propietario, la diferencia puede estribar en que el código fuente en el Software libre está disponible para todos y cualquiera puede aportar una solución y por lo general al poco tiempo de detectarse una vulnerabilidad (a veces en cuestión de horas) se puede disponer de una solución para la misma. Además, al tener acceso al código fuente se puede detectar si alguien introdujo código malicioso a una determinada aplicación.

Pero de todos es sabido, que los usuarios de Software de código abierto, como por ejemplo los que de manera habitual trabajan con equipos comandados por sistemas Linux, por regla general se sienten orgullosos de la seguridad que estos programas aportan con respecto a los sistemas cerrados propios de otras firmas, dígase Microsoft Windows o Mac de Apple.

¿Es Seguro el Software Libre? En primer lugar definiremos el concepto de "seguridad" como salvaguarda de las propiedades básicas de la información. Entre las características que debe cumplir para ser seguro, encontramos la integridad, es decir, que sólo los usuarios autorizados pueden crear y modificar los componentes del sistema, la confidencialidad, sólo estos usuarios pueden acceder a esos componentes, la disponibilidad, que todos los componentes estén a disposición de los usuarios siempre que lo deseen y el "no repudio", o lo que es lo mismo, la aceptación de un protocolo de comunicación entre el servidor y un cliente, por ejemplo, mediante certificados digitales.

Entre las diferencias de seguridad entre un Software Libre y el Software Propietario, podemos destacar:

- Seguridad en el Software Propietario: En el caso de tener "agujeros de seguridad", puede que no nos demos cuenta y que no podamos repararlos. Existe una dependencia del fabricante, retrasándose así cualquier reparación y la falsa creencia de que es más seguro por ser oscuro (la seguridad por oscuridad determina los fallos de seguridad no parcheados en cada producto).

- Seguridad en el Software Libre: Por su carácter público y su crecimiento progresivo, se van añadiendo funciones y se nos permite detectar más fácilmente los agujeros de seguridad para poder corregirlos. Los problemas tardan mucho menos en ser resueltos por el apoyo que tiene de los Hackers y una gran comunidad de desarrolladores y al ser un Software de código libre, cualquier empresa puede aportar soporte técnico.

Sin embargo esta es una pregunta sobre la que los expertos al día de hoy, tras muchos años de discusiones, siguen sin ponerse de acuerdo. ¿Es más seguro el Software de código abierto que los programas cerrados, o viceversa? Lo cierto es que, en términos generales, ambos bandos tienen sus razones con las que defender sus argumentos. Por un lado, los usuarios de las aplicaciones y sistemas de código abierto, defienden que, al estar el código fuente disponible a los ojos de todo el mundo, es mucho más fácil localizar posibles agujeros de seguridad y vulnerabilidades que pongan en peligro los datos de los usuarios.

Por otro lado, aquellos que consideran que los sistemas cerrados son más seguros en este sentido, afirman que al tener acceso tan solo los expertos al código fuente de sus aplicaciones, es más complicado que se produzcan

filtraciones o inserciones de Software malicioso en este tipo de sistemas. Hay que tener en cuenta que, por ejemplo, Google premia a las personas que descubren fallos de seguridad en su Software como Chrome, aunque no es el único gigante de la tecnología en utilizar estas tácticas.

De hecho muchas empresas están gastando miles de millones de dólares y/o euros en hacer que sus propuestas sean lo más seguras posible, argumentando que la seguridad de sus proyectos es una de sus prioridades, todo con el fin de intentar frenar que los atacantes vulneren sus sistemas. Por otro lado, otros aseguran que cuando el código fuente es público, más ojos están disponibles para detectar posibles vulnerabilidades o errores en dicho código, por lo que siempre será más rápido y sencillo poner soluciones con el fin de ganar en seguridad.

Sea como sea, en cualquiera de los dos casos, lo que ha quedado más que demostrado es que la seguridad no está garantizada en ningún momento, ya sean propuestas de código abierto, o no. Pero también es cierto que lo que se procura es que los riesgos de ser atacados se reduzcan en medida de lo posible. Los sistemas Linux son considerados desde hace mucho tiempo como un sistema operativo seguro, en buena parte debido a las ventajas que ofrece su diseño. Dado que su código está abierto, son muchas las personas que incorporan mejoras de las que el resto de usuarios de Linux se benefician, a diferencia de las propuestas de Windows o MacOS, donde estas correcciones generalmente se limitan a las que detectan Microsoft y Apple.

No obstante, en nuestra defensa del Software libre, diremos que su código abierto permite que los errores sean encontrados y solucionados con mayor rapidez, por lo que determinamos que es el Software más recomendable.

En general, puede afirmarse que el Software libre es más seguro, ya que debido a su carácter abierto y distribuido, un gran número de programadores y personas expertas pueden estar atentas al código fuente -especialmente en los grandes proyectos-, lo cual permite hacer auditorías con objeto de detectar errores y puertas traseras (Backdoor, en inglés) que pongan en riesgo nuestros datos.

Así, los grandes programas y proyectos de Software libre, con una extensa comunidad de desarrollo y usuarios que lo respalden, presentan niveles muy altos de seguridad, un alto grado de protección y una rápida respuesta a posibles vulnerabilidades.

2.4 Tipos de Licencias

Tanto la Open Source Initiative como la Free Software Foundation mantienen en sus páginas Web (véase [9], [10], y [5]) listados oficiales de las licencias de Software libre que aprueban.

Una licencia es aquella autorización formal con carácter contractual que un autor de un Software da a un interesado para ejercer "actos de explotación legales". Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el licenciatario. Desde el punto de vista del Software libre, existen distintas variantes del concepto o grupos de licencias:

Licencias GPL Una de las más utilizadas es la Licencia Pública General de GNU (GNU GPL). El autor conserva los derechos de autoría (Copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del Software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que sea imposible crear un producto con partes no licenciadas GPL, el conjunto tiene que ser GPL.

En la práctica, esto hace que las licencias de Software libre se dividan en dos grandes grupos, aquellas que pueden ser mezcladas con código licenciado bajo GNU GPL (y que inevitablemente desaparecerán en el proceso, al ser el código resultante licenciado bajo GNU GPL) y las que no lo permiten al incluir mayores u otros requisitos que no contemplan ni admiten la GNU GPL y que por lo tanto no pueden ser enlazadas ni mezcladas con código gobernado por la licencia GNU GPL.

GPL Versión 1 la versión 1 de GNU GPL, fue presentada el 25 de febrero de 1989, impidió lo que eran las dos principales formas con las que los distribuidores de Software restringían las libertades definidas por el Software libre. El primer problema fue que los distribuidores publicaban únicamente los archivos binarios, funcionales y ejecutables, pero no entendibles o modificables por humanos. Para prevenir esto, la GPLv1 estableció que cualquier proveedor de Software libre además de distribuir el archivo binario debía liberar a su vez código fuente entendible y que pudiera ser modificado por el ser humano bajo la misma licencia (secciones 3a y 3b de la licencia).

El segundo problema era que los distribuidores podían añadir restricciones adicionales, añadiendo restricciones a la licencia o mediante la combinación del Software con otro que tuviera otras restricciones en su distribución. Si

esto se hacía, entonces la unión de los dos conjuntos de restricciones sería aplicada al trabajo combinado, entonces podrían añadirse restricciones inaceptables. Para prevenir esto, GPLv1 obligaba a que las versiones modificadas en su conjunto, tuvieran que ser distribuidas bajo los términos GPLv1 (secciones 2b y 4 de la licencia). Por lo tanto, el Software distribuido bajo GPLv1 puede ser combinado con Software bajo términos más permisivos y no con Software con licencias más restrictivas, lo que entraría en conflicto con el requisito de que todo Software tiene que ser distribuido bajo los términos de la GPLv1.

GPL Versión 2 según Richard Stallman, el mayor cambio en GPLv2 fue la cláusula "Liberty or Death" («libertad o muerte»). Esta sección dice que si alguien impone restricciones que le prohíben distribuir código GPL de tal forma que influya en las libertades de los usuarios (por ejemplo, si una ley impone que esa persona únicamente pueda distribuir el Software en binario), esa persona no puede distribuir Software GPL. La esperanza es que esto hará que sea menos tentador para las empresas el recurrir a las amenazas de patentes para exigir una remuneración de los desarrolladores de Software libre.

En 1991 se hizo evidente que una licencia menos restrictiva sería estratégicamente útil para la biblioteca C y para las bibliotecas de Software que esencialmente hacían el trabajo que llevaban a cabo otras bibliotecas comerciales ya existentes. Cuando la versión 2 de GPL fue liberada en junio de 1991, una segunda licencia Library General Public License fue introducida al mismo tiempo y numerada con la versión 2 para denotar que ambas son complementarias. Los números de versiones divergieron en 1999 cuando la versión 2.1 de LGPL fue liberada, esta fue renombrada como GNU Lesser General Public License para reflejar su lugar en esta filosofía.

GPL Versión 3 A finales de 2005, la Free Software Foundation (FSF) anunció estar trabajando en la versión 3 de la GPL (GPLv3). El 16 de enero de 2006, el primer borrador de GPLv3 fue publicado y se inició la consulta pública. La consulta pública se planeó originalmente para durar de nueve a quince meses, pero finalmente se extendió a dieciocho meses, durante los cuales se publicaron cuatro borradores. La GPLv3 oficial fue liberada por la FSF el 29 de junio de 2007.

Según Stallman los cambios más importantes se produjeron en el campo

de las patentes de Software, la compatibilidad de licencias de Software libre, la definición de código fuente y restricciones a las modificaciones de Hardware. Otros cambios están relacionados con la internacionalización, cómo son manejadas las violaciones de licencias y cómo los permisos adicionales pueden ser concedidos por el titular de los derechos de autor. También añade disposiciones para quitar al DRM su valor legal, por lo que es posible romper el DRM (Digital Rights Management) en el Software de GPL sin romper leyes como la DMCA (Digital Millennium Copyright Act).

GPLv2 vs GPL v3 GPLv3 contiene la intención básica de GPLv2 y es una licencia de código abierto con un Copyleft estricto. Sin embargo, el idioma del texto de la licencia fue fuertemente modificado y es mucho más completo en respuesta a los cambios técnicos, legales y al intercambio internacional de licencias.

La nueva versión de la licencia contiene una serie de cláusulas que abordan preguntas que no fueron o fueron cubiertas de manera insuficiente en la versión 2 de la GPL. Las nuevas regulaciones más importantes son las siguientes:

- GPLv3 contiene normas de compatibilidad que hacen que sea más fácil combinar el código GPL con el código que se publicó bajo diferentes licencias. Esto se refiere en particular al código bajo la licencia de Apache v. 2.0.

- Se insertaron normas sobre gestión de derechos digitales para evitar que el Software GPL se modifique a voluntad, ya que los usuarios recurrieron a las disposiciones legales para protegerse mediante medidas técnicas de protección (como la DMCA o la directiva sobre derechos de autor).

- La licencia GPLv3 contiene una licencia de patente explícita, según la cual las personas que licencian un programa bajo licencia GPL otorgan derechos de autor y patentes, en la medida en que esto sea necesario para utilizar el código que ellos otorgan. Por lo tanto, no se concede una licencia de patente completa. Además, la nueva cláusula de patente intenta proteger al usuario de las consecuencias de los acuerdos entre los titulares de patentes y los licenciatarios de la licencia pública general que solo benefician a algunos de los licenciatarios (correspondientes al

acuerdo Microsoft / Novell). Los licenciatarios deben garantizar que todos los usuarios disfrutan de tales ventajas (licencia de patente o liberación de reclamos) o que nadie puede beneficiarse de ellos.

- A diferencia de la GPLv2, la GPLv3 establece claramente que no es necesario divulgar el código fuente en un uso ASP (Application Service Provider) de los programas GPL, siempre que no se envíe una copia del Software al cliente. Si el efecto Copyleft debe extenderse al uso de ASP, debe aplicarse la Licencia pública general de Affero, versión 3 (AGPL) que solo difiere de la GPLv3 en esta consideración.

Licencias Estilo BSD Llamadas así porque se utilizan en gran cantidad de Software distribuido junto a los sistemas operativos BSD. El autor, bajo tales licencias, mantiene la protección de Copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU GPL con quienes son compatibles. Puede argumentarse que esta licencia asegura "verdadero" Software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al Software y que puede decidir incluso redistribuirlo como no libre.

Licencia Copyleft Hay que hacer constar que el titular de los derechos de autor (Copyright) de un Software bajo licencia Copyleft puede también realizar una versión modificada bajo su Copyright original y venderla bajo cualquier licencia que desee, además de distribuir la versión original como Software libre. Esta técnica ha sido usada como un modelo de negocio por una serie de empresas que realizan Software libre (por ejemplo MySQL); esta práctica no restringe ninguno de los derechos otorgados a los usuarios de la versión Copyleft.

Licencia estilo MIT Las licencias MIT son de las más permisivas, casi se consideran Software de dominio público. Lo único que requieren es incluir la licencia MIT para indicar que el Software incluye código con licencia MIT.

Licencia Apache License La licencia Apache trata de preservar los derechos de autor, incluir la licencia en el Software distribuido y una lista de

los cambios realizados. En modificaciones extensivas del Software original permite licenciar el Software bajo otra licencia sin incluir esas modificaciones en el código fuente.

Licencia Mozilla Public License MPL Esta licencia requiere que los archivos al ser distribuidos conserven la misma licencia original pero pueden ser usados junto con archivos con otra licencia, al contrario de la licencia GPL que requiere que todo el código usado junto con código GPL sea licenciado como código GPL. También en caso de hacer modificaciones extensivas permite distribuirlas bajo diferentes términos y sin incluir el código fuente en las modificaciones.

Licencia Código de Dominio Público Es un código que no está sujeto a derechos de autor que puede utilizarse sin restricciones.

Licencia Creative Commons Las licencias de Creative Commons son más utilizadas para cualquier creación digital que para el Software, entendiendo como creación digital desde fotos, artículos en blogs, música, vídeos, este trabajo, etc. Hay varios tipos de licencias de Creative Commons diferenciando entre permitir modificaciones a la obra original, solicitando crédito de la creación o permitiendo un uso comercial de la obra.

Licencias de Código Abierto Las licencias de código abierto son un intermedio entre las licencias privativas y las licencias de Software libre. Las licencias de código abierto permiten el acceso al código fuente pero no todas se consideran licencias de Software libre al no otorgar otros derechos que se requieren para considerar un Software como Software libre como el derecho al uso o con cualquier propósito, modificación y distribución.

Dado el éxito del Software libre como modelo de desarrollo de Software algunas empresas cuyo Software era privativo pueden decidir hacerlo de código abierto con la intención de suplir algunas carencias de Software privativo pero sin perder ciertos derechos que son la fuente de sus ingresos como la venta de licencias.

Las expresiones «Software libre» y «código abierto» se refieren casi al mismo conjunto de programas. No obstante, dicen cosas muy diferentes acerca de dichos programas, basándose en valores diferentes. El movimiento del Software libre defiende la libertad de los usuarios de ordenadores, en un

movimiento en pro de la libertad y la justicia. Por contra, la idea del código abierto valora principalmente las ventajas prácticas y no defiende principios. Esta es la razón por la que gran parte de la comunidad de Software libre está en desacuerdo con el movimiento del código abierto y nosotros no empleamos esta expresión en este texto.

Licencia Microsoft Public License La Microsoft Public License es una licencia de código abierto que permite la distribución del Software bajo la misma licencia y la modificación para un uso privado. Tiene restricciones en cuanto a las marcas registradas.

En caso de distribuir el Software de forma compilada o en forma de objeto binario no se exige proporcionar los derechos de acceso al código fuente del Software compilado o en forma de objeto binario. En este caso esta licencia no otorga más derechos de los que se reciben, pero si permite otorgar menos derechos al distribuir el Software (compilado o en forma de objeto binario).

Modelo de Desarrollo de Software Bazar y Catedral El tipo de licencia no determina qué Software es mejor o peor, si el privativo o el Software libre, la diferencia entre las licencias está en sus características éticas y legales. Aunque el modelo de desarrollo con una licencia de código abierto a la larga suele tener un mejor desarrollo y éxito que el Software privativo, más aún con un medio como internet que permite colaborar a cualquier persona independiente de donde esté ubicada en el mundo.

Comparación con el Software de Código Abierto Aunque en la práctica el Software de código abierto y el Software libre comparten muchas de sus licencias, la Free Software Foundation opina que el movimiento del Software de código abierto es filosóficamente diferente del movimiento del Software libre. Los defensores del término «código abierto (Open Source)», afirman que éste evita la ambigüedad del término en ese idioma que es «Free» en «Free Software».

Mucha gente reconoce el beneficio cualitativo del proceso de desarrollo de Software cuando los desarrolladores pueden usar, modificar y redistribuir el código fuente de un programa. El movimiento del Software libre hace especial énfasis en los aspectos morales o éticos del Software, viendo la excelencia técnica como un producto secundario de su estándar ético. El movimiento de código abierto ve la excelencia técnica como el objetivo prioritario, siendo

el compartir el código fuente un medio para dicho fin. Por dicho motivo, la FSF se distancia tanto del movimiento de código abierto como del término «código abierto (Open Source)».

Puesto que la «iniciativa de Software libre Open Source Initiative (OSI)» sólo aprueba las licencias que se ajustan a la «definición de código abierto (Open Source Definition)», la mayoría de la gente lo interpreta como un esquema de distribución, e intercambia libremente "código abierto" con "Software libre". Aún cuando existen importantes diferencias filosóficas entre ambos términos, especialmente en términos de las motivaciones para el desarrollo y el uso de tal Software, raramente suelen tener impacto en el proceso de colaboración.

Aunque el término "código abierto" elimina la ambigüedad de libertad frente a precio (en el caso del inglés), introduce una nueva: entre los programas que se ajustan a la definición de código abierto, que dan a los usuarios la libertad de mejorarlos y los programas que simplemente tienen el código fuente disponible, posiblemente con fuertes restricciones sobre el uso de dicho código fuente. Mucha gente cree que cualquier Software que tenga el código fuente disponible es de código abierto, puesto que lo pueden manipular, sin embargo, mucho de este Software no da a sus usuarios la libertad de distribuir sus modificaciones, restringe el uso comercial, o en general restringe los derechos de los usuarios.

2.4.1 Licencias Creative Commons

Las Licencias[7] Creative Commons (CC) de forma general no tienen una definición oficial, sin embargo, entre las muchas definiciones aceptadas están la de la UNESCO, la cual expresa la siguiente descripción:

Las Licencias Creative Commons (CC) son modelos de contratos que sirven para otorgar públicamente el derecho de utilizar una publicación protegida por los derechos de autor. Entre menos restricciones implique una licencia, mayores serán las posibilidades de utilizar y distribuir un contenido. Las Licencias CC permiten a cualquier usuario descargar, copiar, distribuir, traducir, reutilizar, adaptar y desarrollar su contenido sin costo alguno.

Sin embargo, en la Web oficial de la Organización Creative Commons se nos dice sobre las mismas lo siguiente:

[7] Las licencias de Creative Commons son más utilizadas para cualquier creación digital que para el Software, entendiendo como creación digital desde fotos, artículos en blogs, música, vídeos, este trabajo, etc.

Las Licencias Creative Commons (CC) brindan a todos, desde creadores individuales hasta grandes instituciones, una forma estandarizada de otorgar permiso al público para usar su trabajo creativo bajo la ley de derechos de autor. Desde la perspectiva del reutilizador, la presencia de una licencia Creative Commons sobre una obra protegida por derechos de autor responde a la pregunta: ¿Qué puedo hacer con esta obra?.

Las «Licencias Creative Commons» que hoy en día pertenecen a la organización mundial Creative Commons[8], y buscan regularizar y mantener, de forma equilibrada y satisfactoria, todo lo relacionado con el derecho de utilizar una publicación protegida por los derechos de autor a nivel mundial, han logrado un buen trabajo, sin duda alguna. Y seguramente en el tiempo, se irán adaptando a las nuevas realidades sociales y tecnológicas para poder seguir manteniendo de forma armónica las posibilidades de utilizar y distribuir cualquier contenido libre y abierto sobre la Internet, y más allá.

¿Cuáles son y cómo funcionan o para qué se usan? Las 7 distintas Licencias Creative Commons son las siguientes:

CC BY Esta licencia permite a los reutilizadores distribuir, remezclar, adaptar y desarrollar el material en cualquier medio o formato, siempre que se otorgue la atribución al creador. La licencia permite el uso comercial.

CC BY-SA Esta licencia permite a los reutilizadores distribuir, remezclar, adaptar y desarrollar el material en cualquier medio o formato, siempre que se otorgue la atribución al creador. La licencia permite el uso comercial. Si remezcla, adapta o construye sobre el material, debe licenciar el material modificado bajo términos idénticos.

CC BY-NC Esta licencia permite a los reutilizadores distribuir, remezclar, adaptar y desarrollar el material en cualquier medio o formato, únicamente con fines no comerciales y siempre que se otorgue la atribución al creador.

[8]Una organización mundial sin ánimo de lucro que permite compartir y reutilizar la creatividad y el conocimiento mediante el suministro de herramientas legales gratuitas. Y cuyas herramientas legales (licencias) ayudan a quienes quieren fomentar la reutilización de sus obras ofreciéndolas para su uso bajo términos generosos y estandarizados; a quienes quieren hacer usos creativos de las obras; y a quienes quieren beneficiarse de esta simbiosis.

CC BY-NC-SA Esta licencia permite a los reutilizadores distribuir, remezclar, adaptar y desarrollar el material en cualquier medio o formato, únicamente con fines no comerciales y siempre que se otorgue la atribución al creador. Si remezcla, adapta o construye sobre el material, debe licenciar el material modificado bajo términos idénticos.

CC BY-ND Esta licencia permite a los reutilizadores copiar y distribuir el material en cualquier medio o formato únicamente en forma no adaptada, y siempre y cuando se otorgue la atribución al creador. La licencia permite el uso comercial.

CC BY-NC-ND Esta licencia permite a los reutilizadores copiar y distribuir el material en cualquier medio o formato únicamente en forma no adaptada, únicamente con fines no comerciales y siempre que se otorgue la atribución al creador.

CC0 (CC Cero) Esta licencia es una herramienta de dedicación pública que permite a los creadores renunciar a sus derechos de autor y poner sus obras en el dominio público mundial. CC0 permite a los reutilizadores distribuir, remezclar, adaptar y desarrollar el material en cualquier medio o formato, sin condiciones.

2.4.2 Nuevas Licencias para Responder a Nuevas Necesidades

El mundo de la tecnología avanza mucho más rápido que las leyes y estas tienen que esforzarse para alcanzarlo. En el caso del Software libre y de código abierto, tanto la Free Software Foundation como la Open Source Initiative (los organismos encargados de regular las diferentes licencias) enfrentan periódicamente el problema de cómo mantener sus principios y al mismo tiempo evitar que alguien se aproveche indebidamente.

En el último tiempo, la Open Source Initiative le dio el sello de aprobación a otras nuevas licencias para propósitos específicos.

Nuevas Licencias de Código Abierto

- Cryptographic Autonomy License version 1.0 (CAL-1.0)

Fue creada en el 2019 por el equipo del proyecto de código abierto Holochain, esta licencia fue desarrollada para ser utilizada con aplicaciones criptográficas distribuidas. El inconveniente con las licencias tradicionales es que no obligaba a compartir los datos. Esto podría perjudicar el funcionamiento de toda la red. Por eso la CAL también incluye la obligación de proporcionar a terceros los permisos y materiales necesarios para utilizar y modificar el Software de forma independiente sin que ese tercero tenga una pérdida de datos o capacidad.

- Open Hardware Licence (OHL)

De la mano de la Organización Europea para la Investigación Nuclear (CERN) llegó esta licencia con tres variantes enfocadas en la posibilidad de compartir libremente tanto Hardware como Software.

Hay que hacer una aclaración. La OSI fue creada en principio pensando en el Software por lo que no tiene mecanismos para la aprobación de licencias de Hardware. Pero, como la propuesta del CERN se refiere a ambos rubros, esto posibilitó la aprobación:

- CERN-OHL-S es una licencia fuertemente recíproca: El que utilice un diseño bajo esta licencia deberá poner a disposición las fuentes de sus modificaciones y agregados bajo la misma licencia.

- CERN-OHL-W es una licencia débilmente recíproca: Sólo obliga a distribuir las fuentes de la parte del diseño que fue puesta originalmente bajo ella. No así los agregados y modificaciones.

- CERN-OHL-P es una licencia permisiva: Permite a la gente tomar un proyecto, relicenciarlo y utilizarlo sin ninguna obligación de distribuir las fuentes.

Hay que decir que la gente del CERN parece haber encontrado la solución a un problema que viene afectando a algunos proyectos de código abierto. Una gran empresa utiliza ese proyecto para comercializar servicios y no solo hace ningún aporte al proyecto original (ya sea con código o dando apoyo financiero) si no que también compite en el mismo mercado.

Post Open Zero-Cost en el año 2024 se desarrolla la licencia «Post Open Zero-Cost» con la cual busca abordar los desafíos surgidos en la interacción entre desarrolladores de código abierto y empresas comerciales, especialmente en lo que respecta a la compensación justa por el uso comercial del código.

La característica distintiva de la licencia «Post-Open» en comparación con las licencias abiertas existentes, como la GPL, es la introducción de un componente contractual que puede ser rescindido en caso de violación de los términos. Esta licencia ofrece dos tipos de acuerdos contractuales: gratuitos y de pago. El acuerdo de pago permite negociar derechos adicionales para la distribución comercial de productos o modificaciones sin requerir su divulgación pública.

Las situaciones que podrían llevar a la terminación del acuerdo contractual incluyen: violación de los términos de la licencia; reclamaciones por infracción de patentes; imposición de condiciones adicionales (como sanciones en contratos con clientes por divulgación de información sensible); cambios sujetos a leyes de control de exportaciones; ocultamiento de información sobre vulnerabilidades; y uso del código para entrenamiento de modelos de aprendizaje automático bajo términos no permitidos. Las relaciones contractuales no se rescinden de inmediato, sino que se notifica la infracción y se otorgan 60 días para corregirla antes de la rescisión efectiva del acuerdo.

Uno de los problemas que la nueva licencia busca abordar está relacionado con las limitaciones de la GPL, la cual se centra en otorgar derechos sin la capacidad de revocarlos, lo que permite a las empresas encontrar formas de eludir sus requisitos, especialmente en lo que respecta al acceso al código fuente. Estas lagunas son utilizadas para restringir la disponibilidad del código subyacente en productos comerciales mediante la imposición de términos contractuales adicionales con los usuarios finales.

Un claro ejemplo es el de RHEL, la cual los clientes firman un acuerdo con Red Hat que limita la redistribución del código fuente al imponer condiciones sobre la coincidencia de las copias instaladas y compradas de RHEL. Esto coloca a los usuarios en la disyuntiva entre su libertad para disponer del software y mantener su estatus de cliente de Red Hat. Aunque la GPL permite la distribución de parches que solucionan vulnerabilidades en el código de RHEL, esto podría interpretarse como una violación del acuerdo con Red Hat y podría resultar en la terminación de los servicios por parte de la empresa.

2.5 Implicaciones Económico-Políticas del Software Libre

Una vez que un producto de Software libre ha empezado a circular, rápidamente está disponible a un costo muy bajo. Al mismo tiempo, su utilidad no decrece. El Software, en general, podría ser considerado un bien de uso inagotable, tomando en cuenta que su costo marginal es pequeñísimo y que no es un bien sujeto a rivalidad -la posesión del bien por un agente económico no impide que otro lo posea-.

2.5.1 Software Libre y la Piratería

Puesto que el Software libre permite el libre uso, modificación y redistribución, a menudo encuentra un hogar entre usuarios para los cuales el coste del Software no libre es a veces prohibitivo, o como alternativa a la piratería. También es sencillo modificarlo localmente, lo que permite que sean posibles los esfuerzos de traducción a idiomas que no son necesariamente rentables comercialmente, además:

- Porque así no se condiciona a los usuarios a usar siempre lo mismo.
- Porque así no se fomenta la piratería en los usuarios al no pagar licencias.
- Porque así no se obliga a usar una solución concreta y se ofrece libertad de elección a los usuarios.
- Porque es mucho más seguro ya que el Software libre es público y se puede ver qué hace exactamente sin recelos.

La mayoría del Software libre se produce por equipos internacionales que cooperan a través de la libre asociación. Los equipos están típicamente compuestos por individuos con una amplia variedad de motivaciones y pueden provenir tanto del sector privado, del sector voluntario o del sector público. En los últimos años se ha visto un incremento notable de grandes corporativos (como IBM, Microsoft, Intel, Google, Samsung, Red Hat, etc.) que han dedicado una creciente cantidad de recursos humanos y computacionales para desarrollar Software libre, ya que esto apoya a sus propios negocios.

2.5.2 ¿Cuánto Cuesta el Software Libre?

En esta sección intentaremos dar una idea de cuál es el costo del desarrollo del Software Libre, por supuesto que no se tratará más que de una conjetura aproximada basada en las cifras proporcionadas por desarrolladores de Software comercial (al año 2020).

Gratis no Significa Gratuito Supongamos que todos los recursos humanos participantes en el desarrollo de un proyecto de Software libre lo hagan de forma voluntaria. De todas formas tenemos lo que los contables llaman «Costo de oportunidad» esto es, los ingresos que podrían haber generado esas personas si hubieran dedicado el tiempo y los conocimientos invertidos en el proyecto a uno en el que les pagaran. Así, el calcular el costo promedio por hora que cobra un programador, por la cantidad de horas invertidas al proyecto, nos da un razonable costo mínimo. Lo mismo puede hacerse con los voluntarios dedicados a la difusión en las redes. El costo de una campaña de marketing digital puede estimarse fácilmente.

Muchos proyectos de código abierto como una distribución Linux, son construidos a partir de la integración de otros proyectos, por los que sus costos de desarrollo también deberían sumarse.

Por otra parte, necesitamos recursos físicos. Aún cuando los voluntarios trabajen desde su casa, siguen teniendo que comprar y mantener sus equipos, además de pagar la electricidad que los hace funcionar.

Bases para el Cálculo Hay muchos factores que determinan el costo de desarrollar una pieza de Software. En un extremo tenemos una aplicación simple que requiere muy poca interacción del usuario o procesamiento del lado del servidor. Tal es el caso de un cliente de escritorio para redes sociales. Por el otro sistemas operativos que deben operar en múltiples plataformas realizando múltiples tareas (por ejemplo Debían que aspira a ser el sistema operativo universal). Sin embargo, el costo de una aplicación simple puede elevarse debido a que tiene múltiples pantallas diferentes. Por ejemplo un juego desarrollado con HTML5 y Javascript.

Los dos aspectos claves son la cantidad de horas de trabajo necesarias y las tecnologías involucradas. Para una aplicación de escritorio como un procesador de textos con las prestaciones habituales, optimizado para un determinado escritorio Linux, se estima que se tendría que contar con al menos el equivalente a 42,000 euros en trabajo voluntario. Un gestor de contenidos

para comercio electrónico con seguimiento de pedidos e integración con las principales plataformas de pago implicaría desembolsar unos 210,000 euros o su equivalente en trabajo voluntario.

Tomando en cuenta que este cálculo incluye lo que costó el desarrollo de las bibliotecas y otros proyectos libres y de código abierto incluidos, pero no los gastos que efectivamente deben desembolsarse en efectivo como la compra de equipos, Software de seguridad y desarrollo y el pago de electricidad e internet.

Por otro lado, el proceso de medición de costes del Software es un factor realmente importante en el análisis de un proyecto. Hay distintos métodos de estimación de costes de desarrollo de Software (también conocido como métrica del Software). La gran mayoría de estos métodos se basan en la medición del número de Líneas de Código que contiene el desarrollo (se excluyen comentarios y líneas en blanco de los fuentes).

Desarrollo de Fedora 9 La Linux Foundation ha calculado que costaría desarrollar el código de la distribución Fedora 9 que fue puesta a disposición del público el 13 de mayo de 2008, en el informe citado "Estimating the Total Development Cost of a Linux Distribution" se calcula que Fedora 9 tiene un valor de 10.8 mil millones de dólares y que el coste únicamente del Kernel (2.6.25 con 8,396,250 líneas de código) tendría un valor de 1.4 mil millones de dólares.

Esta distribución tiene unas 205 millones de líneas de código, el proyecto debería ser desarrollado por 1000 - 5000 desarrolladores (el trabajo invertido por una única persona desarrollándolo se alargaría durante unos 60.000 años) y esa estimación no va muy desencaminada ya que en los 2 últimos años del desarrollo de esa versión contribuyeron unos 3,200 desarrolladores aunque el número de trabajadores en la historia de la distribución es mucho mayor.

¿Qué pasa con GNU/Linux? en el año 2015 (las estadísticas más actuales que conseguimos) la Linux Foundation analizó el costo de desarrollo del núcleo. Combinando el aporte de los recursos humanos (voluntarios y de pago) y los desembolsos necesarios, la cuenta sumó 476,767,860,000.13 euros.

Todos sabemos que el hecho de tener desarrolladores asalariados no garantiza necesariamente Software de calidad. Pero, tener desarrolladores que pueden dedicar toda su atención a un proyecto en lugar de hacerlo en sus horas libres si lo hace. Lamentablemente, por el momento el único modo de

lograr eso es obtener el apoyo de corporaciones (Intel, Google, IBM, AMD, Sun Microsystems, Dell, Lenovo, Asus, HP, SGI, Oracle, RedHat, etc.) que solo lo hacen con los proyectos que son de su interés como el Kernel de Linux, hay que notar que para el Kernel de Linux un porcentaje importante (más del 10 %) lo hacen programadores independientes.

Costes Recordemos que la segunda de las cuatro libertades de un programa para ser Software libre es:

- Libre redistribución

y esta puede ser a través de un pago o sin costo. Es por ello que existen distintas empresas, organizaciones y usuarios que pueden apoyar a los usuarios finales en el desarrollo y soporte de algún programa de Software libre o una distribución personalizada de Linux por un costo determinado.

Mucha gente, en especial ejecutivos de empresas, se acercan a Linux bajo la promesa de que es una solución de bajo costo -muchos piensan que incluso es gratis-. Pero la realidad es que detrás de Linux y los programas de Software libre (y aquí la traducción correcta de la palabra inglesa, Free es libre, no gratis) pueden llevar una serie de costos ocultos que deben ser considerados al momento de decidir si se implementa una solución propietaria o una libre.

Los costos ocultos aparecen cuando se intenta instalar y capacitar en el uso de algún Software y se necesita la ayuda de un informático, al que se tiene que pagar, o alguna empresa quiere personalizar la interfaz de un programa y necesita la ayuda de un programador, que también tienen un coste, por lo que finalmente el comentario suele ser "el Software libre no es barato".

El primer punto a considerar al evaluar ambas alternativas es el costo de la licencia. Los productos de Software libre no suelen tener un costo de licencia asociado, que sí existe en los programas propietarios. De hecho es allí en donde los fabricantes de Software recargan sus costos de investigación y desarrollo, de producción e incluso sus ganancias. En este primer punto el ganador claro es el Software libre y es lo que los adeptos de este esquema publicitan: "su compañía puede ahorrar miles de dólares al año usando Software libre".

El segundo punto a considerar es el costo de instalación, configuración y capacitación. Dependiendo de su complejidad, algunos productos comerciales no contemplan costos extra por este concepto y otros -como Windows, por ejemplo- son tan populares que se puede encontrar numerosas opciones

de instalación -a través de empresas o profesionales- donde escoger en el mercado. A veces en el Software libre la configuración puede implicar recompilar el producto con algunas opciones particulares, algo que sólo pueden realizar técnicos con un nivel adecuado de conocimientos y que puede que no sean tan fáciles de encontrar. Aquí por lo general la ventaja va hacia el Software comercial.

Una vez instalado el Software, toca realizar actualizaciones de mantenimiento. Si bien es cierto lo que algunos de los fanáticos del Software libre dicen, que nadie lo obliga a mejorar el Software con que cuenta, la realidad -especialmente en lo que a seguridad se refiere- obliga a las empresas a mantener su Software actualizado. Aún así, los costos de actualizar Software libre suelen ser significativamente más bajos que los de productos comerciales y suelen ser menos exigentes con el Hardware necesario para ejecutarlos. La mayoría de las veces la ventaja es para el Software libre, pero hay que evaluar ya que varía dependiendo de cada caso.

Por último hay algunas casas que desarrollan productos de Software libre -dan el código y permiten que cualquiera lo modifique o reutilice- pero fijan contratos con cargos mensuales o anuales para mantenimiento del mismo, algo que se parece mucho a un cobro por licencia, por lo que hay que estar seguro de conocer todas las condiciones cuando una empresa nos ofrece una solución propia y la califica como Software libre.

Además de estas consideraciones hay una que debe sumarse eventualmente a esta evaluación: el costo de migrar a otra solución. En Software libre suelen usarse estándares abiertos para almacenar los datos, lo que facilita las migraciones. En cambio muchas soluciones propietarias suelen tener formatos propietarios que pueden dejar "amarrados" los datos de la empresa a una aplicación específica.

Sólo después de evaluar estos aspectos del Software, que pueden tener implicaciones importantes en el presupuesto, es que un CIO (Chief Information Officer) puede decir si una solución de Software libre le conviene más a una empresa o no, algo que va más allá de que la aplicación sea gratis o no.

2.5.3 La Nube y el Código Abierto

Desde hace años se han creado nuevos desafíos para el código abierto que plantea la nube, un término que para el usuario promedio puede significar cosas diferentes, pero que para la empresa se resume en servicios. Y es que los beneficios económicos que genera el mero Software de código abierto no

son comparables a los que se obtienen cuando se ofrece ese mismo Software a través de servicios, más allá -pero incluyendo- del soporte.

Este hecho diferencial lleva tiempo provocando fricciones entre desarrolladores y proveedores y hay quien adelantó incluso el fin del modelo de desarrollo del código abierto tal y como lo conocemos. ¿Quién tiene la razón?, ¿es para tanto la situación?.

En sus exposiciones, representantes de compañías y proyectos de código abierto muy populares en el ámbito empresarial, explican el supuesto perjuicio que les ocasiona el uso que los grandes proveedores de servicios en la nube hacen del Software que ellos desarrollan y cómo algunos han considerado y aplicado un enfoque más cerrado para sus productos con el fin de evitar lo que denominan como expolio. Hay declaraciones que merecen ser rescatadas para dotar de contexto a la discusión:

- El papel que juega el código abierto en la creación de oportunidades comerciales ha cambiado, durante muchos años les permitimos que las empresas de servicios tomaran lo que se ha desarrollado y ganasen dinero con ello.

- Empresas como Amazon Web Services, Azure de Microsoft, etc. Han ganado cientos de millones de dólares ofreciendo a sus clientes servicios basados en Software libre sin contribuir tanto a la comunidad de código abierto que construye y mantiene ese proyecto. Es imposible saber exactamente de cuánto dinero estamos hablando, pero es cierto que los proveedores de la nube se benefician del trabajo de los desarrolladores de código abierto que no emplean.

- Hay un mito ampliamente instalado en el mundo de código abierto que dice que los proyectos son impulsados por una comunidad de contribuyentes, pero en realidad, los desarrolladores pagados contribuyen con la mayor parte del código en la mayoría de los proyectos de código abierto modernos.

En resumen, todas estas voces se quejan de dos cosas: los grandes beneficios que obtienen los proveedores de servicios en la nube con su Software sin retribuirles en consecuencia, y la falta de colaboración manteniendo los productos con los que lucran. Sin embargo, no nos engañemos, el quid de la cuestión está principalmente en el dinero: la opinión generalizada de la

comunidad es que el Software de código abierto nunca fue pensado para que las empresas de servicios en la nube lo tomasen y lo vendieran.

Por otro lado, si es posible bifurcar un proyecto libre que se cierra, ¿no hubiese sido mejor colaborar con él antes y haber evitado el cierre? Si no se invierte y se mantiene con salud aquello que da beneficios, puede terminar por desaparecer. A medio camino entre el depredador y el parásito: así es como ven muchas desarrolladoras de código abierto a los proveedores de servicios en la nube.

Vale la pena retomar ahora la frase «el Software de código abierto nunca fue pensado para que las empresas de servicios en la nube lo tomasen y lo vendieran». ¿Para qué fue pensado el código abierto entonces? No hay ninguna licencia de código abierto o Software libre reconocida por la Free Software Foundation o la Open Source Initiative que prohíba hacer negocio con el Software. Lo que prohíben es la discriminación en la capacidad y alcance de su uso en función de la parte, se trate de un individuo o de la mayor multinacional imaginable.

¿Cuál es la solución a un embrollo de tamaña envergadura? Lo único claro es que no es una cuestión de blancos y negros y las consideraciones son demasiadas como para seguir ahondando: empresas que cotizan en bolsa quieren más dinero de otras compañías -que también cotizan en bolsa y por mucho más-, que además del Software ponen la infraestructura sobre la que distribuyen sus ofertas y que tienen la capacidad de clonar tu producto en un abrir y cerrar de ojos, no solo porque tienen el capital, sino porque tienen la experiencia necesaria tras contribuir técnica, pero también en muchos casos, económicamente, durante largo tiempo.

Pese a ello, esta situación está alterando el paradigma actual, en el que el modelo de desarrollo del código abierto se ha impuesto como impulsor de la innovación en el sector empresarial y ya hay quien habla de que nos acercamos al fin, o al principio del fin de la Era del Open Source, cuya preponderancia estaría sentenciada por la revolución de la nube, a la postre el mayor estímulo que haya tenido el código abierto hasta la fecha.

El futuro, pues, pasaría por el Shared Source Software, bajo el cual diferentes compañías con intereses alineados colaborarán en el desarrollo de proyectos concretos, pero limitando su explotación comercial a sí mismas. Todavía no estamos ahí, no obstante, y no parece tampoco que el relevo se vaya a dar en breve. De suceder, será muy llamativo: la muerte del código abierto por un éxito mal entendido.

2.5.4 El Código Abierto como Base de la Competitividad

En septiembre del 2021, se publicó un amplio y detallado informe llevado a cabo por el Fraunhofer ISI y por OpenForum Europe para la Comisión Europea, «The impact of open source software and hardware on technological independence, competitiveness and innovation in the EU economy», cuantifica la importancia económica del código abierto aplicado tanto al Software como al Hardware, su efecto en la contribución al producto interior bruto generado, la reducción en aspectos como el coste total de propiedad, dependencia del proveedor y autonomía digital; lanza una serie de recomendaciones específicas de políticas públicas destinadas a lograr un sector público digitalmente autónomo, una investigación y desarrollo abierto que fomente el crecimiento europeo y una industria más digitalizada y competitiva.

En las estimaciones del informe se apunta que las empresas europeas invirtieron alrededor de mil millones de euros en Software de código abierto en 2018, lo que resultó en un impacto en la economía europea de entre 65,000 y 95,000 millones de euros. El análisis estima una relación costo-beneficio superior a 1:4 y predice que un aumento del 10% de las contribuciones de a repositorios de código abierto sería susceptible de generar anualmente entre un 0.4% y un 0.6% adicional en el PIB, así como más de seiscientas nuevas empresas tecnológicas en la Unión Europea.

El análisis de las contribuciones a repositorios de Software de código abierto en la Unión Europea revela que el ecosistema tiene una naturaleza diferente frente al norteamericano, con un volumen de contribuciones que provienen sobre todo de empleados de compañías pequeñas o muy pequeñas, frente a un escenario en los Estados Unidos en el que predominan grandes compañías tecnológicas que se benefician en sus modelos de negocio de la gran cantidad y de la rápida mejora del Software disponible. En Europa, los contribuyentes individuales ascendieron a más de 260,000, lo que representa el 8% de los casi 3.1 millones de empleados de la UE en el sector del desarrollo de Software en 2018. En total, los más de 30 millones de desarrollos consolidados en repositorios en los estados miembros de la Unión Europea representan una inversión de personal equivalente a casi mil millones de euros, que han pasado a estar disponibles en el dominio público y que, por lo tanto, no tienen que ser desarrollados por otros actores.

Según el análisis, cuanto más pequeña es la empresa, mayor es la inversión relativa en Software de código abierto (las empresas con 50 empleados o menos asumieron casi la mitad de los desarrollos en la muestra de las com-

pañías más activas). Aunque más del 50% de los contribuyentes pertenecen a la industria tecnológica (el 8% del total de sus empleados participaron en estos desarrollos), también hubo participación significativa de empresas de consultoría, científicas, técnicas y, en menor medida, de distribuidores, minoristas y empresas del ámbito financiero.

¿Puede una filosofía de desarrollo como el código abierto, disponible para todo el mundo, llegar a convertirse en una fuente de ventajas diferenciales para el resto de los países, que se ha visto tradicionalmente muy superado en su relevancia en el entorno tecnológico por los gigantes tecnológicos de Estados Unidos o de China? El informe afirma que su uso puede llegar a incidir en gran medida en el desarrollo de una independencia tecnológica superior, de una mayor competitividad y de más innovación. Veremos si llegamos a ver en el resto del mundo políticas que incentiven el uso del código abierto como una variable estratégica clave para ello. La idea, capitalizar la tecnología de una forma más orientada al procomún y al desarrollo colaborativo, suena sin duda atractiva e interesante.

2.5.5 Software Libre en Empresas y Corporaciones

En esta sección exploraremos algunas de las claves por las cuales el Software libre está hoy en el punto de mira de todo tipo de empresas y grandes corporaciones (algunas de las cuales ayer eran sus acérrimos enemigos). Pero hay que empezar destacando que corporativos como Google, Amazon Web Services, Azure de Microsoft, Microsoft, IBM, entre otras, en los últimos años han ganado miles de millones de dólares ofreciendo a sus clientes servicios y/o productos basados en Software libre y con una queja recurrente por su pobre o nula contribución a la comunidad de código abierto que construyó y mantiene esos proyectos.

Si bien es imposible saber exactamente de cuánto dinero estamos hablando, pero es cierto que empresas y corporaciones se benefician diariamente del trabajo de los desarrolladores de código abierto que no emplean.

Grandes Equipos de Programadores GNU/Linux ha demostrado que los equipos de desarrolladores grandes, distribuidos, aunque desorganizados pueden crear Software viable.

Antes de la llegada de GNU/Linux, la mayoría del Software era desarrollado por pequeños equipos de programadores que trabajaban en estrecha coordinación entre sí. Ese era el enfoque recomendado por informáticos de hace

unos decenios, que advertían que añadir más programadores a un proyecto tendía a disminuir su eficiencia. Y estaban muy equivocados.

Desde el principio, el Kernel de Linux se desarrolló con un enfoque diferente, en el que programadores de todo el mundo, que en la mayoría de los casos no se conocían, escribieron e integraron el código de forma rápida y poco organizada. Gracias a la publicación temprana y frecuente, consiguieron que funcionara y hoy día es el Kernel más usado en informática en supercomputadoras y dispositivos móviles.

Pero actualmente hay un mito ampliamente instalado en el mundo de código abierto, que dice que los proyectos son impulsados por una comunidad de contribuyentes gratuitos, pero en realidad, los desarrolladores pagados contribuyen con la mayor parte del código en la mayoría de los proyectos de código abierto modernos de los cuales las corporaciones pueden sacar provecho. Claro ejemplo es el propio Kernel de Linux, en el cual una gran cantidad de desarrolladores actuales pertenecen o son subvencionados por empresas, fundaciones o corporaciones (actualmente cientos de ellas), como en el caso de Linus Torvalds que trabaja bajo los auspicios de la Fundación de Linux.

Reutilización de Software Parte de la razón por la que Linux se hizo muy popular entre los ingenieros de Software con relativa rapidez fue que Linux -y el Software libre en general- facilita la reutilización del código escrito por otras personas.

Hoy en día, la reutilización de Software de terceros es habitual, incluso entre los equipos de desarrollo cuyos productos no son de código libre. Es difícil imaginar la construcción de una aplicación hoy en día sin hacer uso de las bibliotecas de Software de origen, las API de terceros u otros recursos externos a su propio proyecto.

Es cierto que proyectos como GNU, que precedió al Kernel de Linux en siete años, promovían la reutilización de código antes de que apareciera el núcleo. Pero, podría decirse que Linux fue el proyecto que trajo las prácticas de codificación libre a la corriente que tanto parece interesar a ciertas grandes empresas, ayudando a crear el modelo de ingeniería de Software de componentes de Software modulares y reutilizables.

Gestión Actual del Código Fuente Linus Torvalds, que creó el núcleo de Linux cuando era estudiante en Helsinki, es el más famoso por ese trabajo.

Pero un hecho a menudo olvidado es que Torvalds es también el padre de Git, el masivamente popular gestor de código fuente libre.

Torvalds creó Git para ayudar a gestionar el código fuente de Linux. Si Linux no existiera, tampoco existiría Git. Tampoco existiría GitHub, ni GitLab, ni GitOps. Y, lo que es más importante, sin la idea de Software libre y colaborativo de Richard Stallman, tampoco existiría la cultura de intercambio y colaboración abierta que sostienen estas tecnologías.

Estrategias de Despliegue de Software "App Store" Apple puede atribuirse el mérito de haber lanzado la primera App Store, un lugar donde los desarrolladores pueden compartir aplicaciones y los usuarios pueden instalarlas fácilmente, utilizando un catálogo Online centralizado.

Pero al igual que con muchas cosas que ha hecho Apple, el concepto de App Store (que ahora es una estrategia de despliegue de Software apilado como servicio, especialmente pero no sólo en el ecosistema móvil) se parece mucho a lo que los desarrolladores de GNU/Linux estaban haciendo a través de los repositorios de Software mucho antes de que las tiendas de aplicaciones se convirtieran en algo común en el mundo del Software propietario, como también lo es la Tienda de Windows.

Los repositorios de Software en GNU/Linux hacen más o menos lo mismo que las tiendas de aplicaciones: Permiten a los usuarios seleccionar las aplicaciones que quieren de una lista centralizada y en línea, y luego instalarlas con unos pocos clics o bien órdenes de terminal (como el famoso *apt* de Debian).

Es cierto que empresas como Apple parecen tener el mérito de crear tiendas de aplicaciones muy fáciles de usar de hacer clic e ir, pero no es un invento de ellos. Y la historia del concepto de tienda de aplicaciones en general implica a más actores que sólo la comunidad de Apple. Aún así, creo que se podría argumentar con fuerza que, sin GNU/Linux y los repositorios de Software de GNU/Linux, las tiendas de aplicaciones tal y como las conocemos hoy no existirían.

Formatos Abiertos para Intercambio de Información Hay una gran variedad de tecnologías disponibles para producir y almacenar datos. Como son: hojas de cálculo, bases de datos, Software estadístico más específico y más. Esto genera una enorme diversidad de formatos, a veces esto es por decir lo menos caótico.

La ventaja de los archivos de formatos abiertos, es que permiten a los de-

sarrolladores producir varios paquetes de Software y servicios utilizando esos formatos. Esto entonces reduce al mínimo los obstáculos para la reutilización de la información que contienen.

El advenimiento del Software libre ha generado algunos de los formatos abiertos más usados para el intercambio de información, pero los entes generadores de información no siempre se adecuan a los niveles de apertura deseados, algunos de estos formatos abiertos son: XML, JSON, YAML, RDF, REBOL, PDF, CSV, ODF, OOXML, TXT, HTML, HDF.

Pero, incluso si la información se proporciona en formato electrónico, formato legible por máquina y en detalle, puede existir problemas relacionados con el formato del archivo en sí (principalmente el generado por los diversos sistemas operativos). Los formatos en los cuales la información es publicada -en otras palabras, la base digital en la cual la información es almacenada- puede ser "abierta" o "cerrada".

Un formato abierto es aquel donde las especificaciones del Software están disponibles para cualquier persona, de forma gratuita, así cualquiera puede usar dichas especificaciones en su propio Software sin ninguna limitación en su reutilización que fuere impuesta por derechos de propiedad intelectual.

Si el formato del archivo es "cerrado", esto puede ser debido a que el formato es propietario y sus especificaciones no están disponibles públicamente, o porque el formato es propietario y aunque las especificaciones se han hecho públicas, su reutilización es limitada. Si la información es liberada en un formato de archivo cerrado, esto puede causar grandes obstáculos para reutilizar la información codificada en él, forzando a aquellos que deseen usar la información a comprar Software innecesario.

El uso de formatos de archivo con propiedad, para el que la especificación no está disponible públicamente, puede crear dependencia de Software de terceros o de los titulares de licencias de los formatos de archivos. En el peor de los casos, esto puede significar que la información sólo se puede leer con cierto Software específico, que puede ser caro, o que puede quedar obsoleto.

La preferencia del término Gobierno de Datos Abiertos, es que la información se publicará en formatos de archivo abiertos, los cuales son de lectura mecánica y esto es una aportación más del Software libre.

Ciencia Abierta La ciencia abierta (Open Science) es el movimiento creciente para hacer que la ciencia sea abierta. La ciencia en sí misma se utilizó como un ejemplo principal de la eficacia del movimiento de código abierto, ci-

tando prácticas como la difusión abierta de información, métodos y revisión por pares de la literatura científica. Podría decirse que la ciencia abierta comenzó en el siglo XVII con el advenimiento de la revista científica y la práctica de repetir los experimentos presentados en los artículos académicos. Estas revistas se imprimirían y distribuirían en todo el mundo, a menudo supervisadas por sociedades científicas como la Royal Society.

¿Qué impulsó la necesidad de un movimiento de ciencia abierta? La Royal Society tenía el famoso lema "Nullius in verba", traducido de forma aproximada como "no tome la palabra de nadie". Esto encarnaba un principio general en la ciencia de que todas las teorías están abiertas a ser cuestionadas y los resultados declarados deben ser repetibles. De hecho, es una práctica generalizada que fue realizada por la sociedad en esos primeros años. En los últimos años esta práctica no ha sido tan común, con más y más ciencia confiando en elementos cerrados, lo que en última instancia conduce a errores que son más difíciles de detectar sin un intercambio completo de información: datos, métodos y publicaciones.

El movimiento de ciencia abierta afirma en términos generales que la ciencia debe realizarse de manera abierta y reproducible donde todos los componentes de la investigación estén abiertos. Muchas revistas permanecen estancadas en un formato en el que se imprimían físicamente, a pesar de que en la actualidad se distribuyen en gran medida en línea. A menudo, todavía utilizan archivos PDF como una forma de "papel electrónico" con publicaciones fijas, procesos cerrados de revisión por pares y poco o ningún acceso a los datos. Este fue sin duda el modo más eficiente de difundir el conocimiento científico antes de los albores de internet, pero ahora un número cada vez mayor lo considera lejos de ser el óptimo.

La ciencia abierta encarna una serie de aspectos, en el núcleo esto incluye acceso abierto, datos abiertos, código abierto y estándares abiertos que ofrecen una diseminación sin restricciones del discurso científico. Estas cosas permiten una ciencia reproducible al brindar acceso completo a los componentes principales de la investigación científica. Hay una serie de componentes adicionales que también se están explorando, como la revisión por pares abierta, donde los revisores de publicaciones científicas publican revisiones abiertamente con su nombre adjunto y la ciencia de libreta abierta donde las libretas (tradicionalmente cerradas) se publican abiertamente en línea a medida que se realiza la investigación.

¿Por qué la ciencia abierta es tan importante en la era digital? También existe una creciente comprensión de que, dado que la investigación científica

depende cada vez más del código informático para simulaciones, cálculos, análisis, visualización y procesamiento de datos en general, es importante tener acceso a este código tal como tradicionalmente ha sido importante mostrar (y derivar) cualquier nueva técnica matemática introducida para el análisis. Hay revistas como PLOS ONE y F1000 que exploran el significado de las publicaciones, ya sea que se deben congelar en el tiempo o se pueden actualizar. Los repositorios de datos también están ganando importancia a medida que las agencias de financiación requieren la publicación y preservación de los datos generados por la investigación financiada.

En esencia, la ciencia abierta se trata de volver a esos valores fundamentales inculcados por algunos de los primeros científicos de que no debemos confiar en la palabra de nadie, que es esencial que todos los elementos pertinentes a un descubrimiento pretendido se publiquen para que los resultados puedan repetirse y validarse. El movimiento de la ciencia abierta varía en el grado en que lo requiere, pero están surgiendo patrones. Se están estableciendo recomendaciones sobre licencias, como CC0 para datos, CC-BY para publicaciones, licencias compatibles con OSI para código fuente y formatos abiertos para datos. En última instancia, se trata de empoderar a todos para que participen en la ciencia, con internet como vehículo principal para la amplia difusión de este conocimiento.

Este movimiento está cambiando la forma en que se hace la ciencia, está recibiendo el respaldo de muchas agencias de financiamiento, ya que requieren planes de gestión de datos, planes de distribución de código fuente y una mayor validación de los resultados a través del acceso abierto a estos resultados para todos. Esto también mejora la transferencia de conocimientos de la academia a la industria, ya que se brinda acceso completo en el momento de la publicación o después de un período de embargo. El movimiento de la ciencia abierta se limita en gran medida a la investigación que está financiada por las agencias de financiación nacionales de todo el mundo y exige que todos los que financian la investigación tengan acceso total e igualitario a ella.

Open Hardware El concepto de Software libre también se permeó al Hardware. El término Open Hardware u Open Source Hardware, se refiere al Hardware cuyo diseño se hace públicamente disponible para que cualquiera pueda estudiarlo, modificarlo y distribuirlo, además de poder producir y vender Hardware basado en ese diseño. Tanto el Hardware como el Software

que lo habilita, siguen la filosofía del Software libre. Hoy en día, el término "hágalo usted mismo" (DIY por sus siglas en inglés) se está popularizando en el Hardware gracias a proyectos como Arduino que es una fuente abierta de prototipos electrónicos, una plataforma basada en Hardware flexible y fácil de utilizar que nació en Italia en el año 2005.

El movimiento de Hardware abierto o libre, busca crear una gran librería accesible para todo el mundo, lo que ayudaría a las compañías a reducir en millones de dólares en trabajos de diseño redundantes. Ya que es más fácil tener una lluvia de ideas propuesta por miles o millones de personas, que por solo una compañía propietaria del Hardware, tratando así de que la gente interesada entienda cómo funciona un dispositivo electrónico, pueda fabricarlo, programarlo y poner en práctica esas ideas en alianza con las empresas fa-bricantes, además se reduciría considerablemente la obsolescencia programada y en consecuencia evitaríamos tanta basura electrónica que contamina el medio ambiente. Al hablar de Open Hardware hay que especificar de qué tipo de Hardware se está hablando, ya que está clasificado en dos tipos:

- Hardware estático. Se refiere al conjunto de elementos materiales de los sistemas electrónicos (tarjetas de circuito impreso, resistencias, capacitores, LEDs, sensores, etcétera).

- Hardware reconfigurable. Es aquél que es descrito mediante un HDL (Hardware Description Language). Se desarrolla de manera similar a como se hace Software. Los diseños son archivos de texto que contienen el código fuente.

Para tener Hardware reconfigurable debemos usar algún lenguaje de programación con licencia GPL (General Public License). La licencia GPL, al ser un documento que cede ciertos derechos al usuario, asume la forma de un contrato, por lo que usualmente se le denomina contrato de licencia o acuerdo de licencia. La Organización Europea para la investigación Nuclear (CERN) publicó el 8 de julio de 2011 la versión 1.1 de la Licencia de Hardware Abierto.

Existen programas para diseñar circuitos electrónicos y aprender de la electrónica como EDA (Electronic Design Automation) y GEDA (GPL Electronic Design Automation), son aplicaciones de Software libre que permiten poner en práctica las ideas basadas en electrónica.

Es posible realizar el ciclo completo de diseño de Hardware reconfigurable desde una máquina con GNU/Linux, realizándose la compilación, simulación,

síntesis y descarga en una FPGA (Field Programmable Gate Arrays). Para la compilación y simulación se puede usar GHDL (https://ghdl.free.fr) junto con GTKWave (https://gtkwave.sourceforge.net) y para la síntesis el entorno ISE de Xilinx. Este último es Software comercial pero existe una versión gratuita con algunas restricciones.

Sabemos que tanto en el caso del Software como el Hardware, libre no es lo mismo que gratis. Específicamente, en el caso del Hardware, como estamos hablando de componentes físicos que son fabricados, la adquisición de componentes electrónicos puede ser costosa. Aun así, es un campo que no solo es apasionante sino que también tiene mucho futuro y representa grandes oportunidades.

Entusiasmo de la Comunidad Por último, pero no menos importante, probablemente el mayor impacto duradero de GNU/Linux en el modelo de ingeniería de Software se reduce a lo que podría llamarse entusiasmo de la comunidad. Me refiero a la forma en que GNU/Linux en particular, y el Software libre en general, ha animado a los desarrolladores de todo tipo a considerar las contribuciones a la comunidad como uno de sus objetivos finales y esto ahora es notorio no solo en Software, sino en Hardware abierto, obras literarias, escritos técnicos (como este trabajo), imágenes, vídeo, música y un largo etc.

En un mundo de código libre en el que las contribuciones a los proyectos de código pueden ser aceleradores de carrera y el código de licencia libre se reutiliza ampliamente, los desarrolladores entienden que hay un valor real en la construcción de Software que puede beneficiar a tantos usuarios como sea posible.

Tal vez los desarrolladores valorarían a la comunidad en su conjunto si GNU/Linux y el código libre nunca hubieran aparecido. Pero me cuesta imaginar un mundo en el que corporaciones como Microsoft y Google trabajarán juntos en la construcción de Software para GNU/Linux si GNU/Linux no hubiera popularizado el concepto de proyectos de Software impulsados por la comunidad que nadie posee realmente, pero que todos pueden utilizar.

Si bien, es innegable que todo lo anterior puso en la mira de las empresas de todos los tamaños y de las grandes corporaciones el Software libre, la principal razón es el poder utilizar una gran cantidad Software funcional, depurado y ampliamente usado para ofrecer servicios y/o productos basados en Software libre y así beneficiarse económicamente de ello.

Por otro lado, se ha visto a través de múltiples estudios, el impacto y la cuantificación de la importancia económica del código abierto aplicado tanto al Software como al Hardware, su efecto en la contribución al producto interior bruto generado, la reducción en aspectos como el coste total de propiedad, dependencia del proveedor y autonomía digital. Además de generar políticas públicas destinadas a lograr un sector público digitalmente autónomo, una investigación y desarrollo abierto que fomente el crecimiento de los países y una industria más digitalizada y competitiva.

Retomando la frase «el Software de código abierto nunca fue pensado para que las empresas de servicios lo tomasen y lo vendieran». ¿Para qué fue pensado el código abierto, entonces? No hay ninguna licencia de código abierto o Software libre reconocida por la Free Software Foundation o la Open Source Initiative que prohíba hacer negocio con el Software. Lo que prohíben es la discriminación en la capacidad y alcance de su uso en función de la parte, se trate de un individuo o de la mayor multinacional imaginable.

Pese a ello, esta situación está alterando el paradigma actual, en el que el modelo de desarrollo del código abierto se ha impuesto como impulsor de la innovación en el sector empresarial, a la postre el mayor estímulo que haya tenido el código abierto hasta la fecha.

¿Puede una filosofía de desarrollo como el código abierto, disponible para todo el mundo, llegar a convertirse en una fuente de ventajas diferenciales para el resto de los países, que se ha visto tradicionalmente muy superado en su relevancia en el entorno tecnológico por los gigantes tecnológicos de Estados Unidos o de China? Se afirma que su uso puede llegar a incidir en gran medida en el desarrollo de una independencia tecnológica superior, de una mayor competitividad y de más innovación. La idea, capitalizar la tecnología de una forma más orientada al procomún y al desarrollo colaborativo, suena sin duda atractiva e interesante pero no libre de inconvenientes para algunos sectores de desarrolladores de Software libre.

2.6 Código Abierto y las Organizaciones Internacionales

Aunque la Organización de las Naciones Unidas (ONU) ha hablado previamente bien del desarrollo del código abierto, varios eventos recientes muestran que la ONU está tomando medidas definitivas para presentar al mundo entero el camino del código abierto. En julio del 2021, el Consejo Económico y Social de la ONU (ECOSOC) adoptó un proyecto de resolución presentado por el representante de Pakistán titulado: Tecnologías de fuente abierta para

el desarrollo sostenible.

2.6.1 Las Naciones Unidas y el Código Abierto

El ECOSOC destacó la disponibilidad de tecnologías de código abierto que pueden contribuir a los Objetivos de Desarrollo Sostenible (ODS). El consejo invitó al Secretario General a "desarrollar propuestas específicas sobre formas de aprovechar mejor las tecnologías de código abierto para el desarrollo sostenible basadas en las aportaciones de los Estados Miembros interesados y otras partes interesadas".

El desarrollo de tecnología de código abierto puede ser una herramienta rápida y eficaz para la innovación. Aplicarlo a tecnologías apropiadas para ayudar a alcanzar los ODS es extremadamente prometedor. Las "tecnologías apropiadas" abarcan opciones y aplicaciones tecnológicas que son a pequeña escala, económicamente asequibles, descentralizadas, energéticamente eficientes, ambientalmente racionales y fácilmente utilizadas por las comunidades locales para satisfacer sus necesidades.

Existe un caso particularmente fuerte para las tecnologías apropiadas de código abierto OSAT (Outsourced Semiconductor Assembly and Test). OSAT podría ayudar a todos a salir de la pobreza y alcanzar un estado sostenible aprovechando el mismo tipo de desarrollo que hace que el Software de código abierto sea un éxito rotundo.

La Declaración Ministerial del Foro político de alto nivel sobre desarrollo sostenible también destacó la importancia de "tecnologías no patentadas que pueden contribuir a los Objetivos de Desarrollo Sostenible, a través de diversas fuentes de acceso abierto". Pidió "el desarrollo y la puesta en funcionamiento de una plataforma en línea en el marco del Mecanismo de facilitación de la tecnología para establecer un mapeo integral y servir como puerta de entrada a la información sobre iniciativas, mecanismos y programas de ciencia, tecnología e innovación existentes, dentro y fuera de las Naciones Unidas".

Es un pequeño paso, pero muy emocionante, porque las Naciones Unidas no se demoran una vez que ven formas de ayudar a sus Estados Miembros y a las personas que los integran. Ahora, el Departamento de Asuntos Económicos y Sociales de las Naciones Unidas (DESA) está trabajando para que esto suceda. DESA está utilizando una Nota sobre una base de datos centralizada propuesta de las Naciones Unidas de tecnologías apropiadas de código abierto publicada por la Conferencia de las Naciones Unidas sobre Comercio

y Desarrollo (UNCTAD) para hacerlo.

La Nota de la UNCTAD aboga por una base de datos centralizada de OSAT para acelerar el descubrimiento y la innovación en todos los sectores asociados con los ODS al tiempo que se minimizan los obstáculos legales o financieros. Esto es importante para la difusión del acervo mundial de conocimientos, especialmente en los países en desarrollo.

Actualmente, no existe un repositorio completo o una base de datos central de OSAT y Appropedia.org, quizás sea el mejor ejemplo. Sin embargo, la Nota de la UNCTAD dice: "Muchas organizaciones, organizaciones sin fines de lucro y empresas con fines de lucro están desarrollando OSAT y manteniendo bases de datos existentes a pequeña escala. Si bien hay muchos OSAT disponibles, se encuentran dispersos en varias bases de datos para tecnologías particulares. Mientras tanto, sigue existiendo una clara necesidad de aumentar la tasa de uso de OSAT.

Por lo tanto, existe una necesidad urgente de una base de datos de código abierto centralizada global (COSD) confiable. Al tener un alcance global, un repositorio de COSD proporcionaría una ventanilla única a la que todos pueden acceder para resolver los desafíos locales".

Concluye: "La ONU está bien posicionada para liderar el establecimiento de un COSD dado su papel bien establecido en la promoción de la tecnología de código abierto a través de varios foros y publicaciones intergubernamentales. En particular, 2030 Connect es una plataforma tecnológica en línea de la ONU que se desarrolló como parte del trabajo del Equipo de Trabajo Interinstitucional de la ONU. El COSD podría mejorarlo".

Con el liderazgo de la ONU, quizás no estemos demasiado lejos de cuándo, sí tiene un problema local (sin importar en qué parte del mundo se encuentre), pueda descargar una solución de código abierto examinada y probada. Quizás, esta es la potencia de fuego que necesitamos para alcanzar los ambiciosos Objetivos de Desarrollo Sostenible.

2.6.2 La Comisión Europea se Compromete a Liberar Todo el Software que Pueda Beneficiar a la Sociedad

A finales del 2021, la Unión Europea (UE) y su órgano legislativo, la Comisión Europea siguen avanzando en su estrategia digital con el Software de código abierto como uno de los pilares fundamentales. En esta ocasión ha sido esta última la que anuncia novedades para con la distribución del Software desarrollado para cubrir necesidades internas de la organización.

De acuerdo a la información publicada, la Comisión Europea ha aprobado una nueva regulación que favorece el libre acceso al Software que producen siempre y cuando existan beneficios potenciales para «los ciudadanos, las empresas u otros servicios públicos», lo que de la teoría a la práctica bien puede abarcar todo lo que se desarrolle bajo su tejado.

Esta nueva disposición se apoya a su vez en un reciente estudio realizado también por la Comisión sobre el impacto del Software de código abierto en áreas como la independencia tecnológica, la competitividad y la innovación en la economía de la Unión Europea. El objetivo, hallar evidencias sólidas con las que conformar las políticas europeas de código abierto para los próximos años.

En términos económicos, de hecho, los cálculos son de lo más optimistas y apuntan un impacto económico contundente, de miles de millones de euros de ahorro al año -a modo de ejemplo, se estimó entre 65 y 95 mil millones de euros solo en 2018- y con un incremento mínimo en la apuesta, se podría dar un crecimiento del PIB de la UE de en torno a los 100,000 millones de euros.

Con semejante escenario, no es de extrañar que la misma Comisión Europea esté interesada en promover las soluciones de código abierto dentro y fuera de las instituciones y no solo se basan en el beneficio económico directo: son muchas otras las ventajas del modelo también recogidas en el informe, tal y como se ha mencionado: independencia, competitividad, innovación... y en el caso de las administraciones públicas, colaboración, reutilización y transparencia.

En palabras de Johannes Hahn, comisario de Presupuesto y Administración: «El código abierto ofrece grandes ventajas en un ámbito en el que la UE puede desempeñar un papel de liderazgo. Las nuevas normas aumentarán la transparencia y ayudarán a la Comisión, así como a los ciudadanos, las empresas y los servicios públicos de toda Europa, a beneficiarse del desarrollo de Software de código abierto. Poner en común los esfuerzos para mejorar el Software y la creación conjunta de nuevas funciones reduce los costes para la sociedad, ya que también nos beneficiamos de las mejoras realizadas por otros desarrolladores. Esto también puede mejorar la seguridad, ya que especialistas externos e independientes comprueban los fallos y las deficiencias de seguridad de los programas informáticos».

La comisaría de Innovación, Investigación, Cultura, Educación y Juventud, Mariya Gabriel, ha declarado: «La Comisión pretende, con su ejemplo, estar al frente de la transición digital en Europa. Con las nuevas normas, la

Comisión aportará un valor significativo a las empresas, también las emergentes, a los innovadores, a los ciudadanos y las administraciones públicas, poniendo a su disposición el código abierto de sus soluciones informáticas. Esta decisión también ayudará a estimular la innovación, gracias al código de la Comisión disponible públicamente».

Como muestra del Software desarrollado bajo el amparo de la Comisión Europea que va a ser liberado se incluyen proyectos como eSignature, «un conjunto de normas, herramientas y servicios gratuitos que ayudan a las administraciones públicas y a las empresas a acelerar la creación y verificación de firmas electrónicas jurídicamente válidas en todos los Estados miembros de la UE»; o LEOS (Legislation Editing Open Software), «el Software utilizado en toda la Comisión para elaborar textos jurídicos. LEOS, escrito originalmente para la Comisión, se está desarrollando en estrecha colaboración con Alemania, España y Grecia».

Esta nueva iniciativa de la Comisión Europa contempla asimismo la creación de un repositorio centralizado para facilitar el descubrimiento, el acceso y la reutilización del Software incluido, el cual se sumará a todos los proyectos realizados por las diferentes administraciones públicas comunitarias en base al mismo modelo de desarrollo. Y viene de lejos este impulso, aun cuando comienza a unificarse ahora.

Sin ir más lejos, hace años que la propia Comisión Europea puso en marcha el programa Interoperable Delivery of European eGovernment Services to Public Administrations, Businesses and Citizens que dio origen al observatorio JoinUp (https://joinup.ec.europa.eu/), en cuyas páginas se recogen casi 3,000 soluciones de Software abierto, 133 colecciones de recursos y cuantiosa información relacionada.

Más tarde, de 2014 a 2017 se inició la «primera fase» en la estrategia de código abierto de la Unión Europea, especialmente dentro de la propia Comisión, estableciendo determinados requisitos en materia de Software de código abierto; actualmente se está desarrollando la nueva «estrategia de código abierto 2020-2023», con la que la Comisión Europea pretende ampliar y afianzar los objetivos de la estrategia digital y la contribución al programa Europa Digital.

2.6.3 El Software Codigo Abierto, un sector de 7.700 millones de dólares anuales en 2024

El aumento en popularidad que está registrando el Software código abierto en todo el mundo lo ha convertido en un sector multimillonario que atrae inversores de empresas de todo el mundo. Lo deja claro la Encuesta de financiación de Software Open Source de 2024, realizada por Github, la Fundación Linux y la Universidad de Harvard. De sus resultados se desprende que en todo el mundo, la contribución de las empresas al sector suma nada menos que 7.700 millones de dólares cada año.

Eso sí, no toda la inversión que las compañías llevan a cabo en Software código abierto es económica. De hecho, solo un 14% se realiza en forma de contribuciones financieras a los distintos proyectos. La mayoría del aporte de las empresas al sector es en tiempo de trabajo de sus empleados. Un 86% de las inversiones en Software código abierto se realizan de esta manera.

Los investigadores que han realizado el informe utilizaron la información que les proporcionaron 501 fuentes, cuya aportación total a proyectos de Software código abierto cada año suma unos 1.700 millones de dólares. La mediana que las empresas invierten en Software código abierto cada año supera los 520,000 dólares. De ellos, 345,000 son en forma de trabajo y los 175,000 dólares restantes, en contribuciones económicas directas.

Un 57% de las contribuciones financieras se destinaron a financiar el trabajo de Freelances, y menos de un quinto de la cantidad se destinó tanto a proyectos concretos (un 17%) como a fundaciones (16%). Las comunidades (4%), mantenedores de Software código abierto (4%) y las plataformas (1%) también recibieron inyecciones económicas, aunque en mucha menor medida.

Además de ofrecer luz sobre las inversiones que se realizan en el sector, la encuesta también ha dejado al descubierto la existencia de zonas oscuras. Así, un 68% de las empresas donantes no saben cuánto apoyo económico prestaron a sus distintos objetivos de Software código abierto, y un 78% no quisieron, o no pudieron, decir qué parte del presupuesto de su empresa se destinaba a Software código abierto.

Un 44% de las empresas que contestaron a la encuesta en la que se ha basado el informe tienen su sede en Estados Unidos o Canadá, mientras que otro 39% están ubicadas en Europa. Otro 11% lo está en la región de Asia-Pacífico. Solo un 4% están en África, y un 2% en México, Centroamérica o Sudamérica.

En todas hay distintos niveles de experiencia en código abierto. A cerca

del 44% les habría gustado crear una Oficina de proyectos código abierto. Un 24% utiliza este tipo de programas, mientras que solo un 21% contribuyen a los proyectos. Otro 18% ha publicado proyectos código abierto, y un 16% de los encuestados tiene influencia en los proyectos por ocupar puestos de liderazgo o mantenimiento de los mismos.

En cuanto a las iniciativas relacionadas con la seguridad en el sector, se centran sobre todo en la corrección de errores y en el mantenimiento. Solo un 6% de las entidades aseguraron tener como prioridad la realización de auditorías de seguridad completas.

Entre las dificultades que está experimentando el sector, además de no saber en concreto qué tipo de contribuciones realizan las compañías a este tipo de proyectos, está la naturaleza descentralizada de las contribuciones de las empresas. En general, no hay grupos centralizados o políticas que animen a organizar este tipo de esfuerzos, o que se lancen a hacerlo. Por tanto, informar sobre el estado del sector es más complejo.

Recomendaciones para organizaciones que contribuyen a proyectos open source sin los datos ni la intención suficiente para recopilarlos, recoger el alcance de una inversión puede ser complicado, según la Fundación Linux. Por eso, sus miembros recomiendan que se pongan en marcha políticas y prácticas destinadas a animar a los empleados de las empresas que participan en el avance del Software código abierto a que informen ellos mismos de sus contribuciones. También a que utilicen sus direcciones de correo electrónico corporativo, de manera que queden huellas de las aportaciones de su empresa, además de la suya propia, a los proyectos a los que contribuyan.

Además, la fundación sugiere que el trabajo en código abierto quede consolidado bajo una única división en una empresa, como un grupo denominado por ejemplo Oficina de programa Open Source (OSPO). También aconsejan incorporar una monitorización de las contribuciones al flujo de trabajo de la organización. Un 64% de los encuestados no cuenta en su organización con una oficina o área de este tipo. Otro 7% no sabía si la tenían o no.

Mientras, menos de un tercio de las organizaciones que han participado en la encuesta obliga a los empleados a enviar commits a los proyectos utilizando sus direcciones de trabajo de empresas, y poco más de una tercera parte anima a hacerlo de esta manera de forma activa.

Las empresas son más propensas a contribuir a proyectos código abierto si lo hacen a través de repositorios que gestionan directamente (38%), y en

proyectos si tienen dependencias de cliente a servidor (34%). Un 39% de las empresas contribuye a estos proyectos a diario, mientras que un 60% lo hacen al menos una vez a la semana.

No parece que haya ninguna correlación sólida entre el tamaño de la empresa y la frecuencia de las contribuciones al código de los proyectos. Por otro lado, algunos encuestados señalaron que no contribuyen con frecuencia por falta de recursos o por compromisos con el cumplimiento de otros objetivos.

3 Seguridad y Privacidad en el Software

Tal como se observa en los principios que guían este trabajo, garantizar la seguridad, la confiabilidad, la resiliencia y la estabilidad de las aplicaciones y servicios de internet es fundamental para fomentar la confianza en su uso. Como usuarios de dispositivos interconectados en internet, debemos tener un alto grado de confianza en que internet, sus aplicaciones y los dispositivos conectados a la red son lo suficientemente seguros como para realizar en línea toda la gama de actividades que deseamos en relación con la tolerancia al riesgo asociado con tales actividades.

En este sentido, el uso de internet desde nuestros dispositivos que esta proliferando actualmente no es diferente y está fundamentalmente relacionada con la capacidad de los usuarios de confiar en su entorno. Si los usuarios no creen que los dispositivos que tienen conectados y su información están razonablemente seguros contra el mal uso o los daños, la erosión de la confianza resultante provoca una renuencia a usar internet.

Esto tiene consecuencias globales para el comercio electrónico, la innovación técnica, la libertad de expresión y prácticamente para todos los demás aspectos de las actividades en línea. En efecto, para garantizar la seguridad en los productos y servicios basados en internet, el sector desarrollador de productos digitales debe considerar la seguridad como una de sus máximas prioridades. A medida que conectamos cada vez más dispositivos a internet, surgen nuevas oportunidades para explotar vulnerabilidades potenciales de seguridad.

Los dispositivos mal asegurados pueden servir como puntos de entrada para ciberataques, permitiendo que personas malintencionadas puedan reprogramar un dispositivo o perjudicar su funcionamiento. Los dispositivos mal diseñados pueden exponer los datos de los usuarios a robos, dejando los flujos de usuarios sin una protección adecuada. Los dispositivos defectuosos o que no funcionan bien también pueden crear vulnerabilidades.

Estos problemas son tanto o más graves en el caso de los dispositivos inteligentes pequeños, baratos y ubicuos en internet. Los desafíos que imponen la competitividad de los costos y las limitaciones técnicas hacen que para los fabricantes de estos dispositivos no sea fácil diseñar funciones de seguridad adecuadas, potencialmente generando, a largo plazo, vulnerabi-lidades en la seguridad y dificultades en el mantenimiento superiores a las computadoras tradicionales.

Junto con posibles deficiencias en el diseño de la seguridad, el enorme au-

mento del número y la variedad de los dispositivos conectados a la red podría aumentar las oportunidades de ataque. Sumado a la naturaleza altamente interconectada de los dispositivos inteligentes, cada dispositivo mal asegurado conectado en línea potencialmente afecta la seguridad y la resistencia de internet a nivel global, no solo a nivel local. Por ejemplo, un refrigerador o un televisor sin protección infectado con Malware que se encuentra en Estados Unidos pueden enviar miles de correos electrónicos no deseados dañinos a destinatarios de todo el mundo usando la conexión Wi-Fi de la casa.

Para complicar todavía más las cosas, en un mundo hiperconectado, nuestra capacidad de funcionar diariamente sin dispositivos o sistemas conectados a internet probablemente disminuirá. De hecho, es cada vez más difícil comprar ciertos dispositivos sin conexión a internet, ya que algunos fabricantes solo ofrecen productos conectados. Cada vez estamos más conectados y dependemos más de los dispositivos para muchos servicios esenciales, por lo que necesitamos que los dispositivos sean seguros.

Pero también reconocemos que ningún dispositivo puede ser absolutamente seguro. Este creciente nivel de dependencia de los dispositivos y de los servicios de internet con los cuales interactúan también aumentan las formas que tienen los delincuentes para acceder a los dispositivos interconectados a la red. Si se ven comprometidas en un ataque cibernético, quizá podríamos desenchufar nuestros televisores conectados a internet, pero no es tan fácil apagar un medidor inteligente de energía eléctrica, un sistema de control de tráfico o un marcapasos si estos dispositivos son víctimas de un ataque malicioso. Esta es la razón por la cual la seguridad de los dispositivos y servicios debe ser un importante punto de discusión y un tema crítico por atender. Dependemos cada vez más de estos dispositivos para servicios esenciales, por lo que su comportamiento puede tener un alcance y un impacto globales.

3.1 Consideraciones de Seguridad

Al pensar en los dispositivos conectados a internet, es importante entender que la seguridad de estos dispositivos no es absoluta. La seguridad de los dispositivos no es una proposición binaria de tipo seguro/inseguro. Por el contrario, resulta útil conceptualizar la seguridad de los dispositivos como un espectro de vulnerabilidad. El espectro va desde dispositivos totalmente desprotegidos sin ninguna función de seguridad hasta sistemas muy seguros con múltiples capas de elementos de seguridad.

En un constante juego de gato y ratón, a medida que las nuevas amenazas de seguridad evolucionan, los fabricantes de dispositivos y los operadores de redes responden para hacer frente a las nuevas amenazas. La seguridad general y la resiliencia de los dispositivos interconectados al internet dependen de cómo se evalúen y gestionen los riesgos de seguridad.

La seguridad de un dispositivo está en función del riesgo en que un dispositivo se vea comprometido, del daño que tal compromiso provocaría, tiempo y los recursos necesarios para lograr cierto nivel de protección. Si un usuario no puede tolerar un alto grado de riesgo (por ejemplo, un operador de un sistema de control de tráfico o una persona a quien se le ha implantado un dispositivo médico que está conectado a internet), puede que para dicho usuario sienta que se justifica gastar una cantidad considerable de recursos para proteger el sistema o el dispositivo contra un ataque.

Del mismo modo, si una persona no le preocupa que su refrigerador pueda ser Hackeado y utilizado para enviar Spam, puede que no se sienta obligada a pagar por un modelo que tenga un diseño de seguridad más sofisticado si esto hace que el dispositivo sea más costoso o complicado. En esta evaluación y cálculo de la mitigación de los riesgos influyen diferentes factores. Estos factores incluyen una comprensión clara de los riesgos de seguridad actuales y posibles riesgos futuros, la estimación de los costos económicos y otros tipos de daños si los riesgos se hacen realidad y el costo estimado de la mitigación de estos.

Si bien este tipo de concesiones de seguridad muchas veces se realizan desde la perspectiva de los usuarios individuales y las organizaciones, también es importante tener en cuenta la interrelación de los dispositivos interconectados como parte de un ecosistema mayor. La conectividad en red de los dispositivos significa que las decisiones de seguridad que se toman a nivel local con respecto a un dispositivo pueden tener impactos globales sobre otros dispositivos.

Como cuestión de principio, quienes desarrollan objetos inteligentes para internet tienen la obligación de garantizar que estos dispositivos no expongan los bienes de sus propios usuarios ni de otras personas a potenciales daños. Como cuestión de negocios y de economía, los fabricantes desean reducir sus costos, su complejidad y su tiempo de comercialización. Por ejemplo, son cada vez más comunes los dispositivos de alto volumen y bajo margen de ganancia y que ya representan un costo adicional para los productos en los que están embebidos; añadir más memoria y un procesador más rápido para implementar medidas de seguridad podría hacer que el producto ya no fuera

competitivo.

En términos económicos, el resultado de la falta de seguridad en los dispositivos digitales es una externalidad negativa, donde una o más partes imponen un costo sobre otras. Un ejemplo clásico es la contaminación del medio ambiente, donde los costos de los daños y la limpieza (externalidades negativas) resultantes de las acciones de quien contamina son asumidos por otras partes. El hecho es que el costo de la externalidad impuesto a los demás normalmente no se considera en el proceso de toma de decisiones, a menos que, como es el caso de la contaminación, se aplique un impuesto que sirva de aliciente para reducir la contaminación.

De acuerdo con Bruce Schneier, en el caso de la seguridad de la información surge una externalidad cuando el proveedor que crea el producto no corre con los costos que ocasionan las potenciales inseguridades; en este caso, una ley de responsabilidad puede convencer a los vendedores para que tomen en cuenta la externalidad y desarrollen productos más seguros. Estas consideraciones de seguridad no son nuevas en el contexto de la tecnología, pero la magnitud de los desafíos que pueden surgir en las implementaciones de dispositivos digitales de gran volumen las vuelve extremadamente significativas. Estos desafíos se describen a continuación.

Desafíos de Seguridad que son Exclusivos de los Dispositivos Interconectados. Las diferencias entre los dispositivos digitales, las computadoras y los dispositivos informáticos tradicionales suelen desafiar la seguridad:

- Muchos dispositivos digitales interconectados (por ejemplo, los sensores y los artículos de consumo) están diseñados para ser desplegados a una escala masiva que es varios órdenes de magnitud superior a la de los dispositivos tradicionalmente conectados a internet. Por consiguiente, la cantidad potencial de enlaces interconectados entre estos dispositivos no tiene precedentes. Además, muchos de estos dispositivos podrán establecer enlaces y comunicarse con otros dispositivos por sí mismos, de manera impredecible y dinámica. Por lo tanto, puede ser necesario considerar nuevamente las herramientas, métodos y estrategias existentes asociadas con la seguridad de los dispositivos.

- Muchos despliegues de dispositivos consistirán en colecciones de dispositivos idénticos o prácticamente idénticos. Esta homogeneidad amplifica el potencial impacto de cualquier vulnerabilidad de seguridad

simplemente por la gran cantidad de dispositivos que tienen las mismas características. Por ejemplo, una vulnerabilidad en el protocolo de comunicación de una marca de bombillas de luz conectadas a internet se podría extender a todas las marcas y modelos de dispositivos que utilizan el mismo protocolo o que comparten características clave de diseño o fabricación.

- Muchos de los dispositivos digitales que se van a desplegar tendrán una vida útil anticipada superior a la que típicamente se espera para los equipos de alta tecnología. Además, estos dispositivos se podrían desplegar en circunstancias que los harían difíciles o imposibles de reconfigurar o actualizar; o bien estos dispositivos podrían sobrevivir a la empresa que los creó, lo que los dejaría huérfanos y sin apoyo a largo plazo. Estos escenarios ilustran que los mecanismos de seguridad que son adecuados en el momento del despliegue podrían no ser adecuados durante toda la vida útil del dispositivo y a medida que las amenazas a la seguridad evolucionen, esta situación podría crear vulnerabilidades que persistirían por mucho tiempo. Esto contrasta con el paradigma de los sistemas de computadoras tradicionales en los cuales normalmente se aplican actualizaciones al sistema operativo durante toda la vida de servicio de los equipos para hacer frente a las amenazas de seguridad. El apoyo y la gestión a largo plazo de los dispositivos digitales interconectados representa un importante reto de seguridad.

- Muchos dispositivos digitales están diseñados intencionadamente sin ninguna posibilidad de actualización; en otros, el proceso de actualización es engorroso o poco práctico. Por ejemplo, consideremos el retiro de 1.4 millones de automóviles Fiat Chrysler 2015 para arreglar una vulnerabilidad que potencialmente permitiría Hackear el vehículo en forma inalámbrica. Estos vehículos se deben llevar a un concesionario Fiat Chrysler para que les realicen una actualización manual del Software, o bien los propietarios deben actualizar el Software por sí mismos usando una memoria USB. La realidad es que un alto porcentaje de estos automóviles probablemente no se actualizarán porque el proceso de actualización representa un inconveniente para los propietarios, esto los deja permanentemente vulnerables a las amenazas de seguridad cibernética, sobre todo porque el automóvil parece estar funcionando muy bien.

- Muchos dispositivos digitales funcionan de modo que es escasa o nula la visibilidad que tiene el usuario de su funcionamiento interno o de los flujos de datos que producen. Si un usuario cree que un dispositivo está ejecutando ciertas funciones pero en realidad está ejecutando funciones no deseadas o recogiendo más información que lo que el usuario desea, se crea una vulnerabilidad. Las funciones del dispositivo también podrían cambiar sin previo aviso cuando el fabricante ofrece una actualización, lo que deja al usuario vulnerable a cualquier cambio que este realice.

- Algunos dispositivos digitales probablemente serán desplegados en lugares donde sea difícil o imposible lograr la seguridad física. Los atacantes pueden tener acceso físico directo a los dispositivos. Para garantizar la seguridad será necesario considerar el uso de protección contra manipulaciones y otras innovaciones de diseño.

- Al igual que muchos sensores ambientales, algunos dispositivos digitales han sido diseñados para ser integrados discretamente en su entorno, donde los usuarios apenas se den cuenta de su presencia o monitoreen su funcionamiento. Además, los dispositivos pueden no tener una forma clara de alertar al usuario cuando surge un problema de seguridad, por lo que es difícil para un usuario saber que la seguridad de un dispositivo digital ha sido vulnerada. Esta situación podría persistir por mucho tiempo antes de ser detectada y corregida; incluso podría darse el caso de que no fuera posible o práctico implementar una corrección o mitigación. Del mismo modo, el usuario podría no ser consciente de que existe un sensor en su entorno, por lo que potencialmente un fallo de seguridad podría persistir por mucho tiempo sin ser detectado.

- Los primeros modelos digitales serán producto de grandes empresas de tecnología privadas y/o públicas. Sin embargo, en el futuro "construir su propia internet de las cosas" (Build Your Own Internet Of Things) podría convertirse en algo habitual, como lo demuestra el crecimiento de las comunidades de desarrolladores de Arduino y Raspberry Pi. Estos despliegues podrán o no aplicar los estándares de mejores prácticas de seguridad de la industria.

Preguntas Relacionadas con la Seguridad de los Dispositivos Interconectados Se han planteado una serie de preguntas con respecto a

los problemas de seguridad que plantea el uso de dispositivos digitales interconectados. Muchas de estas preguntas ya existían antes del crecimiento explosivo de los dispositivos interconectados, pero su importancia ha aumentado debido a la magnitud del despliegue de los dispositivos utilizados. A continuación veremos las preguntas más importantes:

Buenas Prácticas de Diseño ¿Cuáles son las mejores prácticas que los ingenieros y desarrolladores deben utilizar al diseñar dispositivos digitales para que sean más seguros?, ¿cómo se recogen y transmiten las lecciones aprendidas a partir de los problemas de seguridad de los dispositivos a las comunidades de desarrolladores para mejorar las futuras generaciones de dispositivos?, ¿qué formación y recursos educativos se pueden utilizar para enseñar a los ingenieros y desarrolladores para diseñar una gama de dispositivos más segura?.

Equilibrio Entre Costo y Seguridad ¿De qué manera las partes interesadas toman decisiones informadas con respecto a los dispositivos digitales interconectados considerando la relación costo-beneficio?, ¿cómo se pueden cuantificar y evaluar con precisión los riesgos de seguridad?, ¿qué motivará a los diseñadores y fabricantes de dispositivos para que acepten el costo adicional que implica el diseño de dispositivos más seguros, en particular, para que asuman la responsabilidad por el impacto de cualquier externalidad negativa derivada de sus decisiones de seguridad?, ¿cómo se van a conciliar las incompatibilidades entre la funcionalidad, la facilidad de uso y la seguridad?, ¿cómo nos aseguramos de que las soluciones de seguridad para los dispositivos digitales interconectados soporten oportunidades para la innovación y de crecimiento económico?.

Estándares e Indicadores ¿Qué papel desempeñan los estándares técnicos y operativos en el desarrollo y despliegue de dispositivos digitales interconectados seguros y de buen funcionamiento?, ¿cómo se pueden identificar y medir las características de seguridad de los dispositivos digitales interconectados?, ¿cómo se puede medir la efectividad de las iniciativas y medidas de seguridad implementadas?, ¿cómo se puede asegurar la implementación de mejores prácticas de seguridad?.

Confidencialidad de los Datos, Autentificación y Control de Acceso ¿Cuál es el papel óptimo del cifrado de los datos con respecto a los dispositivos digitales?, ¿utilizar tecnologías de cifrado, autenticación y control de acceso en los dispositivos digitales es una solución adecuada para evitar intentos de espionaje y secuestro de los flujos de datos que producen estos dispositivos?, ¿qué tecnologías de cifrado y autenticación se podrían adaptar para los dispositivos digitales y cómo se podrían aplicar considerando las limitaciones de costo, tamaño y velocidad de procesamiento de los dispositivos?, ¿cuáles son los problemas de gestión que se espera deberán ser abordados como resultado del cifrado a una escala de la magnitud de los dispositivos digitales interconectados?, ¿se están abordando las preocupaciones con respecto a cómo gestionar el ciclo de vida de las claves criptográficas y el período durante el cual se espera que un algoritmo dado permanezca seguro?, ¿los procesos de extremo a extremo son lo suficientemente seguros y simples como para que los utilicen los usuarios típicos?.

Capacidad de Actualización en Campo Dado que se espera que muchos de los dispositivos digitales tendrán una vida útil prolongada, ¿estos dispositivos deben diseñarse considerando su mantenimiento y su capacidad de actualización in situ de modo que puedan adaptarse a las nuevas amenazas de seguridad?. Si cada dispositivo tiene integrado un agente de gestión de dispositivos, en los dispositivos digitales interconectados se podría instalar y configurar nuevo Software. Pero los sistemas de gestión aumentan los costos y la complejidad, ¿habrá otros enfoques para actualizar el Software de los dispositivos que sean más compatibles con el uso masivo de los dispositivos digitales?, ¿existe alguna clase de dispositivos de bajo riesgo y que por lo tanto no justifique este tipo de características?. En general, ¿las interfaces de usuario de los dispositivos interconectados (por lo general mínimas) se están analizando adecuadamente, tomando en cuenta la gestión de los dispositivos (por parte de cualquier persona, incluso por el usuario)?.

Responsabilidad Compartida ¿Cómo se puede fomentar la responsabilidad compartida y la colaboración entre todas las partes interesadas en pos de la seguridad de los dispositivos digitales interconectados.

Regulación ¿Se debe sancionar a los fabricantes de dispositivos por la venta de Software o Hardware con fallos de seguridad conocidas o descono-

cidas?, ¿cómo se podrían adaptar o ampliar las leyes de responsabilidad de producto y protección del consumidor para que abarquen las externalidades negativas relacionadas con los dispositivos digitales interconectados?, ¿sería posible hacerlo en un entorno transfronterizo?, ¿la regulación podrá seguir el ritmo y mantener su eficacia en vista de la evolución de la tecnología de los dispositivos y la evolución de las amenazas a la seguridad?, ¿cómo se debe equilibrar la regulación con las necesidades de la innovación sin pedir permiso, la libertad en internet y la libertad de expresión?.

Obsolescencia de los Dispositivos ¿Qué enfoque se debe adoptar con respecto a los dispositivos digitales obsoletos a medida que internet evoluciona y cambian las amenazas a la seguridad?, ¿se debe exigir que los dispositivos tengan una funcionalidad de "final de vida" integrada que los inactive?. En el futuro, este tipo de requisito podría obligar a sacar de servicio a los dispositivos más antiguos que no son interoperables y a reemplazarlos por dispositivos más seguros e interoperables. Esto ciertamente sería muy difícil en un mercado abierto. ¿Qué implicaciones tiene la inactivación automática de los dispositivos digitales interconectados?.

La amplitud de estas preguntas es representativo de la variedad de las consideraciones de seguridad asociadas con los dispositivos digitales interconectados. Sin embargo, es importante recordar que, cuando un dispositivo está en internet también es parte de internet, lo que significa que solo se pueden lograr soluciones de seguridad eficaces y apropiadas si todas las partes involucradas con estos dispositivos aplican un enfoque de seguridad colaborativo.

Tanto entre la industria como entre los gobiernos y las autoridades públicas, el modelo colaborativo aparece como un enfoque eficaz para ayudar a asegurar a internet y al ciberespacio. Este modelo incluye una serie de prácticas y herramientas que incluyen el intercambio de información bidireccional y voluntario, herramientas de aplicación eficaces, preparación para incidentes y ejercicios cibernéticos, creación de conciencia y capacitación, acuerdo sobre las normas de comportamiento internacionales, desarrollo y reconocimiento de prácticas y estándares internacionales.

Es necesario continuar trabajando para que sigan evolucionando los enfoques colaborativos y basados en la gestión de riesgos, a manera de lograr que se adapten bien a la escala y la complejidad de los desafíos de seguridad de los dispositivos digitales interconectados.

3.2 Consideraciones Sobre la Privacidad

El respeto por las expectativas y los derechos de privacidad es fundamental para asegurar la confianza en internet; además, también afecta la capacidad de las personas de hablar, conectarse y escoger de formas significativas. Estos derechos y expectativas se suelen enmarcar en términos del manejo ético de los datos, que hacen hincapié en la importancia de respetar las expectativas de privacidad del individuo y el uso legítimo de sus datos. El uso de dispositivos digitales masivo puede desafiar estas expectativas tradicionales de privacidad. El uso masivo de dispositivos digitales suele referirse a una amplia red de dispositivos con sensores diseñados para recopilar datos acerca de su entorno, que muchas veces incluyen datos relacionados con las personas.

Estos datos presumiblemente proporcionan un beneficio al propietario del dispositivo, pero muchas veces también benefician al fabricante o proveedor. La recopilación y el uso de los datos se convierte en una consideración de privacidad cuando las expectativas de privacidad de quienes son observados por los dispositivos digitales difieren de las de quienes recogerán y usarán estos datos. También hay combinaciones de flujos de datos aparentemente inocentes que también pueden poner en riesgo la privacidad.

Cuando se combinan o correlacionan flujos de datos individuales, el retrato digital que se obtiene de las personas suele ser más invasivo que el que se puede obtener a partir de un flujo de datos individual. Por ejemplo, un cepillo de dientes con conexión a internet puede recoger y transmitir información sobre los hábitos de cepillado de una persona, algo bastante inocuo. En cambio, si el refrigerador de este mismo usuario informa el listado de los alimentos que consume y si además el dispositivo que el usuario utiliza para llevar cuenta de su actividad física también informa los datos correspondientes, la combinación de estos flujos de datos pinta una descripción mucho más detallada y privada de la salud general de la persona.

Este efecto de agregación de los datos puede ser particularmente potente en el caso de los dispositivos digitales interconectados, dado que muchos producen otros metadatos como por ejemplo marcas de tiempo e información de geolocalización, lo que aumenta aún más la especificidad del usuario. En otras situaciones, el usuario puede no ser consciente de que un dispositivo está recogiendo datos sobre su persona y potencialmente compartiéndolos con terceros. Este tipo de recolección de datos es cada vez más frecuente en los dispositivos de consumo, como por ejemplo en los televisores inteligentes y las consolas de videojuegos. Este tipo de productos tienen características

de reconocimiento de voz o de visualización que permanentemente escuchan las conversaciones u observan la actividad en una habitación y selectivamente transmiten los datos recogidos a un servicio en la nube para su procesamiento, donde a veces participa un tercero.

Una persona podría estar en presencia de este tipo de dispositivos sin saber que sus conversaciones o actividades están siendo monitoreadas o que sus datos están siendo registrados. Estos tipos de características pueden ser de beneficio para un usuario informado, pero pueden plantear un problema de privacidad para quienes no son conscientes de la presencia de estos dispositivos y no pueden influir significativamente sobre la forma en que se utiliza la información recogida.

Sin importar si el usuario está al tanto de que los dispositivos digitales recogen y analizan sus datos, estas situaciones ponen de relieve el valor que tienen estos flujos de datos personalizados para empresas y organizaciones que buscan recoger y sacar provecho de la información obtenida a través de los dispositivos digitales interconectados a internet. La demanda de esta información deja al descubierto los desafíos legales y regulatorios que enfrentan las leyes de protección de datos y privacidad.

Es fundamental abordar estos tipos de problemas de privacidad, dado que tienen implicaciones sobre nuestros derechos básicos y nuestra capacidad colectiva de confiar en internet. Desde una perspectiva más amplia, las personas reconocen que su privacidad es un valor intrínseco y tienen expectativas con respecto a los datos personales que se pueden recoger y cómo estos datos pueden ser utilizados por terceros. Esta noción general acerca de la privacidad también vale para los datos recogidos por los dispositivos digitales interconectados a internet, pero estos dispositivos pueden socavar la capacidad del usuario de expresar y hacer cumplir sus preferencias de privacidad. Si el hecho de que sus preferencias de privacidad no sean respetadas por los dispositivos digitales, hace que los usuarios pierdan la confianza en internet, entonces podría disminuir el mayor valor que este tiene.

En general, la forma en que los dispositivos digitales interconectados aumentan la viabilidad y el alcance de la vigilancia y el seguimiento amplifica las preocupaciones relativas a la privacidad. Las características de los dispositivos digitales interconectados y las formas en que se utilizan redefinen el debate sobre los temas de privacidad, ya que modifican drásticamente cómo se recogen, analizan, utilizan y protegen los datos personales. Por ejemplo:

- El modelo tradicional de privacidad de "notificación y consentimiento"

en que los usuarios hacen valer sus preferencias de privacidad interactuando directamente con información que aparece en la pantalla de una computadora o dispositivo móvil (por ejemplo, haciendo clic en "Acepto") deja de funcionar cuando los sistemas no le ofrecen al usuario ningún mecanismo de interacción. Muchas veces los dispositivos digitales no tienen una interfaz de usuario para configurar las preferencias de privacidad y en muchas configuraciones los usuarios no tienen conocimiento ni controlan la forma en que se recogen y utilizan sus datos personales. Esto provoca una brecha entre las preferencias de privacidad del usuario y el comportamiento de recolección de datos del dispositivo. Si consideran que los datos recopilados no son datos personales, es posible que los proveedores de dispositivos se sientan menos incentivados a ofrecer a los usuarios un mecanismo para que expresen sus preferencias de privacidad. Sin embargo, la experiencia demuestra que, en realidad, los datos que tradicionalmente no se consideran personales podrían ser o convertirse en datos personales si se combinan con otros.

- Suponiendo que se pudiera desarrollar un mecanismo eficaz que permitiera que un usuario expresara de manera informada sus preferencias de privacidad, este mecanismo debería poder manejar la gran cantidad de dispositivos digitales interconectados que debe controlar cada usuario. No es realista pensar que un usuario interactuará directamente con cada uno de los dispositivos con que se encuentre a lo largo del día para expresar sus preferencias de privacidad. Por el contrario, las interfaces de privacidad se deben poder escalar de acuerdo con el tamaño del problema, sin dejar de ser completas y prácticas desde la perspectiva del usuario.

- Los dispositivos digitales interconectados pueden poner en peligro las expectativas de los usuarios con respecto a la privacidad en situaciones comunes. Las normas sociales y expectativas de privacidad difieren en los espacios públicos frente a los espacios privados; los dispositivos digitales interconectados desafían estas normas. Por ejemplo, las tecnologías de vigilancia que utiliza cámaras de vigilancia o los sistemas de trazabilidad de ubicación que normalmente funcionan en espacios públicos están migrando hacia espacios tradicionalmente privados como el hogar o los vehículos particulares, donde nuestras expectativas de

privacidad son muy diferentes. Al hacerlo, desafían lo que muchas sociedades reconocen como el derecho a la privacidad en el hogar o los espacios privados. Además, las expectativas de las personas con respecto a su privacidad en los espacios que consideran públicos (parques, centros comerciales, estaciones de tren, etc.) también están siendo desafiadas por el aumento de la naturaleza y el alcance de la vigilancia en tales espacios.

- Muchas veces los dispositivos digitales interconectados funcionan en contextos donde la proximidad expone a múltiples personas a una misma actividad de recolección de datos. Por ejemplo, el sensor de seguimiento por geolocalización de un automóvil podría registrar los datos de localización de todos los ocupantes del vehículo, sin importar si estas personas desean que lo haga o no. Incluso podría realizar un seguimiento de las personas que viajan en otros vehículos cercanos. En este tipo de situaciones podría ser difícil o imposible distinguir -mucho menos respetar- las preferencias de privacidad individuales.

- El análisis de datos personales consolidados a gran escala de por sí representa un riesgo sustancial de invasión a la privacidad y potencial discriminación. Este riesgo se amplifica en los dispositivos digitales interconectados debido a la escala y a la mayor intimidad de la recolección de datos personales. Los dispositivos digitales interconectados pueden recoger información personal con un grado de especificidad y penetración sin precedente; agregar y correlacionar estos datos permite crear perfiles personales detallados que pueden generar un riesgo potencial para la discriminación y otros daños. La sofisticación de esta tecnología puede crear situaciones que expongan al individuo a daños físicos, penales, financieros o de reputación.

- La ubicuidad, familiaridad y aceptación social de muchos dispositivos digitales interconectados pueden crear una falsa sensación de seguridad y alentar a las personas a divulgar información confidencial o privada sin pleno conocimiento o apreciación de las posibles consecuencias.

Preguntas Relacionadas con la Privacidad de los Dispositivos interconectados Estas preguntas referidas a la privacidad serían un desafío incluso si estuvieran bien alineados los intereses y motivaciones de todas las

partes involucradas en el ecosistema de los dispositivos digitales interconectados. Sin embargo, sabemos que las relaciones y los intereses de quienes están expuestos a la recolección de sus datos personales y quienes agregan, analizan y utilizan los datos pueden ser desequilibrados o injustos.

La fuente de datos puede ver una intrusión no deseada a su espacio privado, muchas veces sin consentimiento, control, elección o incluso conciencia. No obstante, quien recoge los datos podría considerarlos un recurso beneficioso que puede añadir valor a sus productos y servicios y proporcionar nuevas fuentes de ingresos. Dado que los dispositivos digitales interconectados desafían nuestras nociones de privacidad de formas nunca antes vistas, al reevaluar los modelos de privacidad en línea en el contexto de los dispositivos digitales interconectados, es necesario responder ciertas preguntas clave. Algunas de las preguntas que se han planteado incluyen las siguientes:

Legitimidad en la Recopilación y el Uso de los Datos en el contexto de los dispositivos digitales interconectados, ¿cómo se resuelve la relación de mercado entre las fuentes de los datos y quiénes los recogen?. Los datos personales tienen un valor personal y comercial diferente según se consideren desde el punto de vista de las fuentes o de los recolectores, tanto indivi-dualmente como en su conjunto; por lo tanto, ambas partes tienen intereses legítimos que podrían estar en conflicto. ¿Cómo se pueden expresar estos diferentes intereses de una manera que conduzca a reglas en materia de acceso, control, transparencia y protección que sean justas y consistentes, tanto para las fuentes como para los recolectores?.

Transparencia, Expresión y Cumplimiento de las Preferencias de Privacidad ¿cómo se puede hacer que las políticas y prácticas de privacidad sean de fácil acceso y comprensibles en el contexto de los dispositivos digitales interconectados?, ¿cuáles son las alternativas al modelo tradicional de privacidad de "notificación y consentimiento" que podrían abordar los aspectos únicos de los dispositivos digitales interconectados?, ¿cuál sería un modelo eficaz para expresar, aplicar y hacer cumplir las preferencias de privacidad individuales y las preferencias multipartitas?, ¿se podría construir un modelo multipartito de este tipo?. De ser así, ¿qué aspecto tendría?, ¿cómo se podría aplicar a circunstancias concretas que impliquen las preferencias de privacidad individuales?, ¿existe un mercado para tercerizar la gestión de la configuración de la privacidad a servicios comerciales diseñados para

implementar las preferencias de los usuarios?, ¿es necesario que exista un Proxy de privacidad que exprese y haga cumplir las preferencias del usuario a través de una serie de dispositivos, al tiempo que elimine la necesidad de interacción directa con cada uno de ellos?.

Gran variedad de Expectativas de Privacidad las normas y expectativas de privacidad están estrechamente relacionadas con el contexto social y cultural del usuario, que puede variar de una nación o de un grupo a otro. Muchos escenarios de los dispositivos digitales interconectados implican el despliegue de dispositivos y actividades de recopilación de datos de alcance multinacional o global que atraviesan fronteras sociales y culturales. ¿Qué implicará esto para el desarrollo de un modelo de protección de la privacidad que se pueda aplicar ampliamente a los dispositivos digitales interconectados?, ¿cómo se pueden adaptar los dispositivos y sistemas de dispositivos digitales interconectados para que reconozcan y respeten la variedad de expectativas de privacidad de los usuarios y las diferentes legislaciones?.

Privacidad por Diseño ¿cómo se pueden adaptar los dispositivos y sistemas digitales para que reconozcan y respeten la variedad de expectativas de privacidad de los usuarios y las diferentes legislaciones?, ¿cómo se puede animar a los fabricantes de dispositivos digitales interconectables para que incorporen los principios de la privacidad por diseño a sus valores fundamentales?, ¿cómo se puede fomentar la inclusión de consideraciones sobre la privacidad de los consumidores en todas las fases de desarrollo y operación de los productos?, ¿cómo se pueden conciliar los requisitos de funcionalidad y privacidad?. En principio, los fabricantes deberían anticipar que, a largo plazo, los productos y las prácticas que respeten la privacidad se ganarán la confianza y la satisfacción de los clientes y generarán lealtad hacia la marca. ¿Es esta motivación suficiente para competir con los deseos de simplicidad en el diseño y velocidad en el mercado?, ¿los dispositivos se deberían diseñar con una configuración predeterminada para el modo de recopilación de datos más conservador (es decir, no recopilación de datos por defecto)?.

Identificación ¿cómo debemos proteger los datos recogidos por los dispositivos digitales interconectados que parecieran no ser personales donde se recogen o que han sido "desidentificados", pero que en algún momento futuro podrían llegar a ser datos personales (por ejemplo, porque podrían ser

re-identificados o combinados con otros datos).

El uso de dispositivos digitales interconectados genera desafíos únicos para la privacidad que van más allá de los problemas que existen en la actualidad. Es necesario desarrollar estrategias para respetar las opciones de privacidad individuales considerando un amplio espectro de expectativas, sin dejar de fomentar la innovación en nuevas tecnologías para los dispositivos digitales.

3.3 Software Libre e Infraestructura Crítica

Si bien, el ecosistema de Software libre es una de las empresas más grandiosa en la historia de la humanidad, en los últimos tiempos los gobiernos, industrias de todos los sectores económicos y personal académico se ha visto en la apremiante necesidad de comprender el Software más importante que corre en las infraestructuras críticas de todos los ámbitos de nuestra vida, si bien existe una gran cantidad de Software libre, algunos han llegado a ser el pilar de nuestra vida tecnológica.

El Software libre se ejecuta en una gran cantidad de dispositivos (desde teléfonos inteligentes, tabletas, computadoras, dispositivos de interconexión de red, etc.) del planeta y mantiene en funcionamiento la infraestructura crítica de gran parte del mundo. Es por ello que agencias de todo el mundo (entre ellas DARPA), están preocupadas por cuanto se puede confiar en dicho Software.

No es una gran exageración decir que gran parte de la infraestructura del mundo está construida sobre Software libre y en particular sobre el Kernel de Linux, aunque la mayoría de la gente nunca ha oído hablar de él. Es uno de los primeros programas que se cargan cuando la mayoría de los dispositivos de cómputo en cuanto se encienden. Permite que el Hardware que ejecuta la máquina interactúe con el Software, gobierna el uso de recursos y actúa como la base del sistema operativo.

Es el bloque de construcción central de casi toda la computación en la nube, prácticamente todas las supercomputadoras, todo el Internet de las cosas, miles de millones de teléfonos inteligentes y más.

Pero el Kernel también es de código abierto, lo que significa que cualquiera puede escribir, leer y usar su código. Y eso tiene a los expertos en seguridad cibernética del mundo seriamente preocupados. Su naturaleza de código abierto significa que el Kernel de Linux, junto con una gran cantidad de otras

piezas de Software crítico de código abierto, está expuesto a una manipulación hostil en formas que apenas entendemos.

La gente apenas se está dando cuenta ahora que literalmente gran parte de lo que hacemos está respaldado por Linux. Esta es una tecnología fundamental para nuestra sociedad, no comprender la seguridad del Kernel significa que no se podrá asegurar la infraestructura crítica.

De los proyectos que han salido a la luz pública, la agencia gubernamental DARPA (el brazo de investigación del ejército de EE. UU.) quiere comprender la colisión de código y comunidad que hace que estos proyectos de código abierto funcionen, para comprender mejor los riesgos que enfrentan. El objetivo es poder reconocer de manera efectiva a los actores maliciosos y evitar que interrumpan o corrompan el código de código abierto de importancia crucial antes de que sea demasiado tarde.

Por ejemplo, el programa "Social Cyber" de DARPA es un proyecto multimillonario de 18 meses de duración que combinará la sociología con los avances tecnológicos recientes en inteligencia artificial para mapear, comprender y proteger estas comunidades masivas de código abierto y el código que crean. Es diferente de la mayoría de las investigaciones anteriores porque combina el análisis automatizado tanto del código como de las dimensiones sociales del Software de código abierto.

Amenazas al Software Libre Gran parte de la civilización moderna ahora depende de un corpus de código abierto en constante expansión porque ahorra dinero, atrae talento y facilita mucho el trabajo. Pero si bien el movimiento de código abierto ha generado un ecosistema colosal del que todos dependemos, no lo entendemos completamente, argumentan expertos. Hay innumerables proyectos de Software, millones de líneas de código, numerosas listas de correo, foros y un océano de colaboradores cuyas identidades y motivaciones a menudo son oscuras, lo que dificulta responsabilizarlos.

Eso puede ser peligroso. Por ejemplo, los piratas informáticos han insertado discretamente códigos maliciosos en proyectos de código abierto en numerosas ocasiones en los últimos años. Las puertas traseras pueden escapar durante mucho tiempo a la detección y en el peor de los casos, se han entregado proyectos completos a malos actores que se aprovechan de la confianza que las personas depositan en las comunidades y el código de código abierto.

A veces hay interrupciones o incluso tomas de control de las mismas redes

sociales de las que dependen estos proyectos. El seguimiento de todo ha sido principalmente, aunque no del todo, un esfuerzo manual, lo que significa que no coincide con el tamaño astronómico del problema.

Varios autores argumentan que necesitamos el aprendizaje automático para digerir y comprender el universo en expansión del código, lo que significa trucos útiles como el descubrimiento automatizado de vulnerabilidades, así como herramientas para comprender la comunidad de personas que escriben, corrigen, implementan e influyen en ese código.

El objetivo final es detectar y contrarrestar cualquier campaña maliciosa para enviar código defectuoso, lanzar operaciones de influencia, sabotear el desarrollo o incluso tomar el control de proyectos de código abierto.

Para hacer esto, los investigadores utilizarán herramientas como el análisis de sentimientos para analizar las interacciones sociales dentro de las comunidades de código abierto, como la lista de correo del Kernel de Linux, que debería ayudar a identificar quién es positivo o constructivo y quién es negativo y destructivo.

Los investigadores quieren conocer qué tipos de eventos y comportamientos pueden perturbar o dañar las comunidades de código abierto, qué miembros son confiables y si hay grupos particulares que justifiquen una vigilancia adicional. Estas respuestas son necesariamente subjetivas. Pero en este momento hay pocas formas de encontrarlos.

A los expertos les preocupa que los puntos ciegos de las personas que ejecutan el Software de código abierto hagan que todo el edificio esté listo para posibles manipulaciones y ataques. Para algunos investigadores, la principal amenaza es la perspectiva de un "código no confiable" que ejecute la infraestructura crítica del mundo, una situación que podría generar sorpresas desagradables.

Preguntas Sin Respuesta Así es como funciona el programa "Social Cyber": DARPA ha contratado a varios equipos de lo que llama "intérpretes", incluidos pequeños talleres de investigación de ciberseguridad boutique con habilidades técnicas profundas.

Uno de estos actores es Margin Research, con sede en Nueva York, que ha reunido a un equipo de investigadores muy respetados para la tarea. Él ha dicho que existe una necesidad desesperada de tratar a las comunidades y proyectos de código abierto con un mayor nivel de cuidado y respeto, ya que mucha de la infraestructura existente es muy frágil porque depende del código

abierto, que asumimos que siempre estará ahí porque siempre ha estado ahí. Esto es alejarse de la confianza implícita que tenemos en las bases de código y Software de fuente abierta.

Muchos investigadores se han enfocado en el Kernel de Linux en parte porque es tan grande y crítico que tener éxito aquí, a esta escala, significa que puede hacerlo en cualquier otro lugar. El plan es analizar tanto el código como la comunidad para visualizar y finalmente comprender todo el ecosistema.

El trabajo de los investigadores mapea quién está trabajando en qué partes específicas de los proyectos de código abierto. Por ejemplo, Huawei es actualmente el mayor contribuyente al Kernel de Linux. Otro colaborador trabaja para Positive Technologies, una empresa rusa de ciberseguridad que, al igual que Huawei, ha sido sancionada por el gobierno de EE. UU. También se ha mapeado código escrito por empleados de la NSA, muchos de los cuales participan en diferentes proyectos de código abierto.

Este tipo de investigación también tiene como objetivo encontrar la inversión insuficiente, es decir, Software crítico ejecutado en su totalidad por uno o dos voluntarios. Es más común de lo que podría pensar, tan común que una forma común en que los proyectos de Software actualmente miden el riesgo es el "factor del autobús": ¿Todo este proyecto se desmorona si solo una persona es atropellada por un autobús?

Si bien la importancia del Kernel de Linux para los sistemas informáticos del mundo puede ser el problema más apremiante para "Social Cyber", también abordará otros proyectos de código abierto. Ciertos artistas se centrarán en proyectos como Python, un lenguaje de programación de código abierto utilizado en una gran cantidad de proyectos de inteligencia artificial y aprendizaje automático.

La esperanza es que una mayor comprensión facilite la prevención de un desastre futuro, ya sea que sea causado por una actividad maliciosa o no. Prácticamente dondequiera que mires, encuentras Software de código abierto, incluso cuando miras el Software propietario, un estudio reciente mostró que en realidad es 70% o más de código abierto.

Este es un problema de infraestructura crítica y no se tiene el control sobre eso. Y una gran cantidad de investigadores cree que se debe controlar. El impacto potencial es que los Hackers malintencionados siempre tendrán acceso a las máquinas Linux. Eso incluye a los teléfonos. Es así de simple.

Google Ofrece su Software de Código Abierto para Preservar la Seguridad La seguridad informática y el Software de código abierto continúan siendo una gran preocupación para los desarrolladores y las organizaciones. En este sentido, un estudio realizado en 2022 por Synopsys alertó sobre la inquietud que producía la ciberseguridad, y donde se señalaba que el 84% de las bases de código de Software de código abierto contenían alguna vulnerabilidad conocida, un dato que suponía un aumento del 4% respecto al año 2021.

En vista a esto, en el pasado mes de mayo de 2022, Google anunció la salida de su nuevo servicio Assured Open Source Software (OSS asegurado), un servicio de Software de código abierto que permitiría a las empresas defenderse de los ataques a la seguridad. El gerente de soluciones de Software de Synopsys, Mike McGuire, explicó el especial interés que tiene Google en que la comunidad de código abierto sea lo más segura posible.

El lanzamiento de Assured OSS ha venido motivado por la creciente cantidad de ataques cibernéticos dirigidos a proveedores de código abierto, según admitió el gerente de productos, seguridad y privacidad de Google, Andy Chang. Por entonces, se informó de un aumento del 650% de estos ataques a la cadena de suministro de Google ofrece su Software de código abierto para preservar la seguridad. Chang dijo por entonces que "Google está en una posición única para ayudar en esta área, ya que somos colaboradores, mantenedores y usuarios de Software de código abierto desde hace mucho tiempo, y hemos desarrollado un sólido conjunto de tecnología, procesos, capacidades de seguridad y controles".

El proceso, anuncia ahora la compañía especializada en productos y servicios relacionados con internet, Software, dispositivos electrónicos y otras tecnologías, se realizaría escaneando y analizando de forma periódica algunas bibliotecas de Software más conocidas a nivel mundial, en busca de vulnerabilidades.

El lanzamiento de Google es una realidad, y Assured OSS se pone a disponibilidad de los usuarios del sector público y empresarial de forma gratuita, permitiendo que éstos incorporen los mismos paquetes de Software de código abierto que emplea Google.

Assured OSS en Respuesta a las Amenazas Ocultas El servicio Assured Open Source Software llega para los ecosistemas Java y Pynthon gratis, después de que durante mucho tiempo dependiera de bibliotecas a

terceros y tras conocer realmente el carácter de las amenazas cuando la Casa Blanca se vio afectada. Tras este hecho, Google se tomó muy en serio el asunto de la ciberseguridad en la cadena de suministro de Software.

En su comunicado oficial, Google ha asegurado que mantendrá de forma constante las bibliotecas actualizadas buscando ventanas abiertas de vulnerabilidad y detectando nuevas que puedan desarrollarse. Una vez identificado algún peligro, ejecutará sus correcciones para dar solución al peligro en el menor tiempo posible. De esta manera, la analista de ESG Melinda Marks, afirmó categóricamente que "a medida que las organizaciones utilizan cada vez más OSS para ciclos de desarrollo más rápidos, necesitan fuentes confiables de paquetes seguros de código abierto".

Además Marks añadió que "sin la investigación y verificación adecuadas o los metadatos para ayudar a rastrear el acceso y el uso de OSS, las organizaciones corren el riesgo de exponerse a posibles vulnerabilidades de seguridad y otros riesgos en su cadena de suministro de Software. Al asociarse con un proveedor confiable, las organizaciones pueden mitigar estos riesgos y garantizar la integridad de su cadena de suministro de Software para proteger mejor sus aplicaciones comerciales".

Para poder tener acceso a este nuevo servicio, Google ha informado al respecto, que los desarrolladores y las organizaciones sólo deberán registrarse y posteriormente integrar Assured OSS en su proceso de desarrollo. Con este sencillo proceso tendrán toda la seguridad que proporciona el nuevo paquete de la compañía.

La OpenELA (Open Enterprise Linux Association), una asociación formada el año 2023 por CIQ (Rocky Linux), Oracle y SUSE se ha unido para garantizar la compatibilidad con RHEL (Red Hat Enterprise Linux). Dentro de este marco, han presentado el proyecto Kernel-lts, el cual proporcionará soporte adicional para algunas ramas obsoletas de Kernels LTS después de que dejen de recibir soporte oficial.

Con el lanzamiento de este proyecto, la versión 4.14, será la primera rama del Kernel que recibirá este soporte adicional (esta versión del Kernel fue lanzada en noviembre de 2017 y ha recibido soporte durante 6 años). En enero 2024, el equipo de desarrollo del Kernel dejó de mantener esta rama y OpenELA ha asumido el mantenimiento y las actualizaciones para el Kernel 4.14 se lanzarán al menos hasta diciembre de 2024. Tras la última versión del Kernel de Linux 4.14.336, el equipo de OpenELA ha lanzado

las actualizaciones extendidas 4.14.337-openela, 4.14.338-openela y 4.14.339-openela.

El mantenimiento proporcionado por OpenELA seguirá las mismas reglas y procesos que se aplican a los Kernels LTS estables normales. No habrá restricciones adicionales, como la vinculación a equipos o productos específicos. Las actualizaciones se publicarán basadas en el trabajo de seguimiento de correcciones en las ramas actuales del kernel y su migración a las ramas LTS extendidas mantenidas por OpenELA.

Además, la Fundación Linux proporciona ramas SLTS (Super Long Term Support) basadas en los Kernels 4.4, 4.19, 5.10 y 6.1. Estas ramas SLTS se mantienen por separado y reciben soporte durante períodos extendidos de 10 a 20 años. El proyecto Civil Infrastructure Platform (CIP) es responsable de mantener estas ramas SLTS, con la participación de empresas como Toshiba, Siemens, Renesas, Bosch, Hitachi, MOXA, mantenedores de las ramas LTS del núcleo principal, desarrolladores de Debian y el proyecto KernelCI. Estas ramas SLTS están diseñadas para su aplicación en sistemas técnicos de infraestructura civil y en sistemas industriales críticos.

4 Consideraciones y Comentarios Finales

Los paquetes comerciales -de Software privativo- en general proveen un ambiente integrado de trabajo ideal para las empresas de todo tipo, pero además puede ser usado en la preparación de estudiantes para aplicar sus conocimientos al egresar en las diversas áreas de las carreras universitarias, esto les permite laborar en empresas pequeñas, medianas y grandes con un mínimo de capacitación técnica adicional.

En un mercado tan competitivo como el actual, las organizaciones actuales focalizan sus recursos en las estrategias más adecuadas para conducir a la compañía hacia el éxito. Los paquetes comerciales y los incipientes paquetes de Software libre pueden ayudar a conseguir este objetivo, completando la inversión ya realizada en sistemas operacionales.

Pero el hecho de que las organizaciones actuales, manejan una gran cantidad de información, la cual puede o no estar dispersa en sus múltiples sistemas operacionales, requiere usar paquetes que tengan integrado el manejo de las grandes bases de datos distribuidas o centralizadas, esta integración ofrece beneficios adicionales.

Por otro lado, notemos que, una vez que un producto de Software libre ha empezado a circular, rápidamente está disponible a un costo muy bajo. Al mismo tiempo, su utilidad no decrece. El Software libre, en general, podría ser considerado un bien de uso inagotable, tomando en cuenta que su costo marginal es pequeño y que no es un bien sujeto a rivalidad (la posesión del bien por un agente económico no impide que otro lo posea).

Puesto que el Software libre permite el libre uso, modificación y redistribución, a menudo encuentra un hogar entre usuarios para los cuales el coste del Software no libre es a veces prohibitivo, o como alternativa a la piratería (véase 2.5). También es sencillo modificarlo localmente, lo que permite que sean posibles los esfuerzos de traducción a idiomas que no son necesariamente rentables comercialmente.

La mayoría del Software libre se produce por equipos internacionales que cooperan a través de la libre asociación. Los equipos están típicamente compuestos por individuos con una amplia variedad de motivaciones, y pueden provenir tanto del sector privado, del sector voluntario o del sector público.

En México el Software libre nació en las Universidades y los Centros de Investigación. Es por eso que, desde hace cuatro décadas, los estudiantes y los profesores usan Software libre para fines didácticos y de investigación. Las universidades suelen optar por el uso de Software libre en vez de utilizar

Software privativo porque satisface de una mejor manera sus necesidades de cómputo, dada su naturaleza de apertura del código y la libertad de compartir los resultados obtenidos. De forma colateral, no se tienen gastos adicionales derivados del pago de licenciamientos.

Computólogos, físicos, químicos, matemáticos, otros profesionistas y científicos utilizan Software libre como herramienta de investigación y creación. Un claro ejemplo de ello es la llamada Delta Metropolitana, que es una red de supercomputadoras que están en varios puntos de la Ciudad de México, en el CINESTAV, el IPN, la UAM y la UNAM. Esa red de supercómputo utiliza Software libre para consolidar sus recursos, hacer investigación y generar conocimiento.

Por otro lado, dadas las características del Software de código cerrado, un usuario común ignora absolutamente el contenido del mismo y por tanto si existe dentro de las líneas del código alguna amenaza contra su equipo o su información, el usuario no sólo tiene prohibido el intentar eliminar o cambiar esa parte del código sino que puede ser perseguido por la ley por el hecho de intentar conocer si existe tal amenaza en dicho Software.

Además, en una sociedad de la información, el Software se ha convertido en una herramienta importante de productividad y una licencia de Software privativo constituye un acuerdo o contrato entre dos sujetos jurídicos que voluntariamente acuerdan las condiciones de uso de un programa, pero el costo económico de dicha licencia es cada vez más alto y en el caso de instituciones educativas, gubernamentales y sociedades civiles es en la mayoría de los casos -por decir lo menos- prohibitivo.

Hace un tiempo, en varios periódicos de circulación nacional (véase [13]) fue publicado el siguiente anuncio:

> El Instituto Mexicano de la Propiedad Industrial (IMPI) anunció que en las próximas semanas dará inicio una serie de clausuras a negocios que utilicen Software licenciado de manera ilegal; esto como parte del acuerdo que tiene la dependencia con The Software Alliance (BSA) desde el 2002, el cual busca fomentar el uso de programas informáticos legales y disminuir el índice de piratería en el país.
>
> De acuerdo a la BSA, el porcentaje de Software ilegal utilizado en el territorio aún se ubica en un 56 por ciento, cifra considerablemente menor a lo visto en el 2005, cuando el número ascendía

> a más del 65 por ciento. Sin embargo, México continúa siendo uno de los mayores compradores de piratería a nivel mundial, y lo que se busca con este tipo de acciones en el 2013 es disminuir, al menos, un punto porcentual.
>
> "Si como consecuencia de una visita de inspección completa se encuentra la existencia de Software ilegal, se procede a la sanción. En el 2012 incrementaron hasta un 200% las sanciones por el uso ilegal de Software", dijo Kiyoshi Tsuru, director general en México de la BSA.
>
> Aquí es donde algunos se preguntarán, ¿y qué autoridad tiene The Software Alliance para ejecutar estas determinaciones? Pese a que cuenta con el apoyo de empresas como Microsoft, Apple, Autodesk, Adobe, Aveva, AVG, CISCO, Dell, Hewlett Packard, IBM, SAP y Symantec, lo cierto es que la BSA no puede clausurar legalmente ningún negocio. La verdadera autoridad llega en su acuerdo con el IMPI, el cual sí tiene las facultades para aplicar sanciones.

Además, la UNAM, desde junio 9 del 2009 firmó un acuerdo (véase [14]):

> Con el objetivo de fomentar la cultura de la legalidad en lo que se refiere al uso del Software entre los estudiantes, la Universidad Nacional Autónoma de México y la Business Software Alliance (BSA) firmaron un convenio general de colaboración.
>
> Mediante este acuerdo, se promoverá el uso ético de las tecnologías de la información y comunicación, y se compartirán conocimientos en materia de propiedad intelectual y Software, a fin de apoyar el desarrollo y explotación de bienes digitales en la UNAM, así como definir programas para contribuir al avance de un mundo digital seguro, informaron ambas organizaciones en un comunicado.
>
> El secretario general de la máxima casa de estudios, Sergio M. Alcocer Martínez de Castro, reconoció que la UNAM necesita hacer un esfuerzo en el ámbito institucional y en cada una de las entidades que la conforman, para brindar educación a sus alumnos, profesores y trabajadores en este campo.

> "Se pretende", destacó, "que el convenio sea operativo y que se desarrolle en cada una de las entidades con la impartición de cursos y capacitación y en una rendición de cuentas para que el Software que se utilice sea legal".
>
> El funcionario reconoció a los miembros de Business Software Alliance en Latinoamérica, y apuntó que la Universidad Nacional hará lo necesario para facilitar el uso responsable, ético y seguro del Software.
>
> Informó también que ambas partes se reunirán seis meses después del inicio de este convenio de colaboración para analizar los avances.
>
> En tanto, el director General de BSA-México, Kiyoshi Tsuru Alberú, resaltó que con la firma de este convenio podrán impulsar un plan conjunto relacionado con alta tecnología, ética y legalidad "Estamos seguros que en el mediano plazo se tendrán resultados importantes y se podrá hacer la diferencia", señaló.
>
> Por su parte, el abogado general, Luis Raúl González Pérez, comentó que la UNAM busca difundir estos valores entre su comunidad, y llegar especialmente a los estudiantes que inician el bachillerato, porque desde edad temprana es importante fomentar la cultura de la legalidad.

Ante este escenario, una alternativa viable podría ser optar por el Software libre, aunque, pese a su incipiente desarrollo es seguro que en un futuro podría alcanzar a suplir todas las necesidades básicas de los usuarios, dejando la adquisición de paquetes especializados sólo para los cursos avanzados que justifique el uso de Software privativo.

4.1 El Cómputo en Instituciones Educativas

Hace algunos años la disposición de un equipo de cómputo por cada estudiante era algo difícil de satisfacer para las instituciones educativas. Ahora, las cosas son distintas, cada vez más estudiantes disponen y tienen acceso a dispositivos de cómputo -computadoras de escritorio, portátiles, tabletas, y teléfonos inteligentes- que en principio pareciera que permitirían satisfacer la creciente demanda de recursos computacionales de los estudiantes.

Pero una computadora requiere de un sistema operativo además de los diversos paquetes de Software -que estén disponibles para esa versión del sistema operativo- que permitan resolver los problemas para los cuales usa el equipo de cómputo. Aquí es donde empiezan los problemas para los usuarios de equipos de cómputo, puesto que hay una gran cantidad de equipos de cómputo con diversas tecnologías y recursos que soportan alguna versión de sistema operativo acorde a los recursos computacionales del equipo adquirido que no necesariamente soportan a todos y cada uno de los programas de cómputo que el usuario requiere.

Ante la creciente necesidad de programas de cómputo podríamos pensar en que cada usuario que requiera hacer uso de ellos tenga acceso a un equipo de cómputo adecuado, conjuntamente con el sistema operativo que lo soporte. Pero esto dista mucho de la realidad, puesto que la gran mayoría de los usuarios no pueden hacer esos gastos y menos una institución educativa y sus respectivos estudiantes.

¿Entonces qué opciones tenemos para satisfacer la creciente demanda de recursos computacionales?

- Por un lado, si ya disponemos de un equipo de cómputo con su respectivo sistema operativo, entonces hacer uso de sólo aquellos programas de cómputo que nuestro equipo soporte, teniendo cuidado de no instalar programas de cómputo antagonistas.

- Otra opción es, si ya disponemos de un equipo de cómputo, entonces tener dos o más versiones de sistema operativo que permitan instalar una mayor diversidad de programas de cómputo y tener el cuidado de no instalar programas de cómputo incompatibles. Así, dependiendo de nuestras necesidades podemos hacer uso de uno u otro sistema operativo y sus respectivos programas.

- La opción más viable, es una que conjugue las dos anteriores. Pero además, podríamos emular Hardware del que no disponemos mediante el uso de máquinas virtuales, escritorios remotos y virtuales que nos permitirían en un sólo equipo de cómputo usar simultáneamente diversos sistemas operativos para distintas arquitecturas y sus respectivos programas que ahora es posible instalar en las máquinas virtuales programas de cómputo incompatibles de forma aislada unos de otros.

Usando esta última opción es posible satisfacer en un sólo equipo de cómputo una gran variedad de necesidades computacionales. Esto permite que a nivel de usuario (estudiante, ayudante y profesor) o institución educativa, el equipo de cómputo usando Software de virtualización pueda proporcionar un marco que permita satisfacer las diversas y crecientes necesidades computacionales. Pero hay que notar que aún esta opción no está exenta de problemas legales y técnicos, pero en principio es una opción viable para la gran mayoría de los usuarios y la institución educativa.

Tomando esto en cuenta, es viable tener una cantidad adecuada de paquetes de cómputo, que permitieran satisfacer las necesidades especializadas de la gran mayoría de los cursos y estos estar instalados en aquellos espacios en los cuales se asignarían los cursos, además de las áreas comunes de cómputo en la que los estudiantes requiriesen hacer uso de dichos paquetes. Además, de proporcionar un mecanismo para que los profesores y ayudantes que requieran enseñar algo con alguna versión privativa que no se disponga, sea implementada -en medida de lo posible- en los paquetes disponibles.

Pero hay que hacer notar, que no todas aquellas funciones que hace una versión particular de un paquete, es posible hacerlas con otras versiones o paquetes alternativos. Esto es muy común con ciertas actividades especializadas -al hacer cálculo simbólico, cálculo numérico, manejo de datos y trabajar en entornos de desarrollo-. Ello implicaría, por un lado restringir el Software instalado en los equipos de cómputo o por el otro instalar todas y cada una de las solicitudes de Software, aún cuando se requiera más de una versión de un paquete particular.

El restringir el Software instalado, impediría al profesor -que así lo requiera por la libertad de cátedra- enseñar aquello que considera que es necesario -en particular el manejo de uno o más paquetes especializados de cómputo- para proporcionar las herramientas básicas a sus alumnos y que estos deben de dominar para aprobar su curso.

En el caso de dar flexibilidad, para que cada profesor solicite la instalación del paquete o los paquetes que requiera para sus cursos, implica que el Software solicitado puede o no contar con licencia adecuada de uso. Así, se estaría permitiendo que se tenga instalado Software del que se viola la licencia de uso.

En cuanto a tener la lista definitiva de Software que usarán todos y cada uno de los profesores o ayudantes de los cursos asignados a un espacio es difícil tener antes del inicio del curso -por la constante evolución del Software y las cambiantes necesidades de la enseñanza-, además de depender de la forma de

asignación de estos en los laboratorios y talleres de cómputo. En cuanto a la solicitud para hacer la instalación correspondiente, se requiere tener certeza de en qué espacio serán asignados todos y cada uno de los cursos.

Por ello se han buscado opciones[9] -no siempre las más adecuadas o lícitas (véase 2.5)- para que sin importar en qué espacio sea asignado el curso -siempre y cuando el equipo de cómputo lo soporte- se tenga desde los primeros días de uso del espacio el paquete solicitado y en casos excepcionales el tiempo de espera sea menor a unos horas o días sin importar la plataforma -Windows o Linux- o el tipo de Software solicitado -libre o privativo-.

Por ejemplo, se puede optar por la virtualización[10], usando como sistema operativo base Debian GNU/Linux estable, instalando como paquete de virtualización a KVM/QEMU. Aquí, se montarían las múltiples máquinas virtuales que serían ejecutadas según las necesidades del usuario -para cualquier versión de Windows, Linux u otro sistema operativo de cualquier arquitectura de Hardware soportada por QEMU-. Para controlar la actualización de las máquinas virtuales sin que se requiera intervención del usuario, se usaría RSYNC tunelizado mediante SSH que sincronizaría las máquinas virtuales y la configuración del equipo base de forma remota.

Para tener la flexibilidad anteriormente comentada, es necesario poder contar con distintas versiones de sistemas operativos, de cada una de las versiones -en caso de Windows, tener independientemente los Service Pack-. De tal forma que sea posible instalar cada versión de Software solicitada en la plataforma adecuada, teniendo en cuenta que muchas versiones del Software son mutuamente excluyentes para ser instaladas en una misma versión del sistema operativo simultáneamente.

Por todo lo anterior, el uso de máquinas virtuales -que permiten tener múltiples versiones de sistemas operativos independientemente, así como de una versión particular tener por separado cada una de ellas con los respectivos Service Pack- es una opción viable para proporcionar el servicio de

[9] En el caso que el equipo sólo tenga un sistema operativo sin virtualización, es necesario esperar a que las asignaciones de los cursos y sus respectivas peticiones de uso de paquetes de cómputo estén completas, para entonces proceder a realizar instalación del Software que no sean antagónicos. Nótese que, por lo general, los cursos requieren el uso de los equipos de cómputo y el Software solicitado de forma inmediata, por lo cual esperar tiempo (días) para tener acceso al mismo no es una opción viable.

[10] Una vez creada la máquina virtual, esta es un archivo que puede ser copiado o descargado de la red, por ello el usuario -estudiante, ayudante o profesor- puede llevarse la máquina virtual para hacer uso de ella en el equipo al que tenga acceso, teniendo como único requisito tener instalado el programa de virtualización.

instalación centralizada de los diversos paquetes de cómputo solicitados por los profesores de las diversas carreras universitarias. Esta opción minimiza los tiempos de espera para la instalación de un paquete en particular y agiliza las prestaciones a todos y cada uno de los grupos que se atienden semestralmente en los cientos de equipos en los laboratorios y talleres de cómputo.

4.2 Integración del Cómputo en Ciencias e Ingenierías

El uso de programas de cómputo está integrado a las carreras de Ciencias e Ingenierías desde hace mucho tiempo, pero la gran mayoría se realiza con productos propietarios, lo cual no representa ningún problema técnico, pero sí un problema para la institución y estudiantes, ya que las versiones actualmente usadas, no son del todo compatibles entre sí, ello implica que se requiere o tener la última versión del producto o diferentes versiones del mismo para trabajos cotidianos en una misma computadora.

El uso de programas de cómputo de Software libre está cada día más integrado al uso cotidiano que hacen profesores, ayudantes y estudiantes en la carreras de Ciencias e Ingenierías, pero todavía para el Sistema Operativo Windows, así como para paquetes de uso común, no ha sido posible encontrar un adecuado reemplazo, los más comunes son MATLAB, Mathematica, Maple, SPSS, SAS y Microsoft Office.

Para las Universidades, el contar con las licencias necesarias para que cada máquina a la que los alumnos tienen acceso cuente con una, es en extremo prohibitivo por el costo. Esto mismo sucede en el caso de los estudiantes, pues el costo de una sola licencia para uso académicos es onerosa más si consideramos la diversidad de programas requeridos para una sola meteria y esto pasa con cada uno de los cursos de la carrera.

Es por ello que el uso de herramientas de Software libre se visualiza como un reemplazo natural a los paquetes propietarios, pero la realidad dista de ser tan simple. Ya que, actualmente no es posible obtener las características mínimas en Software libre para que puedan ser un reemplazo real de los paquetes de propietarios. Este hecho ha ocasionado que existe un uso cada vez más generalizado entre profesores y alumnos a usar Software sin la licencia respectiva (véase 2.5).

por ejemplo, en la UNAM, a través de la Dirección General de Cómputo y de Tecnologías de la Información y Comunicación se dispone de un restringido número de paquetes y versiones que son puestos a disposición de la comunidad universitaria para usar en los equipos personales sin aparente costo para el

usuario final -pero el costo de dichos paquetes son deducidos por la empresa como una donación, lo cual sí implica un costo real que se deduce en el ejercicio fiscal de la empresa donante y éste repercute en los ingresos que el gobierno no recaudará por motivo de impuestos-.

4.3 Ventajas, Desventajas y Carencias del Software Libre

Notemos que la ventaja de tener múltiples herramientas para realizar operaciones elementales y avanzadas de paquetes de cálculo numérico, simbólico, estadístico y ofimático es en sí misma una gran ventaja. Para los centros universitarios y usuarios ocasionales, las herramientas de Software libre son una herramienta invaluable, en el caso de empresas que requieren usar opciones avanzadas o generadas por terceros, los paquetes propietarios destacan como herramientas de trabajo óptimas. Pero para todos los casos, hay que destacar:

- Funcionalidades básicas: Todos los paquetes implementan las funcionalidades básicas, ya que todos los paquetes llevan años desarrollándose.
- Funcionalidades avanzadas: Por mucho, los paquetes propietarios tienen implementadas cientos de funciones avanzadas que pueden ser muy útiles para usuarios avanzados, pero rara vez son usados por los usuarios noveles o cotidianos.
- Fiabilidad: En los paquetes en desarrollo son comunes las caídas del programa, pero en los de Software propietario se destaca por ser más fiable que los demás.
- Información: El Software propietario son paquetes con una abundante bibliografía y la propia ayuda del programa.
- Facilidad de Manejo: Ninguno de los programas presenta grandes dificultades a la hora de su uso. Pero en menor o mayor medida, todos los paquetes del Software libre presentan entornos de desarrollo funcional, pero perfectible.
- Costo: El costo de las diversas versiones de Software propietario suele ser prohibitivo para instituciones educativas y usuarios ocasionales, en

el caso del Software libre, los paquetes se pueden descargar de la red sin más costo que el acceso a Internet y los medios de instalación cuando son requeridos.

El Software libre es aún joven, en los miles de proyectos actuales se está trabajando a diario en mejorar la parte computacional de los algoritmos involucrados en el paquete, haciendo y puliendo interfaces gráficas, generando ayuda en línea así como la documentación necesaria para que usuarios noveles y avanzados usen la mayor cantidad de opciones programadas en los paquetes.

Para muestra de este maravilloso avance, tomemos el proyecto del Kernel de Linux y su uso en los sistemas operativos Android, Ubuntu, Debian GNU/Linux, que actualmente se ejecuta en millones de equipos y contiene miles de aplicaciones y están soportados por una gran cantidad de usuarios y empresas comerciales. Estos han logrado desplazar a muchos de sus competidores por sus múltiples bondades y bajo costo de desarrollo, al reusar miles de aplicaciones ya existentes que usan Software libre y permitir desarrollar otro tanto de aplicaciones bajo una plataforma que se ejecutan en los más diversos procesadores.

Así también, en los últimos años, muchos proyectos han pasados de ser simples programas en línea de comandos a complejas aplicaciones multiplataforma -ejecutan en distintos sistemas operativos como son Windows, Linux y Mac- con ambientes gráficos multimedia que en muchos casos han superado a sus contrapartes comerciales, por ejemplo los navegadores Web tipo FireFox y la suite ofimática tipo LibreOffice, entre muchos otros.

4.4 Comentarios Finales

A diferencia de otros paquetes, SPSS, SAS, Microsoft Office, etc. Ofrecen soluciones en forma de una suite completa para la gestión de información para encontrar el llamado poder del conocimiento, pero el costo de las versiones completas y aún las educativas es prohibitivo para la gran mayoría de las instituciones educativas, en particular para la UNAM. Por ello, el resto de los paquetes propietarios y libres ofrecen una ventaja competitiva, al permitir al profesor y sus estudiantes contar con versiones completas y funcionales en las que pueden ser aplicados los conocimientos adquiridos en los diversos cursos de la carrera.

Por otro lado, para reforzar la apropiación del Software libre por parte de la comunidad de la UNAM, es necesario proporcionar a la comunidad

demostraciones y cursos cortos de las herramientas de Software libre, iniciando con mostrar el uso de sistemas operativos libres basados en Linux. Ello es posible haciendo uso de los sistemas llamados Live[11], ya que cada alumno puede probar y usar el sistema operativo en conjunto con cientos de herramientas libres, sin la necesidad de instalar Software en la máquina que utilice para practicar. Cuando el alumno se sienta cómodo con el sistema, es posible ayudarlo a instalar mediante tutoriales en línea y/o presenciales el sistema en su equipo de cómputo.

Lo mismo es posible hacer, al preparar demostraciones del Software que puede reemplazar paquetes muy difundidos en la comunidad como son: MATLAB, Mathematica, Maple, SPSS, SAS y Microsoft Office. Estos cursos no necesariamente se centrarían en las similitudes o diferencias entre paquetes libres y propietarios, más bien, para cautivar a usuarios noveles y futuros ayudantes a dar cursos completos de las herramientas libres mostrando su

[11] Un Live CD/DVD o USB, más genéricamente Live Distro, es un sistema operativo almacenado en un medio extraíble, tradicionalmente un CD/DVD o USB (de ahí sus nombres), que puede ejecutarse directamente en una computadora.

En la historia más reciente de Linux, las llamadas distribuciones Live Distro se han vuelto muy populares porque le permiten probar una distribución de Linux sin siquiera instalarla en el equipo. Esto es excelente porque no tiene todas las molestias de volver a particionar el disco o instalarlo sobre su sistema operativo (Windows/Mac OS). Simplemente puede colocar el CD/DVD o USB para una distribución en vivo e iniciar la computadora desde ahí. Por lo general, obtiene la mayor parte de la funcionalidad principal de la distribución, por lo que realmente puede evaluar si la distribución es para usted antes de elegir instalarla de verdad.

Normalmente, una versión Live viene acompañado de un par de aplicaciones. Algunos Live CD/DVD o USB incluyen una herramienta que permite instalarlos en el disco duro. Otra característica es que por lo general no se efectúan cambios en la computadora utilizada.

Para usar una versión Live es necesario obtener uno (muchos de ellos distribuyen libremente una imagen ISO que puede bajarse de Internet y grabarse en disco/USB) y configurar la computadora para que arranque desde la unidad lectora, reiniciando luego la computadora con el disco en la lectora o USB, con lo que el que el sistema Live se iniciará manualmente.

Uno de los mayores inconvenientes de este sistema es el mal uso de una gran cantidad de memoria RAM, una parte para su uso habitual y otra para funcionar como el disco virtual del sistema. En el arranque, se le pueden dar distintos parámetros para adaptar el sistema a la computadora, como la resolución de pantalla o para activar o desactivar la búsqueda automática de determinado Hardware.

Otro inconveniente es el rendimiento de la Live Distro, pues la velocidad de transferencia de las unidades lectoras CD/DVD o USB es muy inferior a la de los discos duros. Una vez instalada en la computadora se apreciará la velocidad real de la distribución.

aplicabilidad en diferentes ramas de las matemáticas aplicadas.

Para realizar dichos cursos, se cuenta con todos los recursos necesarios. Por un lado, se dispone de laboratorios y talleres con Software libre instalado en los equipos de cómputo, además, se pueden usar los sistemas "Live" que pueden ser proporcionados en DVDs o en unidades flash USB. Estas últimas, proporcionan mejor rendimiento, pueden ser actualizadas y reutilizados tantas veces como sea necesario para conocer uno o más sistemas operativos. Estos sistemas "Live" pueden ser generados por el propio usuario, usando las decenas de paquetes disponibles en Windows o Linux que generan sistemas "Live" a partir de las imágenes ISO bajadas de la red -por ejemplo, de sistemas operativos como Ubuntu, Debian, etc.-.

De esta forma, se puede coadyuvar a que alumnos, ayudantes y profesores conozcan el mundo del Software libre, para que con el tiempo se adopte su uso, sin dejar de lado, el proporcionar cuando sea necesario, cursos de Software privativo pero siempre teniendo en cuenta que se puede -en medida de lo posible- trabajar con paquetes alternativos, como los que proporciona el Software libre.

Además, el Software libre ofrece una ventaja competitiva, al permitirle al profesor y sus estudiantes contar con versiones completas y funcionales en las que pueden ser aplicados los conocimientos adquiridos en los diversos cursos de las carreras de Ciencias e Ingenierías, dejando el manejo especializado de paquetes a cursos avanzados o para cuando el educando realice sus prácticas profesionales. De esta forma se pueden preparar a los estudiantes para aplicar sus conocimientos al egresar en diversas áreas de la carreras de Ciencias e Ingenierías y con pocos conocimientos técnicos adicionales puedan laborar en pequeñas, medianas y grandes empresas.

5 Bibliografía

Este texto es una recopilación de múltiples fuentes, nuestra aportación -si es que podemos llamarla así- es plasmarlo en este documento, en el que tratamos de dar coherencia a nuestra visión de los temas desarrollados.

En la realización de este texto se han revisado -en la mayoría de los casos indicamos la referencia, pero pudimos omitir varias de ellas, por lo cual pedimos una disculpa- múltiples páginas Web, artículos técnicos, libros, entre otros materiales bibliográficos, los más representativos y de libre acceso los ponemos a su disposición en la siguiente liga:

Herramientas
http://132.248.181.216/Herramientas/

Referencias

[1] https://www.gnu.org/philosophy/free-sw.es.html 22

[2] https://es.wikipedia.org/wiki/Software_libre 22

[3] https://www.hispaLinux.es/SoftwareLibre 22

[4] https://es.wikipedia.org/wiki/Software_propietario 20

[5] Diferentes Tipos de Licencias para el Software, https://www.gnu.org/licenses/license-list.html 22, 32

[6] Proyectos de Software Sourceforge, http://sourceforge.net/ 7, 8, 9

[7] Google Code, http://code.google.com 8, 9

[8] Software proyecto GNU, http://www.gnu.org/Software/Software.es.html 7

[9] FSF, Free Software Fundation, http://www.fsf.org/ 7, 22, 32

[10] GNU Operating System, http://www.gnu.org/ 22, 32

[11] GCC, the GNU Compiler Collection, http://gcc.gnu.org/ 7

[12] The Linux Kernel Archives, http://www.Kernel.org/ 7

[13] El economista, https://eleconomista.com.mx/tecnociencia/2013/01/22/clausuraran-negocios-mexico-uso-ilegal-Software 90

[14] PCworld, http://www.pcworld.com.mx/UNAM-y-BSA-promueven-el-uso-de-Software-legal/

91



Declaramos terminado este trabajo sufrido, ideado y llevado a cabo entre los años 2020 al 2026, aún y a pesar de impedimentos tales como: la mala suerte, la desventura, el infortunio, la incomprensión, la gripe, el COVID-19, la migraña, las horas de frío y calor, la tristeza, la desesperanza, el cansancio, el presente, el pasado y nuestro futuro, el que dirán, la vergüenza, nuestras propias incapacidades y limitaciones, nuestras aversiones, nuestros temores, nuestras dudas y en fin, todo aquello que pudiera ser tomado por nosotros, o por cualquiera, como obstáculo en este tiempo de mentiras, verdades, de incredulidad e ignorancia o negación de la existencia real y física de la mala fe.

Atentamente

Antonio Carrillo Ledesma
Karla Ivonne González Rosas

