

Software Libre, Fundamentos, Licencias y Mitos

Jon Ander Hernández



`EHUko Software Librea sustapenerako taldea`

14 de Mayo de 2009



1 Introducción

- ¿Qué es el Software Libre?
- ¿Qué es el Open Source?
- Conclusiones

2 Licencias

- GNU GPL
 - GNU GPL 3
 - Medidas contra las Patentes de Software
 - Medidas contra la TiVolización
- Licencias Permisivas BSD/MIT/X11
 - Compatibilidad con la GPL
 - Licencias permisivas contra licencias Copyleft
- Creative Commons
 - Escenarios de las Creative Commons
 - Tabla de derechos de las Creative Commons
 - Ejemplo
 - Información y términos legales desde la web
- Licenciamiento múltiple
 - Problemas de los licenciamientos múltiples
- Compatibilidad

3 Escenarios interesantes

- Linux
- Webkit
- Solaris
- BeagleBoard

4 Referencias



Software Libre



FREE SOFTWARE
F O U N D A T I O N



¿Qué es el Software Libre?

Es software Libre¹ es un movimiento *ético, social y político*, no *técnico*.

Cuyo fin es el promover la libertad de los usuarios.

La filosofía del software libre, hereda muchos de los elementos de la llamada “*cultura hacker*” de los años 70.

El movimiento nació a manos de Richard M. Stallman² a principios de los años 80 *a causa de una impresora* mientras trabajaba en el “MIT Artificial Intelligence Laboratory”

¹http://en.wikipedia.org/wiki/Free_software

²http://en.wikipedia.org/wiki/Richard_Stallman



Y todo comenzó con una impresora...

En la era de los años 70 Stallman y el resto sus compañeros del MIT trabajan en un entorno acostumbrados a :

- ▶ A compartir el código de sus programas con el resto del equipo y universidades.
- ▶ A aprender del trabajo de los demás.
- ▶ A debatir e intercambiar soluciones abiertamente.
- ▶ A devolver nuevamente las mejoras realizadas a los programas de los demás.

Estaban acostumbrados a trabajar como una *comunidad*.



Un buen día, compraron una nueva impresora, muy buena, muy rápida... pero al igual que la antigua *frecuentemente se atascaba*.

¡Y ojo! La impresora estaba en otro piso! ;-)

Eran programadores, no podían mejorar el mecanismo.
Pero habían *mejorado el software* de la antigua impresora :

- ▶ Cuando se imprimía correctamente un documento → enviaba un aviso al usuario para que pasara a recogerlo.
- ▶ Cuando se atascaba → enviaba un aviso a todos los usuarios con documentos pendientes.

No resolvía el problema de los atascos, pero resolvía parte del problema.

Stallman se puso en contacto con Xerox para obtener el código del driver, y por bajo su sorpresa se lo negaron.

(Ahora pensemos en interminables viajes arriba y abajo, descubriendo frecuentemente que la impresora llevaba tiempo atascada y con una enorme cola de trabajos)

“Se sintió prisionero de su software, y que estaban limitados a soportar el problema sin poder hacer nada...”



2 Años más tarde descubrió que una persona en la universidad de Carnegie-Mellon disponía del código fuente. Cuando fue a pedírsela descubrió que había firmado un NDA³, el cual supuso su primer contacto con un NDA.

Se quedó lleno de enojo y pensativo, y concluyó que los *NDA*s tienen “victimas”.

“Había hecho un compromiso de no cooperar con sus colegas del MIT.”

“La misión de la ciencia, y la tecnología, es crear herramientas útiles para los humanos. Si la guardamos como secretos, en vez de publicarla, la estamos traicionando. No es ético, debemos trabajar hacia ampliar y compartir el conocimiento humano”.

³un acuerdo de no divulgación

El golpe definitivo vino cuando el P10 sobre el que trabajaban se declaró obsoleto, junto con el sistema operativo *libre* que ellos mismo habían escrito.

No pudo continuar su trabajo como antes, *el mundo cambiaba* y había dejado de comportarse como una *comunidad*, con herramientas libres. Todos los sistemas operativos para ordenadores modernos eran *privativos* y eran obligados a firmar NDAs. Y no podía continuar su trabajo en el MIT sin traicionar sus principios.

Se planteó 2 opciones :

- ▶ Aceptar el cambio y adaptarse al mundo, y firmar NDAs y programas privativos. Y traicionar sus principios.
- ▶ Abandonar el MIT.

Y así nació el proyecto **GNU**⁴ :

Gnu It's Not Unix!

Un sistema operativo 100% libre.



⁴<http://en.wikipedia.org/wiki/GNU>

Las 4 libertades

RMS concluyó que para crear una comunidad libre, el software debería cumplir 4 libertades esenciales⁵ :

- ▶ Libertad 0: libertad de usar el programa, con cualquier propósito.
- ▶ Libertad 1, libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades. *Libertad para ayudarse a uno mismo.*
- ▶ Libertad 2, libertad de distribuir copias, con lo que puedes ayudar a los demás. *Libertad para ayudar al prójimo.*
- ▶ Libertad 3, libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. *Libertad para ayudar a tu comunidad.*

⁵http://en.wikipedia.org/wiki/The_Free_Software_Definition

Licencias

- ▶ Vaya lio...entonces...
 - ▶ ¿Puedo copiar software libre tal cual sin restricciones?
 - ▶ ¿Puedo *vender* software libre? ¿o siempre tiene que ser gratis?
 - ▶ ¿Qué hay de la GPL o eso?
 - ▶ Tengo un programa que viene con su código ¿Es por tanto libre?



¿El software libre es siempre gratis?

¿El software libre es siempre gratis?

- ▶ La gratuidad no está recogida en ninguna de las libertades
- ▶ Yo puedo venderte por el precio que yo quiera un software
 - ▶ En el momento en el que tú lo compras, por la libertad 2 podrás distribuirlo

“Free as in free speech, not as in free beer”



Patentes de Software⁶

Free software projects cannot agree to patent licences that include any kind of *per-copy fee*, adding any *requirements to pay* or to *notify* someone each time a copy is made.

- ▶ Patent Pools

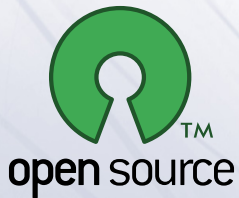
In 2005, IBM, Novell, Philips, Red Hat, and Sony founded the Open Invention Network (OIN). OIN is a company that acquires patents and offers them royalty free "to any company, institution or individual that agrees not to assert its patents against the Linux operating system or certain Linux-related applications"

- ▶ Promises from patent holders

Some software companies who hold significant patent portfolios have made non-aggression pledges to the free software community. These have varied in scope and have received a variety of responses. IBM,[17] Sun, and Nokia[18] are three examples. These have been described by Richard Stallman as "significant", "not really anything", and "next to nothing", respectively.

⁶http://en.wikipedia.org/wiki/Software_patents_and_free_software

Open Source



¿Qué es el Open Source⁷?

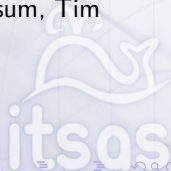
El término *Open Source Software* nace de la mano de la *Open Source Initiative* fundada por Jon "maddog" Hall, Bruce Perens y Eric S. Raymond.

"Dump the moralizing and confrontational attitude that had been associated with 'free software' in the past and sell the idea strictly on the same pragmatic, business-case grounds."

Advocating free software exclusively on the grounds of technical superiority (a claim previously made by Raymond in his essay *The Cathedral and the Bazaar*).

Gente afín al OSS : Linus Torvalds, Larry Wall, Guido van Rossum, Tim O'Reilly...

⁷http://en.wikipedia.org/wiki/Open_source_software



The Cathedral and the Bazaar⁸

Open source evangelist Eric S. Raymond suggests a model for developing OSS known as the Bazaar model.

Traditional development methodologies to building a cathedral :

- ▶ “carefully crafted by individual wizards or small bands of mages working in splendid isolation.”
- ▶ Development takes place in a centralized way.
- ▶ Roles are clearly defined. Roles include people dedicated to designing (the architects), people responsible for managing the project, and people responsible for implementation.



⁸http://en.wikipedia.org/wiki/The_Cathedral_and_the_Bazaar

All software should be developed using the bazaar style, which he described as "a great babbling bazaar of differing agendas and approaches."

- ▶ Roles are not clearly defined
- ▶ Users should be treated as co-developers
- ▶ Early releases
- ▶ Frequent integration
- ▶ Several versions
- ▶ High modularization
- ▶ Dynamic decision making structure

OSI's Open Source Definition⁹

- ▶ Free Redistribution
- ▶ Source Code
- ▶ Derived Works
- ▶ Integrity of The Author's Source Code
- ▶ No Discrimination Against Persons or Groups
- ▶ No Discrimination Against Fields of Endeavor
- ▶ Distribution of License
- ▶ License Must Not Be Specific to a Product
- ▶ License Must Not Restrict Other Software
- ▶ License Must Be Technology-Neutral

⁹http://en.wikipedia.org/wiki/Open_Source_Definition

Conclusiones

Free Software vs Open Source

“We disagree on the basic principles, but agree more or less on the practical recommendations. So we can and do work together on many specific projects.”

El Software Free Open Source (FLOSS) es la sinergia de ambas filosofías, y es parte de la razón por la que es un modelo

A efectos prácticos tan solo existe **una** licencia que apruebe la OSI, que la FSF no apruebe.

La APSL 1.0 (Apple Public Source Licence), fue rechazada por FSF, posteriormente fue revisada y la APSL 2.0 fue finalmente aprobada por la FSF.

http://en.wikipedia.org/wiki/Comparison_of_free_software_licences



Introducción

Una de las primeras licencias libres registrada :

```
; *****  
;  
; TINY BASIC FOR INTEL 8080  
; VERSION 2.0  
; BY LI-CHEN WANG  
; MODIFIED AND TRANSLATED  
; TO INTEL MNEMONICS  
; BY ROGER RAUSKOLB  
; 10 OCTOBER, 1976  
; @COPYLEFT  
; ALL WRONGS RESERVED  
;  
; *****
```

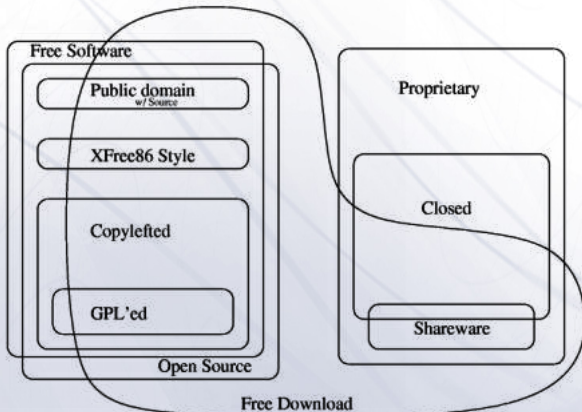
Principales licencias libres :

- ▶ GNU General Public License
- ▶ GNU Lesser General Public License
- ▶ BSD License
- ▶ Mozilla Public License
- ▶ MIT License
- ▶ Apache License

Licencias víricas y no víricas

- ▶ Una licencia vírica:
 - ▶ Exige que cuando alguien la distribuya, la distribuya bajo los mismo términos
 - ▶ Yo puedo copiarte, pero automáticamente todo mi código estará licenciado bajo la misma licencia, por lo que tú podrás copiarme
 - ▶ Ejemplo: GNU GPL
- ▶ Una licencia no vírica:
 - ▶ No tiene estas exigencias
 - ▶ Yo puedo copiarte, mejorar el código y distribuirlo bajo la licencia que yo quiera, y sólo tengo que poner tu nombre.
 - ▶ Ejemplo: BSD

Mapa de categorías de licencias



GNU GPL¹¹

Free as in Freedom

La licencia Copyleft usada por excelencia, y que se ciñe a la definición de Free Software

GPL (Hard Copyleft) → Todo lo que enlace al programa GPL debe ser GPL.

LGPL (Weak Copyleft)¹⁰ → Todo lo que enlace al programa LGPL debe ser LGPL excepto cuando enlazamos dinámicamente.

A key dispute related to the GPL is whether or not non-GPL software can dynamically link to GPL libraries. The GPL is clear in requiring that all derivative works of code under the GPL must themselves be under the GPL. However, it is not clear whether an executable that dynamically links to a GPL code should be considered a derivative work (see Weak Copyleft).

¹⁰http://en.wikipedia.org/wiki/Copyleft#Strong_and_weak_copyleft

¹¹http://en.wikipedia.org/wiki/GNU_GPL

GNU GPL3¹²

La versión 3.0 de la GPL ha sido fuente de bastantes controversias, ya que extiende el número de restricciones abordando problemas como :

- ▶ Las Patentes de Software
- ▶ Trusted Computing y “TiVolización”

Una enorme cantidad de proyectos han anunciado su intención de *no* actualizar a la versión 3.0 de la licencia, incluido el propio proyecto Linux.



¹²http://en.wikipedia.org/wiki/GNU_GPL#Version_3

Medidas contra las Patentes de Software

Medida ante posibles juicios por patentes :

“Create a penalty so as to *discourage the licensee*¹³ from *suing the licensor*¹⁴ for patent infringement by *terminating the licence* upon the initiation of such a lawsuit.”

¹³the user/recipient of the software

¹⁴the provider/author of the software

Medidas contra las Patentes de Software 2

Medida para aprovechar acuerdos como Novell-Microsoft para tratar de extender el acuerdo al resto de usuarios :

“You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license.”

This aims to make future such deals ineffective. The license is also meant to cause Microsoft to extend the patent licenses it grants to Novell customers for the use of GPLv3 software to all users of that GPLv3 software; this is possible only if Microsoft is legally “conveyor” of the GPLv3 software.

Tivoization¹⁵

Is the creation of a system that incorporates software under the terms of a copyleft software license, but uses hardware to prevent users from running modified versions of the software on that hardware.

- ▶ Stallman believes TiVo circumvented this goal by making their products run programs only if the program's digital signature matches those authorized by the manufacturer of the TiVo. So while TiVo has complied with the GPL v2 requirement to release the source code for others to modify, any modified software will not run on TiVo's hardware.
- ▶ Linus Torvalds, the author of the Linux kernel, has argued that it is appropriate for TiVo to use digital signatures to limit what software may run on the systems that they sell. Torvalds has stated that he believes the use of private digital signatures on software is a beneficial security tool. Torvalds also believes that software licenses should attempt to control only software, not the hardware on which it runs.

The term came about in reference to TiVo's use of GNU GPL licensed software on the TiVo brand digital video recorders (DVR)

¹⁵<http://en.wikipedia.org/wiki/Tivoization>

Licencia BSD/MIT/X11



FreeBSD®

La mayor parte de las licencias permisivas siguen este esquema¹⁶:

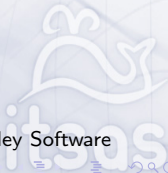
Unlimited redistribution for any purpose as long as its copyright notices and the license's disclaimers of warranty are maintained.

- ▶ Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- ▶ Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- ▶ Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

http://en.wikipedia.org/wiki/Permissive_free_software_licence

http://en.wikipedia.org/wiki/BSD_licenses

¹⁶El nombre deriva del sistema Unix de la universidad de Berkeley, "Berkeley Software Distribution" (BSD)



Compatibilidad con la GPL

Actualmente la mayor parte del software con licencia BSD, usa una versión *sin la 4^o cláusula* original que la hace totalmente compatible con la GPL.

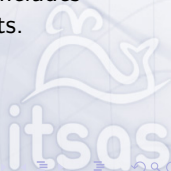
Some permissive free software licences contain a 4th clause that require advertising materials to credit the copyright holder.

Licences with an advertising clause include the 4-clause BSD license, the PHP License, and the OpenSSL License. These licences, although they are permissive free software licences, are incompatible with the widely used GNU General Public License.



Licencias permisivas contra licencias CopyLeft

- ▶ The GPL requires any derivative work that is released to be released according to the GPL while the BSD license does not.
- ▶ BSD code can find its way into proprietary software that only acknowledges the authors. For instance, the IP stack in Microsoft Windows and Mac OS X are derived from BSD-licensed software.
- ▶ Supporters of the BSD license argue that it is more free than the GPL because it grants the right to do anything with the source code, second only to software in the public domain. This includes incorporating the BSD-licensed code in proprietary products.



Creative Commons¹⁷



Licencias enfocadas a otras cuestiones que no sean Software.

Orientadas a gente que desea un tipo de licencia que les permita compartir reservando algunos derechos.



¹⁷en.wikipedia.org/wiki/Creative_Commons_licenses

Escenarios de las Creative Commons

- ▶ Un músico desea que se distribuya libremente su obra, pero que nadie la altere, ni se lucre económicamente de ella.
- ▶ Un fotógrafo desea compartir un enorme book de buenas fotos que descansan en su cajón, desea que se le atribuya su autoría, y no desea que nadie se lucre con ellas.
- ▶ Un escritor desea poner a disposición de todo el mundo un libro suyo actualmente descatalogado, pero no desea que nadie plagie contenido del mismo.

Para cubrir todas estas necesidades distintas, las Creative Commons proponen un sistema donde el autor decida que derechos desea preservar o revertir, y en función de estos obtenga una licencia.



	Attribution (<i>by</i>)	Licensees may copy, distribute, display and perform the work and make derivative works based on it only if they give the author or licensor the credits in the manner specified by these.
	NonCommercial (<i>nc</i>)	Licensees may copy, distribute, display, and perform the work and make derivative works based on it only for noncommercial purposes.
	No Derivative Works (<i>nd</i>)	Licensees may copy, distribute, display and perform only verbatim copies of the work, not derivative works based on it.
	ShareAlike (<i>sa</i>)	Licensees may distribute derivative works only under a license identical to the license that governs the original work.

Ejemplo

Creative Commons - By - NonCommercial - NonDerivative

La licencia limita el uso de la obra de modo que :

- ▶ Se mantenga una atribución al autor.
- ▶ No sea usada con fines comerciales.
- ▶ La obra se redistribuya de forma íntegra.

La licencia permite que :

- ▶ La obra se distribuya con otra licencia que mantenga las anteriores restricciones.

Combinaciones posibles :

1. Attribution alone (*by*)
2. Attribution + Noncommercial (*by-nc*)
3. Attribution + NoDerivs (*by-nd*)
4. Attribution + ShareAlike (*by-sa*)
5. Attribution + Noncommercial + NoDerivs (*by-nc-nd*)
6. Attribution + Noncommercial + ShareAlike (*by-nc-sa*)

Información y términos legales accesibles para cada licencia desde su web:

- ▶ Explicación intuitiva de las posibilidades de la licencia cc-by-sa

<http://creativecommons.org/licenses/by-sa/3.0/>

- ▶ Código legal de la licencia

<http://creativecommons.org/licenses/by-sa/3.0/legalcode>

Licenciamiento Dual

Otra práctica habitual suele el uso de varias licencias sobre un mismo proyecto.

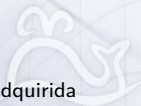
Ejemplo, Trolltech y su librería QT :

Licenciada como GPL y privativa¹⁸ :

- ▶ Si alguien desea usar la librería para un proyecto libre, puede usar la versión licencia GPL, la cual obliga a que el proyecto se distribuya a su vez con la GPL (licencia vérica).
- ▶ Si una compañía desea usar la librería para un software propietario¹⁹ deberá comprar una licencia tradicional a Trolltech.

¹⁸Esta ha sido su licencia y su modelo de negocio hasta que Trolltech fue adquirida por Nokia, y relicenciada a LGPL, lo que ahora permite ser usada en proyectos privativos

¹⁹Como por ejemplo Google Earth



Problemas de los licenciamientos múltiple

Los proyectos con estas licencias, exigen que los autores de parches distribuyan sus parches bajo todas las licencias oficiales del proyecto.

Cambios de Licencia:

La elección de una licencia correcta es importante, entre otras razones por la complejidad de un proceso de relicenciamiento.

Este proceso supondría la consulta a todos los autores, de todas las modificaciones o contribuciones realizadas al proyecto.



Casos famosos han sido :

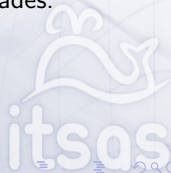
- ▶ Relicenciamiento de Mozilla de MPL/NPL a GPL/LGPL/MPL, supuso un complejo proceso que tardó más de 2 años en completarse, y supuso la eliminación y reescritura de todas aquellas contribuciones de aquellos autores con los que no consiguieron ponerse en contacto.
- ▶ Relicenciamiento de la Wikipedia de la GFDL²⁰ a una licencia más universal y copyleft como la cc-by-sa, finalmente la solución ha venido de una modificación de la GFDL para permitir que todos los contenidos licenciados con esa licencia, puedan ser también distribuidos bajo la cc-by-sa.

Conclusión

Existen *demasiadas* licencias libres en este mundo, y es costoso comprobar :

- ▶ Su compatibilidad con el resto de licencias.
- ▶ Si judicialmente tiene *validez*
 - ▶ Licencias como la GPL han sido extensamente probadas en casos judiciales con resultados favorables.

Lo óptimo es usar una existente en función de nuestras necesidades.



Escenarios Interesantes

Reflexión sobre algunos escenarios exitosos de software libre, y las claves de su éxito.



Linux

Variantes imposibles sin la libertad :

El sistema más portable existente. Incluso a plataformas bastante dispares :

- ▶ Grandes superordenadores.
- ▶ Sistemas de tiempo real.
- ▶ Sistemas empotrados uClinux.
- ▶ Consolas como la PlayStation 3, con los procesadores CELL.



Comunidad

Congresos :

Otawa Linux Simposium

<http://www.linuxsymposium.org>

Fosdem

<http://fosdem.org>

CE Linux Forum

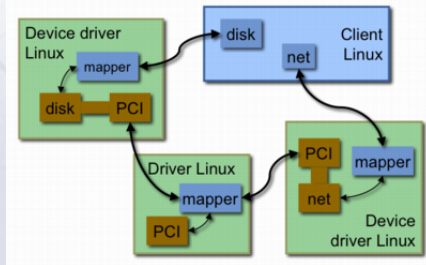
Embedded Linux Conference



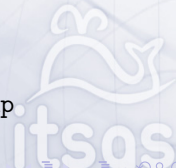
Linux sobre uKernel L4

Ejemplo interesante :

Driver Isolation Architecture



<http://l4ka.org/projects/virtualization/drivers.php>



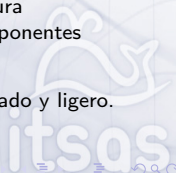
Webkit

Apple después del lanzamiento de Mac Os X estaba dispuesta a conquistar el sector de Escritorio, pero tenía una gran dependencia de Microsoft :

El Microsoft Internet Explorer 5.5 el cual llevaba tiempo abandonado por Microsoft, convirtiéndose en un dolor para usuarios y desarrolladores Web.

Solución romper la dependencia y desarrollar un navegador propio, opciones :

- ▶ Comprar Opera → Probablemente no hubiera sido una compra barata.
- ▶ Usar el motor Gecko de Mozilla (GPL/MPL/NPL) → Una arquitectura demasiado cargada (múltiples capas de portabilidad, sistema de componentes propio, sistema de widgets propio...).
- ▶ KHTML (LGPL) → En estado prematuro, pero funcional, bien diseñado y ligero.



Primeramente :

- ▶ Apple contrato a 3 o 4 desarrolladores de Mozilla para trabajar en KHTML.
- ▶ El desarrollo se realizó cerrado de puertas para adentro.
- ▶ Cuando salió el producto, liberaron un tar.gz con todo el fuente.
- ▶ Problemas :
 - ▶ Difícil de reintegrar de nuevo en KHTML.
 - ▶ Cumplía la licencia, pero no revertía sobre la comunidad.
 - ▶ Derroche de energías corrigiendo bugs por duplicado.
 - ▶ Aún había intercambio de código entre dos proyectos cada vez más diferentes.

En una segunda fase :

- ▶ Abrieron su desarrollo, abrieron listas de correo para las discusiones técnicas.
- ▶ Abrieron una web `http://webkit.org`, con un wiki, blog, etc...
- ▶ Abrieron un subversion público.
- ▶ Problemas :
 - ▶ Seguía siendo difícil para KHTML, y otros dos nuevos ports (Nokia gdkwebkit y Nokia S60 Browser) integrar los cambios del proyecto.
 - ▶ Era complicado seguir el estado de los bugs y la revisión de los parches.



En una tercera fase :

- ▶ Abrieron el bugzilla para gestionar los bugs y los parches.
- ▶ Comenzaron a conceder accesos al subversión.
- ▶ Nombraron supervisores a desarrolladores externos a Apple.

Y se convirtieron en el mejor Navegador de la actualidad.



Navegadores basados en Webkit :

- ▶ Apple Safari (Mac Os X, iPhone).
- ▶ Google Chrome.
- ▶ Adobe Air.
- ▶ Nokia Symbian S60 Browser
- ▶ QWebkit → empotrable en aplicaciones QT
- ▶ GtkWebkit → empotrable en aplicaciones GTK+
- ▶ wxWebkit → empotrable en aplicaciones wxWidgets
- ▶ Clutter Webkit → empotrable en aplicaciones OpenGL-ES con Clutter
- ▶ EFLWebkit (Enlightenment)

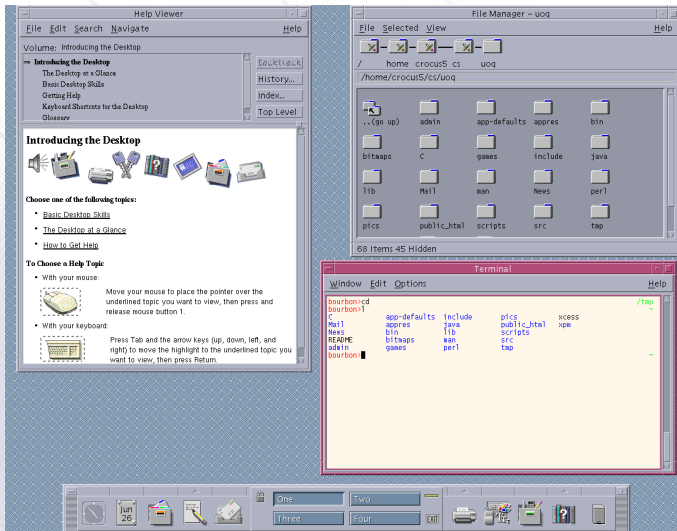
Auténtica revolución técnica :

- ▶ Interpretes JavaScript V8 (Google) y SquirrelFish (Apple).
- ▶ Interprete de JavaScript 2.0 (donado por Adobe).



Solaris

Sun requería un “Desktop” moderno para sustituir al viejo CDE²¹.



²¹Common Desktop Environment

Y tenían un fuerte requisito :

Sun's accessibility strategy for the Solaris OS is designed to meet the requirements of Section 508 of the U.S. Rehabilitation Act. According to this statute, federal agencies are required to acquire, information and/or electronic technology, documentation, support, and services that allow comparable access for employees and members of the public with disabilities.

Solución :

Usar el escritorio libre GNOME.

Beneficios :

- ▶ GNOME 2.0 era un escritorio completo, moderno → No era necesaria la inversión en implementar un nuevo Escritorio.
- ▶ GNOME contaba con un buen mantenimiento → Un gran ahorro en costes de mantenimiento.



Pero GNOME carecía de un framework para accesibilidad, así que Sun contribuyó a esa parte y obtuvo un Escritorio completo y accesible.

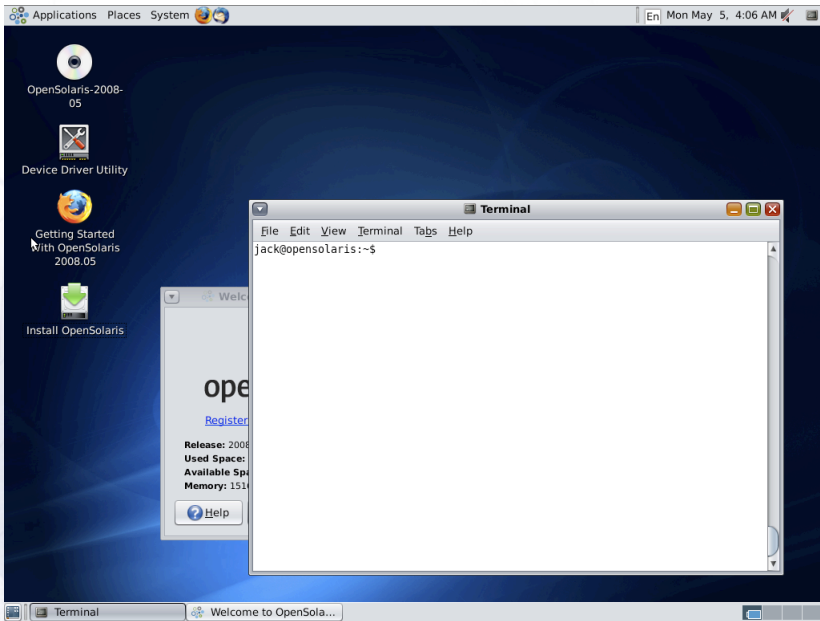
“Deliver the GNOME 2.0 Desktop with built-in accessibility solutions as the next user environment for the Solaris OE. GNOME 2.0 provides an accessible desktop, applications, utilities, and accessories, as well as assistive technologies that adapt to the needs of users with disabilities.”

- ▶ ATK (Accessibility Toolkit Application Programming Interface)
- ▶ AT-SPI (Assistive Technology Service Provider Interface)

http:

[//www.sun.com/software/star/gnome/accessibility/generalfaq.xml](http://www.sun.com/software/star/gnome/accessibility/generalfaq.xml)





BeagleBoard

Proyecto de Hardware libre²² de la mano de Texas Instrument.

Laptop-like performance

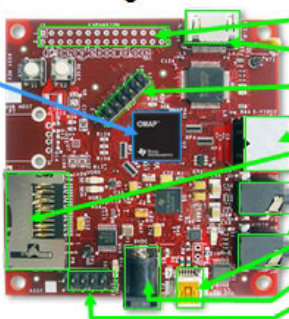
TI OMAP3530

- 600 MHz superscaler ARM[®] Cortex™-A8
- More than 1200 Dhrystone MIPS
- Up to 10 Million polygons per sec graphics
- HD video capable C64x+™ DSP core

Memory

- 128MB LPDDR RAM
- 256MB NAND flash

3"



Flexible expansion

- I²C, I²S, SPI, MMC/SD
- DVI-D
- JTAG
- S-Video
- SD/MMC+
- Stereo Out
- Stereo In
- USB 2.0 HS OTG
- Alternate Power
- RS-232 Serial

²²<http://beagleboard.org/hardware/design>

²³La revisión C2 actual posee una memoria de 256MB LPDDR RAM

Capacidades del dispositivo :

- ▶ 'HD capable' TMS320C64x+ core (75MHz upto 720p @30fps)
- ▶ Imagination Technologies PowerVR SGX 2D/3D graphics processor supporting dual independent displays
- ▶ Rendimiento equivalente a un Intel Atom N280 "downgradeado" a 800mhz, con 2W/ de consumo.
- ▶ Placa de desarrollo por \$149

Motivación de TI :

- ▶ Potenciar el sector de los sistemas empotrados.
- ▶ El OMAP 3530 es un buen candidato para Internet Tablets con Maemo²⁴ o Android, ambos basados en Linux.
- ▶ Tal vez esten comprobando hasta donde puede llegar el modelo FLOSS

²⁴Sistema Operativo de Nokia para sus Internet Tablets

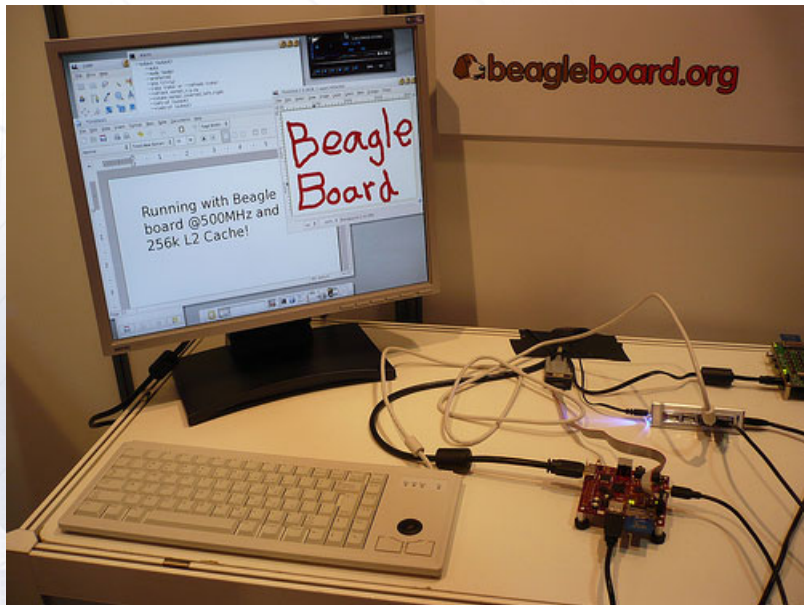
Respuesta del mundo FLOSS a TI

Sistemas operativos disponibles para BeagleBoard :

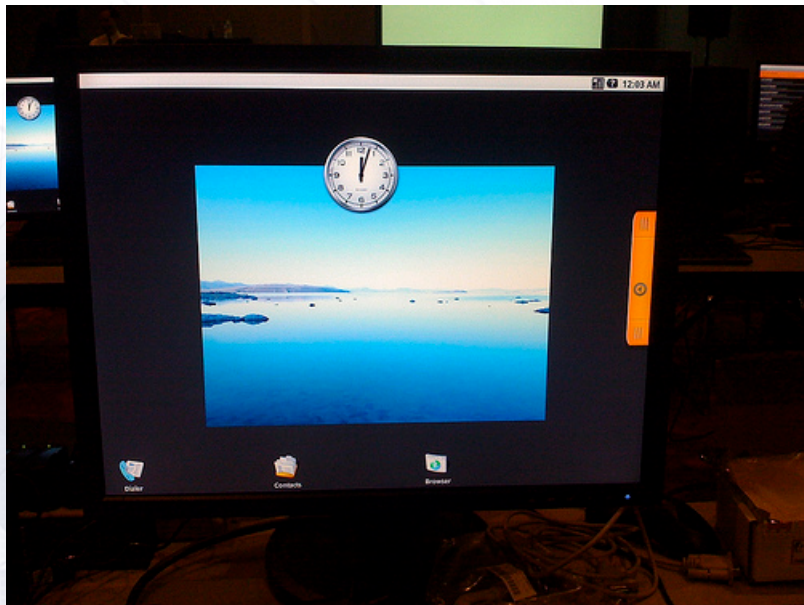
- ▶ Ångström Distribution
- ▶ Ubuntu on the BeagleBoard
- ▶ Android for BeagleBoard
- ▶ QNX Neutrino
- ▶ Montavista
- ▶ Maemo 5
- ▶ Mer



Ångström Distribution

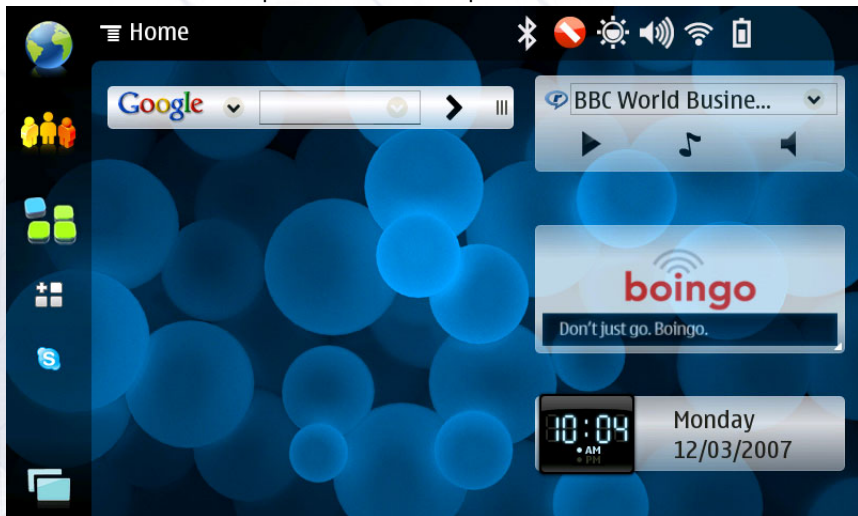


Google Android



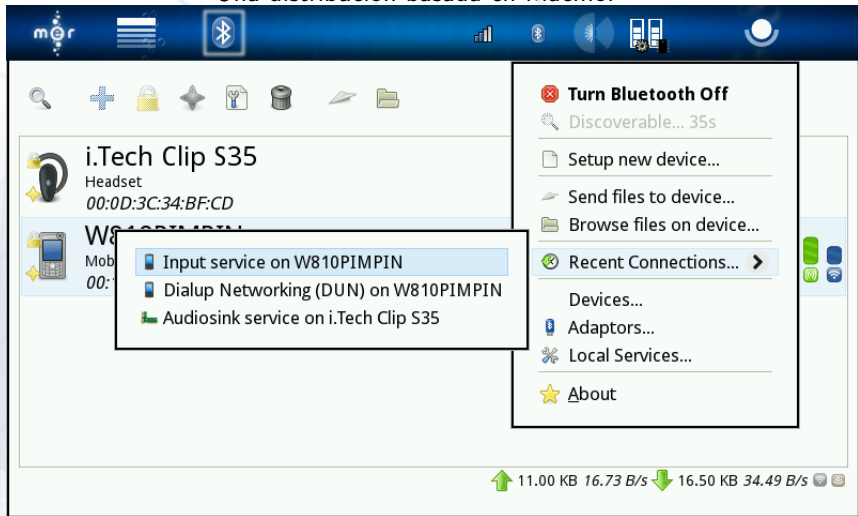
Maemo

El Sistema operativo de Nokia para Internet Tablets.



Mer

Una distribución basada en Maemo.



¿Preguntas?



¡Gracias a todos por venir! ;-)

Referencias

- ▶ | ♥ <http://en.wikipedia.org> 0:-)



- ▶ “Introducción al Software Libre”

por Arkaitz Jimenez, Pablo Orduña y Álvaro Uría.

<http://nctrun.e-ghost.net/#softlibreMarzo2006>

