



## ¿Como iniciar un proyecto de Software Libre?

El texto que vas a leer a continuación es el guión que se ha empleado para la charla facilitada por CodeKo.com en las II Jornadas de Software Libre de la UGR.

Antes de hablar acerca de cómo iniciar un proyecto de software libre es importante conocer qué es exactamente eso del software libre y en que puede beneficiarte. Vayamos por partes.

### ¿Que es el software libre?

El software libre **NO** es necesariamente software gratuito (aunque puede serlo). Para el un software sea libre debe reunir las siguientes libertades.

- Libertad 0 - La libertad de usar el programa, con cualquier propósito.
- Libertad 1 - La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades.
- Libertad 2 - La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo.
- Libertad 3 -La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.

Para poder ejercer las libertades 1 y 3 es necesario el acceso al código fuente para poder estudiar y modificar el software .

### Repercusiones del software libre ... o por qué es beneficioso usar SL

En esta charla no voy a entrar a valorar moralmente el uso del software libre, la dimensión ética de esta elección es algo evidente pero a algunos puede resultarle tan poco tangible como la promesa de una vida eterna. La retribución al ego de un programador de SL esta muy bien pero no da de comer, es por ello que voy a centrarme en los aspectos mas "egoistas" del uso de SL

### El software libre y yo

Una vez tenemos claro más o menos que eso del Software Libre vamos a ver **en qué me beneficia a mi cómo individuo.**

- El usuario no se mantiene preso. Al disponer del código siempre puedes cambiar de proveedor
- Posibilidad de adaptación, al disponer de los fuentes puedes realizar cambios profundos, superficiales, de idioma, etc..
- Posibilidad de auditar el código y verificar su robustez y seguridad.
- Un código visible no oculta procesos que pueden ser dañinos, comprometer mis datos, etc..

### El software libre y el programador

Esta claro que como persona es una ventaja, pero como programador **¿que saco yo del software libre?**

En que me beneficia como programador

- Posibilidad de colaborar en proyectos reales. Es importante no sólo para adquirir experiencia personal en un ámbito de trabajo real, si no también de cara a demostrar experiencia en futuras contrataciones. Seguro que todos conocéis los famosos "se requieren dos años de experiencia demostrable" ahí tenéis una forma de adquirirlos haciendo algo que además os guste.
- Ver el código de otros para aprender soluciones diferentes a las propias. No mejores ni peores pero si diferentes maneras de afrontar el mismo problema.
- Posibilidad de reutilización de código. ¿Por qué volver a programar algo que ya esta hecho y funciona bien? Reutilizar código es bueno (*siempre que respetes las licencias, claro*).
- Posibilidad de dejar constancia de tu trabajo, demostrando que eso lo has hecho tu.





Ahora estarás pensando que está muy bien, que como programador y como usuario todo son ventajas, pero ¿realmente una empresa puede vivir de esto del SL?

Si. Se puede no sólo es viable sino que además es rentable, para ello es importante tener claro que las empresas que se dedican al desarrollo de SL no venden licencias, venden productos y servicios pero no licencias.

En Codeko.com no sólo trabajamos con SL que ya existe adaptándolo al cliente sino que también creamos nuestros propios productos y los distribuimos como SL.

A veces el cliente no desea que el producto sea liberado, en este caso, SIEMPRE facilitamos los fuentes de la aplicación al cliente, de esta manera el producto pasa a ser parte del cliente (aunque no renunciamos a nuestro derecho de autoría) y el puede elegir si liberarlo o no.

### En que me beneficia el SL si soy una empresa de desarrollo de Software

- Posibilidad de ofrecer productos muy terminados a bajo coste y en tiempo breve, con la ventaja frente al software de código cerrado de que es adaptable.
  - \* Puedo ofrecer productos terminados como Wordpress, tiendas virtuales, etc
  - \* Posibilidad de trabajar sobre frameworks robustos como Django
  - \* En los desarrollos completamente a medida el uso de librerías permite centrarse en la lógica de la aplicación y no en detalles como un login, o una validación de campos
- Clientes no cautivos. Una empresa que hace bien su trabajo mantiene a los clientes, por la calidad no por la licencia. Además se reciben clientes de empresas que no realizan bien su trabajo. A veces este punto es discutible, desde nuestra experiencia nunca hemos perdido un cliente por trabajar con SL, al contrario, sabemos que los clientes que mantenemos es por que ellos así lo desean, de manera que no mantenemos un cliente a pesar suya que probablemente *boicotearía* sus desarrollos. Es realmente una gran ventaja.
- Las nuevas empresas cuentan desde el inicio con una cartera de productos que pueden adaptar a las necesidades del cliente. Es cierto que es posible disponer de esa cartera mediante el software privativo, pero en este caso no puedes realizar las adaptaciones necesarias para lograr un producto a medida.

### Pues en el fondo esto del SL resulta que está bien. ¿Que mas ventajas tiene?

- El código es conocimiento y como tal debe ser libre. Ocultar o intentar privatizar el conocimiento es un sinsentido. El conocimiento es el motor que hace que la sociedad avance y debe de estar en manos de todos.
- Si eres un programador y trabajas en el SL puedes elegir en que proyecto trabajar e incluso ganar dinero con ello. Si no lo haces, al menos te habrás divertido. Es una opción mejor a las prácticas no remuneradas en empresas.
- Mejores productos a un coste mas económico. El uso de librerías y fragmentos de código ya existentes, o el de aplicaciones ya desarrolladas hacen que no sea necesario invertir tanto tiempo en el desarrollo de manera que los costes, lógicamente, se reducen.
- Una administración pública / empresa que trabaje con SL es una administración mas segura.
- Uso de estándares, si no practicas la ocultación de código no te importará adherirte a un estándar, ni te preocupas en no ser compatible

Una vez que ya tenemos claro que es esto del SL vamos ver como se puede iniciar un proyecto de SL y no morir en el intento





## Como iniciar un proyecto de software libre.

A continuación voy a exponer una serie de pasos de cómo iniciar un proyecto de software libre

- El primer paso de todos es **tener clara la idea**

Preguntate ¿qué es lo que quieres hacer? ¿Existe ya?

Muchas veces es mejor unirse a un proyecto existente que iniciar uno propio. De esta manera aunas esfuerzos y colaborarás con gente para que tu proyecto salga adelante.

- Una vez tengas claro qué vas hacer y cómo **elige que licencia vas a utilizar**, es muy importante ya que esto afectará, por ejemplo, a que librerías puedes utilizar.

Además la licencia va a determinar que uso puede tener el programa. (Uso comercial, o no por ejemplo)

Deja muy claro a los usuarios desde el principio cual va a ser la licencia.

Si no sabes que licencia utilizar la Oficina de Software Libre de la Universidad de Granada puede asesorarte <http://osl.ugr.es>

- **Preguntate ¿existe mas gente interesada en tu proyecto?** Busca socios, o colaboradores, esto no se trata de que Google apoye tu proyecto, si no de buscar si hay alguien que esté ya en tu misma situación y que podáis colaborar juntos.

- Antes de empezar a escribir código, elegir la forma de trabajo, nosotros recomendamos y usamos SCRUM! Una metodología de desarrollo ágil, pero existen muchas otras. Para conocer de que va esto del Scrum! Puedes leer el libro "Scrum desde las trincheras" (<http://go4.es/01DD>)

- **¡Reutiliza código!** Antes de pensar que es lo que vas a programar piensa que es lo que NO tienes que programar. No reinventes la rueda.

- **Presta atención a los usuarios**, documenta la instalación, documenta tu código, manen una linea de buenas practicas que faciliten el uso de tu aplicación. Muchos proyectos acaban "muriendo" por qué no se ha documentado correctamente la instalación, o no se ha facilitado ésta a los usuarios. Esto **NO** quiere decir que no liberes tu código hasta que no este documentado o perfecto, pero si que tengas en cuenta esto.

- **Pide que la gente se implique**. Seguramente recibirás e-mails, o propuestas de usuarios diciendo *¿Tu programa no hace X? ¿tu aplicación no esta traducida a Y? Tu diseño no es todo lo bueno que podría ser ...* Ante esta situación pide que los usuarios se impliquen, el desarrollo de un programa no es sólo código hay otras capacidades importantes que hay que tener en cuenta en las que los usuarios pueden ayudarte, diseñadores, aportación de ideas, consejos de usabilidad.

**Es software libre, cualquier puede colaborar!**

- **Utiliza una forja, un sistema de versiones, y manen un registro de bugs** Son puntos muy importantes para mantener activo y sobre todo, organizado un proyecto.

A continuación tienes una lista muy breve con algunos ejemplos.

Forjas que puedes utilizar:

- <https://forja.rediris.es/>

- <http://code.google.com>

Sistemas de versiones:

- Subversion

- Mercurial

- Git

Sistemas de control de bugs

- Bugzilla

- Tu propio e-mail





***Y recuerda que si te cansas de tu proyecto, no te preocupes es software libre, mientras el código este disponible siempre habrá posibilidad de que alguien lo continúe y éste siga vivo.***

Este documento se distribuye bajo licencia Creative Commons <http://creativecommons.org/licenses/by-sa/3.0/es/>

**Usted es libre de:**

Copiar, distribuir y comunicar públicamente la obra  
Hacer obras derivadas

**Bajo las condiciones siguientes:**

Reconocimiento — Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).

Compartir bajo la misma licencia — Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Para esta obra, reconozca a CodeKo.com

***Gracias por haber leído hasta aquí.***

CodeKo

[Http://www.codeko.com](http://www.codeko.com)

[codeko@codeko.com](mailto:codeko@codeko.com)

