



# **La computación y sus versiones : El Debate del Software Software Libre vs. Software Privativo**

Pablo De Nápoli

[pdenapo@dm.uba.ar](mailto:pdenapo@dm.uba.ar)

# ¿Porqué debería interesarme...?

Un número creciente de nuestras actividades está controlada por las computadoras, incluyendo:

- Las comunicaciones interpersonales.
- Las transacciones comerciales y bancarias.
- Los trámites ante organismos del gobierno.

Las computadoras están controladas por el código (el software).

Pero ... ¿Quién controla el código?

¿De quién recibe órdenes tu computadora?

¿Tenemos derecho a saber exactamente qué hace nuestra computadora?



**¿Qué es el Software Libre?**

# ¿Qué es el Software Libre?

Según la definición propuesta por **Richard Stallman** un programa es software libre si el usuario tiene (de acuerdo a la **licencia** del programa) las siguientes libertades o derechos:

- 0) La libertad de **ejecutar el programa**, con cualquier propósito.
- 1) La libertad de **estudiar cómo funciona el programa**, y **adaptarlo** a sus necesidades.
- 2) La libertad de **distribuir copias**, con lo que puede ayudar a su vecino.
- 3) La libertad de mejorar el programa y **hacer públicas las mejoras** a los demás, de modo que toda la comunidad se beneficie.

# Una cuestión de Derechos

- El software libre es una **cuestión de derechos**, no una cuestión de tecnología.
- Lo que decide si un programa es o no libre, es la **licencia** bajo la cual el programa se distribuye.

Una licencia de software es un **contrato** entre el titular del copyright (derecho de copia) sobre un software, y el usuario, que establece que cosas el usuario puede hacer con el programa (y cuales no).

# Una cuestión ética

Según Stallman, las cuatro libertades se fundamentan en **razones éticas**.

Las dos primeras libertades: ejecutar el programa con cualquier propósito y estudiar como funciona el código y adaptarlo a tus necesidades son necesarias para que tengas el control de tu computadora.

Las otras dos libertades: posibilidad de distribuir copias y versiones modificadas, son la que permiten la **cooperación** entre los usuarios, permitiendo que formen una **comunidad**.

# Un Dilema Moral

“Si usas un programa sin la libertad de distribuir copias, pronto te enfrentarás a un dilema moral cuando un amigo te diga: Ese programa que tienes me sería útil. ¿Podrías darme una copia?” (Richard Stallman)

# Libre $\neq$ Gratuito

- El software que no es libre se denomina **software propietario** o **software privativo**.
- Software libre (free software)  $\neq$  software gratuito (freeware).

“El software libre es una cuestión de libertad, no de precio”.

Es posible vender software libre.

Existe software gratuito que no es libre.

- Libre  $\neq$  Software No comercial

Existe software libre desarrollado por empresas comerciales como IBM o Sun Microsystems.



# Acceso al Código Fuente

Para poder ejercer las libertades:

- 1) La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades.
- 3) La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

se requiere (es una condición necesaria) poder acceder al **código fuente** del programa.

El **código fuente** es el texto del programa en un lenguaje de programación comprensible para los humanos.



# Ejemplos de Software Libre

# El Proyecto GNU

El **proyecto GNU** fue iniciado en 1984 por Richard Stallman con la misión de construir un **sistema operativo** que fuera completamente libre. GNU significa “**GNU is not Unix**” (GNU no es Unix)

Un **sistema operativo** es el programa más importante de una computadora: sin él la computadora es absolutamente inútil.

Sus funciones incluyen: administrar los recursos del sistema tales como espacio en disco y memoria, aceptar órdenes del usuario y proporcionar un ambiente para que las aplicaciones puedan ejecutarse.

Por ejemplo son sistemas operativos **privativos**: MS-Windows, MS-Dos, OS/2 y Mac OS.

# GNU/Linux

El proyecto GNU tenía muchos programas útiles, pero le faltaba un **núcleo** para que fuera un sistema operativo completo y pudiera utilizarse.

Ese núcleo fue escrito por **Linus Torvalds**, con la cooperación de muchos otros a través de Internet. Se denomina **Linux**.

La combinación **GNU+Linux** constituye el sistema operativo libre más comúnmente utilizado (conocido más popularmente como **Linux** a secas).

# Otros Ejemplos de S.L.

- Los sistemas operativos **FreeBSD**, **OpenBSD** y **netBSD** derivados de la versión de Unix de la Universidad de California, Berkeley.
- El sistema de procesamiento de textos **T<sub>E</sub>X**, creado por Donald Knuth.
- La suite de oficina **Openoffice.org**, desarrollada originalmente por Sun Microsystems.
- El navegador web **Mozilla Firefox**.
- El web server **Apache** (que se utiliza en aproximadamente el 66 % de los sitios de la red).
- Lenguajes de programación: compilador **GCC** (C/C++/Fortran/Java/Pascal), Perl, Python, PHP.
- Bases de datos: Postgres, MySQL.



# La Filosofía del Software Libre

# Comunidad

En el modelo del Software Libre, el software tiene **autores**, pero no tiene **propietarios**. El código pertenece a la **comunidad**.

Nadie tiene el tiempo y los conocimientos técnicos para auditar todo el código o para hacer todas las modificaciones que quisiera.

El software libre propone hacer esto en forma comunitaria.

# ¿Cómo podemos colaborar?

- Quienes saben programar pueden hacer algunas modificaciones por sí mismos, y contribuir las a la comunidad.
- Quienes no saben programar, pueden contratar a cualquier programador, para que haga las modificaciones que necesitan.
- Todos podemos colaborar aportando el conocimiento, el "know-how" sobre como debería funcionar el programa, ayudando a encontrar errores y a mejorar la funcionalidad general del programa.
- También podemos colaborar escribiendo o traduciendo **documentación**.



# S.L. y Método Científico

El modelo de desarrollo del software libre es similar al **método científico**.

Al estar el código fuente disponible para todos, cualquier programador puede examinar que es lo que el programa hace, corregir sus fallos, controlar que no haya problemas de seguridad importantes, código malicioso, etc.; y expandir sus posibilidades

(**libre examen** por parte de los pares de la **comunidad científica**).

# S.L. y Acceso al Conocimiento

El **conocimiento** es **incremental**: cada programador de software libre, puede aprovechar el código desarrollado por otros. No es necesario reinventar la rueda cada vez.

El software libre es el único que nos permite **acceder al conocimiento** que hay detrás del software.

# El S.L. es Fiable

Como vimos, en el modelo del software libre, los usuarios pueden participar del desarrollo. Esto ayuda a que los errores se descubran y se corrijan más rápido y que el software sea más fiable. Esto fue observado en el clásico documento "La catedral y el Bazar" de Eric Raymond (1997). Existen trabajos científicos que procuran fundamentar esta afirmación, en el análisis de modelos matemáticos de ingeniería del software, y simulaciones numéricas.

Enrique A. Chaparro - "Aproximaciones a la Fiabilidad del Software Libre" (2003). Damien

Challet - Yann Le Du- "Microscopic Model of Software Bug Dynamics: Closed Source vs. Open Source". (2003)

# S.L. y Formatos de Archivos

- El software privativo suele usar formatos de archivos, cerrados o propietarios cuya especificación no está públicamente disponible.
- Esto crea incompatibilidades forzadas entre los sistemas, y dependencia de un proveedor; ya que dependemos del software de una determinada empresa para poder acceder a nuestros datos.
- En cambio, el software libre emplea formatos abiertos (cuya especificación está públicamente disponible y que cualquiera puede implementar). Esto garantiza la interoperatividad y la persistencia de los datos en el tiempo.

# La Licencia GPL de GNU

“ La mayor parte de las licencias de software están diseñadas para quitarte tu libertad de copiarlo y modificarlo. En cambio, la

**Licencia Pública General de GNU (GPL)**

intenta **garantizar tu libertad** de compartir y distribuir software libre, y asegurarse que el software sea libre para todos sus usuarios” [...]

“Protegemos tus derechos mediante dos pasos:

- Ponemos el software bajo **copyright**.
- Te **ofrecemos** esta licencia que te da permiso legal para copiar, distribuir y/o modificar el software.”

(traducción del preámbulo de la GPL)

# Copyleft

“[...], si distribuyes copias del programa (ya sea gratuitamente o por un precio), debes dar a todos los receptores **los mismos derechos que tu tienes**.

Debes asegurarte de que ellos también reciban o puedan obtener el **código fuente**.

Y debes mostrarle los términos de esta licencia, para que conozcan sus derechos.”

(traducción del preámbulo de la GPL)

# S.L. y Modelo de Negocios

- El software libre promueve un mercado informático basado en los **servicios asociados al software**, en vez de la venta de licencias de software.
- El software libre puede generar nuevas oportunidades de trabajo para los programadores. Con el software privativo, sólo las grandes corporaciones pueden programar. Con el software libre, **todos podemos programar**.
- Cualquier empresa puede ofrecer soporte de un programa libre.
- El software libre permite un mercado informático **no monopolístico** y abre nuevas oportunidades para las Pymes y empresas argentinas.

# ¿Qué son las patentes?

Las patentes consisten en un monopolio temporal sobre una invención susceptible de uso industrial, basado en un privilegio otorgado por el gobierno y derivado de su poder de coacción, a cambio de dar a conocer detalles públicamente de la invención.

Patentes y copyright (derecho de autor) son regímenes legales diferentes.



# El Peligro de las Patentes

- Si en un país se implementa un régimen de patentes de software, un programador puede ser demandado judicialmente por distribuir el código que él mismo escribió.
- Las patentes de Software son una constante amenaza al desarrollo del software (tanto libre como propietario). No promueven el desarrollo sino que lo imposibilitan.
- En Argentina, afortunadamente el software no es patentable (por ahora!) (artículo 6, inciso c), ley 24.481).

# Referencias

- Proyecto GNU - Free Software Foundation  
<http://www.gnu.org>
- Mi Página personal  
<http://mate.dm.uba.ar/~pdenapo/linux.html>
- Asociación Civil SoLAr - Software Libre Argentina  
<http://www.solar.org.ar>