

# Master Universitario Oficial en Sistemas Telemáticos e Informáticos (URJC)

Jesús M. González Barahona y Gegorio Robles

jgb@gsyc.es grex@gsyc.es  
GSyC/LibreSoft, Universidad Rey Juan Carlos

Febrero 2010



©2001-2010 Jesús M. González Barahona, Gregorio Robles,  
Joaquín Seoane

Algunos derechos reservados.  
Este artículo se distribuye bajo la licencia  
“Reconocimiento-CompartirIgual 3.0 España” de Creative  
Commons, disponible en  
<http://creativecommons.org/licenses/by-sa/3.0/es/deed.es>

Este documento (o uno muy similar)  
está disponible en (o enlazado desde)  
<http://gsync.es/moodle>

# Breve historia del software libre

## Los años 1950, 1960

- Relación con el software muy parecida a la que hoy tenemos con el software libre
- Software incluido en el catálogo del fabricante, “viene con el hardware”
- Grupos de usuarios comparten software, y lo mejoran
- Código fuente a disposición de quien lo pide
- En general, no se considera el software como un elemento independiente
- Todo cambia con el “unbundling” de hardware, software y servicios de IBM (1969)

## Los años 1970, los primeros 1980

- 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” (ej: costumbre en la comunidad matemática)
  - A veces, motivos “prácticos” (ej: difusión de una nueva tecnología)
- Richard Stallman, GNU, FSF.
  - Fundamentos “legales” (GPL) y filosóficos.
  - Infraestructura básica: editor (Emacs), compilador (GCC), depurador (GDB), etc.
  - Meta: construir un sistema completo alternativo a Unix.
  - Trabajo muy estructurado y con metas claras.

## Los años 1970, los primeros 1980 (2)

- CSRG de Berkeley:
  - Importancia de compartir fuentes (cultura Unix “original”).
  - Coartado por la licencia ATT (poco desde el punto de vista práctico, todos la tenían).
  - Énfasis en el sistema operativo (kernel, utilidades, etc.).
  - Utilizado por mucho software propietario (SunOS, Ultrix, etc.)
- Primera Internet:
  - Implementaciones de referencia, disponibles para todos.
  - La Red como herramienta de cooperación (News, ftp, correo-e).
  - La comunidad de usuarios proporciona el mejor soporte.

## Final de los 1980, primeros 1990

- Entornos completos sobre Unix (SunOS, Solaris, etc.):
  - Muchas aplicaciones son las mejores en su campo (utilidades Unix, compiladores, etc.).
  - Especialmente interesante: XWindow.
  - Sólo falta el kernel...
- 386BSD, NetBSD, FreeBSD, OpenBSD:
  - Bill Jolitz hace lo que le faltaba al kernel.
  - Rápidamente: sistemas completos, similares a SunOS en funcionalidad.
  - Licencia BSD (puede usarse para distribución como software propietario).

## Final de los 1980, primeros 1990 (2)

- GNU/Linux:
  - Linus Torvalds hace su “Minix libre”.
  - Cientos de desarrolladores se vuelcan sobre él, e integran con él todo el software de GNU.
  - Adaptaciones de aplicaciones, nuevas aplicaciones, etc.
  - Casi todo bajo GPL: sólo puede redistribuirse con fuentes.
  - Un kernel, muchas distribuciones (Slackware, Debian, RedHat, Suse, etc.).
  - Gran éxito “popular”.

## Finales de los 1990

- Anuncio de Netscape.
- GNU/Linux y FreeBSD compiten con Windows NT.
- Más y más cerca del usuario “normal”: KDE, GNOME.
- GNU/Linux penetra en Universidades (y en casa de los estudiantes).
- La mejor opción es libre en muchos ámbitos (Apache, infraestructura de Internet, XFree, GCC, Gnat).
- Empresas como RedHat consiguen capital-riesgo.
- La prensa comienza a atender al software libre.
- Grandes empresas (Corel, Apple, IBM) estudian cómo tratar con el software libre.

## Principios de los 2000

- El software libre empieza a estar listo para el escritorio (GNOME 2.x, KDE 3.x, OpenOffice), y es simple de instalar por el usuario final
- El software libre se incorpora a la estrategia de grandes empresas (IBM, HP, Sun)
- Otras (como Microsoft) prefieren una estrategia de enfrentamiento parcial
- Dificultades financieras como resultado de la crisis de las puntocom
- Comienza la penetración en administraciones públicas y grandes empresas
- Aumento grande del número de desarrolladores, de la cantidad de software libre disponible, etc.

## Principios de los 2000 (2)

- Nuevas disciplinas estudian el software libre: comenzamos, poco a poco, a entender cómo funciona
- Comienzan a verse efectos de la “deslocalización” del desarrollo de software libre: países “periféricos” hacen cosas interesantes.
- Ciertos mercados, ciertos sectores, ya consideran al software libre como una opción “natural”
- El entorno legal está cambiando: ¿se convertirá en hostil para el software libre?

## Finales de los 2000

- Software libre estratégico para muchas empresas (ej: Google)
- Conjuntos de aplicaciones muy completos para muchos entornos
- Empresas probando nuevos modelos de colaboración (ej: ObjectWeb, Morfeo)
- Software libre como propuesta para dominar mercados (ej: Android, Symbian, Maemo en móviles)
- Nuevos modelos de negocio, modelos para nuevos negocios
- Software libre como posibilitador en muchas industrias
- Software libre parte del análisis de competencia en sectores (ej: MySQL en la compra de Sun por Oracle)
- El software libre se va convirtiendo en algo “normal”

## El futuro: ¿una carrera de obstáculos?

La evolución futura del software libre puede encontrarse varios obstáculos:

- Técnicas FUD (miedo, desconocimiento, duda): hasta ahora han mostrado no ser muy problemáticas.
- “Disolución” (sistemas que pueden confundirse con el software libre): división de la comunidad, pérdida de las ventajas del modelo.
- Desconocimiento (pérdida de visión): ¿por qué es interesante el software libre?
- Impedimentos legales: patentes de software, mecanismos de control de acceso a la información, etc.
- ¿Cuánto de sostenible es el desarrollo de software libre?

# ¿Cómo serán las cosas dentro de 10 años?