

# El software libre en el sector público

David Megías (coordinador)  
Francesc Rambla i Marigot  
Antonio Rodil Garrido

PID\_00142691

Material docente de la UOC



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)

**David Megías**

Ingeniero en Informática por la UAB. Máster en Técnicas avanzadas de automatización de procesos por la UAB. Doctor en Informática por la UAB. Profesor de los Estudios de Informática y Multimedia de la UOC.

**Francesc Rambla i Marigot**

Ingeniero técnico en Telecomunicaciones. Ha trabajado más de diez años como consultor en el ámbito de las tecnologías de la información y las comunicaciones, actividad que ha compaginado con la docencia en el ámbito de la formación profesional. Desde el año 2007, es el responsable de los proyectos de software libre de la Secretaría de Telecomunicaciones y Sociedad de la Información de la Generalitat de Cataluña. Es consultor del máster de Software libre de la UOC.

**Antonio Rodil Garrido**

Ingeniero de Informática por la UOC. Máster en Software libre y máster en Dirección y gestión de la calidad por la UOC. Doctorando en Informática en la Universidad de Cádiz. Tutor y consultor del máster de Software libre de la UOC. En la actualidad presta sus servicios profesionales en EPICSA (Empresa Pública de Informática de Cádiz, S. A., entidad pública de la Excelentísima Diputación Provincial de Cádiz).

Primera edición: febrero 2010  
© Francesc Rambla i Marigot, Antonio Rodil Garrido  
Todos los derechos reservados  
© de esta edición, FUOC, 2010  
Avda. Tibidabo, 39-43, 08035 Barcelona  
Diseño: Manel Andreu  
Realización editorial: Eureka Media, S. L.  
ISBN: 978-84-692-9596-0  
Depósito legal: B-8.144-2010

© 2009, FUOC. Se garantiza permiso para copiar, distribuir y modificar este documento según los términos de la GNU Free Documentation License, Version 1.2 o cualquiera posterior publicada por la Free Software Foundation, sin secciones invariantes ni textos de cubierta delantera o trasera. Se dispone de una copia de la licencia en el apartado "GNU Free Documentation License" de este documento.

## Contenidos

### Módulo didáctico 1

#### **La Administración pública como impulsora de acciones y fomento del software libre**

Francesc Rambla i Marigot

1. Principales argumentos de impulso del software libre
2. Planes directores y estratégicos

### Módulo didáctico 2

#### **La Administración pública como receptora de proyectos internos de software libre**

Francesc Rambla i Marigot

1. Principales argumentos para el uso de software libre en los sistemas de una administración pública
2. Principales argumentos para liberar el software producido por las administraciones públicas
3. Liberación del software

### Módulo didáctico 3

#### **Estudio de casos reales de migración de administraciones públicas**

Antonio Rodil Garrido

1. Fases de una migración a software libre
2. Casos de estudio

### Módulo didáctico 4

#### **Plan de migración de una administración pública**

Antonio Rodil Garrido

1. El plan de migración



# La Administración pública como impulsora de acciones y fomento del software libre

Francesc Rambla i Marigot

PID\_00154682



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



# Índice

<b>Introducción</b> .....	5
<b>Objetivos</b> .....	8
<b>1. Principales argumentos de impulso del software libre</b> .....	9
1.1. Independencia tecnológica de la Administración pública .....	9
1.2. Independencia de proveedor .....	11
1.3. Ahorro económico .....	12
1.4. Apuesta por la industria local de las TIC .....	18
1.5. Libertad de localización y traducción .....	19
1.6. Ayuda a la cohesión digital de la población .....	21
<b>2. Planes directores y estratégicos</b> .....	22
2.1. Principales objetivos de los planes directores de la sociedad de la información .....	23
2.1.1. La administración electrónica .....	23
2.1.2. La cohesión digital .....	24
2.1.3. El crecimiento de la economía a través de las TIC .....	25
2.2. El software libre en los planes directores .....	25
<b>Resumen</b> .....	28
<b>Bibliografía</b> .....	31



## Introducción

Las administraciones públicas que ejercen competencias en la sociedad de la información tienen como uno de sus principales hitos evitar una situación de escisión en el conocimiento de las tecnologías de la información y la comunicación, llamadas TIC, que desemboque en el analfabetismo digital de una parte de la sociedad. Lógicamente, hay todo un conjunto de colectivos que por cuestiones socio-económicas y geográficas se encuentran en una situación de mayor riesgo de exclusión digital, ya sea por la simple dificultad técnica de conseguir conectividad, o bien por cuestiones de marginalidad que requieren de actuaciones específicas que ayuden a minimizar este riesgo.

Este objetivo lleva a las administraciones públicas a impulsar políticas orientadas a combatir esta realidad, que se articulan en diferentes estrategias y líneas de trabajo, a menudo muy condicionadas por las características socio-económicas del territorio en cuestión. Por una parte, lo que puede parecer evidente en países del primer mundo es que uno de los primeros campos donde hay que trabajar es en facilitar la conectividad con banda ancha a todos los puntos del territorio para que nadie quede excluido de la oferta. Ahora bien, estas medidas no tienen mucho sentido en países del tercer mundo, donde la dificultad a menudo empieza en servicios más básicos, como el suministro eléctrico. Pero otras iniciativas relacionadas con la difusión de las TIC, y de todo aquello que pueden ofrecer, son fundamentales para que esta oferta resulte atractiva y que, por lo tanto, se genere una demanda y sean aplicables a prácticamente cualquier punto del planeta. Superadas estas primeras barreras, que a menudo son francamente complejas, las administraciones se encuentran ante la necesidad de formar a la ciudadanía para que sea capaz de aprovecharse de la oferta disponible. Y, entrados en este punto, es cuando una administración pública se empieza a plantear si aquello más conveniente para los administrados es que se les haga usuarios de una solución tecnológica concreta, sobre todo cuando esta solución tiene un coste nada despreciable y beneficia fundamentalmente a una empresa extranjera.

A partir de mediados de los años 90, se empezaron a consolidar algunas soluciones de software libre en determinados ámbitos, como el de los servidores de páginas web, que es dominado por el Apache desde el año 1996; pero todavía no había una oferta lo bastante madura con respecto a las soluciones de escritorio. A finales de los años 90, sin embargo, coinciden una serie de circunstancias: el nacimiento de los proyectos KDE y GNOME con el fin de construir entornos de escritorio más próximos al usuario; la liberación del código del Navigator por parte de Netscape y la posterior creación del proyecto Mozilla; la adquisición y posterior liberación de StarOffice por parte de Sun Microsystems con la creación de OpenOffice.org, que marca una inflexión en este campo.

### Lecturas obligatorias

**Junta de Andalucía** (2006). *Plan Andalucía Sociedad de la Información 2007-2010*.

**Generalitat de Catalunya** (2006). *Pla de Serveis i Continuits de la Generalitat de Catalunya*.

### Lecturas recomendadas

*Extremadura en la Sociedad de la Información, del Conocimiento y de la Imaginación*. Dossier de prensa.

A partir de este punto, entre finales de los años 90 y los primeros años del siglo XXI, algunas administraciones públicas empiezan a ver que el software libre puede suponer una oportunidad para llevar a cabo algunas iniciativas que no habrían sido abordables con las soluciones privativas existentes. Así, por ejemplo, fruto del compromiso electoral de dotar a las escuelas públicas de Extremadura de un ordenador por cada dos estudiantes, nace el proyecto gnuLinux y la estrategia de software libre de Extremadura, una de las más conocidas y observadas desde todo el mundo. Según fuentes de la propia Junta de Extremadura, este compromiso no habría sido económicamente viable desde una solución basada en sistemas privativos.

Ante la existencia de esta oferta de soluciones, que se disponen con un coste irrelevante tanto para la administración como para el ciudadano y que, además, se pueden adaptar y traducir a las necesidades específicas de una sociedad concreta, las administraciones públicas empiezan a identificar un vehículo magnífico sobre el que construir buena parte de sus acciones de alfabetización digital. Así pues, volviendo a aquello que se comenta un poco más arriba en esta introducción, las administraciones públicas empiezan a tomar conciencia de que el hecho de introducir las TIC a los ciudadanos a través de soluciones de software libre las hace más asequibles, especialmente por el ahorro en licencias. Ahora bien, éste no es el único beneficio: alcanzar las competencias básicas para desarrollarse con una solución de software libre, que a menudo no es la que se encuentra en el puesto de trabajo o en el ordenador de casa, da una visión más amplia a la persona que se introduce en el conocimiento digital, que le permite adaptarse mejor a las soluciones, libres o privativas, que hay de utilizar en otros entornos.

Otra de las líneas de trabajo que tienen muchos gobiernos en sus estrategias para la sociedad de la información es la de impulsar los sectores locales de las TIC como una forma de potenciar y apostar por nuevos ámbitos de crecimiento económico. El mundo del software ha sido dominado de forma prácticamente absoluta por empresas de los Estados Unidos, ya sea con los modelos tradicionales de producción y comercialización de software, que encontramos sobradamente representados por grandes corporaciones como Microsoft, Sun Microsystems, Adobe, Oracle, IBM o Novell, o bien, con los nuevos modelos que ha traído la red basados en los servicios en línea, y que ejemplarizan empresas como Google, Yahoo, Facebook o AOL. Este predominio, especialmente cuando se produce en torno a soluciones privativas, pone en una situación muy débil a las compañías locales, que se acaban convirtiendo en simples subsidiarias de unas tecnologías sobre las cuales no tienen ningún tipo de control.

En este sentido, en cambio, el software libre ofrece muchas oportunidades a los integradores y desarrolladores de soluciones informáticas y, generalmente, les puede permitir acabar influyendo o interviniendo en las decisiones que

afecten al futuro de las herramientas que utilizan. Así pues, otra de las líneas de trabajo en el impulso del software libre es la de fomentar su uso por parte del sector TIC.

Mencionemos que estas políticas han sido asumidas por administraciones públicas de todo el mundo, de todos los colores y modelos, de las derechas a las izquierdas, tanto en gobiernos democráticos como en dictaduras. Las políticas de impulso del software libre que han llevado a cabo los gobiernos de Brasil, Malasia o la ya citada, comunidad de Extremadura son destacados ejemplos. Dentro de la bibliografía recomendada, podéis encontrar una muestra de las iniciativas en estos y otros ámbitos del impulso de software libre que han emprendido administraciones públicas de todo el mundo.

En este módulo, por una parte, nos centramos en el papel que desempeña o puede desempeñar la Administración pública para promover el uso del software libre en la sociedad y los argumentos que justifican que se tome esta apuesta y, por otra parte, echamos una mirada a la manera como diversas administraciones han reflejado estas estrategias en los planes que dirige la sociedad de la información.

#### Web recomendada

Center for Strategic and International Studies (Washington DC) (2008). *Government Open Source Policies*. [www.csis.org/media/csis/pubs/0807218\\_government\\_opensource\\_policies.pdf](http://www.csis.org/media/csis/pubs/0807218_government_opensource_policies.pdf)

## Objetivos

Con el estudio de este módulo, el estudiante tiene que satisfacer los siguientes objetivos:

1. Identificar y contextualizar los principales argumentos políticos, sociales y económicos que adoptan las administraciones públicas para impulsar el software libre.
2. Analizar los planes que dirige la sociedad de la información y conocimiento y otros planes estratégicos que recogen las principales líneas de impulso y de difusión del software libre.
3. Establecer un análisis de oportunidades, ventajas y detección de posibles carencias respecto a este tipo de iniciativas avaladas por instituciones públicas.

# 1. Principales argumentos de impulso del software libre

Este primer apartado recoge una exposición de los principales argumentos que impulsan las políticas de impulso del software libre que están llevando a cabo administraciones públicas de todo el mundo.

Los principales argumentos que estudiamos son:

- La independencia tecnológica de la Administración pública.
- La apuesta de las TIC por la industria local.
- La independencia del proveedor.
- El ahorro económico.
- La libertad de localización y traducción.
- La cohesión digital de la población.

La bibliografía disponible en torno al software libre y a la Administración pública se centra mucho más en los procesos de adopción del software libre, generalmente en las llamadas migraciones, y que se trata en los módulos "Estudio de casos reales de migración de administraciones públicas" y "Plan de emigración de una administración pública" de la asignatura, o bien en el papel de la administración como receptora/impulsora de proyectos de desarrollo de software que se cubre en el módulo "La Administración pública como receptora de proyectos internos de software libre", que en el papel de impulsora de acciones y fomento del software libre.

## 1.1. Independencia tecnológica de la Administración pública

Las administraciones públicas de una cierta dimensión tienen actualmente una absoluta dependencia de sus sistemas de información. Servicios básicos como la sanidad pública, la policía o los bomberos, quedarían en gran medida interrumpidos si en un momento dado los sistemas de información que les dan soporte quedaran fuera de servicio. Por este motivo, estos tipos de sistemas se construyen sobre plataformas tecnológicas que garanticen un funcionamiento continuado con sistemas de alta disponibilidad y totalmente redundados.

Habitualmente, los sistemas que soportan estos servicios se desarrollan a medida para la propia administración. Generalmente, por lo menos en el Estado español, en todos los contratos para la construcción de soluciones de software se suele requerir a los proveedores la entrega del código fuente y el traspaso de todos los derechos de explotación de la obra a la administración. Así pues, generalmente la administración es la propietaria de las aplicaciones que sostienen la mayor parte de sus sistemas.

### Lecturas obligatorias

Marcelo D'Elia Branco (2004). *Software libre en la Administración Pública Brasileña*.

The South African view on Open Source (2005). *SA National Open Source Strategy*.

Por otra parte, en algunos casos, existen soluciones en el mercado que ya se adaptan lo bastante bien a los requerimientos concretos de un sistema de información y, en vez de hacer un desarrollo nuevo, se opta por comprar la licencia de uso de esta solución, en el caso de que no sea libre, y parametrizarla para que se adapte o se adecue a las necesidades concretas de tal caso de utilización.

Ahora bien, el software de infraestructura sobre el que funcionan estas aplicaciones –el sistema operativo, los servidores de bases de datos, los servidores de aplicaciones y los servidores de páginas web, entre otros– es prácticamente siempre de terceros que han cedido o vendido una licencia de uso a la administración. En la mayor parte de los casos, este software es privativo; y en la mayoría de aplicaciones intervienen piezas que, generalmente, han sido desarrolladas por diferentes empresas.

¿Qué pasaría si el fabricante de una determinada solución sobre la que está construida nuestra aplicación con alta disponibilidad decidiera dejar de prestarle soporte, o incluso discontinuar dicho producto? En un principio podemos pensar que si ya tenemos la licencia comprada, pues no hay problema, y desde un punto de vista meramente técnico eso probablemente sea así. Muchas veces, sin embargo, en la práctica esta situación hipotética nos llevaría, en un período más o menos largo de tiempo, a la necesidad de buscar un software equivalente a aquel que se ha dejado de ofrecer y a adaptar nuestras aplicaciones para funcionar con esta nueva pieza, y, claro está, eso suponiendo que lo encontráramos.

Determinados sistemas de información de una administración tienen que funcionar y evolucionar de forma continuada con independencia de las decisiones estratégicas que tomen los fabricantes de las soluciones sobre las cuales están contruidos. La independencia tecnológica de la administración se entiende como la capacidad de decidir a corto, medio y largo plazo la propia estrategia tecnológica, sin estar sometidos a las decisiones de una tercera entidad.

La única forma de conseguir esta independencia a un coste razonable es con la adopción y el uso de soluciones de software libre como infraestructura para construir los sistemas de información de una administración pública, ya que, ni aun haciendo una inversión al mantener piezas de software que se hayan dejado de distribuir, se puede marcar la dirección en la que evoluciona la tecnología utilizada.

Es bastante conocido el caso del Ayuntamiento de Múnich, que decidió emprender un costoso proyecto de migración de sus sistemas informáticos hacia soluciones de software libre. La principal motivación para sacar adelante este proyecto fue la pérdida de soporte del sistema operativo que el personal del ayuntamiento tenía instalado en las estaciones del cliente, el Windows NT Workstation. Esta pérdida de soporte implicaba la obligación de actualizar todas sus licencias a versiones posteriores de este sistema, con todas las con-

#### Lectura recomendada

Daniel Reina (2006). *Criterios de migración a Linux en las administraciones locales de la UE. Los casos de Múnich y Newham*. UOC Papers, número 2.

secuencias que eso pudiera suponer: renovación de hardware, adaptación de aplicaciones al nuevo sistema o la formación del personal, por citar algunas. En este caso, el principal criterio para afrontar tal proceso, que desde un punto de vista económico habría sido difícil de justificar, fue la decisión estratégica de garantizar que esta administración tendría la capacidad de escoger el rumbo que tenían que tomar sus TIC en el futuro: la independencia tecnológica.

Ahora bien, la independencia tecnológica no es sólo cosa de sistemas y aplicaciones, también tiene mucho que ver con las personas. La dependencia que genera tener todo el personal de una organización mayor, como es el caso de muchas administraciones públicas, formado o acostumbrado a utilizar unas soluciones concretas, a menudo es una barrera más difícil de salvar que la propia tecnología; por este motivo, es importante velar para que las competencias en las TIC de los ciudadanos, en general, y de los trabajadores de la administración, en particular, sean en gran medida independientes de la solución tecnológica con las que se han alcanzado. Lógicamente, eso no siempre es posible, pero uno de los objetivos fundamentales de la formación en TIC que se promueva desde las administraciones públicas tendría que ser este.

## **1.2. Independencia de proveedor**

Como las administraciones públicas gestionan fondos públicos, los procesos de contratación requieren una transparencia particular y un cuidado especial en la forma en que se utilizan. En general (dejando de lado pequeñas contrataciones en las cuales eso sería del todo ineficiente), cuando una administración pública inicia un proceso de compra se tiene que garantizar que todos los actores del mercado con capacidad para vender aquel producto o servicio podrán contribuir abiertamente a esta licitación. Naturalmente, eso implica necesariamente que los procesos de compra no se restrinjan a un producto o marca concretos, sino que partan de unas características o especificaciones técnicas mínimas que los participantes de la licitación habrán de satisfacer con su producto y exponer en su oferta.

Ahora bien, a menudo sucede que algunos productos acaban ejerciendo una hegemonía tal en su mercado, que lo acaban monopolizando. Tal situación se produce en algunos ámbitos bastante conocidos de la informática y las comunicaciones, que acaba poniendo en peligro la libre competencia a las licitaciones públicas para que, si bien muchos actores pueden concurrir a la licitación, todos ellos lo acaben haciendo con el producto de un mismo fabricante. Eso se produce aunque existan alternativas funcionalmente equivalentes que podrían formar parte de una oferta válida, pero no se tienen en cuenta por el hecho de no estar lo bastante implantadas o extendidas.

Cuando estos productos son privativos, a menudo la capacidad de participar en la licitación va vinculada a una relación de colaboración con el fabricante (*partnership*) que, por otra parte, no suele ser gratuita. De esta manera, el fabricante del producto monopolizado se beneficia de vender su producto con

la oferta ganadora de la licitación y de establecer las relaciones con todas las empresas locales que quieren licitar, por las que a menudo les hace pagar unas cuotas elevadas que las convierten en auténticas aliadas y promotoras de la compañía fabricante y de su tecnología.

El software libre no necesariamente comporta la independencia del proveedor, especialmente cuando nos encontramos ante soluciones muy especializadas, e incluso se puede dar una situación aparentemente contraria. A veces es difícil encontrar a algún proveedor local que tenga los conocimientos necesarios para presentarse a una licitación cuando se pide una solución tecnológica basada con productos de software libre. Pero, a pesar de eso, la pericia y el conocimiento de un producto de software libre están al alcance de todas las compañías del mercado, suscriban o no contratos de colaboración con sus fabricantes. Y el hecho de que exista una oferta de servicios suele depender únicamente de que exista una suficiente demanda.

Si una administración pública licita repetidamente sobre ciertas soluciones de software libre, ello ejerce una influencia en el mercado que hará que cada vez más proveedores obtengan el conocimiento necesario para participar, pero incluso si no fuera así, entonces la propia administración podría emprender acciones dentro del sector para fomentar y promover el conocimiento de estas soluciones.

### **1.3. Ahorro económico**

El argumento de que las implantaciones de soluciones de software libre suponen un ahorro económico es uno de los que más estudiados y de los que más discusiones ha suscitado. Generalmente, estos estudios llegan a conclusiones contrarias según la entidad que los haya promovido o pagado, y a menudo eso depende en gran medida de los conceptos y los plazos que se tengan en cuenta a la hora de realizar los cálculos.

Generalmente, todos estos estudios hablan de conceptos económicos como el TCO (*total cost of ownership*, coste total de propiedad) que se refiere a aquello que hace falta tener en cuenta a la hora de comprar un producto, además del coste mismo de éste (costes de implantación, mantenimiento, soporte, etcétera) o el ROI (*return of investment*, retorno de la inversión) que es el tiempo necesario para recuperar la inversión que supone la apuesta por una solución concreta.

Todo ello hace que el ahorro económico de la elección de software libre, que aparentemente sería uno de los argumentos más claros debido a la inexistencia de coste de licencias, se tenga que matizar y justificar muy bien, porque, de lo contrario, puede ser fácilmente rebatido.

En primer lugar, pues, hay que tener muy presente que las administraciones públicas suelen ser organizaciones muy grandes que utilizan intensivamente las TIC desde hace años. En la mayoría de los casos, las soluciones desplegadas actualmente no son ni están basadas en software libre y, muchas veces, ni tan solo se desarrollaron pensando en la posibilidad de funcionar en un entorno diferente a lo que había en la administración en aquel momento. Así pues, en general, la adopción de nuevas soluciones diferentes a las ya existentes requerirá un importante esfuerzo en la gestión del cambio, tanto con respecto al personal técnico y a los responsables de informática, como para los usuarios de la solución.

Antes de ver detalladamente cada uno de los conceptos que se utilizan para medir el impacto económico del despliegue de una solución informática es necesario remarcar el tópico de que el software libre no es gratuito, las licencias generalmente no impiden el cobro por su distribución, pero las libertades que lo definen acaban suponiendo la gratuidad de la licencia en la mayoría de casos. Los modelos de negocio que se construyen alrededor del software libre (y si son de negocio es porque efectivamente algo se ha de pagar) siempre giran en torno a la prestación de servicios sobre este software.

Uno de los argumentos que los estudios exponen es que el coste total de propiedad de una solución de software libre es superior a la de una solución privativa más conocida y extendida. Con el fin de evaluar esta afirmación, vamos a suponer el escenario de la implantación de una solución o aplicación informática. En la evaluación de sus costes habrá que tener en cuenta como mínimo los siguientes conceptos:

- la licencia o licencias de la solución;
- la instalación, integración y adaptación al entorno informático de la organización donde se utilizará;
- los costes de formación del personal en la aplicación concreta;
- el soporte a los usuarios, desarrolladores y administradores de la aplicación;
- el mantenimiento correctivo y evolutivo de la aplicación.

Lógicamente, el coste exacto de cada uno de estos conceptos dependerá de cada caso, pero vamos a seguir haciendo cálculos. Los fabricantes de software privativo afirman que, en el despliegue de soluciones, el coste de las licencias suele suponer entre un 15 y un 20% del total. Así pues, y dando por buena esta cifra, si el coste total del despliegue de esta aplicación con software privativo fuera 100, podríamos considerar bastante válida la siguiente distribución de costes:

- licencia: 20% (de los cuales 5% son con cariz anual en concepto de actualizaciones),
- instalación y adaptaciones: 25%,
- formación: 15%,

- soporte anual: 15%,
- mantenimiento anual: 25%.

Vale la pena remarcar que los dos últimos conceptos supondrían un gasto anual.

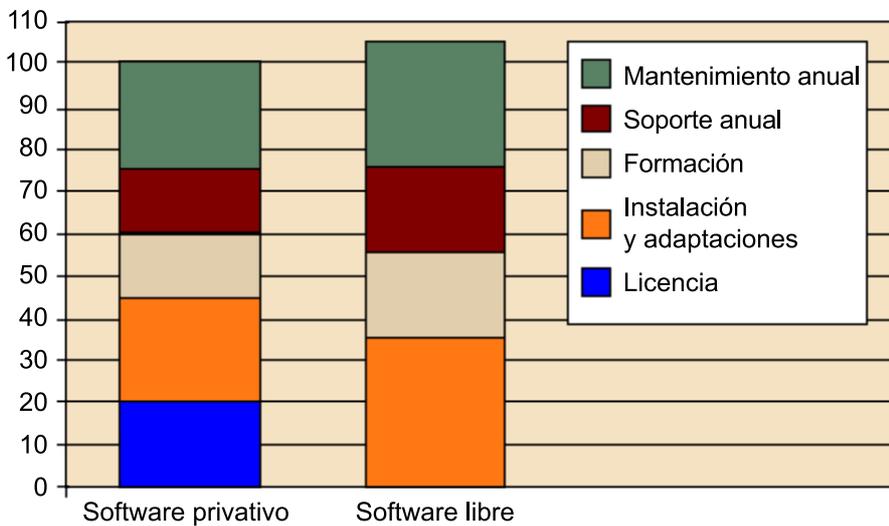
Si ahora hacemos la misma distribución suponiendo que la aplicación se construya a partir de soluciones de software libre y siguiendo los argumentos que los estudios impulsados por las empresas de software privativo esgrimen, podemos suponer que los costes de todos los conceptos, excepto las licencias, serán más elevados por el hecho de que son productos menos conocidos, y eso provocará una menor **concurrencia**<sup>1</sup>, que acabará suponiendo una reducción menos importante de su importe. Así pues, tendremos:

<sup>(1)</sup>Tal como se ha comentado en el argumento "independencia de proveedor", incluso se podría dar el caso de que sólo un proveedor tuviera la capacidad de prestar este servicio, de manera que la falta de competencia haría subir el precio.

- licencia: 0%,
- instalación y adaptaciones: 35%,
- formación: 20%,
- soporte anual: 20%,
- mantenimiento anual: 30%.

El siguiente gráfico ilustra la diferencia entre las dos situaciones:

Figura 1. Distribución de los costes en un proyecto de desarrollo construido con software libre y con software privativo



Dado que hemos utilizado estos cálculos según los principales argumentos que se hacen habitualmente como barrera para emprender proyectos de implantación de software libre, encontramos que, efectivamente, el coste total de propiedad es superior a lo que obtendríamos con una solución privativa. Los cálculos en un caso real son bastantes más complejos, pero este ejemplo es lo bastante ilustrativo sobre la forma como se distribuye el gasto en un proyecto de este tipo. En este sentido, podemos observar que el coste de licencias que tenemos con una solución privativa (que generalmente acabarán suponiendo una transferencia de capital al extranjero) se distribuye en el coste de las otras

partidas con una solución de software libre. Pero precisamente estos servicios son los que hay que prestar in situ, de manera que acaban repercutiendo en una mayor facturación de las empresas locales.

Sin embargo, ¿qué sucede con el periodo de retorno de la inversión, aquello que se llama ROI? Podemos seguir con el mismo supuesto para ver qué sucederá en los próximos años. Consideraremos que el segundo año los precios de los servicios de soporte y mantenimiento permanecen inalterables por el hecho habitual de que la administración licita este tipo de provisión con carácter prorrogable. A partir del segundo año, sin embargo, hará falta una nueva licitación para continuar disponiendo de estos mismos servicios.

En el primer caso, considerando la solución de software privativo, aunque es posible conseguir rebajar los costes, la verdad es que probablemente sólo será posible mantenerlos, ya que los servicios se contrataron en una situación de libre concurrencia sobre una solución madura en el mercado, en la que diferentes proveedores podían acreditar su experiencia, y no parece muy razonable que sea posible obtener una gran rebaja. Por otra parte, consideraremos también que las cuotas de mantenimiento de actualizaciones no aumentan.

Con respecto a la solución de software libre, en cambio, hemos partido del supuesto de que la falta de concurrencia, y por lo tanto de competencia, era la causante de un incremento de los costes. Así pues, parece que no sería demasiado extraño que otros proveedores hayan empleado la pericia necesaria para proveer estos servicios; de aquí que este argumento pierda peso; con lo cual sería lógico obtener una rebaja, como mínimo hasta situar el importe de estos servicios al mismo nivel que los que obtenemos con la solución privativa.

La tabla siguiente resume la evolución de estas dos situaciones.

Tabla 1. Evolución anual de los costes de un proyecto de desarrollo construido con software libre y software privativo

		año 1	año 2	año 3	año 4	...
Software privativo	Licencias	20	5	5	5	
	Instalación y adaptaciones	25	0	0	0	
	Formación	15	0	0	0	
	Soporte anual	15	15	15	15	
	Mantenimiento anual	25	25	25	25	
	Coste año	<b>100</b>	<b>45</b>	<b>45</b>	<b>45</b>	
	Coste acumulado	<b>100</b>	<b>145</b>	<b>190</b>	<b>235</b>	

		año 1	año 2	año 3	año 4	...
	Coste acumulado de los servicios	80	120	160	200	
Software libre	Licencias	0	0	0	0	
	Instalación y adaptaciones	35	0	0	0	
	Formación	20	0	0	0	
	Soporte anual	20	20	15	15	
	Mantenimiento anual	30	30	25	25	
	Coste anual	105	50	40	40	
	Coste acumulado	105	155	195	235	
	Coste acumulado de los servicios	105	155	195	235	

Como se puede ver, el coste superior inicial de la solución de software libre (la inversión necesaria para desarrollar la aplicación como software libre) quedaría absorbido por el coste recurrente del mantenimiento de las licencias del software privativo en un plazo de cuatro años. En este mismo periodo, el importe destinado a servicios, que representaría la parte de la contratación que beneficiaría directamente a las empresas locales, sería un 17,5% mayor que en el caso de haber construido la solución con software privativo.

Ahora bien, todo este supuesto se ha hecho basándose en el escenario del despliegue de una aplicación donde el coste de licencias supone una parte muy pequeña del global. Sin embargo, ¿qué sucedería si la implantación fuera de soluciones ya listas en un determinado ámbito de la administración o la sociedad: una suite ofimática, un antivirus, un cliente de correo, un editor de gráficos o una impresora PDF? ¿Qué pasaría si el producto tuviera un número de usuarios potenciales y, por lo tanto, de licencias, muy elevado? A partir del análisis efectuado en el supuesto anterior, se puede ver que cuanto más peso tenga el coste de la licencia sobre el global del proyecto, más ahorro comportará la utilización de soluciones de software libre. Por otra parte, también evidencia que cuanto mayor sea el impacto de esta implantación (especialmente sobre las personas pero también sobre otras aplicaciones o sistemas), mayor será la inversión inicial, dada la necesidad de realizar adaptaciones o formar al personal en las nuevas aplicaciones.

Habría que considerar que el escenario donde se desarrolla este supuesto es plausible a corto plazo, pero que si una administración pública empieza a licitar servicios sobre soluciones de software libre, las empresas de su entorno tratarán de adquirir la pericia suficiente en estos entornos para poder competir

en mejores condiciones en estas licitaciones, lo que llevará a desmontar el argumento del sobrecoste por desconocimiento sobre el que hemos construido el supuesto anterior a partir de las sucesivas licitaciones en estas tecnologías.

Aparte del ahorro económico vinculado a la gratuidad de las licencias del software libre, continuamente se habla del hecho de que las nuevas versiones de los entornos de software libre no suelen requerir actualizaciones de hardware, lo cual suele darse en los sistemas operativos privativos por el hecho de que los fabricantes llegan a acuerdos para vender estos sistemas a los nuevos equipos. Generalmente, un equipo que ya ha llegado al final de su vida útil con un sistema privativo porque las aplicaciones actuales ya no funcionan puede seguir siendo útil en un entorno de software libre, ya sea utilizando un sistema ligero, o bien actuando como terminal ligero en un sistema de terminales. Este tipo de re-utilización no es muy frecuente en sistemas corporativos o empresariales, pero sí en los contextos escolares, de telecentros o de aulas de alfabetización digital.

### **Telecentros**

Un telecentro es un espacio orientado al desarrollo de las habilidades digitales y al acceso a las nuevas tecnologías. Los telecentros están dotados de equipamiento informático y de ayuda personalizada (a diferencia de un locutorio o un cibercafé), y ofrecen a los usuarios diversos servicios, aparte de la simple conectividad, que les permite mejorar su capacitación tecnológica y les facilita incorporarse a la sociedad de la información y superar la fractura digital.

La mayoría de las consideraciones que hemos expresado hasta ahora se apoyan en la Administración pública como receptora de desarrollos internos de software libre. ¿Qué sucede si nos ponemos en el papel de la Administración pública como impulsora del uso?

La mayor parte de legislaciones protegen el software con las leyes de propiedad intelectual<sup>2</sup> y en la mayoría de casos<sup>3</sup> el uso de software privativo sin la adquisición de la correspondiente licencia es ilegal. Es responsabilidad de las sociedades de autores denunciar y de la justicia, perseguir estos actos. Desde el punto de vista del impulso de la sociedad de la información, ello supone una discriminación para las clases menos favorecidas, ya que las coloca en el dilema de escoger entre quedarse al margen de la digitalización, o bien acceder ilegalmente. En este caso, la gratuidad de las licencias que encontremos en la práctica totalidad del software libre hace muy difícil buscar argumentos que justifiquen que no sea la opción más adecuada en iniciativas como la incorporación de las TIC al sistema educativo, las redes de telecentros y las actividades de alfabetización digital, las acciones para fomentar la digitalización de las empresas y cualquier actividad, en general, que busque una mayor penetración de las TIC en la sociedad.

<sup>(2)</sup> *Copyright* en la terminología anglosajona.

<sup>(3)</sup> La excepción sería el software privativo gratuito (*freeware*, en inglés).

#### 1.4. Apuesta por la industria local de las TIC

Las administraciones públicas son especialmente responsables de la manera en que se efectúan las compras; y todavía más cuando tienen un cierto tamaño, ya que sus decisiones acaban influyendo ampliamente en el mercado. Hemos visto que algunos de los argumentos que impulsan las administraciones para promover el uso del software libre son la "independencia tecnológica" y la "independencia del proveedor", que han de garantizar que una administración será capaz de establecer en qué sentido tienen que evolucionar las tecnologías que utilizan para su funcionamiento y que podrán comprar esta tecnología al proveedor que más le convenga. Aunque también sería deseable que esta capacidad de compra de la administración beneficiara de forma especial a sus ciudadanos y ciudadanas por medio de la adquisición de productos o servicios prestados por empresas locales.

Como ya se ha comentado, la industria de las TIC ha sido dominada por grandes corporaciones multinacionales que ofrecen sus servicios a las administraciones públicas de todo el mundo, ya sea directamente, o bien indirectamente a través de sus redes de colaboradores (*partners*). Por esta razón el sector de las TIC, fuera de estos países y especialmente en Europa, está formado por pequeñas y medianas empresas que, en la mayoría de los casos, ofrecen servicios con un elevado grado de especialización sobre tecnologías de terceras compañías. Pero el principal capital de la sociedad de la información es el conocimiento digital. Quienes no tienen este conocimiento se encuentran excluidos de esta sociedad y quienes no tienen capacidad de crear se encuentran relegados a ser simples observadores. Si las empresas del sector TIC de un determinado territorio se limitan a ser subsidiarias de la tecnología de otros, obviando la capacidad creativa, será difícil que este sector crezca más allá de la capacidad de compra que tenga su entorno.

Las administraciones públicas de los países que no son importantes creadores de conocimiento digital apuestan por mejorar la productividad y la competitividad de las empresas locales en los sectores relacionados con las TIC como una medida para modernizar sus economías.

El software libre supone una muy buena oportunidad para este tipo de empresas, porque dispone una gran cantidad de código a punto para ser utilizado en sus proyectos. Según refleja el informe desarrollado para la comisión europea en el año 2006 sobre el impacto económico del software libre en los sectores TIC europeos, volver a escribir la cantidad de software libre de calidad existente en aquel momento habría supuesto un coste de doce billones de euros, y, además, esta base se había ido doblando cada 18-24 meses desde 1998. Toda esta base, y mucha más que se ha desarrollado desde entonces, está a disposición de las empresas que lo deseen y que tengan la capacidad de acceder.

#### Lectura recomendada

*Economic impact of FLOSS on innovation and competitiveness of the EU ICT sector (2006).*

El modelo de construcción en red que se utiliza en muchos proyectos de software libre se adecua mucho a las estructuras pequeñas y especializadas de estas empresas, que, si bien no tienen la masa suficiente para construir una gran aplicación, pueden contribuir a crear pequeños módulos de su especialidad que permiten que el conjunto de la aplicación se desarrolle y mejore.

Por otra parte, vistos los beneficios y las oportunidades que el software libre ofrece a las administraciones públicas para la construcción de soluciones de software, es bastante clara la conveniencia de que las empresas locales estén capacitadas para ofrecer este tipo de servicios sobre aplicaciones de software libre. Este motivo justificaría por sí solo la necesidad de emprender acciones que faciliten que el sector local de las TIC tenga conocimiento y capacidad de trabajar con estas soluciones tecnológicas.

### **1.5. Libertad de localización y traducción**

En una herramienta informática, el término localización se refiere al conjunto de modificaciones necesarias para adaptar esta herramienta a las características locales de un país o región. Estas adaptaciones suelen incluir su traducción –tanto de la interfaz como de la documentación–, la adaptación al marco legal local, la divisa utilizada en la expresión de importes económicos, los diccionarios, la forma de expresar números, fechas o importes, etc.

El principal objetivo que mueve a la industria del software es, fundamentalmente, tanto para el software libre como para el privativo, conseguir el máximo rendimiento de la inversión realizada. Pero, dejando de lado este objetivo, los modelos de negocio en las dos industrias a menudo son bastante diferentes. Mientras que en el primer caso el fabricante obtiene una gran parte de los beneficios de la venta de licencias de su producto, en el segundo los obtiene de vender servicios a su alrededor. En los dos modelos el beneficio del fabricante es mayor cuantos más usuarios tenga su producto, pero en el modelo del software privativo todas las modificaciones y adaptaciones del producto corren a su cargo.

La realidad a la que conduce este hecho es que la gran mayoría del software privativo sólo está disponible en la lengua original en que se desarrolló y, en todo caso, en las lenguas de aquellos territorios donde su fabricante identifica un mercado lo bastante interesante como para que compense el esfuerzo de emprender esta traducción. ¿Qué sucede con lenguas minoritarias como el catalán, el gaélico, el vasco, el gallego, el occitano, el flamenco, etc.? ¿Qué pasa con lenguas que requieren de adaptaciones de la interfaz, como el árabe, el chino o el hebreo? Pues depende de cada caso. En determinadas ocasiones los fabricantes pueden autorizar la traducción por parte de terceros, o incluso llevarla a cabo a cambio de alguna contraprestación. También puede ocurrir que el fabricante no autorice la localización. Hay que destacar que los fabricantes tienen derecho a hacerlo, ya que supone una modificación del software y, por

lo tanto, un incumplimiento de la licencia. Así que, finalmente, la situación más habitual es que los hablantes de estas lenguas acaben utilizando el software con un idioma diferente al suyo.

En muchas ocasiones, las administraciones públicas han empujado a los fabricantes de software a traducirlo, ya sea directamente por medio de una contratación, o bien indirectamente como contrapartida por la adquisición de un determinado volumen de licencias. Desgraciadamente, el software evoluciona y a veces estas acciones quedan reducidas a una versión concreta. Según las condiciones con las que se haya efectuado una operación de este tipo, podría ser necesario, incluso, volver a traducir toda la aplicación de cada una de las versiones posteriores.

Con respecto a la localización, actualmente los entornos gráficos de los sistemas operativos con mayor implantación han resuelto muy bien muchos de los aspectos de este proceso (la expresión de fechas, números, importes económicos, etc.), así que la mayor parte de aplicaciones delegan estas tareas al sistema operativo. En cuanto al resto de adaptaciones, las aplicaciones se desarrollan de una forma lo bastante genérica para que sean válidas, con independencia de estas particularidades y, cuando eso no es posible, como ocurriría con los diccionarios, muchas veces se opta por soluciones externas bien documentadas que presten estos servicios y que puedan ser provistas por terceros.

En el software libre, en cambio, las diferentes licencias existentes garantizan la libertad de modificar el software y de redistribuir las copias modificadas; por lo que la libertad de traducción es una de sus características inherentes. De hecho, muchos proyectos de software libre tienen comunidades de traducción y localización que se ocupan de mantener las versiones de los diferentes idiomas y que están abiertas a la incorporación de nuevas lenguas siempre que alguien se encargue de ello.

En contextos de software libre, es muy frecuente que todas las aplicaciones compartan los mismos motores de corrección; de esta manera, se consigue una mejor integración y coherencia entre las diferentes aplicaciones.

Últimamente, las administraciones públicas que tienen competencias en territorios con lenguas minoritarias participan o colaboran en los procesos de localización de aquellas aplicaciones de software libre que utilizan, ya sea directamente, con proyectos internos o bien ofreciendo a la ciudadanía acciones de difusión.

## 1.6. Ayuda a la cohesión digital de la población

Uno de los principales hitos de las administraciones públicas con responsabilidades en el campo de la sociedad de la información es la cohesión digital de su población. La cohesión digital se refiere a la situación deseable de una sociedad en la que no haya excluidos digitales, es decir, donde ninguna persona quede desvinculada del conocimiento y de la utilización de las TIC.

Los principales instrumentos de los que disponen las administraciones públicas para evitar esta fractura digital de la sociedad son:

- el sistema educativo,
- las escuelas de adultos,
- las escuelas de administración pública,
- los cursos de formación ocupacional,
- las redes de telecentros.

Cada uno de estos instrumentos se dirige a colectivos diferentes. No tienen como misión exclusiva la cohesión digital de la sociedad, a excepción quizás, de las redes de telecentros.

El software libre ayuda a promover la cohesión digital de la sociedad fundamentalmente porque posibilita que todos estos instrumentos lleguen a un número mayor de ciudadanos, gracias a que su despliegue es más económico –tanto por la gratuidad de las licencias como por la posibilidad de reutilizar equipos al límite de su vida útil–. Por otro lado, dado que en determinados entornos el idioma puede suponer una barrera, la posibilidad de adaptar y traducir el software a las características locales de los ciudadanos suele suponer que el software libre sea más asequible.

Así, en principio es más sencillo introducir a alguien en las TIC por medio de soluciones libres, antes que por medio de las privativas, por lo menos sobre el papel. Si además añadimos que el ciudadano que se ha introducido con una de estas soluciones puede llevársela a casa legalmente y sin ningún coste significativo –sólo el del soporte o el ancho de banda necesario para descargarlo–, parece evidente que el software libre desempeña o puede desempeñar un notable papel en el campo de la cohesión digital de la sociedad.

Las experiencias más relevantes en este ámbito se han realizado hasta ahora en el entorno educativo, donde sería discriminatorio ofrecer la preparación en el ámbito TIC sobre soluciones que no estuvieran al alcance de toda la comunidad educativa, y en torno a los telecentros, donde se trata de concentrar la inversión en los servicios, especialmente los de formación y difusión, por encima de la infraestructura del centro, ya sea de hardware o de software.

## 2. Planes directores y estratégicos

La rápida evolución que presentan las tecnologías de la información y las comunicaciones, las TIC, y los efectos que estas tecnologías tienen en los modelos de relación entre las personas, las empresas y las administraciones públicas son difíciles de prever, pero a corto plazo se identifican riesgos y barreras que muchas administraciones públicas intentan salvar con el fin de garantizar que las sociedades que administran vayan adaptándose a este progreso.

Muchas administraciones públicas han identificado las oportunidades potenciales de crecimiento económico y de ocupación que las TIC han supuesto para países con economías emergentes y han visto oportunidades directas para sus territorios. Sin embargo, las mejoras que la adopción de estas tecnologías aportan en la productividad y competitividad del resto de sectores son igualmente relevantes y no se pueden dejar de lado, ya que no es posible promover el crecimiento del sector de las TIC sin un impulso paralelo de estas tecnologías en empresas de otros ámbitos.

Si bien es necesario que las administraciones hagan un importante esfuerzo presupuestario para garantizar las infraestructuras necesarias que dan acceso a servicios como la banda ancha o la televisión digital disponibles en todo el territorio, las infraestructuras no lo son todo. También es de especial importancia invertir allí donde la educación del futuro ciudadano lo capacite para moverse ágil y cómodamente por los entornos digitales, y evitar que parte de la sociedad quede digitalmente excluida, al ofrecer unos servicios de administración electrónica que sirvan al mismo tiempo como proyectos tractores de los sectores locales de las TIC y como dinamizadores de la demanda de servicios digitales, entre otros.

Uno de los objetivos de la estrategia de Lisboa en la Comisión Europea era convertir Europa en la economía basada en el conocimiento más potente del mundo. Como forma de relanzar estos objetivos se promovió la iniciativa i2010 que identifica los tres ejes prioritarios siguientes:

- la creación de un espacio de información europeo único,
- el incremento de la inversión en la investigación en las TIC,
- el fomento de la inclusión de las TIC con servicios públicos digitales de calidad.

La mayoría de las administraciones públicas de Europa que tienen competencias en la sociedad de la información han construido estrategias y planes alineados con estos ejes. Las administraciones públicas exponen sus políticas y

### Lecturas obligatorias

Junta de Andalucía (2006). *Plan Andalucía Sociedad de la Información 2007-2010*.

Generalitat de Catalunya (2006). *Pla de Serveis i Continguts de la Generalitat de Catalunya*.

### Lectura recomendada

*Extremadura en la sociedad de la información, del conocimiento y de la imaginación*. Dossier de prensa.

### Web recomendada

[www.europarl.europa.eu/summits/lis1\\_en.htm](http://www.europarl.europa.eu/summits/lis1_en.htm)

estrategias en un campo concreto en planes directores, hojas de ruta o planes estratégicos que, una vez aprobados por el gobierno, vienen acompañados de una dotación presupuestaria durante un periodo de tiempo.

Es sencillo establecer sinergias entre los objetivos que persiguen estos planes y los argumentos que se exponen en el apartado anterior para promover los proyectos de software libre desde la Administración pública. En este apartado, nos centraremos en el estudio de los planes directores de la sociedad de la información en diferentes administraciones, sus principales objetivos y cómo influyen o confluyen en las políticas de impulso del software libre.

## **2.1. Principales objetivos de los planes directores de la sociedad de la información**

En esta parte de la documentación, se repasan los objetivos que persiguen los planes directores de la sociedad de la información.

Naturalmente, estos objetivos suelen contextualizarse en el entorno particular de cada administración pública y dependen de ese mismo entorno en gran medida. En cualquier caso, generalizando y tratando de buscar los elementos comunes, podemos destacar los tres ejes que se identifican en la mayor parte de los casos:

- la administración electrónica,
- la cohesión digital de la sociedad,
- el crecimiento de la economía a través de la industria de las TIC.

### **2.1.1. La administración electrónica**

Hoy en día, la mayoría de las administraciones públicas tienen líneas de trabajo para impulsar o promover la administración electrónica. Pero ¿qué queremos decir con "administración electrónica"?

Con el término "administración electrónica" nos referimos a los modelos de administración pública basados en la utilización de las tecnologías de la información y de la comunicación. Los principales objetivos que se persiguen con la aplicación de estos modelos son mejorar la eficiencia interna, las relaciones entre administraciones y las relaciones de la administración con las personas, las empresas y las organizaciones.

Dado que la presencia de las TIC en la vida de los ciudadanos y ciudadanas es cada vez mayor, es natural que para las administraciones se cree una demanda de servicios basados en estas tecnologías. Pero, aparte de eso, la aplicación de las TIC en todos los procesos que marcan la actividad de la Administración

pública también supone una buena oportunidad para que lleguen a ser más eficientes y ágiles. Así pues, la administración electrónica aporta beneficios importantes tanto para la ciudadanía como para la misma administración.

Para la ciudadanía, la administración electrónica facilita el acceso a los servicios públicos, desde cualquier lugar, cualquier día del año, a cualquier hora. Aporta una mayor simplicidad en las tramitaciones con la administración. Todo se produce con una mayor rapidez y agilidad en la obtención del servicio requerido, sin necesidad de desplazarse físicamente a las dependencias de la administración.

Ahora bien, eso nos lleva al primero de los requisitos: la garantía de los principios básicos de no exclusión a fin de que el acceso a estos servicios se produzca para toda la ciudadanía con igualdad de condiciones, con independencia de su situación geográfica, social y económica.

Con respecto a las administraciones públicas, la administración electrónica permite ofrecer una mejora del servicio y, por lo tanto, de la propia imagen, por una parte; pero también permite una mayor eficiencia interna, que se consigue con la integración de los diferentes canales de prestación de los servicios y el fomento del uso generalizado de las nuevas tecnologías. Estas iniciativas acaban suponiendo un ahorro económico, ya que el coste por trámite se reduce, y al mismo tiempo suponen un argumento de peso para introducir cambios con los que modernizar los procesos internos de una administración pública.

Dada la lógica responsabilidad de las administraciones públicas en su propia eficacia interna, por un lado, y que la administración electrónica permite mejorar el acceso a los servicios de la administración a los ciudadanos, a la vez que supone un elemento dinamizador de las TIC, parece lógico que uno de los principales objetivos de los planes directores de la sociedad de la información sea impulsar y promover las iniciativas de administración electrónica.

### **2.1.2. La cohesión digital**

La rápida evolución y crecimiento de la oferta de servicios digitales está revolucionando las relaciones entre las personas en todos los ámbitos de la vida: el doméstico, el laboral, el educativo, etc. Ahora bien, esta revolución no está exenta de riesgos, y si bien se pueden identificar muchos beneficios en la irrupción de las TIC en nuestras sociedades, también es cierto que ésta comporta un importante grupo de responsabilidades para las administraciones públicas con el fin de evitar que una parte de la sociedad quede desvinculada de este proceso.

La cohesión de la sociedad a través de las tecnologías de la información y la comunicación es lo que llamamos cohesión digital. Este concepto se podría explicar como aquella situación ideal en la cual ningún miembro de una sociedad quedaría excluido del acceso y el conocimiento de las TIC por motivo

de su situación geográfica, económica o social. En las sociedades basadas en el conocimiento digital, el fracturado o analfabeto digital es aquel que no tiene las competencias suficientes para desarrollarse, si tenemos en cuenta lo que supone el analfabetismo en las sociedades de la era industrial.

El objetivo de la cohesión digital es básico para evitar que los desequilibrios que se producen dentro de las sociedades actuales se reproduzcan en las sociedades del futuro y forma parte fundamental de la mayor parte de planes directores de la sociedad de la información.

### **2.1.3. El crecimiento de la economía a través de las TIC**

La creencia en que los modelos económicos que se construyan a partir de las TIC permitirán, a medio y largo plazo, modelos de crecimiento de la economía más sólidos que los que se basan en los sectores industriales ha llevado a muchas administraciones a emprender líneas de trabajo para propiciar el desarrollo de estos sectores.

Este tipo de acciones se centran, por una parte, en dinamizar la demanda, tratando, de hacer más atractiva la oferta por medio de acciones de difusión, formación y asesoramiento en el uso de las TIC. De esta manera, se trata de conseguir que el mercado interior de éstas sea más amplio y, por lo tanto, atraiga una mayor masa laboral.

Por otra parte, y de forma paralela, hay que preparar la oferta para ser capaz de responder de forma eficiente a los requerimientos de la misma. En este sentido, se emprenden acciones para aumentar el número de profesionales cualificados que trabajan en el sector y para mejorar la preparación de los que ya están trabajando.

Finalmente, para poder analizar la evolución tanto de la oferta como de la demanda, es necesario acompañar todo este proceso de servicios de observación que permitan tomar el pulso a la situación actual del sector y evaluar el impacto de las medidas realizadas.

El objetivo último de todas estas acciones es que los sectores productivos tradicionales sean más competitivos con una aplicación eficiente de las TIC y que el sector de las mismas tenga la capacidad de ofrecer servicios de mayor valía inicialmente en el mercado interior, pero también de exportarlos.

## **2.2. El software libre en los planes directores**

Muchas administraciones que han elaborado planes directores de la sociedad de la información han identificado en el software libre un medio muy válido para llevar a cabo una parte importante de sus objetivos estratégicos.

Entre los diferentes argumentos que utilizan las administraciones públicas para impulsar la promoción del uso del software libre en la sociedad, identificamos algunos que se alinean totalmente con los ejes sobre los que se construyen los planes directores.

La utilización de software libre en los proyectos de administración electrónica entraría más en el conjunto acciones que hay que emprender para que la Administración pública se convierta en receptora de proyectos internos de software libre. En cualquier caso, sin embargo, la necesidad inherente de estos sistemas de interoperar con otras administraciones y con la ciudadanía, requiere que se establezcan unos criterios de neutralidad con respecto a las tecnologías que se utilicen; es decir, los sistemas de administración electrónica se tienen que construir y ofrecer con independencia de la tecnología que utilice el ciudadano u otra administración para acceder a la misma, y no tendrían que discriminar a nadie por motivo de su elección tecnológica.

Con respecto a la cohesión digital, la utilización del software libre en las acciones de alfabetización digital permite reducir algunas de las barreras de entrada que supone la adopción de las TIC por diversos motivos. Por una parte, la posibilidad de localizar y adaptar el software a las características específicas de una sociedad lo hace más próximo. El software libre funciona en equipos de prestaciones más modestas; de manera que muchas veces es posible reutilizar ordenadores que han llegado al final de su vida útil en un entorno empresarial para el uso doméstico con soluciones de software libre. Finalmente, tal como también se comenta en este módulo, la gratuidad de las licencias facilita la utilización legal del software libre a cualquier ciudadano sin discriminación.

### **gnuLinux en Extremadura**

Aunque es una decisión criticada desde algunos sectores, en la distribución educativa gnuLinux utilizada en el sistema educativo extremeño, se optó por cambiar los nombres de las aplicaciones por los de personajes extremeños de relevancia. Así, por ejemplo, se decidió nombrar la herramienta de edición de imágenes del GNOME, el GIMP, con el nombre del pintor Zurbarán, y el procesador de textos con el del escritor Espronceda. El único motivo de esta decisión fue hacer la distribución más próxima a la sociedad.

Con respecto a la promoción del uso de las TIC como motor de impulso de la economía, estudios como los que encontramos en *Economic impact of FLOSS on innovation and competitiveness of the EU ICT sector (2006)*, muestran el potencial que el software libre supone para la mejora de la competitividad del sector. La existencia de una gran cantidad de código disponible para ser utilizado –o reutilizado– permite reducir los costes de implantación de soluciones, o bien destinar estos recursos a mejorar la lógica de negocio con que se pueden obtener productos más adaptados a las necesidades de cada cliente o sector. Por lo que se refiere directamente al sector de las TIC, la posibilidad de ser más productivo y competitivo, a partir de esta base existente, le permite hacer más atractiva su oferta y, en definitiva, vender más.

#### **Estándares abiertos**

La recomendación más habitual es que la interoperabilidad de estos sistemas se construya entorno a estándares abiertos, y sólo alternativamente se utilicen los formatos de amplia implantación: los llamados *estándares de facto*.

Así pues, la mayoría de planes directores de la sociedad de la información enfatizan en las posibilidades que el software libre supone en estos campos y en otros. Por este motivo, muchos planes incluyen líneas de trabajo para impulsar el uso del software libre en la sociedad como medida complementaria que ayuda a la consecución de sus objetivos estratégicos.

## Resumen

Las administraciones públicas con competencias en las TIC, y concretamente en la sociedad de la información, han identificado toda una serie de oportunidades en el impulso de los proyectos de software libre, que se justifican, entre otros, con el siguiente conjunto de argumentos:

- la independencia tecnológica de la Administración pública,
- la apuesta por la industria local de las TIC,
- la independencia del proveedor,
- el ahorro económico,
- la libertad de localización y traducción,
- la cohesión digital de la población.

En cualquier caso, es importante comprender que el software libre por sí mismo sólo es un medio a disposición de las administraciones públicas, para llevar a cabo de una forma más efectiva las políticas que dan respuesta a las competencias asignadas.

Las administraciones públicas grandes hacen usos intensivos de soluciones TIC, tienen una alta capacidad de influencia en el mercado y se organizan con estructuras muy atomizadas, poco avezadas con la *filosofía colaborativa* del software libre. Todo eso, que por una parte puede dificultar la adopción de soluciones internas de software libre, también supone una oportunidad para liderar y promover el uso de soluciones de software libre en la sociedad.

Los principales ámbitos de la sociedad en los cuales se ha impulsado desde las administraciones públicas la utilización del software libre han sido: el entorno educativo, con numerosas experiencias de utilización de software libre, ya esté sobre sistemas privativos o libres; las acciones de alfabetización digital de la población, a menudo apoyadas por las redes de telecentros y, finalmente, en las actividades para promover la informatización de las pequeñas y medianas empresas.

Generalmente, las administraciones públicas basan sus líneas de actuación sectoriales en planes directores que identifican los ejes de actuación que hay que emprender para alcanzar una serie de objetivos estratégicos. La mayoría de administraciones que han desarrollado planes para impulsar la sociedad de la información en los primeros años de nuestro siglo XXI han trabajado sobre los siguientes ejes:

- la administración electrónica, término con el que entendemos cualquier modelo de administración que se base en las tecnologías de la información y la comunicación;
- la cohesión de digitalización, como paso indispensable para garantizar la igualdad de oportunidades de toda la ciudadanía ante los retos de la sociedad del conocimiento;
- el crecimiento de la economía con las TIC, entendido desde los objetivos complementarios de mejorar la competitividad de las empresas de cualquier sector por medio de una utilización eficiente de las TIC, pero también de las propias empresas del sector de las TIC, para que impulsen la transición de la sociedad industrial hacia una sociedad del conocimiento y ayuden a modernizar la economía local.

Muchas administraciones públicas han incluido líneas de impulso del software libre como un medio para apoyar estos ejes estratégicos.



## Bibliografía

**Branco, Marcelo D'Elia** (2004). *Software Libre en la Administración Pública Brasileña*. <[http://www.alfa-redi.com/apc-aa-alfaredi/img\\_upload/9507fc6773bf8321fcad954b7a344761/branco.pdf](http://www.alfa-redi.com/apc-aa-alfaredi/img_upload/9507fc6773bf8321fcad954b7a344761/branco.pdf)>  
[Consulta: noviembre 2009].

**Center for Strategic and International Studies (Washington DC)** (2008). *Government Open Source Policies*. <[http://www.csis.org/media/csis/pubs/0807218\\_government\\_opensource\\_policies.pdf](http://www.csis.org/media/csis/pubs/0807218_government_opensource_policies.pdf)>  
[Consulta: noviembre 2009].

*Economic impact of FLOSS on innovation and competitiveness of the EU ICT sector* (2006). <[http://ec.europa.eu/enterprise/sectors/ict/files/2006-11-20-flossimpact\\_en.pdf](http://ec.europa.eu/enterprise/sectors/ict/files/2006-11-20-flossimpact_en.pdf)>  
[Consulta: noviembre 2009].

*Extremadura en la Sociedad de la Información, del Conocimiento y de la Imaginación*. Dossier de prensa. <[http://www.juntaex.es/consejerias/economia-comercio-innovacion/dg-telecomunicaciones-sociedad-informacion/common/dossier\\_prensa\\_11\\_junio\\_07.pdf](http://www.juntaex.es/consejerias/economia-comercio-innovacion/dg-telecomunicaciones-sociedad-informacion/common/dossier_prensa_11_junio_07.pdf)>  
[Consulta: noviembre 2009].

**Junta de Andalucía** (2006). *Plan Andalucía Sociedad de la Información 2007-2010*. <[http://www.juntadeandalucia.es/innovacioncienciayempresa/descarga/contenidos/cice/SSI-3316410/noticia/CICE280307d/Plan\\_ASI\\_2007\\_2010\\_0702081.pdf](http://www.juntadeandalucia.es/innovacioncienciayempresa/descarga/contenidos/cice/SSI-3316410/noticia/CICE280307d/Plan_ASI_2007_2010_0702081.pdf)>  
[Consulta: noviembre 2009].

**Generalitat de Catalunya** (2006). *Pla de Serveis i Continguts de la Generalitat de Catalunya*. <<http://www.lafarga.cat/sites/default/files/PSiC.pdf>>  
[Consulta: noviembre 2009].

**Reina, Daniel** (2006). *Criterios de migración a Linux en las administraciones locales de la UE. Los casos de Múnich y Newham*. Uocpapers número 2. <<http://www.uoc.edu/uocpapers/2/dt/esp/reina.pdf>>  
[Consulta: noviembre 2009].

**Rishab A. Ghosh (MERIT), Rüdiger Glott (MERIT), Karsten Gerloff (MERIT), Patrice-Emmanuel, Schmitz (UNISYS), Kamini Aisola (UNISYS), Abdelkrim Boujraf (UNISYS)** (2007). *Study on the effect on the development of the information society of European public bodies making their own software available as open source*. <[http://publicsectoross.info/images/resources/17\\_366\\_file.pdf](http://publicsectoross.info/images/resources/17_366_file.pdf)>  
[Consulta: noviembre 2009].

**SA National Open Source Strategy** (2005). *The South African view on Open Source*. <[http://radian.co.za/fossconference/GOSS\\_TTC\\_workingpaper101.pdf](http://radian.co.za/fossconference/GOSS_TTC_workingpaper101.pdf)>  
[Consulta: noviembre 2009].



# La Administración pública como receptora de proyectos internos de software libre

Francesc Rambla i Marigot

PID\_00154680



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



# Índice

<b>Introducción</b> .....	5
<b>Objetivos</b> .....	7
<b>1. Principales argumentos para el uso de software libre en los sistemas de una administración pública</b> .....	9
1.1. Fundamentar la reutilización de componentes .....	10
1.2. Garantizar la independencia tecnológica de los sistemas internos de información .....	11
1.3. Seguridad de los sistemas de información .....	12
1.4. La interoperabilidad. Un objetivo estratégico, un deber ético ....	13
<b>2. Principales argumentos para liberar el software producido por las administraciones públicas</b> .....	15
2.1. El software de la Administración pública: un bien público .....	15
2.2. La reutilización de componentes fuera de la propia administración .....	16
2.3. Interoperabilidad con los administrados .....	17
<b>3. Liberación del software</b> .....	19
3.1. Pliegos y contratación de desarrollos de software .....	19
3.2. Derechos de autoría y explotación del software de la administración .....	21
3.3. Licencias de software libre .....	22
3.4. Repositorios y forjas de desarrollo .....	23
<b>Resumen</b> .....	25
<b>Bibliografía</b> .....	27



## Introducción

Los beneficios que aporta el software libre a la sociedad de la información desde el punto de vista de una administración pública han hecho que muchos gobiernos de todo el mundo hayan emprendido políticas de impulso y promoción del uso del software libre en sus sociedades, pero este software no sólo supone una oportunidad para llevar a cabo políticas que ayuden a encarar la división digital de la población. Muchas organizaciones grandes, ya sean empresas, organizaciones no gubernamentales o administraciones públicas, han visto en el software libre una apuesta interesante de cara a emprender de forma más eficiente la adopción de las tecnologías de la información y la comunicación en sus propias organizaciones.

Muy a menudo se habla de la *neutralidad tecnológica* de la administración en cuanto a las TIC para referirse a la necesidad de que ésta escoja sus soluciones tecnológicas según lo que más se adecue en cada caso y no por principios generales, por ejemplo una apuesta estratégica por el software libre. El concepto es especialmente curioso si tenemos en cuenta que las administraciones públicas no suelen ser neutrales en ningún otro ámbito de la tecnología –se decantan o impulsan determinadas energías antes que otras, se financian determinados ámbitos de investigación e incluso se prohíben otros, se apoyan determinados sistemas de transporte delante de otros, etc–.

En este segundo módulo de la asignatura *El software libre en el sector público*, nos centraremos en el papel de la Administración pública como receptora de proyectos internos de software libre. Este papel tiene una primera vertiente obvia en la capacidad de argumentar la adopción de este tipo de software, incluso en aquellos casos en que, inicialmente, no sea la opción que ofrezca mejores condiciones, ya sean funcionales o económicas.

Pero hay otros aspectos que hay que tener presentes. Las administraciones públicas de una cierta dimensión son grandes productoras de códigos que, potencialmente, se podrían reutilizar en otros proyectos internos, o bien en proyectos de otras administraciones. A menudo, sin embargo, se hace difícil y costoso liberar una aplicación de software que no se ha desarrollado inicialmente teniendo en cuenta esta posibilidad, y, además, muchas veces eso también supone una dificultad añadida para los que quieran reutilizar totalmente o en parte esta aplicación, ya que no se suele disponer del histórico de versiones, el seguimiento del mantenimiento correctivo y evolutivo (*bug tracking*), ni en muchos casos la documentación actualizada de la última versión del código.

Todo nos lleva a ver por qué motivos es justificable que la Administración pública asuma, como parte de sus proyectos TIC, la liberación de los desarrollos que efectúa, y cómo tendría que llevarse a cabo este proceso para que finalmente le suponga un beneficio.

## Objetivos

Con el estudio de este módulo, el estudiante tiene que satisfacer los siguientes objetivos:

- 1.** Analizar planes concretos de impulso del software libre en la administración.
- 2.** Analizar el alcance de los planes mencionados y principales implicaciones dentro de la propia idiosincrasia de cada administración pública: normativa, contratación, gestión del cambio, etc.
- 3.** Establecer un análisis de oportunidades, ventajas y detección, en su caso, de las principales carencias respecto a este tipo de iniciativas desarrolladas dentro de una institución pública.



## 1. Principales argumentos para el uso de software libre en los sistemas de una administración pública

Las administraciones públicas son organismos de dimensiones muy diversas, con una disponibilidad de recursos bastante limitada y ajustada a las competencias que ejercen; sin embargo, el funcionamiento y las necesidades de la mayoría de las administraciones de una dimensión equivalente son similares.

Así pues, si nos referimos a las administraciones locales de municipios o de áreas con poca población, se puede decir que los tipos de trámites y los documentos que se producen y las necesidades del personal y las autoridades son extremadamente parecidos.

Naturalmente, a medida que analizamos administraciones de mayores dimensiones, con un mayor número de competencias y, por lo tanto, con un presupuesto más elevado, podemos observar que su funcionamiento y estructura son cada vez más complejos, por lo que es difícil encontrar los paralelismos. En cualquier caso, si tratamos de huir de la particularidad del procedimiento que acompaña a cada trámite, se puede ver que, conceptualmente, los trabajos que se hacen en las diferentes administraciones de una dimensión equivalente siguen siendo muy parecidos.

### Lecturas obligatorias

**Generalitat de Catalunya** (2008). Full de Ruta de Programari Lliure: resum executiu.

**IDABC** (2007). *The Netherlands in Open Connection. An action plan for the use of Open Standards and Open Source Software in the public and semi-public sector.*

**Ministerio de Administraciones Públicas (MAP)** (2005). *Propuesta de recomendaciones a la AGE sobre la utilización del SL y de fuentes abiertas.*

*Referencia de Migración para Software Libre del Gobierno Federal Brasileño* (2004).

Lógicamente, si las tareas y los requerimientos que tiene cada tipología de administración son tan similares, parecería lógico que los sistemas informáticos que utilizan las administraciones públicas como apoyo al desarrollo de su actividad fueran muy parecidos e incluso los mismos, con pequeñas adaptaciones a la forma de trabajar en cada caso. Ahora bien, aunque los responsables técnicos de las diferentes administraciones son conscientes de esto y, por lo tanto, de las muchas posibilidades de colaborar con otras administraciones en cuanto a las TIC, la verdad es que este tipo de colaboraciones son poco frecuentes y cuestan mucho de articular, muchas veces incluso entre los diferentes ministerios, consejerías o concejalías de una misma administración.

Si la adopción de las TIC en las administraciones públicas se hubiera realizado a partir de un plan perfectamente definido y diseñado, seguramente el proceso habría sido mucho más eficiente y se habría obtenido un rendimiento

mucho mayor de las inversiones realizadas, pero la realidad no es ésta. Cada administración ha ido adoptando estas tecnologías en función de los recursos disponibles en cada momento y de las necesidades más inmediatas, de manera que la informatización se ha realizado de forma progresiva, poco planificada y no muy ordenada. Este hecho ha llevado a menudo a entornos tecnológicos heterogéneos que tienen dificultades para interoperar y que, sobre todo, dificultan mucho la reutilización de aplicaciones.

La responsabilidad de la Administración pública hacia la ciudadanía establece unos requerimientos especiales con respecto a los sistemas de información que utiliza para prestar los servicios que le han sido encomendados. Por ejemplo, la sanidad, la educación o la seguridad son competencia de la Administración pública en muchos países y son servicios que mueven datos sensibles de los ciudadanos, y no pueden detenerse.

En este apartado se ven algunos argumentos que justifican la utilización del software libre en las administraciones públicas, pero también la adopción de las metodologías de desarrollo que rodean a los proyectos de software libre con estructuras TIC que dan soporte a estas organizaciones.

### **1.1. Fundamentar la reutilización de componentes**

Tal como se expone en la introducción de este apartado, los sistemas de información de las diferentes unidades o departamentos de una administración pública generalmente se han construido con soluciones heterogéneas y, además, muy a menudo incompatibles entre sí, hecho que dificulta la colaboración y la reutilización de soluciones ya existentes.

¿Cómo puede ayudar el software libre en todo esto? Pues por sí mismo y si lo planteamos únicamente desde un punto de vista tecnológico, el uso de software libre no puede aportar mucho más que el privativo para mejorar esta situación. Puede contribuir, de alguna manera, en la forma en la que se encaran los desarrollos de las nuevas aplicaciones, partiendo de lo que ya está hecho, colaborando a completarlo con lo que falta y a mejorarlo con lo que no está lo suficiente bien.

Así pues, como es evidente para todo el mundo que la reutilización de componentes es necesaria con el fin de conseguir una gestión más eficiente de las TIC y está claro que la situación de partida no es en absoluto la deseable, muchas administraciones se están replanteando la forma en que tienen que construir sus soluciones.

El primer paso para poder compartir y colaborar es disponer de entornos de construcción y ejecución de aplicaciones comunes. En esta línea, muchas organizaciones grandes, y las administraciones públicas en particular, han empezado a trabajar en los *frameworks* (entornos) de desarrollo, que además, muy a menudo, están contruidos sobre soluciones de software libre. El desarrollo

en este tipo de entornos se orienta hacia la construcción de componentes que van extendiendo la funcionalidad del *framework* y, por lo tanto, hacia la reutilización del mismo. Una vez llegados a este punto, si las diferentes áreas de una administración tienen una plataforma tecnológica común, ya empieza a ser posible la colaboración interna.

Sin embargo, ¿qué pasa con la colaboración entre las administraciones? A priori, volvemos a encontrarnos en la situación inicial, ya que nuevamente las diferentes administraciones tienen sistemas heterogéneos que hacen difícil la compartimentación de soluciones. Pero a pesar de esto, la realidad es que la construcción basada en componentes obliga a realizar ejercicios de abstracción que acaban facilitando la compartimentación, ya sea de los componentes por separado, o bien de las aplicaciones completas.

La reutilización de componentes no es o no tendría que ser, únicamente, un acto de generosidad del creador o del promotor de éstos hacia el resto de la sociedad y administraciones públicas. El software siempre se puede mejorar y evoluciona constantemente. Si un número mayor de organismos o entidades utilizan un determinado componente, la probabilidad de detectar errores o carencias aumenta; y la responsabilidad en su mantenimiento, ya sea correctivo como evolutivo, se distribuye entre los requerimientos y las necesidades de todos ellos, de manera que no sólo se consigue un ahorro por el reaprovechamiento de una pieza ya existente, sino también por una forma más eficiente de mantenerla.

## **1.2. Garantizar la independencia tecnológica de los sistemas internos de información**

Tal como se ha comentado, hoy en día la mayoría de organizaciones, y en particular las administraciones públicas, tienen una gran dependencia de sus sistemas de información. Algunos servicios básicos, como la sanidad, el transporte público, la policía o los bomberos, que a menudo tratan información altamente sensible de la ciudadanía y que prestan servicios públicos de especial relevancia, dejarían de funcionar si los sistemas de información que les dan soporte se detuvieran.

Algunos sistemas de información de una administración tienen que funcionar y evolucionar de forma continuada. Estos complejos sistemas dependen de toda una infraestructura básica formada por sistemas operativos, sistemas de gestión de bases de datos relacionales, servidores de aplicaciones y servidores web, entre otros componentes. Por otra parte, estos sistemas también dependen de los entornos de escritorio con que trabajan los usuarios que finalmente interactúan. Todos estos elementos constituyen los llamados entornos de ejecución de las aplicaciones, y normalmente son compartidos por diferentes sistemas y condicionan en gran medida la forma de construcción de las nuevas aplicaciones.

Así pues, si bien las administraciones públicas suelen tener bastante control sobre las aplicaciones que han desarrollado específicamente para sus sistemas de información, no se puede decir lo mismo de toda esta infraestructura *estándar* que hoy está formada por un entramado de soluciones privativas y libres, y provista y elaborada por diferentes fabricantes.

Pero a pesar de todo esto, es necesario que los sistemas funcionen y evolucionen con independencia de las decisiones que tomen los fabricantes de las soluciones sobre las que están contruidos. La independencia tecnológica de la administración se entiende como la capacidad de decidir la propia estrategia tecnológica a corto, medio y largo plazo, sin que estas decisiones sean sometidas necesariamente a las que tome una tercera entidad.

La única manera de conseguir esta independencia es teniendo el control de todo el entramado tecnológico sobre el que se construyen los sistemas de información; y no escapa a la atención de nadie que, hoy en día, la forma más eficiente de conseguir este control es construyéndolo a partir de productos ya existentes de software libre. Así pues, el hecho de construir las aplicaciones de la administración sobre una infraestructura basada en software libre es una solución muy eficaz para garantizar la independencia tecnológica de los sistemas internos de información.

Pero las administraciones públicas no empiezan a utilizar las TIC desde cero y el parque instalado ha de perdurar durante largos periodos que permitan amortizar la inversión ya realizada. Esto conlleva la necesidad de mantener entornos mixtos, donde se mezclan y tienen que interoperar soluciones libres, con aplicaciones de la propia administración y con soluciones privativas. Tal convivencia puede suponer incompatibilidades con otras piezas de software que dificulten, e incluso imposibiliten, a corto plazo la adopción de otras aplicaciones o entornos en algunas secciones o puestos de trabajo de la administración.

### **1.3. Seguridad de los sistemas de información**

Como ya hemos mencionado, los sistemas de información de las administraciones públicas manejan datos muy confidenciales de sus ciudadanos. Informaciones como el historial médico o policial de un ciudadano, por ejemplo, tendrían que encontrarse perfectamente protegidas, pero también tienen que ser fácilmente accesibles desde diferentes contextos a fin de que la administración pueda prestar sus servicios de forma eficiente. De ahí que la seguridad del software utilizado en determinados ámbitos de la administración sea un aspecto muy relevante en las arquitecturas y las soluciones escogidas.

Si bien las auditorías de seguridad del software se basan en una serie de puntos de control que en la mayoría de los casos no requieren el acceso al código fuente de las aplicaciones, es cierto que el acceso a éste es la única forma de conocer exactamente el funcionamiento de una aplicación. Tanto es así que

los fabricantes de software privativo a menudo ofrecen un acceso al código fuente de sus aplicaciones a las administraciones públicas para que puedan realizar las comprobaciones que crean conveniente. Pero el acceso al código de una aplicación o un entorno que tenga unos requerimientos especiales de seguridad no puede ser suficiente si no podemos garantizar que el código que se ejecuta en nuestro sistema es el que hemos estudiado; eso sólo es posible si se nos permite y se nos dan las instrucciones suficientes para compilarlo, situación que actualmente sólo encontramos garantizada con el software libre.

El acceso generalizado al código fuente de una aplicación se ha probado en muchos casos como una medida mucho más eficaz para detectar y resolver más rápidamente los posibles agujeros de seguridad del software, que el modelo de software privativo con un acceso restringido al código, lo que compensa con creces el hecho de que el estudio del código pueda facilitar sus debilidades a un potencial atacante. Esto, sin embargo, lleva de nuevo a la necesidad de una cantidad suficiente de usuarios y desarrolladores que muy pocas aplicaciones desarrolladas por las administraciones públicas pueden aspirar a conseguir. Así pues, si una administración pública liberara, por ejemplo, aplicaciones propias relevantes para la seguridad de sus datos, podría estar facilitando información muy valiosa a sus potenciales atacantes, sin la contrapartida de una cantidad suficiente de usuarios y desarrolladores para detectarlas y corregirlas en un breve espacio de tiempo. Por todo esto, seguramente no sea una buena política de seguridad liberar o compartir el software desarrollado por la propia administración que tenga que tratar con datos especialmente sensibles.

#### **1.4. La interoperabilidad. Un objetivo estratégico, un deber ético**

La comunicación de documentos electrónicos entre las administraciones públicas, las empresas y los ciudadanos, la llamada *e-administración* o administración electrónica, es cada vez más habitual; pero a pesar de ello, en muchos estados todavía no hay ninguna normativa que establezca de forma general los criterios para decidir los formatos en los que estos documentos se pueden o se tienen que presentar para que sean aceptados por la administración. Por eso, frecuentemente los formatos de los productos con más implantación en el mercado se convierten en *estándares* de hecho, aunque pueda suponer una discriminación para los ciudadanos usuarios de productos o sistemas no tan extendidos.

##### **Estándares**

Quando hablamos de un estándar nos referimos a una especificación técnica que ha sido aprobada y hecha pública por un organismo internacional reconocido de normalización. A menudo los llamados estándares de hecho no han sido aprobados por ninguno de estos organismos y, por lo tanto, no se pueden considerar estrictamente estándares, sino especificaciones técnicas.

Por otra parte, la Administración pública es responsable de velar por la integridad y durabilidad del archivo de los documentos electrónicos utilizados en el ejercicio de su actividad. Y en ningún caso sería deseable que esta responsabilidad se encontrara condicionada por los intereses comerciales del fabricante

##### **Lectura recomendada**

IDABC (2004). *European Interoperability Framework*.

de un producto concreto que pudiera tomar, unilateralmente, la decisión de abandonar el desarrollo del propio producto, o bien abandonar el soporte a un formato determinado de documento.

Definimos la interoperabilidad como la capacidad que tienen los sistemas de información para intercambiar y compartir datos. Los sistemas de información de las administraciones públicas tienen que interoperar habitualmente con otras administraciones y con los ciudadanos, entendidos éstos últimos en el sentido amplio que incluye tanto a personas como empresas u otras entidades. La interoperabilidad con las otras administraciones públicas es un objetivo estratégico general, ya que acaba facilitando en gran medida los trámites interadministrativos; por eso se está trabajando en este campo desde hace años. Pero con independencia de que los sistemas de información de una administración sean capaces o no de interoperar directamente con los de todos los ciudadanos o todas las otras administraciones, una administración pública tiene el deber ético de no discriminar a nadie por su elección tecnológica y, más especialmente, si la elección más sobradamente extendida es privativa y tiene un coste que no es despreciable.

El Marco Europeo de Interoperabilidad es un documento elaborado por la Comisión Europea que recoge un conjunto de recomendaciones para que las diferentes administraciones públicas de la Unión Europea construyan sus propias políticas de interoperabilidad. Entre las diversas recomendaciones que incluye el documento, se insta a las administraciones públicas a basar estas políticas en la utilización de estándares abiertos siempre que sea posible, sin perjuicio de los formatos que se estén utilizando actualmente.

Pero ¿qué aporta el software libre a la interoperabilidad? Pues en sí mismo, nuevamente, el software libre no aporta necesariamente nada a esta discusión. De hecho, muchas soluciones de software libre funcionan con formatos o protocolos cerrados, contruidos a partir de ingeniería inversa con el fin de poder interoperar con sistemas privativos. A pesar de esto, es cierto que, a diferencia de algunas soluciones de software privativo, las aplicaciones de software libre no hacen del *lock-in* tecnológico un argumento para mantener fieles a sus usuarios y que, por lo tanto, cuando un estándar abierto está disponible para una determinada funcionalidad, protocolo o formato, tiende a utilizar la especificación con implementaciones muy cuidadas.

### **Ingeniería inversa**

Llamamos ingeniería inversa al proceso de estudiar y analizar un sistema, formato o protocolo, con el objetivo de determinar parcial o totalmente la especificación técnica desde un conjunto discreto de casos. A partir de esta especificación inferida, se pretende reproducir el funcionamiento o interpretar el contenido.

### **Lock-in**

Con el término *lock-in* solemos referirnos al tipo de cautividad tecnológica que provoca la adopción de una determinada opción. El *lock-in* no es exclusivo de los sistemas privativos, pero los fabricantes de este software a menudo lo han utilizado para fidelizar a sus clientes.

## 2. Principales argumentos para liberar el software producido por las administraciones públicas

Las administraciones públicas producen grandes cantidades de código que muchas veces se podrían reaprovechar, ya sea para aplicaciones de otras administraciones o bien para usos de cariz más o menos general.

### Lectura obligatoria

*A sharing attitude: Programverket in Sweden (2007).*

Los objetivos que han movido a algunas administraciones a publicar su software con licencias libres suelen estar más motivados por consideraciones éticas relativas a la necesidad de poner a la disposición de toda la ciudadanía lo que se ha construido con fondos públicos, que por consideraciones más prácticas que impulsan la mayoría de los proyectos de software libre y constituyen la esencia de su modelo de desarrollo: la existencia de un mayor parque instalado, la atracción de desarrolladores y la creación de una comunidad en torno al producto. Tal hecho comporta que estas liberaciones, aunque puedan tener un cierto valor político, se acaben convirtiendo en poco provechosas para la sociedad, ya que muchas veces no se realizan en un marco suficientemente idóneo: falta de documentación, no se previó su reutilización, no existe información pública de los errores detectados y el proceso de revisión y corrección, etc.

En este apartado veremos los principales argumentos que justifican el hecho de que el software producido por las administraciones se haga público con licencias de software libre.

### 2.1. El software de la Administración pública: un bien público

Si partimos de la consideración de que todas las acciones que emprende una administración pública se llevan a cabo con fondos públicos, parece bastante coherente que los frutos o productos que se produzcan permanezcan a disposición de toda la ciudadanía. Algunas administraciones públicas consideran que este conocimiento sólo tendría que ser de libre disposición por parte de la ciudadanía del territorio que administran, ya que es la que de una forma indirecta ha pagado el desarrollo; pero dejando de lado este matiz, lo que esta consideración comporta es la necesidad de hacer una cesión generalizada de la explotación de una obra.

Siguiendo con esta argumentación, es lógico que el software que desarrolla una administración pública, como el resto de obras, esté disponible para toda la ciudadanía ante cualquier propósito, incluso comercial, y en este sentido se tendría que entender como un bien público.

Si bien existen muchas formas ceder los derechos de explotación de todo este conocimiento, en el campo del software el modelo que hasta el momento parece más eficiente es el de las licencias de software libre, ya que, aparte de ofrecer el uso y la explotación, velan por la forma como se tienen que distribuir las obras derivadas, requiriendo en muchas ocasiones que, aparte de preservar la autoría de la obra original, se mantenga la licencia original. De esta manera se garantiza que el conocimiento que se ha iniciado como un bien público continúe siéndolo junto con las contribuciones que otros terceros aporten.

Pero no olvidemos que otra de las responsabilidades de la Administración pública es velar por la eficiencia de los recursos de que dispone, y hay que tener presente que no siempre es sencillo, desde un punto de vista técnico o legal, realizar la cesión en las condiciones que establece una licencia de software libre, así que será muy importante el balance del coste/beneficio de cada caso antes de apostar por la liberación de una solución.

## **2.2. La reutilización de componentes fuera de la propia administración**

Hemos visto que la reutilización de componentes es uno de los principales argumentos para que la Administración pública adopte los modelos de desarrollo propios de los proyectos de software libre. Generalmente, el beneficio inmediato de la reutilización de componentes se identifica en los proyectos internos de diferentes departamentos de la propia organización, que se ahorran el desarrollo de partes del software que otros ya han elaborado; con lo cual se obtiene un mejor rendimiento de presupuesto de la administración. Sin embargo, la reutilización de estos componentes fuera del propio ámbito de la administración puede aportar beneficios añadidos, derivados de ampliar la base instalada.

En este sentido, aparte del deber ético de poner a disposición de los ciudadanos los desarrollos que la administración haya realizado con dinero público, hay un beneficio asociado a la liberación del software que –por lo menos teóricamente– permite reducir los costes de mantenimiento evolutivo y correctivo del software por el hecho de que los diferentes usuarios se hagan responsables.

Vamos a poner un ejemplo concreto. Supongamos que tenemos un componente de registro de entrada y salida. Fundamentalmente estos componentes introducen una firma temporal a los documentos y mantienen una relación ordenada de todo lo que se ha registrado. Suelen integrarse con sistemas de gestión documental o con tramitadores de expedientes, y son un tipo de componentes imprescindibles para cualquier aplicación de administración electrónica. No sólo una administración pública, sino también diferentes tipos de organizaciones, como las universidades, las entidades bancarias o los colegios profesionales, pueden beneficiarse de una herramienta de estas características. El hecho de poner a disposición de toda la sociedad un componente de este tipo, contribuirá fácilmente a que diferentes organizaciones lo tengan en

cuenta cuando se encuentren con una necesidad similar. Generalmente, la integración de este componente en un sistema diferente para el que fue diseñado requerirá de algún tipo de desarrollo, como por ejemplo, un conector con un gestor documental nuevo o con un motor de bases de datos diferente. El desarrollo de estos conectores contribuirá a extender las prestaciones del producto con una menor inversión global de recursos, ya que cada organización se ocupa sólo de aquellas partes que son necesarias para su configuración particular. Con respecto al mantenimiento correctivo, podemos decir que pasará más o menos lo mismo: si el componente lo utilizan diferentes organizaciones, es más sencillo que se detecten los errores del software y que, por lo tanto, se corrijan más rápidamente.

Así, la reutilización de los componentes de software que ha desarrollado una administración pública fuera de la propia administración ofrece una oportunidad relevante para mejorar las características del software, al distribuir el coste de estas mejoras entre todos los agentes que utilizan tales componentes.

### 2.3. Interoperabilidad con los administrados

En el apartado anterior se destaca la importancia de la interoperabilidad como objetivo estratégico para garantizar la perdurabilidad de la documentación que la administración utiliza en el ejercicio de sus funciones y como ésta tendría que ser una prioridad desde el punto de vista organizativo. Ahora bien, desde el punto de vista del impulso de una sociedad de la información no excluyente, la interoperabilidad con la ciudadanía se convierte en una necesidad al poner al alcance de toda la población los servicios de administración electrónica. No se debería aceptar que ningún ciudadano quedara excluido del acceso a estos servicios por la necesidad de adquirir la licencia de un software, especialmente si existe alguno equivalente que es libre o, como mínimo, gratuito.

Si, a modo de ejemplo, nos centramos en los formatos utilizados en el intercambio de documentos entre la Administración pública y la ciudadanía, tendría que quedar claro que los criterios para escoger unos u otros no pueden basarse exclusivamente en la mayor cuota de mercado, los llamados *estándares de hecho*. En muchos casos, la solución dominante es privativa, y eso podría suponer una discriminación para determinados ciudadanos a causa del coste de la licencia de este software, pero además podría condicionar la evolución del servicio en cuanto a las decisiones del fabricante de esta solución.

Así pues, la principal recomendación para las administraciones públicas (ved el Marco Europeo de Interoperabilidad) siempre pasa por elegir formatos basados en estándares abiertos cuando éstos estén disponibles. Se ha de considerar la cuota de mercado, ya sea ante la inexistencia de un estándar abierto, o bien como formato alternativo, cuando la solución basada en el estándar o estándares abiertos no sea lo suficientemente generalizada.

#### Lectura recomendada

IDABC (2004). *European Interoperability Framework*.

Cuando haya más de un estándar disponible (aunque sería conveniente que las administraciones públicas tuvieran la capacidad de tratarlos todos), a la hora de decantarse por uno u otro, se debería tener en cuenta aspectos como la disponibilidad de implementaciones libres de estos estándares, o bien la cuota de mercado de las soluciones que los tratan.

Por otra parte, la Administración pública tendría que utilizar siempre formatos basados en estándares abiertos para todos los documentos que ponga a disposición de la ciudadanía y, sólo si eso no fuera posible, ofrecer herramientas informáticas libres o gratuitas que puedan tratarlos.

Ligado al concepto de la interoperabilidad con la ciudadanía (aunque no esté totalmente relacionado), hemos de considerar la necesidad de respetar la independencia de los ciudadanos para escoger la solución tecnológica que crean más conveniente. En este sentido, las soluciones que las administraciones públicas implementen tendrían que ser desde el punto de vista tecnológico totalmente neutrales de cara al ciudadano y se deberían construir sobre estándares y protocolos abiertos y sobradamente extendidos, de manera que no discriminen a ningún usuario por motivo de su elección tecnológica. Si bien esto es relativamente sencillo, en muchos campos donde las soluciones basadas en tecnologías web ofrecen de manera natural este marco neutral, hay que tener presente otros ámbitos, como los contenidos multimedia de los medios de comunicación públicos o de los materiales educativos, en los que también habría que tener presentes estos principios, y tratar de evitar los formatos privativos.

La condición de que la relación de la administración con la ciudadanía no se podrá basar siempre en estándares abiertos, o bien que no siempre existirán implementaciones libres de los mismos de la suficiente calidad, la mejor forma de garantizar la interoperabilidad con los administrados sería poniendo a disposición de los ciudadanos el software que puedan necesitar para acceder a cualquier servicio de administración electrónica. Lógicamente, este software deberá ser gratuito, pero por otro lado también sería muy deseable que fuera libre para que quien lo crea conveniente pueda aprovechar las partes necesarias para desarrollar o integrar otras soluciones que se comuniquen con la administración de la misma manera, por ejemplo.

Así, imaginamos un software que permita a las empresas hacer las declaraciones de impuestos de forma telemática. Si las bibliotecas de este software se publican con una licencia libre lo bastante permisiva, cualquier desarrollador de aplicaciones de gestión empresarial las podría aprovechar para incluir directamente desde su aplicación la funcionalidad de presentación telemática de las declaraciones. Esto garantizaría la interoperabilidad con la administración, porque se estaría utilizando el mismo código que ésta ha desarrollado y facilitaría la realización de este trámite a todos los usuarios de esta aplicación, fuera privativa o libre.

#### Licencia LGPL

Es recomendable publicar este tipo de aplicaciones o bibliotecas con una licencia del tipo LGPL (*lesser general public license*, la licencia general pública menor), que mantiene la garantía de las cuatro libertades propias del software libre y al mismo tiempo permite su utilización como parte de un software privativo.

### 3. Liberación del software

En los apartados anteriores se recogen los argumentos que justifican que la administración utilice software libre en sus proyectos internos y que libere las aplicaciones que desarrolla con licencias de software libre. Una vez entendida la importancia de liberar el software, hay que tener presente la liberación en todo el procedimiento de construcción de una aplicación en la Administración pública. De esta manera, se podrán evitar situaciones que acaben impidiéndolo o haciéndolo muy complicado, o bien dificultando en gran medida el reaprovechamiento del código liberado.

En este apartado, se ven las consideraciones necesarias para todo el proceso de construcción de las aplicaciones de una administración pública a fin de que el código generado se pueda acabar publicando con una licencia libre y, por lo tanto, que sea susceptible de ser reutilizado con éxito por otra entidad.

El apartado se centra en los siguientes aspectos del procedimiento:

- El procedimiento de contratación en la Administración pública aplicado especialmente a la contratación de desarrollos de software.
- Los derechos de autoría y de explotación del software, concretamente en la transmisión a una administración pública.
- Las licencias de software libre centrado en los criterios que se deberán seguir para escoger una de ellas.
- Las forjas y los depósitos repositorios como entornos donde finalmente se efectúa la publicación del software.

#### 3.1. Pliegos y contratación de desarrollos de software

Si bien los proyectos en el ámbito TIC de una administración se crean de una manera muy similar a la que podemos encontrar en cualquier otra organización, como ya se comenta en esta documentación, el acto administrativo de la contratación tiene unos requerimientos de transparencia muy específicos que lo caracterizan de forma muy particular y, desgraciadamente, lo convierten en un proceso pesado.

El proceso de la contratación de cualquier servicio que requiera la administración sigue los siguientes pasos:

- Detección de una necesidad y de una solución a ésta.
- Obtención de la financiación.
- Preparación de los pliegos.
- Publicación de la licitación (concurso de proveedores).

#### Lectura obligatoria

Karsten Gerloff (2005). *Making public administration's software public: The Andalusian Software repository*. UNU-MERIT.

#### Lectura obligatoria

IDABC (2005). *London Borough of Camden: public procurement of open source software development*.

- Elaboración y presentación de las ofertas por parte de los proveedores de la administración.
- Valoración de las ofertas y adjudicación.
- Firma del contrato.

Este procedimiento se puede simplificar en muchos casos, especialmente cuando la cuantía de los servicios que se van a contratar no es elevada; en muchos países es incluso posible la contratación directa a un proveedor si el importe del servicio no supera un determinado valor. Sin embargo, este procedimiento es válido en general y permite comprender la complejidad de la contratación.

Uno de los puntos más importantes de todo el procedimiento es la elaboración del pliego, ya que, aparte de establecer las prescripciones técnicas que tendrá que cumplir el objeto del contrato, también se establecen las condiciones en las que será ejecutado. Los aspectos de un desarrollo que no se hayan previsto en esta primera fase, serán difíciles de introducir en las fases posteriores. Por este motivo, es recomendable que, antes de preparar el pliego para el desarrollo de una aplicación, se intente determinar si ésta será susceptible de ser liberada. Si se detecta esta posibilidad (que, por otra parte, en contadas ocasiones se podrá descartar desde el principio), es conveniente introducir algunas condiciones en el pliego que garanticen los siguientes aspectos:

- Los derechos de explotación de los productos resultantes serán propiedad de la Administración pública.
- La aplicación se desarrollará en una forja desde el principio (si la administración tiene una propia se puede requerir que se utilice; de lo contrario se puede proponer una pública).
- Todos los componentes de terceros que se utilicen tendrán una licencia libre.
- La aplicación y su documentación se podrán liberar con una licencia libre.

También es conveniente solicitar un estudio legal sobre las condiciones de liberación de los productos resultantes del contrato.

Con estas precauciones en el pliego, a la finalización de la ejecución de los trabajos se estará en condiciones idóneas para liberar el software desarrollado.

### 3.2. Derechos de autoría y explotación del software de la administración

En la mayor parte de países, el software se suele proteger con las normativas de propiedad intelectual, es decir, con los mismos términos que una obra literaria o una composición musical. Este tipo de protección es la que relacionamos con expresiones como: "todos los derechos reservados", que significan que sólo el autor de una obra puede autorizar determinados usos, tales como la copia o la modificación.

#### Formas de proteger el software

Si bien el software se protege con las normativas de propiedad intelectual, en algunos países también es posible utilizar los mecanismos de propiedad industrial, como las patentes, con el fin de proteger determinados algoritmos o funcionalidades.

Mientras que los fabricantes o autores de software privativo utilizan esta protección para reservar o restringir lo que se puede realizar con sus aplicaciones, el software libre aprovecha estos mecanismos para permitir explícitamente la copia, el estudio, la modificación o mejora y la distribución de estas versiones modificadas del software. Para que el marco en el que se realizan estas cesiones sea homogéneo y con el objetivo de simplificar el tratamiento, normalmente se tiende a utilizar los textos de licencias estándares sobradamente conocidas, como la GPL, con todas sus variantes.

Así, el creador de una obra obtiene la propiedad intelectual por el simple hecho de haberla creado, pero si tiene una relación laboral con una empresa y crea la obra en el marco de esta relación, se entiende que hay una presunta cesión de estos derechos hacia la empresa. Los derechos de explotación reservados exclusivamente al autor/a de un programa de ordenador son para realizar los siguientes actos:

- La reproducción total o parcial: la copia, a través de cualquier medio y en cualquier forma, sea permanente o transitoria.
- La distribución pública: la transmisión del programa a terceras personas, mediante la venta, el alquiler o cualquier otra forma. La doctrina entiende que este derecho incluye la comunicación pública, por ejemplo, la publicación de la obra en un servidor accesible desde Internet.
- Transformación: "traducción, adaptación, arreglo o cualquier otra transformación del programa".

Sólo el titular de estos derechos puede cederlos a terceros, ya sea de forma exclusiva, o bien con una licencia, por ejemplo de software libre. Dado que el software que se desarrolla desde las administraciones públicas no es elaborado por personal interno, es esencial que en el marco de cualquier contrato

#### Lectura recomendada

*Libro blanco: Gestión de la propiedad intelectual (2007).*

#### Web recomendada

Portal i-Legal  
<http://ilegal.cpl.upc.edu>

de desarrollo se requiera la cesión de todos los derechos de explotación de los trabajos realizados para que después puedan acabar siendo publicados con licencias de software libre.

### 3.3. Licencias de software libre

Existe una gran cantidad de licencias de software libre que, a pesar de garantizar las cuatro libertades, establecen diferentes mecanismos de redistribución de las obras derivadas.

A grandes rasgos, cuando una administración hace pública una aplicación con una licencia libre, tendría que escoger una licencia que garantizara que las modificaciones que contribuyan al código liberado se distribuyan con la misma licencia original, de manera que se preserve la inversión realizada y que ésta permanezca disponible para todo el mundo.

Muchos desarrolladores de software libre utilizan licencias que requieren que las obras derivadas mantengan la misma licencia que el original, tal como pasa con la licencia GPL, por ejemplo, que es una de las licencias más utilizadas. Ahora bien, la interpretación que se hace de la obra derivada establece que un nuevo desarrollo independiente, si depende de la obra original o de una parte de ésta, es una obra derivada<sup>1</sup>. Es decir, a modo de ejemplo, si desarrollamos una aplicación que depende de un sistema de gestión de bases de datos publicado con una licencia tipo GPL y la queremos publicar como software libre, nos encontraríamos con que la aplicación desarrollada se tendría que publicar también con una licencia GPL si el autor nos autorizara de forma expresa a hacerlo<sup>2</sup>.

<sup>(1)</sup>Ved la cláusula 2.b) de la licencia GPLv2.

<sup>(2)</sup>Ved la cláusula 10 de la licencia GPLv2.

Actualmente, el proceso de desarrollo de una aplicación informática se aborda a partir de una serie de componentes ya existentes y sobradamente probados, que ofrecen gran parte de la funcionalidad requerida: gestores de bases de datos, servidores de aplicaciones, servidores de páginas web, bibliotecas y entornos (*frameworks*) de desarrollo, entre otros. Todos estos productos se tienen que integrar con el conjunto de reglas de negocio concretas de una aplicación para que se construya la solución definitiva. En algunos casos el proceso requiere de la creación de alguna pieza nueva o de la modificación de alguna ya existente. Muy a menudo estos componentes son de software libre.

Así pues, en buena parte, la construcción de software consiste en juntar un conjunto de piezas. Si este puzle se piensa con una aplicación a medida orientada exclusivamente al uso interno, la mayoría de las consideraciones en torno a los componentes que lo componen o forman parte del mismo se pueden pasar por alto, ya que las principales restricciones que imponen las licencias se refieren a las obras derivadas y a su redistribución. En la medida en que las modificaciones no se redistribuyan, se suele ser muy flexible y tolerante con

<sup>(3)</sup>Ved la cláusula 2 de la licencia GPLv2.

la forma como se hagan y las implicaciones que comporten<sup>3</sup>. Si hablamos de liberar software desde una administración pública, queda claro que el escenario es totalmente diferente.

Si queremos liberar, como uno todo, una aplicación que está formada por elementos con diferentes licencias de software libre, podría suceder que las licencias de los diferentes componentes lo hicieran muy complejo o que, simplemente, no lo permitieran para cláusulas como la que incorpora la GNU/GPL, que requiere que todas las obras derivadas mantengan la misma licencia.

Por todos estos motivos, desde las fases iniciales de un proyecto es necesario prever la posibilidad de liberarlo, porque de lo contrario podemos llegar a situaciones en que eso no sea posible o sea muy costoso.

### 3.4. Repositorios y forjas de desarrollo

Una vez que una administración pública ha hecho un desarrollo, ha previsto su liberación y, finalmente, se dispone a materializarlo publicando el código elaborado, surge la cuestión: ¿en qué lugar tengo que poner esta aplicación para que otros potenciales usuarios la encuentren? De hecho, ésta es una cuestión general que podemos aplicar a cualquier contenido que se pretenda difundir por la red. En el caso de software libre, lo más habitual es depositarlo en lo que llamamos un repositorio. ¿Y qué es un repositorio de software? Es un sistema informático donde se almacena información estructurada con el objetivo de facilitar el acceso.

En general, dado que hay muchos repositorios, será mejor publicar las aplicaciones en aquellos que sean susceptibles de atraer mejor al público objeto de la aplicación. Como es de prever que los usuarios más interesados en el software desarrollado en una administración pública puedan ser de otras administraciones, se trata de publicar estas aplicaciones en repositorios que incluyan otras aplicaciones dirigidas a la administración, ya que de esta manera será más sencillo que alguien las encuentre.

Si se publica una aplicación como software libre, lo que se suele pretender es que otros lo utilicen y que de entre éstos surja una comunidad. Uno de los principales beneficios que se han identificado en la liberación de software es la posibilidad de compartir las tareas de detección, corrección de errores y de evolución del código con la mejora o incorporación de funcionalidades. Las forjas son plataformas que facilitan el desarrollo colaborativo de software en la red. Los principales servicios que ofrece una forja son:

- Los sistemas de gestión de usuarios y permisos.
- Los sistemas de control de versiones (tipo *CVS* o *subversion*).
- Las herramientas de publicación de ficheros y documentos.
- Los sistemas de seguimiento de incidencias o errores (*bug tracking*).

#### Lectura recomendada

CENATIC (2009). *Forjas: entornos de desarrollo colaborativo. Su integración en el ámbito empresarial.*

#### Webs recomendadas

<http://www.osor.eu>  
<http://sourceforge.net>

- Las herramientas para la elaboración colaborativa de documentación (tipo *wiki*).
- Un espacio web.

A menudo se habla de forjas y repositorios como si fueran una misma cosa; y de hecho la confusión es bastante lógica, porque muchas veces las forjas hacen también el papel de repositorios. La clasificación de un portal como el SourceForge como un repositorio o como forja depende fundamentalmente del uso que se haga en cada momento: para el usuario que busca una aplicación hará claramente la función de repositorio, mientras que para el desarrollador de una de las aplicaciones hospedadas ofrecerá el entorno de trabajo donde colabora en la elaboración de esta aplicación, y, por lo tanto, hará el papel de forja.

El principal inconveniente de esta aproximación –que por otra parte es la más habitual en la mayoría de proyectos de software libre– está en que quien crea el proyecto tiene unas inquietudes y un perfil que, normalmente, hacen difícil que se ponga en el lugar del que después lo buscará. Así que, aunque para que se acepte un proyecto en una forja pública sea necesario rellenar un formulario con unos datos básicos que lo describan y lo categoricen, muchas veces la información incluida es poco detallada, porque la persona que introduce esta información no se pone en el lugar del que la buscará.

## Resumen

Muchas administraciones públicas han visto en el software libre una apuesta interesante para emprender de forma más eficiente la adopción de las tecnologías de la información y la comunicación en sus propias organizaciones.

Aparte del evidente ahorro en licencias que puede suponer la utilización de determinadas piezas de software, se identifican los siguientes argumentos para utilizar software libre en una administración pública:

- La reutilización de componentes.
- La independencia tecnológica de la administración.
- La seguridad de los sistemas de información.
- La interoperabilidad.

Ahora bien, las administraciones públicas de una cierta dimensión, además de grandes consumidoras de software, también son grandes productoras. Dado que las necesidades de las diferentes administraciones son muy similares, la probabilidad de que el desarrollo realizado por una administración sea útil para otra es bastante elevada. Si bien este tipo de colaboraciones entre administraciones hace tiempo que se realizan por medio de cesiones de uso, se identifican diferentes argumentos que hacen interesante que el software se comparta por medio de licencias libres:

- El software de la administración como un bien público.
- La reutilización de componentes o de software fuera del ámbito de la administración.
- La interoperabilidad con los administrados.

Las condiciones en las que se desarrolla el software que se produce en las administraciones públicas hacen recomendable una serie de precauciones en los procedimientos de contratación que garanticen que, una vez desarrollada, éste se pueda liberar con garantías. Llegados a este punto, con una solución de software acabada y lista para ser publicada con una licencia libre, también se ha visto cuáles son los entornos más adecuados para publicar el software liberado por las administraciones, de manera que tenga visibilidad y se facilite su futuro desarrollo en comunidad.



## Bibliografía

*A sharing attitude: Programverket in Sweden.* (2007). <<http://ec.europa.eu/idabc/en/document/7315>>

[Consulta: noviembre 2009].

**CENATIC** (2009). *Forjas: entornos de desarrollo colaborativo. Su integración en el ámbito empresarial.* <[http://observatorio.cenatic.es/index.php?option=com\\_rubberdoc&view=doc&id=65&format=raw](http://observatorio.cenatic.es/index.php?option=com_rubberdoc&view=doc&id=65&format=raw)>

[Consulta: noviembre 2009].

**Generalitat de Catalunya** (2008). *Full de Ruta de Programari Lliure: resum executiu.* <<http://www20.gencat.cat/docs/societat-informacio/Documents/Societat%20de%20la%20Informacio/Arxius/P060003-FRPL-PD4-RES-Resultats%20FRPL.pdf>>

[Consulta: noviembre 2009].

**Gerloff, Karsten**, UNU-MERIT (2005). *Making public administration's software public: The Andalusian Software repository.* <[http://www.osor.eu/case-studies-and-idabc-studies/case\\_studies/docs/andalusia-floss-1-pdf](http://www.osor.eu/case-studies-and-idabc-studies/case_studies/docs/andalusia-floss-1-pdf)>

[Consulta: noviembre 2009].

*Guide to Open Source Software for Australian Government Agencies.* (2005). <[http://www.finance.gov.au/publications/guide-to-open-source-software/docs/A\\_Guide\\_to\\_Open\\_Source\\_Software.pdf](http://www.finance.gov.au/publications/guide-to-open-source-software/docs/A_Guide_to_Open_Source_Software.pdf)>

[Consulta: noviembre de 2009]

**IDABC** (2004). *European Interoperability Framework.* <<http://ec.europa.eu/idabc/servlets/Doc?id=19529>>

[Consulta: noviembre 2009].

**IDABC** (2005). *London Borough of Camden: public procurement of open source software development.* <[http://www.osor.eu/case-studies-and-idabc-studies/case\\_studies/docs/london-borough-of-camden/](http://www.osor.eu/case-studies-and-idabc-studies/case_studies/docs/london-borough-of-camden/)>

[Consulta: noviembre 2009].

**IDABC** (2007). *The Netherlands in Open Connection. An action plan for the use of Open Standards and Open Source Software in the public and semi-public sector.* <<http://www.ez.nl/dsresource?objectid=154648&type=PDF>>

[Consulta: noviembre 2009].

*Libro blanco: Gestión de la propiedad intelectual* (2007). <<http://projectes.lafarga.cat/frs/download.php/811/GestioEntornLegal.pdf>>

[Consulta: noviembre 2009].

**Ministerio de Administraciones Públicas (MAP)** (2005). *Propuesta de recomendaciones a la AGE sobre la utilización del SL y de fuentes abiertas.* <<http://www.csi.map.es/csi/pg5s44.htm>>

[Consulta: noviembre 2009].

*Referencia de Migración para Software Libre del Gobierno Federal Brasileño.* (2004). <<http://www.governoeletronico.gov.br/anexos/versao-em-espanhol-do-guia-livre>>

[Consulta: noviembre 2009].



# Estudio de casos reales de migración de administraciones públicas

Antonio Rodil Garrido

PID\_00154679



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



# Índice

<b>Introducción</b> .....	5
<b>Objetivos</b> .....	7
<b>1. Fases de una migración a software libre</b> .....	9
<b>2. Casos de estudio</b> .....	13
2.1. Extremadura .....	15
2.2. El ayuntamiento de Munich .....	15
2.3. El ayuntamiento de Zaragoza .....	16
2.4. El Ministerio de Exteriores alemán .....	17
2.5. El Gobierno de Malasia .....	17
2.6. Brasil .....	18
2.7. Venezuela .....	19
2.8. Bolzano .....	19
<b>Resumen</b> .....	21
<b>Bibliografía</b> .....	23



## Introducción

En este módulo se estudian y analizan algunos de los planes de migración a software libre más conocidos, que han sido aplicados en alguna administración pública.

En el contexto de estos materiales, vamos a hacer el análisis de la migración en el ámbito de las administraciones públicas, ya sean nacionales, regionales o locales.

Si bien muchos consejos son aplicables a cualquier ámbito, no vamos a profundizar en los aspectos técnicos, los cuales varían muy rápidamente y dependen en muchos casos de la infraestructura existente, sino que vamos a hacer énfasis en lo metodológico y en las opciones disponibles para encarar la migración.

En todo caso, siempre será conveniente contar con ayuda de personal técnico que, en buena medida, puede surgir de los integrantes de la misma comunidad de software libre.

Si la Administración pública que usa una aplicación informática no tiene libertad de contratación en lo que se refiere a ampliaciones y correcciones del sistema, se produce una dependencia tecnológica en la que el proveedor está en condiciones de dictar unilateralmente términos, plazos y precios.

La forma en la que se almacena la información es muy importante, ya que si una aplicación utiliza un formato de almacenamiento abierto, el usuario siempre estará seguro de que en el futuro podrá seguir accediendo a los datos. Por el contrario, si éstos se almacenan en un formato cerrado y secreto, los usuarios dependerán de un determinado proveedor, que es el único que puede ofrecer alguna garantía de acceso a ellos.

Por lo tanto, es altamente recomendable que las administraciones públicas utilicen software libre y que los estándares que usen sean abiertos.

Una migración a software libre tiene unos costes elevados, ya que requiere tomar decisiones, mano de obra para implementar el cambio, conversión de datos, formación del personal, desarrollo y tiempo. Éstas son inversiones fijas que se pagan una vez.

El software propietario también tiene sus costes fijos, además de las actualizaciones permanentes, pérdida de interoperabilidad, mantenimiento y el coste extremadamente elevado que tiene para las administraciones públicas la pérdida de las libertades que le garantizan el control de su propia información. Estos costes son permanentes y crecientes a lo largo del tiempo.

Migrar a software libre es un coste a corto plazo, pero un ahorro enorme a largo plazo y, además, produce más y mayores beneficios.

## Objetivos

Los objetivos que se deben alcanzar al finalizar este módulo son los siguientes:

- 1.** Conocer y analizar algunos de los planes de migración a software libre más conocidos y que hayan sido implementados en las administraciones públicas.
- 2.** Identificar cuáles son los rasgos contextuales más importantes que tienen que ver con los planes de migración, más allá de los elementos tecnológicos: análisis previo, formación, estímulo y apoyo, recursos, gestión del cambio, establecimientos de plazos adecuados, etc.
- 3.** Establecer un análisis de oportunidades, ventajas y detección de posibles carencias respecto de este tipo de iniciativas desarrolladas dentro de una institución pública.



## 1. Fases de una migración a software libre

Lo ideal sería partir de cero utilizando software libre, pero si estamos hablando de migración, ése no es el caso. Es decir, vamos a partir de una situación en la cual utilizamos software privado y deseamos pasar a software libre.

Cada proyecto de migración, sea cual sea su tipología, debe realizarse con las máximas precauciones, tomando medidas de seguridad y verificando el proceso en todo momento. Esto conlleva una serie de problemas inherentes a cualquier migración (inclusive de una versión de software privado a otra), otros inherentes a la migración de software privativo a software libre, que presenta algunas características especiales y por último, en el caso de una migración de una versión de software libre a otra.

### **Ejemplo**

Una migración de Debian 4.0 ("etch") a Debian 5.0 ("lenny").

La migración de todo software propietario a software libre debe realizarse con un mayor control, medidas de seguridad y puntos de verificación y pruebas.

Cualquier proyecto de ingeniería debe realizarse por fases y siguiendo un orden lógico y cronológico, aunque en función del proyecto a realizar, algunas fases pueden ejecutarse en paralelo.

Una migración a software libre es considerada un proyecto de ingeniería y debe ser ejecutado por personal especializado en la materia. Si la migración se ejecuta por una empresa externa, como suele ocurrir en las administraciones públicas, el personal interno debe participar activamente en todo el proceso, con la finalidad de que exista una transferencia fluida y gradual del conocimiento.

En la asignatura *Implantación de sistemas de software libre*, se identifican siete fases o etapas en los proyectos de implantación de estos sistemas, que son las siguientes:

- 1) Estudio de la situación actual.
- 2) Estudio de los requisitos de la implantación.
- 3) Análisis de las soluciones en software libre.
- 4) Formalización de la propuesta.

5) Desarrollo.

6) Implantación y migración.

7) Formación, documentación y soporte al usuario.

Sin ánimo de exhaustividad, las fases de un proyecto de migración a software libre son las siguientes:

- 1) Justificación.
- 2) Concienciación.
- 3) Diseño de la migración.
- 4) Ejecución y evaluación de un plan piloto.
- 5) Ejecución del plan de migración.

Las siete fases identificadas para los proyectos de implantación son aplicables a las fases 4 y 5 de los proyectos de migración.

A continuación se describe cada una de las fases de un proyecto de migración.

### 1) Fase 1. Justificación

Consiste en justificar el porqué de la migración. Se efectuará un análisis de costos de la migración y se proyectarán los ahorros factibles, con el paso del tiempo, en pago de licencias.

### 2) Fase 2. Concienciación

La participación de los usuarios es fundamental en todo proceso de migración. Una migración tendrá éxito si se cuenta con defensores del cambio dentro del equipo de trabajo y el apoyo de los usuarios del sistema.

Un instrumento ideal para conseguir la concienciación de todo el personal es la realización de charlas explicativas informando sobre el porqué del cambio y de los beneficios esperados.

### 3) Fase 3. Diseño de la migración

Esta fase consiste en la definición del proyecto y la recogida de información, con indicación de las condiciones iniciales y finales y el proceso a seguir para llegar de unas a otras.

Se elaborará un diagnóstico de la situación de partida o escenario, que incluirá la arquitectura del sistema, configuración de la red, hardware y software utilizado. Además, será necesario considerar los requisitos de los usuarios del sistema, como los conocimientos de informática, la necesidad de capacitación, etc.

#### Lectura recomendada

Amadeu Albós Raya; Óscar David Sánchez Jiménez; David Megías Jiménez (coord.) (2008). *Implantación de sistemas de software libre* (mód. 1, apdo. 2, máster Software libre). UOC.

En las páginas 21 y 22 del capítulo 5, de las *Directrices IDA de migración a software de fuentes abiertas* se establece una interesante lista de datos a recopilar de cada aplicación que se desea migrar, los requisitos en cuanto a los datos y los de seguridad.

**Lectura obligatoria**

Ministerio de Administraciones Públicas (2003). *Directrices IDA de migración a software de fuentes abiertas*.

#### 4) Fase 4. Ejecución y evaluación de un plan piloto

Se elaborará un plan piloto de migración, que se basará en la información recogida en la fase anterior.

Los datos obtenidos indicarán si el plan y la justificación funcionan.

#### 5) Fase 5. Ejecución del plan de migración

Se describirá el plan de migración, teniendo en cuenta la forma de migración por la que vayamos a optar. Las posibilidades son las siguientes:

**a) En varias fases.** Es más simple y segura, aunque también más lenta. Se comenzaría utilizando software libre en el mismo sistema operativo, dejando la migración de este último para el final del proceso.

Un buen ejemplo de lo anterior está en el área de ofimática, en la que el paquete de software libre OpenOffice.org ofrece las mismas funciones que sus equivalentes privativos, al tiempo que sustituye los formatos cerrados (Microsoft Word, Microsoft Excel, Microsoft PowerPoint) por el formato abierto y estándar OpenDocument Format (ODF).

Podríamos seguir con el cliente de correo electrónico, utilizando, por ejemplo, Evolution.

Una de las principales ventajas de esta forma de migración es que se pueden ir pasando de los formatos cerrados (por ejemplo, los archivos de Microsoft Word) a formatos abiertos (como por ejemplo OpenDocument Format) en forma progresiva y sin cambiar el entorno operativo.

Una vez que los usuarios se hayan familiarizado con las aplicaciones libres de uso cotidiano, se puede comenzar la migración del sistema operativo. Eso se puede hacer en forma gradual máquina a máquina y/o por departamentos.

**b) En una única fase.** Es la opción más rápida pero el riesgo es máximo y se necesita personal muy cualificado y convencido de poder realizar el cambio. Se requiere una mayor planificación.

En este caso, se sustituye el sistema operativo y todas las aplicaciones por software libre.

Es fundamental, de acuerdo al tamaño de la organización, contar con un equipo técnico que pueda ofrecer asesoramiento a los usuarios después de la migración, responder a las consultas y solucionar problemas que puedan surgir. También es importante la formación de capacitación posterior a la migración, pero, realmente, la formación debería hacerse en paralelo con el resto de las fases.

La migración de un sistema operativo privativo a otro de software libre debe hacerse con la mayor seguridad, ya que se trata de la base de cualquier sistema informático. La *Guía de Migración a Linux Desktop* trata de exponer las diferentes consideraciones y estrategias que hace falta tener en cuenta ante un proyecto de migración del software de las estaciones de trabajo de una organización a GNU/Linux.

La tercera parte de la citada guía (capítulos 6 a 9, ambos inclusive) es de especial importancia, ya que nos desvela las diferentes fases para llevar a cabo un plan piloto de una migración de estas características.

#### Lectura obligatoria

Chris Almond, Jeroen van Hoof; Nick Lassonde; Ben Li; Kurt Taylor (2006). *Linux Client Migration Cookbook, Version 2. A Practical Planning and Implementation. Guide for Migrating to Desktop Linux.*

<http://www.redbooks.ibm.com/redbooks/pdfs/sg246380.pdf>

## 2. Casos de estudio

El plan de acción eEurope 2002 "Una sociedad de la información para todos" fue ratificado por los jefes de Estado y de Gobierno en junio del 2000. El objetivo de este plan es introducir plenamente a Europa en la era digital. El plan destaca como prioritario el nivel de información de las administraciones públicas.

El proceso de migración, en las administraciones públicas, de software propietario a software libre se promueve en Europa por el plan eEurope 2002.

El plan eEurope 2002 y la Comisión Europea destacan cuatro criterios básicos para la migración a software libre. Éstos son: dependencia, coste, seguridad y transparencia.

Sin embargo, algunos informes nacionales dejan de lado el cuarto criterio, la transparencia:

- El Gobierno Federal Alemán incluye la dependencia del proveedor, el coste y la seguridad.
- El Reino Unido recomienda el control de la dependencia del proveedor con el uso de estándares abiertos, evalúa el coste en función del *total cost of ownership* (TCO) y se mantiene neutral en lo referente a la seguridad.

Todos los informes nacionales publicados a partir del plan eEurope 2002 aconsejan el uso del software libre en la Administración pública o al menos, lo consideran una muy buena alternativa al software propietario.

El plan eEurope 2005 sucedió al de 2002. Uno de los objetivos principales del plan eEurope 2005 fue conseguir que todas las administraciones públicas contasen con conexiones de banda ancha en igualdad de condiciones.

El programa *Interchange of Data between Administrations* (IDA) tiene por objeto facilitar el desarrollo y la aplicación operativa de redes telemáticas transeuropeas de intercambios de datos entre las administraciones de los Estados miembros y las instituciones europeas.

El citado programa ha publicado unas directrices de migración a software libre cuyos objetivos son los siguientes:

- Ayudar a decidir si debe abordarse el proyecto de migración a software libre.

### Lectura recomendada

Comisión de las Comunidades Europeas (2000). *eEurope 2002. Una sociedad de la información para todos. Plan de acción.*

<http://www.ub.edu/integracio/Lleis/UE/eEUROPE%202000.pdf>

### Lectura recomendada

Comisión de las Comunidades Europeas. (2002). *eEurope 2005. Una sociedad de la información para todos. Plan de acción.*

[http://ec.europa.eu/information\\_society/eeurope/2002/news\\_library/documents/eeurope2005/eeurope2005\\_es.pdf](http://ec.europa.eu/information_society/eeurope/2002/news_library/documents/eeurope2005/eeurope2005_es.pdf)

**Lectura recomendada**

Ministerio de Administraciones Públicas (2003). *Directrices IDA de migración a software de fuentes abiertas.*

- Describir, en términos técnicos, cómo llevar a cabo la migración de sistemas de información del sector público a software libre.

Estas directrices se dirigen a los gestores y profesionales de las tecnologías de la información de las administraciones públicas europeas. Su desarrollo ha sido guiado por un equipo de expertos del sector público de Alemania, Dinamarca, España, Finlandia, Italia, Malta, Países Bajos, Suecia y Turquía. El Ministerio de Administraciones Públicas ha elaborado la traducción a la lengua española.

Este documento contiene lo siguiente:

- En primer lugar, elementos de apoyo a la decisión como razones para la migración, recomendaciones de carácter general, orientación metodológica, descripción de los elementos claves organizativos y técnicos de la migración.
- En segundo lugar, recomendaciones detalladas en relación con la migración de elementos funcionales tales como ofimática, correo electrónico, trabajo en grupo, servicios web, gestión de documentos, bases de datos y otros elementos subsidiarios, como sistemas operativos, servidores de ficheros, gestión de autenticación y autorización de usuarios, antivirus y antispam, recuperación y copias de seguridad y gestión de impresión.
- En tercer lugar, un modelo para la comparación de costes entre instalaciones basadas en software propietario o en software de libre.

A continuación se analizan algunos casos reales de administraciones públicas que han tenido una importante relevancia en la migración a software libre.

**Lectura obligatoria**

Ministerio de Administraciones Públicas. (2003). *Directrices IDA de migración a software de fuentes abiertas.*

## 2.1. Extremadura

La Junta de Extremadura fue la primera administración pública europea en poner en marcha una distribución de software libre. Lo hizo con su distribución gnuLinEx en el sistema educativo de su comunidad.

Debido a su carácter pionero, fueron muchas las dificultades encontradas en el camino, fundamentalmente por la falta de información en migraciones similares.

El Consejo de Gobierno de la Junta de Extremadura aprobó que la información electrónica que generen e intercambien los distintos órganos de su administración pública utilizará obligatoriamente formatos estándar. Para la elaboración y proceso administrativo utilizan OpenOffice.org sobre la Norma ISO/IEC DIS 26300 y para la información que garantice la inalterabilidad de visualización, PDF sobre la Norma ISO 19005-1:2005.

Esta decisión fue considerada por organismos internacionales como el paso más importante para favorecer la innovación tecnológica, reducir la dependencia de usuarios, empresas y administraciones públicas respecto a aplicaciones propietarias no compatibles e incrementar la interoperabilidad entre sistemas y aplicaciones a nivel global.

## 2.2. El ayuntamiento de Munich

Ante un anuncio de Microsoft, en diciembre del 2002, donde se informaba de la suspensión del soporte de Windows NT Server 4.0, los responsables del Ayuntamiento de Munich, en Alemania, acordaron la migración a software libre, por lo que desarrollaron uno de los planes de migración más consistentes conocidos hasta la fecha.

El caso del Ayuntamiento de Munich hay que estudiarlo junto al de Newham, en Reino Unido. Se trata de dos casos de administraciones públicas que, ante un mismo problema, optaron por tomar caminos opuestos. En el caso de Newham, sus máximos responsables decidieron aceptar un ventajoso acuerdo con Microsoft.

Anteriormente se han comentado los cuatro criterios básicos de una migración a software libre y es curioso observar que ni en el caso de Munich ni en el de Newham se han tenido en cuenta los criterios de seguridad y transparencia, dando prioridad al coste y a la independencia del proveedor.

En Munich, el coste inicial de la migración no fue un obstáculo y el criterio que más peso específico tuvo fue la independencia del proveedor. En cambio, en el caso de Newham, el criterio más importante fue el coste.

### Lectura recomendada

Marcelo D'Elia Branco; Mónica León Martínez; Alejandro Novo López; Alberto Otero García (2005). *Primera unidad del segundo módulo de la asignatura obligatoria del Máster de Software Libre de la UOC, Implantación de sistemas de software libre.*

[http://cv.uoc.es/cdocent/G\\_FEEP6\\_97BP8H74TAJ3.pdf](http://cv.uoc.es/cdocent/G_FEEP6_97BP8H74TAJ3.pdf)

### Lectura recomendada

ISO (2006). *Information technology – Open Document Format for Office Applications (OpenDocument) v1.0.*

### Lectura obligatoria

Marcelo d'Elia Branco (2006). *Munich: la implantación del software libre avanza.*

[http://wiki.softwarelivre.org/pub/Blogs/BlogPostMarceloBranco20061201174131/Munich\\_informe\\_es.pdf](http://wiki.softwarelivre.org/pub/Blogs/BlogPostMarceloBranco20061201174131/Munich_informe_es.pdf)

### Lectura obligatoria

Daniel Reina (2006). *Criterios de migración a Linux en las administraciones locales de la UE. Los casos de Munich y Newham.*

<http://www.uoc.edu/uocpapers/2/dt/esp/reina.pdf>

### 2.3. El ayuntamiento de Zaragoza

El objetivo global del Ayuntamiento de Zaragoza es que en el año 2010, Zaragoza sea una de las ciudades europeas con mayor desarrollo económico basado en las nuevas tecnologías de la información y en la gestión del conocimiento.

Como parte de este objetivo global se encuentra el software libre. El Ayuntamiento de Zaragoza entiende que es necesario respaldar este fenómeno emergente en la informática mundial por razones de transparencia e independencia tecnológica, así como de conveniencia práctica, como están haciendo ya muchas empresas e instituciones públicas. Pero sobre todo, constituye una oportunidad para impulsar el desarrollo de una industria local de software ante los nuevos mercados que se pueden crear para este tipo de aplicaciones y sistemas. En ese sentido, el compromiso municipal con el software libre se manifestará en tres direcciones:

- Utilización de formatos abiertos y compatibles con todas las plataformas informáticas en las relaciones electrónicas del Ayuntamiento con los ciudadanos.
- Consideración de las ventajas objetivas (en términos de coste, fiabilidad, adaptabilidad y mantenimiento) de las alternativas de software libre a la hora de decidir la contratación de nuevas aplicaciones.
- Creación de un centro de software libre para la homologación y estandarización de aplicaciones para las administraciones públicas, en cooperación con las empresas interesadas y con los grupos de usuarios y desarrolladores independientes de software libre de la ciudad.

Se espera que el ahorro sea de casi un millón de euros con la migración a software libre. El ahorro se verá reflejado en aspectos como la actualización, desarrollo, soporte, mantenimiento del software y la prolongación de la vida útil del hardware.

La migración se lleva a cabo según las recomendaciones nacionales y europeas. Su ejecución está en manos de la Concejalía de Ciencia y Tecnología y se ejecutará en dos fases:

- **Primera fase:** migración de la suite ofimática que venían utilizando a OpenOffice.org, así como el cliente de correo electrónico.
- **Segunda fase:** migración del sistema operativo a GNU/Linux.

La puesta en marcha del proyecto de administración electrónica en el Ayuntamiento de Zaragoza tiene como estrategia básica el uso de software libre.

#### Lectura obligatoria

*El País* (2007). "Zaragoza se pasa a Linux para ahorrar hasta un millón al año".  
[www.lourdesmunozsantamaria.cat/IMG/pdf/ZaragozaLinux.pdf](http://www.lourdesmunozsantamaria.cat/IMG/pdf/ZaragozaLinux.pdf)

### Lectura recomendada

**Ricardo Cavero Arceiz** (2008). *La administración electrónica y los archivos en el Ayuntamiento de Zaragoza*.

<http://portal.aragon.es/portal/page/portal/CULTURA/PUBLIELECTRONICAS/PATRIMONIO-CULTURAL/ARCHIVOS/Actas%20VIII%20Jornadas%20/20%20La%20Administraci%C3%B3n%20Electronica%20Archivos%20Ayto%20Zgz.pdf>

## 2.4. El Ministerio de Exteriores alemán

El Ministerio de Exteriores alemán ha comenzado a migrar sus 11.000 ordenadores a GNU/Linux y a aplicaciones de software libre. Esto ha permitido reducir considerablemente los costes si los comparamos con otros ministerios.

Se pretende que la migración se lleve a cabo en las 230 embajadas y consulados que están repartidos por todo el mundo. En algunos países ha sido muy difícil la migración, como por ejemplo en Japón, ya que ha habido problemas con OpenOffice.org.

La embajada de Alemania en Madrid está utilizando GNU/Linux desde octubre del 2008, y se ha considerado que la migración ha sido todo un éxito.

## 2.5. El Gobierno de Malasia

El 16 de julio del 2004, la Administración pública de Malasia inició el Plan maestro para fomentar, adoptar y desarrollar el uso del software libre en el sector público. Se puso en marcha el Open Source Competency Center (OSCC). El OSCC se estableció como un centro de referencia nacional para dirigir y apoyar la aplicación del software libre en la administración.

El Plan maestro indica cómo llevar a cabo el trabajo de migración a software libre. Es un plan de trabajo a largo plazo y consta de tres fases de ejecución:

1) **Fase 1 (2004-2006)**. Es una fase de establecimiento y aprobación y se ha completado con éxito. Los objetivos de esta fase son los siguientes:

- Determinar normas y directrices.
- Establecer el Plan de ejecución.
- Determinar y aplicar el Plan de gestión del cambio.
- Diseñar y poner en marcha el Programa Nacional de Concienciación.
- Establecer un sistema de I+D.
- Certificación y la formación.
- Poner en práctica proyectos pilotos (rápidos).
- Desarrollar programas de incentivos y financiación.

2) **Fase 2 (2007-2010)**. Fase de adopción acelerada. Los objetivos de esta fase son los siguientes:

- Repetir los proyectos pilotos en los organismos seleccionados.

### Lectura recomendada

**Rolf Theodor Schuster** (2008). *Experiencias del Ministerio Federal alemán de Relaciones Exteriores*

<http://www.davidhammerstein.org/documentos/PresentacionRolfSchuster.pdf>

- Monitorización de tasa de adopción del software libre y el progreso a través de indicadores clave de rendimiento predefinidos.
- Llevar a cabo iniciativas bilaterales con países que adoptan el software libre.

3) **Fase 3 (2010–en adelante)**. Fase de autosuficiencia. Sus objetivos son los siguientes:

- Mejorar las soluciones de software libre con independencia de los organismos.
- Continuidad de la identificación de nuevas oportunidades.

## 2.6. Brasil

Gracias a las iniciativas llevadas a cabo por el Gobierno Federal, Brasil se ha convertido en un claro referente para los países iberoamericanos.

Algunas de las razones por las que el Gobierno Federal de Brasil ha optado por la migración a software libre son las siguientes:

- Necesidad de adoptar un sistema de administración electrónica.
- Seguridad.
- Eliminación de los cambios que el software propietario impone.
- Independencia tecnológica.
- Desarrollo de conocimiento local.
- Posibilidad de auditar los sistemas.
- Independencia de los proveedores.
- Reducción de costes por no tener que pagar licencias de software propietario.

En el punto 1 de estos materiales, "Fases de una migración a software libre", se ha indicado dos formas de hacer una migración. Brasil optó por la más segura, es decir, aquella que se realiza en varias fases, comenzando por los ordenadores personales.

La *Guía de referencia de migración para software libre* del Gobierno Federal brasileño se inspiró en la *Guía IDA* de la Unión Europea.

### Lectura obligatoria

Comité Técnico para Implementación de Software Libre; Comité Técnico de Sistemas Legados y Licencias de Software; Secretaría de Logística y Tecnología de Información (2004). *Referencia de Migración para software libre del Gobierno Federal Brasileño*.

[http://el-directorio.org/Telecentro\\_Comunitario/InformaticaComunitariaGobierno/CasosEstudioLatinoAmerica?action=AttachFile&do=get&target=brasil.pdf](http://el-directorio.org/Telecentro_Comunitario/InformaticaComunitariaGobierno/CasosEstudioLatinoAmerica?action=AttachFile&do=get&target=brasil.pdf)

## 2.7. Venezuela

En diciembre del 2004, Hugo Chávez Frías, presidente de la República Bolivariana de Venezuela, ordenó publicar el Decreto Presidencial 3390, que establecía el proyecto de ley "infogobierno".

En este decreto, se indica que la Administración pública nacional empleará prioritariamente software libre desarrollado con estándares abiertos en sus sistemas, proyectos y servicios informáticos.

A tales fines, todos los órganos y entes de la Administración pública nacional iniciarían los procesos de migración gradual y progresiva de éstos hacia el software libre desarrollado con estándares abiertos.

Además, el artículo 110 de la Constitución de la República Bolivariana de Venezuela reconoce como de interés público la ciencia, la tecnología, el conocimiento, la innovación y los servicios de información, con el objeto de lograr el desarrollo económico, social y político del país. Esta disposición constitucional se expresa con fuerza en los artículos 1.º de la Ley de Telecomunicaciones y 12.º de la Ley Orgánica de la Administración Pública. Con el Decreto N.º 825, emitido el 10 de mayo del 2000, se establece el acceso y el uso de Internet como política prioritaria para el desarrollo cultural, económico, social y político del Estado.

Para el cumplimiento de lo establecido en el decreto anteriormente citado, se propone un **Plan nacional de migración a software libre**.

El objetivo de este plan es guiar a todos los órganos de la Administración pública nacional de Venezuela en su migración a software libre.

Cada uno de estos órganos y entes de la Administración pública nacional deberían elaborar un Plan institucional de migración, que siga las directrices marcadas por el Decreto 3390 y el ya mencionado Plan nacional de migración a software libre.

## 2.8. Bolzano

Bolzano es una provincia del norte de Italia, que cuenta con 500.000 habitantes (datos del 2009) y tiene un área de 7.400 kilómetros cuadrados.

### Lectura recomendada

Hugo Chávez Frías (2004). *Decreto 3.390*.

<http://www.gobiernoenlinea.ve/legislacion-view/sharedfiles/Decreto3390.pdf>

### Lectura obligatoria

Oficina de Tecnologías de Información (2005). *Plan Nacional de Migración a Software Libre de la Administración Pública Nacional*.

[www.softwarelibre.gob.ve/documentos/PLANNACIONALDEMIGRACIONASWL230305.pdf](http://www.softwarelibre.gob.ve/documentos/PLANNACIONALDEMIGRACIONASWL230305.pdf)

En Bolzano han seguido los pasos de la Comunidad de Extremadura, llegando a migrar a software libre todos los ordenadores existentes en las 83 escuelas de la provincia. Antes de la migración, el coste anual era de 269.000 euros, frente a los 27.000 euros actuales por el mantenimiento.

#### Vídeo recomendado

**RAI 3 (2007).** *Uso del software libre en las escuelas de la provincia de Bolzano.*

Subtítulos en español:  
<http://www.youtube.com/watch?v=X7FOq0KbhyE>

Subtítulos en inglés:  
[http://www.youtube.com/watch?v=R2kM6bTh\\_XY](http://www.youtube.com/watch?v=R2kM6bTh_XY)

## Resumen

Cada vez son más las administraciones públicas que muestran su apoyo al software libre. Tanto es así, que algunas ya han optado por migrar parcial o totalmente sus servidores y sistemas de escritorio. Otras administraciones han decidido apoyar el software libre mediante subvenciones.

Las administraciones públicas utilizan numerosos programas informáticos para llevar a cabo sus actividades. Los datos de los ciudadanos se almacenan electrónicamente en varios organismos públicos y hacen uso de ellos según las leyes sobre privacidad. Los ciudadanos se ven obligados, por imperio de la ley, a ceder sus datos, por lo que es lógico que la propia Administración garantice de forma adecuada la gestión de tan valiosa información. Por lo tanto, las administraciones públicas tienen la obligación de resguardar dicha información y mantenerla en formatos que no le aten a iniciativas privadas y formatos cerrados.

Cualquier administración pública, tras una migración a software libre, obtendrá los siguientes beneficios:

- Independencia tecnológica.
- Compatibilidad.
- Interoperabilidad.
- Acceso universal a largo plazo.
- Preservación del patrimonio digital.
- Igualdad y no discriminación.
- Libertad de elección.
- Seguridad de la información.
- Oportunidad de crecimiento y desarrollo local.

Administraciones públicas que ya avanzan en la adopción de la filosofía del software libre son: Alemania, Argentina, Brasil, Chile, China, Colombia, Cuba, Ecuador, España, Francia, India, Italia, Francia, Finlandia, Inglaterra, México, Noruega, Nigeria, Perú, República Dominicana, Rusia, Sudáfrica, Suecia, Venezuela, etc.

En este módulo hemos estudiado las seis fases de las que se compone una migración a software libre: concienciación; diagnóstico y recogida de información; plan piloto; justificación; definición y ejecución del plan; y mantenimiento, asesoramiento y formación.

Se establecen como básicos cuatro criterios para la migración a software libre. Éstos son dependencia, coste, seguridad y transparencia.

De los ocho casos de estudio presentados en este módulo, cabe destacar el del Ayuntamiento de Munich que hemos estudiado junto con el de Newham, en Reino Unido. Ante un mismo problema, el primero optó por migrar a software libre y el segundo obtuvo un ventajoso acuerdo con Microsoft.

Resulta curioso observar que los criterios de seguridad y transparencia no fueron tenidos en cuenta en los casos mencionados en el párrafo anterior.

## Bibliografía

**Almond, Chris; Van Hoof, Jeroen; Lassonde, Nick; Li, Ben; Taylor, Kurt** (2006). *Linux Client Migration Cookbook, Version 2. A Practical Planning and Implementation. Guide for Migrating to Desktop Linux*. [Consulta: noviembre de 2009.]  
<http://www.redbooks.ibm.com/redbooks/pdfs/sg246380.pdf>

**Cavero Arceiz, Ricardo** (2008). *La administración electrónica y los archivos en el Ayuntamiento de Zaragoza*. [Consulta: noviembre de 2009.]  
<http://portal.aragon.es/portal/page/portal/CULTURA/PUBLIELECTRONI>

**Chávez Frías, Hugo** (2004). *Decreto 3.390*. [Consulta: noviembre de 2009.]  
<http://www.gobiernoonlinea.ve/legislacion-view/sharedfiles/Decreto3390.pdf>

**Comisión de las Comunidades Europeas** (2000). *eEurope 2002. Una sociedad de la información para todos. Plan de acción*. [Consulta: noviembre de 2009.]  
<http://www.ub.edu/integracio/Lleis/UE/eEUROPE%202000.pdf>

**Comisión de las Comunidades Europeas** (2002). *eEurope 2005. Una sociedad de la información para todos. Plan de acción*. [Consulta: noviembre de 2009.]  
[http://ec.europa.eu/information\\_society/eeurope/2002/news\\_library/documents/eeurope2005/eeurope2005\\_es.pdf](http://ec.europa.eu/information_society/eeurope/2002/news_library/documents/eeurope2005/eeurope2005_es.pdf)

**Comité Técnico para Implementación de Software Libre; Comité Técnico de Sistemas Legados y Licencias de Software; Secretaría de Logística y Tecnología de Información** (2004). *Referencia de Migración para software libre del Gobierno Federal Brasileño*. [Consulta: noviembre de 2009.]  
[http://el-directorio.org/Telecentro\\_Comunitario/InformaticaComunitariaGobierno/CasosEstudioLatinoAmerica?action=AttachFile&do=get&target=brasil.pdf](http://el-directorio.org/Telecentro_Comunitario/InformaticaComunitariaGobierno/CasosEstudioLatinoAmerica?action=AttachFile&do=get&target=brasil.pdf)

**D'Elia Branco, Marcelo; León Martínez, Mónica; Novo López, Alejandro; Otero García, Alberto** (2005). *Primera unidad del segundo módulo de la asignatura obligatoria del Máster de Software Libre de la UOC, Implantación de sistemas de software libre*. [Consulta: noviembre de 2009.]  
<http://cv.uoc.es/cdocent/9690KDU35DP1RF8OK9B0.pdf>

**D'Elia Branco, Marcelo** (2006). *Munich: la implantación del software libre avanza*. [Consulta: noviembre de 2009.]  
[http://wiki.softwarelivre.org/pub/Blogs/BlogPostMarceloBranco20061201174131/Munich\\_informe\\_es.pdf](http://wiki.softwarelivre.org/pub/Blogs/BlogPostMarceloBranco20061201174131/Munich_informe_es.pdf)

**El País** (2007). *Zaragoza se pasa a Linux para ahorrar hasta un millón al año*. [Consulta: noviembre de 2009.]  
<http://www.lourdesmunozsantamaria.cat/IMG/pdf/ZaragozaLinux.pdf>

**ISO** (2006). *Information technology – Open Document Format for Office Applications (OpenDocument) v1.0*. [Consulta: noviembre de 2009.]  
[http://standards.iso.org/ittf/PubliclyAvailableStandards/c043485\\_ISO\\_IEC\\_26300\\_2006\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c043485_ISO_IEC_26300_2006(E).zip)

**Ministerio de Administraciones Públicas** (2003). *Directrices IDA de migración a software de fuentes abiertas*. [Consulta: noviembre de 2009.]  
[http://www.csi.map.es/csi/pdf/Directrices\\_IDA\\_OSS\\_ESv1\\_r.pdf](http://www.csi.map.es/csi/pdf/Directrices_IDA_OSS_ESv1_r.pdf)

**Oficina de Tecnologías de Información** (2005). *Plan Nacional de Migración a Software Libre de la Administración Pública Nacional*. [Consulta: noviembre de 2009.]  
<http://www.softwarelibre.gob.ve/documentos/PLANNACIONALDEMIGRACIONASWL230305.pdf>

**RAI 3** (2007). *Uso del software libre en las escuelas de la provincia de Bolzano*. [Consulta: noviembre de 2009.]

Subtítulos en español:  
<http://www.youtube.com/watch?v=X7FOq0KbhyE>

Subtítulos en inglés:  
[http://www.youtube.com/watch?v=R2kM6bTh\\_XY](http://www.youtube.com/watch?v=R2kM6bTh_XY)

**Reina, Daniel** (2006). *Criterios de migración a Linux en las administraciones locales de la UE. Los casos de Munich y Newham*. [Consulta: noviembre de 2009.]  
<http://www.uoc.edu/uocpapers/2/dt/esp/reina.pdf>

**Schuster, Rolf Theodor** (2008). *Experiencias del Ministerio Federal alemán de Relaciones Exteriores* [Consulta: noviembre de 2009.]  
<http://www.davidhammerstein.org/documentos/Presentacion Rolf Schuster.pdf>

# Plan de migración de una administración pública

Antonio Rodil Garrido

PID\_00154681



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



# Índice

<b>Introducción</b> .....	5
<b>Objetivos</b> .....	6
<b>1. El plan de migración</b> .....	7
1.1. Introducción .....	8
1.2. Conceptos y consideraciones previas .....	9
1.3. Metodología .....	9
1.3.1. Situación actual .....	9
1.3.2. Inventario .....	9
1.3.3. Costes .....	10
1.3.4. Recursos dedicados .....	11
1.4. Análisis de viabilidad funcional, operativo y técnico .....	11
1.5. Plan de migración .....	11
1.5.1. Propuesta tecnológica inicial .....	11
1.5.2. Fases y detalles del plan .....	12
1.5.3. Calendario .....	12
1.5.4. Postinstalación .....	12
1.5.5. Gestión del cambio .....	13
<b>Resumen</b> .....	14
<b>Bibliografía</b> .....	15



## **Introducción**

En este último módulo de la asignatura se pone en práctica todo lo aprendido durante el curso.

A lo largo de los tres módulos anteriores se estudian los principales argumentos que impulsan a las administraciones públicas a adoptar sistemas de software libre, algunos de los planes directores más destacados, argumentos para el uso del software libre y liberación del mismo por parte de las administraciones, las fases de una migración a software libre y algunos de los casos de mayor relevancia en cuanto a migraciones se refiere.

Con los conocimientos adquiridos, estamos en disposición de elaborar un pequeño plan de migración. Los datos de la administración a migrar pueden ser reales o ficticios, pudiendo escoger también los que se indican en este módulo.

Algunos de los contenidos de este módulo han sido recopilados de las consultas que los estudiantes de cursos anteriores han efectuado a los consultores.

## Objetivos

Los objetivos de este módulo son los siguientes:

- 1.** Poner en práctica todo lo aprendido en los módulos anteriores.
- 2.** Elaborar un plan de migración a software libre.
- 3.** Establecer las normas del plan de migración.
- 4.** Guiar al estudiante por aquellos puntos del plan de migración que puedan resultarle confusos.

## 1. El plan de migración

Un proyecto de migración es, generalmente, un proceso largo. En el módulo 3 se comenta que una migración a software libre es un proyecto de ingeniería, sin embargo, no debemos pensar únicamente en la parte de ingeniería, debemos saber organizar los recursos de los que disponemos, principalmente los humanos, para realizar y gestionar correctamente la migración.

La definición de mecanismos de control será fundamental para poder realizar una migración con éxito. No debemos concentrarnos únicamente en los detalles técnicos, los cuales nos consumirán recursos e impedirán lograr un buen resultado. Sin documentación ni planificación, el proyecto fracasará. Por ejemplo, la migración de un navegador web sin un plan implicará eventos no controlados. Esto provocará pérdidas no cuantificables en horas de trabajo.

En todo proyecto el riesgo debe ser controlado. No basta con la disposición de especialistas en software o hardware, será necesario conocer las necesidades de la organización y también la manera de funcionar.

En el plan de migración deben existir hitos predefinidos, los cuales serán controlados en cada etapa revisando el cronograma y la precedencia de cada actividad.

Las mejoras o cambios en la funcionalidad original no son estrictamente parte de la migración, por lo que debe quedar clara la funcionalidad que se desea obtener al finalizar el proceso de migración. Por ejemplo, migrar el sistema de correo no es solamente cambiar un cliente por otro, debemos dar a los usuarios finales las mismas funcionalidades que tenían con el software reemplazado.

Como parte de la gestión debemos contemplar el proceso de elección del software. El análisis y la elección del software a utilizar influirán en el tiempo y la metodología de la migración.

Los aspectos más importantes en la elección del software que utilizaremos son los siguientes:

- Soporte del proveedor (información pública disponible en la red o servicios pagados).
- Desarrollo anexo necesario para que el paquete de software elegido cumpla con las funcionalidades.
- Compatibilidad y eventual interoperabilidad con otros paquetes de software en uso.

- Evaluar las funcionalidades que no estén en el nuevo paquete de software y revisar si podremos prescindir de ellas.

La tarea de la elección de software se verá facilitada en la medida en que tengamos más antecedentes sobre el software y las versiones disponibles. No debemos olvidar que, cuando revisemos las alternativas de software a utilizar, debemos tener en cuenta la compatibilidad con versiones anteriores. Usualmente tendemos a reemplazar una pieza de software encantados por las nuevas funcionalidades y potencialidades de éste, perdiendo el foco en la integración funcional con todos los componentes. También debemos tener en cuenta la madurez del software que vamos a seleccionar para la migración.

En la madurez están implícitos aspectos que debemos considerar, como por ejemplo:

- Disponibilidad de herramientas de administración.
- Tiempo de vida de la última versión.
- Tasa de errores reportados.
- Tasa de errores corregidos.
- Madurez de los estándares utilizados por el software.
- Popularidad del software, medido en cantidad de usuarios y reportes de éxito.

Estos puntos nos ayudarán a disminuir los riesgos ante errores no corregidos que impliquen un retraso en el proceso de migración.

En los capítulos 10 y 11 del documento *Migración a Software Libre. Guía de Buenas Prácticas*, se puede encontrar una amplia lista de software recomendado. Por una parte tenemos el software recomendado por el Proyecto SourcePyme y por otro lado, un directorio de software que debemos tener en cuenta a la hora de llevar a cabo un proyecto de migración.

El software detallado en el citado documento debe ser revisado para la utilización de la versión correspondiente, siendo necesario revisar también las recomendaciones de los desarrolladores.

A continuación se detallan los puntos que debe tener nuestro plan de migración. El estudiante podrá incorporar nuevos puntos, pero los que aquí se indican no podrán ser eliminados.

### 1.1. Introducción

Deberá hacerse una breve descripción e introducción del proyecto de migración que hay que realizar, con indicación de la justificación y punto de partida, objetivos generales y alcance del proyecto.

#### Lectura obligatoria

Daniel Sáenz; Martín Peris; Ricard Roca; David Anes (2007). *Migración a Software Libre. Guía de Buenas Prácticas*.

<https://discovirtual.aimme.es/docs/sourcepyme07/migracionalsoftwarelibre.pdf>

El estudiante podrá optar por realizar el proyecto para una administración pública real y que conozca en profundidad o para una administración pública ficticia.

## **1.2. Conceptos y consideraciones previas**

En este apartado se incluirá un glosario de términos, acrónimos y abreviaturas, aunque también puede trasladarse a la parte final del documento.

En cuanto a las consideraciones previas, se hará referencia a otros casos de éxito de similares características, indicación de las tareas a realizar (creación de los grupos de trabajo, determinación del responsable de la migración, resultados esperados tras la migración, seleccionar un grupo pequeño de usuarios para hacer pruebas piloto, como por ejemplo, un departamento, etc.).

## **1.3. Metodología**

### **1.3.1. Situación actual**

Realizaremos un estudio de la situación actual, así como de las características de la Administración pública seleccionada para la migración.

Si se trata de una administración local, como por ejemplo un ayuntamiento, es interesante dar información estadística sobre el municipio: situación geográfica, superficie en kilómetros cuadrados, densidad de población, número de habitantes, media de edad de la población, cercanía o lejanía con alguna localidad importante, estimación del número de habitantes con acceso a Internet, si la localidad dispone de telecentros, Wi-Fi en determinadas zonas de la localidad, etc. También es interesante facilitar alguna reseña histórica de interés.

### **1.3.2. Inventario**

En el caso de que la Administración pública seleccionada sea ficticia, se podrá utilizar la siguiente información de base:

- Una administración local de una población mayor a los 10.000 habitantes y con un mínimo de 20 empleados públicos / puestos de trabajo.
- El sistema actual está basado completamente en soluciones Microsoft. Windows 2000, Windows XP, Windows Vista, Windows 7 y MS Office 97 o posterior. Puede parecer extraño pero estas combinaciones de software aún son bastante frecuentes en el ámbito de las administraciones públicas. Servidores Windows 2000 o 2003 Server con ActiveDirectory.

- Sistema de correo ofrecido por un ISP con una cuenta por usuario y acceso POP/IMAP y WebMail.
- Existe un mínimo de dos puestos de trabajo con soluciones basadas en AutoCAD.
- Aplicación de registro de entrada y salida realizada en Microsoft Access.
- Necesidad de sincronización de correo, contactos y agenda con dispositivos móviles tipo Windows Mobile o Blackberry.
- Utilización de soluciones de firma electrónica en los puestos de los cargos electos.
- Las estaciones de trabajo tienen una antigüedad media de 4 años y los servidores de 5.  
Estos criterios reflejan, según nuestra experiencia, los principales problemas en planes reales de migración.

### 1.3.3. Costes

Indicaremos los gastos que tiene la Administración pública antes de la migración, en cuanto a costes de licencias de software, mantenimiento del software y hardware, formación de los empleados públicos, etc.

Deberá realizarse una comparación de los costes antes de hacer la migración y los que se estima tener después de la misma.

En el caso de estar realizando el proyecto de migración de una administración ficticia, tendremos en cuenta que antes de la migración los costes anuales de mantenimiento de las licencias de software de cada uno de estos equipos es de 90 euros para las estaciones cliente y de 750 euros para los servidores.

En cuanto a los costes de personal tenemos los siguientes:

- Coste de formador 30 €/h.
- Coste de consultor senior 60 €/h.
- Coste de consultor júnior 45 €/h.

Si por el contrario se trata de una administración real, será necesario aportar los datos reales.

### **1.3.4. Recursos dedicados**

Debemos determinar, según el tipo de migración que realicemos, el personal que vamos a necesitar, así como si se tratará de personal propio o subcontratado.

Es importante indicar los perfiles de los distintos profesionales y usuarios que intervendrán en la migración.

### **1.4. Análisis de viabilidad funcional, operativo y técnico**

En un proyecto real, deberíamos tener en cuenta el marco regulatorio y de competencias de la Administración pública, sin embargo, para simplificar el trabajo, no lo tendremos en cuenta en nuestro plan de migración.

En este punto podemos emplear la herramienta de análisis DAFO para determinar las debilidades, amenazas, fortalezas y oportunidades.

1) **Viabilidad funcional.** Debe determinarse si la migración es o no posible, aportando los argumentos oportunos. Se deben identificar los posibles riesgos potenciales y tratar de garantizar la migración, siempre que sea posible su realización.

Este apartado se puede abordar desde el estudio de otras migraciones, con la validación por parte del cliente de un piloto inicial que contenga los casos más representativos.

2) **Viabilidad operativa.** La viabilidad operativa siempre dependerá de los recursos humanos que tengamos disponibles para la migración. Debe determinarse si el sistema funcionará y si será aceptado y utilizado una vez se finalice. También es interesante tener en cuenta a los usuarios y si estos aceptarán el cambio.

3) **Viabilidad técnica.** Debemos concretar si disponemos de la tecnología necesaria para realizar la migración. En cualquier caso, debemos indicar la arquitectura técnica, ya sea la disponible o la que se debe adquirir para poder realizar la migración.

### **1.5. Plan de migración**

#### **1.5.1. Propuesta tecnológica inicial**

Llegados a este punto, ya debemos tener el suficiente conocimiento de la Administración pública. Por lo tanto, es hora de realizar una primera propuesta tecnológica.

Debemos indicar cada una de las aplicaciones de software libre que sustituirá a su homóloga de software propietario. Esta lista deberá estar categorizada por tipo. También debe indicarse si es necesaria la adquisición de nuevo hardware, la ampliación de anchos de banda, compras de dominios para nuevas páginas web, etc.

### **1.5.2. Fases y detalles del plan**

Indicaremos el tipo de migración que vamos a realizar. Debemos tener en cuenta la clase de administración y los objetivos de la misma.

Desglosaremos cada una de las fases de nuestro plan, con una explicación del desarrollo de las mismas, indicando el resultado final que esperamos de cada una de las mismas.

### **1.5.3. Calendario**

Se elaborará un calendario para cada una de las fases y subfases de la migración, con indicación de la fecha de inicio, fecha final, duración, participantes, etc. Es importante hacer un diagrama de Gantt de todo el proyecto.

### **1.5.4. Postinstalación**

Una vez hecha la migración, debemos establecer mecanismos de control que ayuden al mantenimiento y máximo aprovechamiento de los sistemas migrados.

Los usuarios deben entrar en una dinámica que les permita familiarizarse con las aplicaciones recién migradas. Si durante la migración se ha utilizado alguna herramienta cooperativa para la coordinación, los usuarios pueden aprovechar esta herramienta para conocer mejor sus aplicaciones, compartiendo experiencias, sugerencias, etc.

## **Mantenimiento y apoyo**

Se indicarán los recursos con que debe contar la administración para poder llevar a cabo el correcto mantenimiento del sistema migrado.

Es posible que necesitemos contratar personal con unos perfiles determinados, desarrollo de una web de apoyo, herramientas colaborativas, sistema de encuestas, sugerencias, foros, etc.

## **Nuevas implementaciones**

Se deben describir las nuevas implementaciones y actualizaciones de software que requerirá la organización, así como el tipo y tamaño de las mismas.

## **Evaluación y conclusiones**

El estudiante deberá realizar su propia evaluación.

### **1.5.5. Gestión del cambio**

En este punto indicaremos todas aquellas acciones formativas que creamos necesarias para que la gestión del cambio y uso de los nuevos productos de software libre sea correcta, ya sea antes, durante o tras la realización del proyecto.

#### **Formación**

Se establecerá un plan de formación, tanto para usuarios como para administradores, indicando el número de horas de formación para ambos grupos.

#### **Elementos de apoyo a la formación**

Indicación de las herramientas con que contamos para la formación, como por ejemplo, plataforma de formación *on-line*, uso de cursos de especificación SCORM, etc.

#### **Apoyo**

Se indicará todo el material de apoyo que consideremos necesario. Sabemos de la abundante información que podemos encontrar en Internet, por lo que un ejemplo claro de lo que podemos incluir en este punto es una lista de aquellas direcciones con recursos de formación.

## Resumen

Se estima que, a nivel mundial, el dinero de los contribuyentes destinado al uso de software privativo asciende a más de 1 billón de dólares al año (unos 677.900 millones de euros, aplicando la conversión establecida el día 2 de noviembre del 2009, que indica que 1 € = 1,4751 USD).

Un sistema operativo privativo tiene un coste aproximado de 140 euros, y una suite ofimática privativa unos 115 euros. Por lo tanto, por cada equipo nuevo se necesita una inversión en software de 255 euros aproximadamente.

En un proyecto de migración a software libre, la planificación puede marcar la diferencia entre el éxito y el fracaso. Esta planificación debe realizarse para tratar de minimizar la repercusión del proyecto en el trabajo diario del personal de la Administración pública. Lo fundamental es el trabajo y no la herramienta.

Ha de dedicarse un tiempo a la formación, no sólo de los usuarios, sino sobre todo para el personal informático.

Uno de los principales problemas a la hora de realizar un proyecto de migración a software libre es el miedo al cambio tecnológico y éste puede llegar a frenar el proyecto. Para vencer este miedo, es importante la formación, planificación y disponer siempre de un "plan B".

## Bibliografía

**Almond, Chris; Van Hoof, Jeroen; Lassonde, Nick; Li, Ben; Taylor, Kurt** (2006). *Linux Client Migration Cookbook, Version 2. A Practical Planning and Implementation. Guide for Migrating to Desktop Linux*. [Consulta: noviembre de 2009.]  
<http://www.redbooks.ibm.com/redbooks/pdfs/sg246380.pdf>

**Sáenz, Daniel; Peris, Martín; Roca, Ricard; Anes, David** (2007). *Migración a Software Libre. Guía de Buenas Prácticas*. [Consulta: noviembre de 2009.]  
<http://discovirtual.aimme.es/docs/sourcepyme07/migracionalsoftwarelibre.pdf>



---

# GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

---

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## **TERMS AND CONDITIONS**

### **0. Definitions.**

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

---

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

## **1. Source Code.**

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically

---

linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## **2. Basic Permissions.**

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

## **3. Protecting Users' Legal Rights From Anti-Circumvention Law.**

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

---

#### **4. Conveying Verbatim Copies.**

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### **5. Conveying Modified Source Versions.**

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

#### **6. Conveying Non-Source Forms.**

---

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a parti-

---

cular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## **7. Additional Terms.**

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

---

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

---

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## **8. Termination.**

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

## **9. Acceptance Not Required for Having Copies.**

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

## **10. Automatic Licensing of Downstream Recipients.**

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

---

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

## **11. Patents.**

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself

---

of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. 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 (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

## **12. No Surrender of Others' Freedom.**

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

## **13. Use with the GNU Affero General Public License.**

---

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

#### **14. Revised Versions of this License.**

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

#### **15. Disclaimer of Warranty.**

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### **16. Limitation of Liability.**

---

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
```

```
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation, either version 3 of the License, or  
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
```

---

```
along with this program. If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<http://www.gnu.org/philosophy/why-not-lgpl.html>>.