



# Principios básicos de Linux y software libre

## Basic Principles of Linux and Free Software

Alfonso Fernández<sup>1</sup>

**Para citar este artículo:** Fernández, A. (2017). Principios básicos de Linux y *software* libre. *TIA*, 5(2), pp. 262-268.

### Resumen

En este artículo se pretende brindar al lector un conocimiento básico sobre todo lo relacionado con el *software* libre, además de abordar cómo se está apoderando poco a poco del mundo y de las diferentes áreas de las TI. Se hablará acerca de su historia, sus inicios, sus fortalezas, debilidades, el estado actual y las oportunidades que brinda, dando así la capacidad al lector de entender a qué se refiere con *software* libre y que no es solamente Linux o Android, mucho menos que es solo para uso de ingenieros informáticos o de sistemas, sino que también es un mundo que se abre más y más al usuario final, con diferentes formas de disfrutar sus bondades y estrategias de una forma segura, económica además de eficiente, cada vez más amigable y de acceso global con gran información en la red para su óptimo manejo.

**Palabras clave:** ccódigo, implementación, Linux, software libre, tecnologías de la información (TI), Unix.

### Abstract

The purpose of this article is to offer to the reader a basic knowledge about everything related with free software and how is empowering little by little of the world and the different IT's areas, we're going to talk about its history, beginning, strengths, weakness, its current state and the opportunities that bring us the free software, giving to the lector the capacity to understand the free software and that is more than Linux, Android, or that is just for engineers, but also is a world that is opening more and more to the final user with different ways to enjoy of its benefits and strategies in a secure, efficient and economic way; everytime more friendly with global access and a wide information about its optimal management.

**Keywords:** code, implementation, information technology's (IT), free software, Linux, Unix.

### ARTÍCULO DE CORTO

**Fecha de recepción:**  
01-06-2017

**Fecha de aceptación:**  
14-07-2017

ISSN: 2344-8288

Vol. 5 No. 2

Julio - diciembre 2017

Bogotá-Colombia

<sup>1</sup> Ingeniero de Sistemas, Universidad INCCA de Colombia; Especialista en Ingeniería de Software, Universidad Distrital Francisco José de Caldas. Ingeniero de sistemas en Arquitectura & Concreto. Correo electrónico: pocho970@hotmail.com

## INTRODUCCIÓN

Este ensayo va orientado a toda persona que le interese saber sobre principios básicos y algo de la historia de Linux, por lo cual se abordarán cinco secciones que hablarán acerca de Linux; teniendo en cuenta lo anterior, se abordarán la historia, los inicios, las fortalezas y debilidades con el usuario, estado actual y las oportunidades que actualmente nos brinda el *software* libre para su correspondiente uso.

## HISTORIA E INICIOS

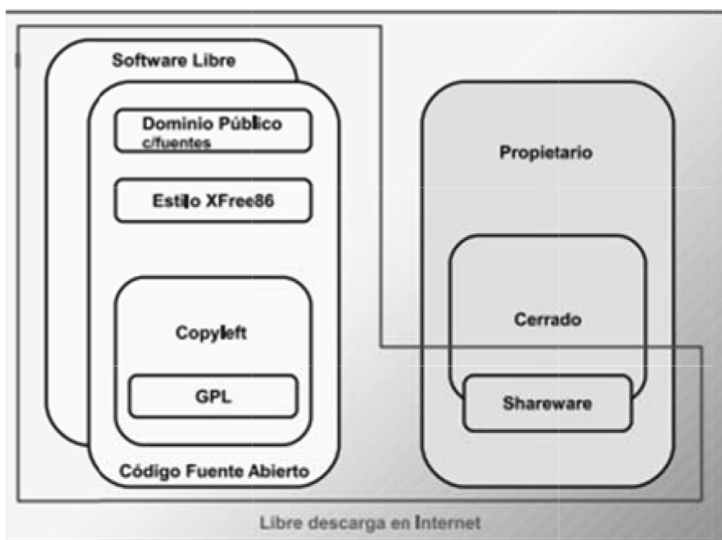
El movimiento de *software* libre surge a principios de 1980 con Richard Stallman, del Laboratorio de Inteligencia Artificial del MIT. Crean en 1985 la Fundación GNU para avanzar el movimiento y fomentar el desarrollo de *software* libre. Las computadoras sin la existencia de herramientas de *software* no son de utilidad, por ello, se enfocarían los esfuerzos a desarrollar programas para hacer

al *hardware* útil. Saliendo de la comunidad académica, la concepción de *software* libre mantiene el espíritu de que todo conocimiento debe ser compartido con el resto del mundo; así, tanto el conocimiento, como el *software*, no deben tener propietarios (Stallman, 1994).

El *software* libre (libre como en libertad) cumple las cuatro libertades:

- Ejecutar el programa con cualquier propósito (privado, educativo, público, comercial, etc.).
- Estudiar y modificar el programa (para lo cual es necesario poder acceder al código fuente).
- Copiar el programa de manera que se pueda ayudar al vecino o a cualquiera.
- Mejorar el programa y hacer públicas las mejoras, de forma que se beneficie toda la comunidad.

Existe otra familia de *software*, el *software* de código fuente abierto (OpenSource.org), que en cierta forma coincide con algunos principios del *software* libre. Para propósitos del presente texto, no se marcarán las diferencias y lo serán incluidos en una sola gran familia (Figura 1).



**Figura 1.** Diferencias entre *software* libre y propietario

Fuente: [1].

## Años setenta

- El *software* empieza a ser privativo “por defecto”.
- Esfuerzos “aislados”: TeX, Spice, etc.
- En general, el objetivo es hacer una herramienta determinada.
- A veces, motivos éticos (por ejemplo, costumbre en la comunidad matemática).
- A veces, motivos prácticos (por ejemplo, difusión de una nueva tecnología).

## El surgimiento de Unix

- El nacimiento de Unix, una auténtica revolución del *software*.
- En 1969 Ken Thompson inventó Unix (mismo año que Arpanet).
- Surge de los deshechos de Multics, en AT&T (Bell Labs).
- Dennis Ritchie inventa un nuevo lenguaje llamado C para usarlo en el Unix de Thompson.
- Primer sistema operativo portable y modular (KISS), frente a anteriores sistemas incompatibles y costosos.
- Se extiende rápidamente y de forma no oficial por AT&T y por Arpanet (*hardware* distinto, gracias a C).
- Acuerdo judicial (*antitrust*) de 1956 impide a AT&T comercializar Unix, debe licenciarlo (con fuentes) a quien se lo solicite.
- CSRG (*Computer Systems Research Group*) de Berkeley.
- Importancia de compartir fuentes (cultura Unix “original”).
- Limitado por la licencia AT&T (poco desde el punto de vista práctico, todos la tenían).
- Financiado por DARPA (DoD).
- Utilizado por mucho *software* propietario (SunOS, Ultrix, etc.).
- Primera internet.
- Implementaciones de referencia, disponibles para todos: la base de los estándares actuales.
- La red como herramienta de cooperación (*news*, *ftp*, *e-mail*).

- La comunidad de usuarios proporciona el mejor soporte.
- Falso mito de los ataques nucleares.

## Años ochenta: el fin de los viejos tiempos

- En 1983, DEC cancela la línea PDP-10. ITS ya no tiene futuro (no era portable).
- Se extienden los acuerdos de no divulgación.
- Comienza a despegar la gran industria del *software* privativo, basada en el secreto (binarios), en la venta de licencias y en la privatización de las fuentes.
- Unix e internet: choca el modelo privativo (AT&T) contra el modelo abierto (BSD).

## Años ochenta: el proyecto GNU

- Stallman abandona el MIT en 1984 para poder dedicarse al proyecto GNU (GNU's Not UNIX!).
- En 1985 Stallman publica el manifiesto GNU: sienta los fundamentos éticos del *software* libre.
- La meta es construir un sistema completo libre, alternativo a Unix.
- Crea la infraestructura básica: editor (emacs), compilador (gcc), depurador (gdb), gmake.
- Crea la Fundación de *Software* Libre (1985) para apoyar el proyecto GNU.
- Fundamentos legales: la GPL (1989).
- Trabajo muy estructurado y con metas claras.
- A principios de los 1990 GNU tenía su sistema casi completo, faltaba el núcleo.

## Los años noventa: el nacimiento de Linux

- Linux es un kernel.
- Lo inicia Linus Torvalds, en 1991, y *just for fun*.
- Existían ya sistemas operativos libres casi completos (GNU y Unix BSD).
- Desde que liberó la primera versión (0.01) se van uniendo cientos de desarrolladores.
- Se adopta la licencia GPL.

- En marzo 1994 aparece versión 1.0.
- Linux es solo un kernel, necesita algo más para funcionar.
- Al proyecto GNU le falta un núcleo en 1990.
- Desarrollo del proyecto Hurd, arquitectura de microkernel (Mach): sin resultados.
- Se adopta temporalmente como núcleo para GNU.
- Proliferan las distribuciones GNU/Linux: Slackware, Debian, Red Hat, SuSE, Gentoo, etc. [2].

## FORTALEZAS Y DEBILIDADES DEL SOFTWARE LIBRE

### Ventajas

- Existen aplicaciones para todas las plataformas (Linux, Windows, Mac Os).
- El precio de las aplicaciones es mucho menor, la mayoría de las veces son gratuitas.
- Libertad de copia.
- Libertad de modificación y mejora.
- Libertad de uso con cualquier fin.
- Libertad de redistribución.
- Facilidad a la hora de traducir una aplicación en varios idiomas.
- Mayor seguridad y fiabilidad.
- El usuario no depende del autor del *software*.

### Inconvenientes

- Algunas aplicaciones (bajo Linux) pueden llegar a ser algo complicadas de instalar.
- Inexistencia de garantía por parte del autor.
- Interfaces gráficas menos amigables.
- Poca estabilidad y flexibilidad en el campo de multimedia y juegos.
- Menor compatibilidad con el *hardware*.

## SOFTWARE PROPIETARIO

### Ventajas

- Facilidad de adquisición (puede venir preinstalado con la compra del equipo, o encontrarlo fácilmente en las tiendas).
- Existencia de programas diseñados específicamente para desarrollar una tarea.
- Las empresas que desarrollan este tipo de *software* son por lo general grandes y pueden dedicar muchos recursos, sobre todo económicos, en el desarrollo e investigación.
- Interfaces gráficas mejor diseñadas.
- Más compatibilidad en el terreno de multimedia y juegos.
- Mayor compatibilidad con el *hardware*.

### Inconvenientes

- No existen aplicaciones para todas las plataformas (Windows y Mac OS).
- Imposibilidad de copia.
- Imposibilidad de modificación.
- Restricciones en el uso (marcadas por la licencia).
- Imposibilidad de redistribución.
- Por lo general son ser menos seguras.
- El coste de las aplicaciones es mayor.
- El soporte de la aplicación es exclusivo del propietario.
- El usuario que adquiere *software* propietario depende al 100% de la empresa propietaria.

Se podría decir que donde una opción flojea la otra cobra más fuerza, por ejemplo, en el terreno multimedia y juegos; no obstante, el *software* libre está en constante crecimiento y evolución, logrando día a día mejorar y eliminar los aspectos en que flojea. Por otro lado, las empresas de *software* propietario empiezan a invertir en el *software* libre, en vista de que este modelo es mucho más eficiente que el modelo tradicional [3].

## ESTADO ACTUAL Y OPORTUNIDADES QUE NOS BRINDA EL SOFTWARE LIBRE EN COLOMBIA

Por una parte, están quienes defienden los beneficios que traería al Gobierno la propuesta, promovida por el congresista Gustavo Petro, por otra, los que la acusan de violar la libertad del mercado.

Para empezar, vale la pena aclarar que el *software* libre, como lo define el documento radicado en el Senado por Petro, es todo aquel que brinda al usuario la libertad de estudiar y manipular su código fuente (el conjunto de reglas que le permiten operar) y de distribuir copias bajo las mismas condiciones que le fueron otorgadas con el programa original. Bajo esta categoría, se encuentran estándares como Linux y Unix, que han servido como base para el desarrollo de infinidad de aplicaciones y herramientas empleadas en el mundo tecnológico actual.

De otro lado, se encuentra el denominado *software* propietario, es decir, todo aquel cuyo autor no está dispuesto a conceder licencias libremente y que cuenta con un código fuente cerrado y secreto; esto significa que distribuir copias de la herramienta sin autorización es ilegal y que, aparte de su desarrollador, nadie conoce cómo funciona su estructura básica. Los programas propietarios más utilizados en el mundo son los diseñados por Microsoft, como las plataformas Windows, el navegador Internet Explorer y las herramientas de Office.

El problema, sostiene Carlos Santanilla, presidente de la Federación Colombiana de la Industria del *software* (Fedesoft), radica en que una ley de esas características violaría algunos de los principios establecidos en la Constitución, como el derecho a la igualdad.

Según el ejecutivo, el proyecto genera discriminación contra los fabricantes y desarrolladores de *software* comercial protegido, sin presentar una justificación técnica o legal

suficientemente fuerte. De otro lado, el artículo 333 de la Constitución consagra la libre competencia económica como un derecho de todas las personas.

La obligación de utilizar exclusivamente *software* libre señala Santanilla, coarta la posibilidad del Estado para escoger programas con base en aspectos normales del mercado, como precio y condiciones de soporte técnico. En opinión de los analistas que se oponen al proyecto, esto último favorecería la aparición de un monopolio de creadores de aplicaciones libres, pues excluiría de entre los proveedores de soluciones para el Estado a todos aquellos desarrolladores de programas de licencia.

Sin embargo, según especialistas en plataformas de código abierto, la propuesta no establece que ciertos tipos de *software* propietario deban salir del mercado gubernamental, lo que la propuesta pretende —explica Ricardo Naranjo, gerente de Skina, una compañía dedicada al desarrollo de aplicaciones en Linux— es proteger los intereses estatales al establecer ciertas reglas en la adopción de programas. Esto no representa ningún tipo de discriminación; de hecho, el proyecto nunca prohíbe las herramientas creadas por firmas como Microsoft, señala Naranjo, pero plantea que se debe levantar el velo que existe sobre su código fuente.

Así las cosas, agrega el experto, analistas designados por los organismos públicos de control podrían escudriñar en sus entrañas para entender su funcionamiento y modificar ciertas características que podrían resultar riesgosas.

Es bien sabido, afirma Naranjo, que programas como Windows cuentan con instrumentos de espionaje que envían información de su computador anfitrión a Microsoft. Si estas acciones son ejecutadas en sistemas gubernamentales, información vital para el Estado estaría en serio peligro. No obstante, Víctor Duque, especialista en sistemas libres y exasesor del Gobierno en materia de tecnología, explica que la seguridad de la información mencionada estaría más comprometida con programas libres.

Desglosar el código fuente de un *software* abierto, sostiene Duque, no representa ninguna dificultad para un programador, esto le permitiría encontrar la forma de romper las barreras que resguardan los datos almacenados en el sistema, algo que sería mucho más complicado si se trata de una aplicación cuya estructura permanece en secreto.

Por otra parte, quienes apoyan la aprobación de la ley aseguran que esta permitirá promover la igualdad de acceso a la información pública por parte de los ciudadanos, lo que evitaría forzarlos a depender de proveedores únicos de aplicaciones.

En algunos casos, comenta Ricardo Laverde, asesor informático de varias compañías nacionales, los documentos puestos en internet por ciertas oficinas públicas se encuentran en formatos específicos, como Microsoft Word 97; así, un usuario que no tenga un procesador de texto compatible con este estándar no podrá acceder a la información. En pocas palabras, concluye Laverde, la ley garantizaría la funcionalidad de estrategias de fiscalización estatal por parte de cualquier usuario del ciberespacio.

Al respecto, Enrique Fajardo, diseñador de páginas virtuales para el Estado, sostiene que, en la actualidad, son pocos los sitios web gubernamentales que no ofrecen soporte para varias plataformas; además, la mayoría de esa información puede ser consultada sin necesidad de programas diferentes al navegador.

De otro lado, Laverde asegura que, de contar con *software* libre en sus equipos, el Gobierno podría garantizar la perennidad de su información. Con su desaparición, afirma, programas como WordPerfect se llevaron grandes cantidades de documentos que nunca podrán recuperarse; no obstante, Duque explica que, según la experiencia de los mercados de tecnología, cada vez que un programa popular deja de ser distribuido, otras aplicaciones empiezan a absorber sus características y brindan soporte a la información almacenada en el formato.

En todo caso, el argumento más popular a la hora de defender las bondades del proyecto de ley radica en los beneficios económicos que su aplicación traería a las oficinas públicas; el costo de licenciamiento para plataformas propietarias comenta Jorge Cardona, experto en soporte técnico para soluciones Linux, puede alcanzar los 200 000 pesos por unidad. Una entidad del Gobierno puede tener 500 computadores, lo que significa, en principio, 100 millones de pesos, solamente por el permiso de utilización del programa.

Según el especialista, el *software* libre es gratuito en muchas oportunidades, o notoriamente más barato que su contraparte comercial en el peor de los casos; esa condición representa un gran ahorro para quienes implementan dicho tipo de *software*, finaliza Cardona.

No obstante, un estudio realizado por la firma Gartner Group, publicado en internet, destaca que los costos de adquisición de *software* por parte de una compañía alcanzan solo 8% de la inversión total de implementación; así, el 92% restante corresponde a capacitación, soporte, mantenimiento y administración del sistema.

Fuera de las implicaciones económicas mencionadas, concluye Duque, la prevalencia de un *software* u otro no debe ser determinado por las leyes del mercado: un buen producto sobrevivirá a la competencia y uno no tan fuerte estará destinado a desaparecer.

Por supuesto, añade Fajardo, Linux y otros estándares abiertos son más estables que plataformas como Windows, pero su conveniencia depende de los requerimientos del usuario, no de una ley del Gobierno.

De la misma forma piensa Oswaldo Alarcón, gerente de operaciones y de programas de gobierno de IBM, compañía que ha concentrado buena parte de sus desarrollos de *software* en sistemas Linux, el Estado no debe obligar a nadie a usar ningún tipo de *software*, ni libre ni propietario. Lo peor que se puede hacer con las plataformas libres, finaliza Alarcón, es regularlas mediante una ley; al igual que todos los programas, estas

estructuras informáticas deben ser regidas por las leyes del mercado.

Colombia no es el primer país en el que se pretende establecer la obligatoriedad del *software* abierto en las entidades estatales, en Malasia el Gobierno decidió adoptar desde las 2001 plataformas de libre distribución, con el propósito de reducir los costos de propiedad, estimular el desarrollo tecnológico local, mejorar la seguridad digital y garantizar estabilidad tecnológica del país. En China, el Estado contrató el desarrollo de sistemas de código abierto para disminuir la dependencia de compañías extranjeras que proveían estándares empleados en el control de la infraestructura digital del país.

También en Argentina se encuentra en curso un proyecto de ley que busca obligar al Estado a implementar exclusivamente programas libres; sin embargo, la propuesta cuenta con varios detractores porque representará un desembolso económico que el Gobierno de ese país no puede costear.

En Perú, un proyecto de ley presentado por el congresista Edgar Villanueva, y que tiene el mismo objetivo que el promovido en Colombia, se ha mantenido en el centro de la opinión pública durante varios meses. El establecimiento de leyes similares es estudiado en otras naciones como Venezuela, Bélgica, Finlandia y Noruega [3].

## CONCLUSIONES

Se puede decir que en la historia de la informática moderna es importante conocer las bondades del

*software* libre, entender que su desarrollo permite más fácil el acceso al usuario común y sirve como pauta para su implementación, no solamente en el ámbito privado sino también en el público. Como uno de los pioneros en la Alcaldía de Bogotá en la implementación de *software* libre en la Secretaria de Gobierno de Bogotá, puedo confirmar que existe un tabú en los usuarios acerca del uso del *software* libre, pero que bajo el acompañamiento del personal capacitado al usuario final es fácil la transición de *software* con propietario al *software* libre, también que brinda mayores ventajas en cuestiones no solo de desarrollo sino de inversión.

Acompañar a la población colombiana es el gran reto que tienen los informáticos e ingenieros de sistemas, pues el acompañar al usuario final en el uso e implementación de las herramientas de *software* libre abrirá nuevos horizontes en el avance tecnológico.

## REFERENCIAS

- [1] Celaya, C. y Martínez, S. (2007). Uso de *software* libre y de internet como herramientas de apoyo para el aprendizaje. *Revista Iberoamericana de Educación a Distancia*, 10(1), 83-100. Recuperado de <https://search.proquest.com/docview/1160531234?accountid=34687>
- [2] Vidal, M (2010, 8 de mayo). *Historia del software libre*. Recuperado de [https://gsync.urjc.es/~mvidal/docs/FLOSS\\_history.pdf](https://gsync.urjc.es/~mvidal/docs/FLOSS_history.pdf)
- [3] Gentegeek.com. (2006, 23 de marzo). *Software Libre y Software propietario, ventajas y desventajas*. <http://www.gentegeek.com/sl-sp-ventajas-desventajas/>