

# La Catedral y el Bazar

Eric S. Raymond

Traducción de José Soto Pérez

21 de mayo de 1997

## Índice

### 1. La Catedral y el Bazar

### 2. El correo tenía que llegar

## Resumen

Analizo un exitoso proyecto de software libre (*fetchmail*), que fue realizado para probar deliberadamente algunas sorprendentes ideas sobre la ingeniería de software sugeridas por la historia de Linux.

Discuto estas teorías en términos de dos estilos de desarrollo fundamentalmente opuestos: el modelo **catedral** de la mayoría de los fabricantes de software comercial contra el modelo **bazar** del mundo Linux. Demuestro que estos modelos parten de puntos de vista contrapuestos acerca de la naturaleza de la tarea de depuración del software.

Posteriormente, hago una argumentación, a partir de la experiencia de Linux, de la siguiente sentencia: “si se tienen las miradas suficientes, todas las pulgas saltarán a la vista”.

Al final, sugiero algunas fructíferas analogías con otros sistemas autoregulados de agentes egoístas, y concluyo con una somera exploración de las implicaciones que pude tener este enfoque en el futuro del software.

## 1. La Catedral y el Bazar

1 Linux es subversivo. ¿Quién hubiera pensado hace apenas cinco años que un sistema operativo de talla mundial surgiría, como por arte de magia, gracias a la actividad “hacker” desplegada en ratos libres por varios miles de programadores diseminados en todo el planeta, conectados solamente por los tenues hilos de la Internet?

Lo que sí es seguro es que yo no. Cuando Linux apareció en mi camino, a principios de 1993, yo tenía invertidos en UNIX y en el desarrollo de software libre alrededor de diez años. Fui uno de los primeros en contribuir con GNU a mediados de los ochenta y he estado aportando una buena cantidad de software libre a la red, desarrollando o colaborando en varios programas (NetHack, los modos VC y GUD de Emacs, xlife y otros) que todavía son ampliamente usados. Creí que sabía cómo debían hacerse las cosas.

Linux vino a trastocar buena parte de lo que pensaba que sabía. Había estado predicando durante años el evangelio UNIX de las herramientas pequeñas, de la creación rápida de prototipos y de la programación evolutiva. Pero también creía que existía una determinada complejidad crítica, por encima de la cual se requería un enfoque más planeado y centralizado. Yo pensaba que el software de mayor envergadura (sistemas operativos y herramientas realmente grandes, tales como Emacs) requería

construirse como las catedrales (ver figura 1), es decir, que debía ser cuidadosamente elaborado por genios o pequeñas bandas de magos trabajando encerrados a piedra y lodo, sin liberar versiones beta antes de tiempo.



Figura 1: Catedral de la Plaza de las Pasiegas (Granada)

El estilo de desarrollo de Linus Torvalds (*libere rápido y a menudo, delegue todo lo que pueda, sea abierto hasta el punto de la promiscuidad*) me cayó de sorpresa. No se trataba de ninguna forma reverente de construir la catedral. Al contrario, la comunidad Linux se asemejaba más a un bullicioso bazar de Babel (ver figura 2), colmado de individuos con propósitos y enfoques dispares (fielmente representados por los repositorios de archivos de Linux, que pueden aceptar aportaciones de “quien sea”), de donde surgiría un sistema estable y coherente únicamente a partir de una serie de artilugios.

El hecho de que este estilo de bazar parecía funcionar, y funcionar bien, realmente me dejó sorprendido. A medida que iba aprendiendo a moverme en ese medio, no sólo trabajé arduamente en proyectos individuales, sino en tratar de comprender por qué el mundo Linux no naufragaba en el mar de la confusión, sino que se fortalecía con una rapidez inimaginable para los constructores de catedrales.

Creí empezar a comprender a mediados de



Figura 2: Gran bazar de Estambul (Turquía)

1996. El destino me dio un medio perfecto para demostrar mi teoría, en la forma de un proyecto de software libre que trataría de realizar siguiendo el estilo del bazar de manera consciente. Así lo hice y resultó un éxito digno de consideración.

En el resto de este artículo relataré la historia de este proyecto, y la usaré para proponer algunos aforismos sobre el desarrollo real del software libre. No todas estas cosas fueron aprendidas del mundo Linux, pero veremos como fue que les vino otorgar un sentido particular. Si estoy en lo cierto, le servirán para comprender mejor qué es lo que hace a la comunidad de linux tan buena fuente de software; y le ayudarán a ser más productivo.

## 2. El correo tenía que llegar

Desde 1993 he estado encargado de la parte técnica de un pequeño ISP de acceso gratuito llamado Chester County InterLink (CCIL), en West Chester, Pennsylvania (fui uno de los fundadores de CCIL y escribí su original software BBS multiusuario, el cual puede verse en [telnet://locke.ccil.org](http://telnet://locke.ccil.org). Actualmente soporta más de tres mil usuarios en 19 líneas). Este empleo me permitió tener acceso a la red las 24 horas del día a través de la línea de 56K de CCIL, ¡de hecho, prácticamente él me lo de-

mandaba!

[...]

Necesitaba un cliente POP3; de tal manera que lo busqué en la red y encontré uno. En realidad hallé tres o cuatro. Usé pop-perl durante un tiempo, pero le faltaba una característica a todas luces evidente: la capacidad de identificar las direcciones de los correos recuperados para que las respuestas pudieran darse correctamente.

El problema era este: supongamos que un tal *monty en locke* me envía un correo. Si yo lo jalaba desde snark y luego intentaba responder, entonces mi programa de correos dirigía la respuesta a un *monty inexistente en snark*. La edición manual de las direcciones de respuesta para pegarles el ‘@ccil.org’, muy pronto se volvió algo muy molesto.

Era evidente que la computadora tenía que hacer esto por mí. (De hecho, de acuerdo con RFC1123, sección 5.2.18, *sendmail* debería de estarlo haciendo.) ¡Sin embargo, ninguno de los clientes POP lo hacía realmente! Esto nos lleva a la primera lección:

*1. Todo buen trabajo de software comienza a partir de las necesidades personales del programador. (Todo buen trabajo empieza cuando uno tiene que rascarse su propia comezón.)*

Esto podría sonar muy obvio: el viejo proverbio dice que “la necesidad es la madre de todos los inventos”. Empero, hay muchos programadores de software que gastan sus días, a cambio de un salario, en programas que ni necesitan ni quieren. No ocurre lo mismo en el mundo Linux; lo que sirve para explicar por qué se da una calidad promedio de software tan alta en esa comunidad.

Por todo esto, ¿pensaran que me lancé inmediatamente a la vorágine de escribir, a partir de cero, el programa de un nuevo cliente POP3 que compitiese con los existentes? ¡Nunca en la vida! Revisé cuidadosamente las herramientas POP que tenía al alcance, preguntándome

“¿cuál se aproxima más a lo que yo necesito?”, porque

*2. Los buenos programadores saben qué escribir. Los mejores, que reescribir (y reutilizar).*

Aunque no presumo ser un extraordinario programador, he tratado siempre de imitar a uno de ellos. Una importante característica de los grandes programadores es la meticulosidad con la que construyen. Saben que les pondrán diez no por el esfuerzo, sino por los resultados; y que casi siempre será más fácil partir de una buena solución parcial que de cero.

Linus, por ejemplo, no intentó escribir Linux partiendo de cero. En vez de eso, comenzó por reutilizar el código y las ideas de Minix, un pequeño sistema operativo (OS) tipo UNIX hecho para máquinas 386. Eventualmente terminó desechando o reescribiendo todo el código del Minix, pero mientras contó con él le sirvió como una importante plataforma de lanzamiento del proyecto en gestación que posteriormente se convertiría en Linux.

[...]

## Créditos

Este extracto de «La Catedral y el Bazar» se ha obtenido de:

<http://biblioweb.sindominio.net/telematica/catedral.html>

La fotografía de la Plaza de las Pasiegas en Granada es de:

<http://imageshack.us/photo/my-images/27/04plazapasiegas.jpg>

La fotografía del bazar de Estambul es de Babak Gholizadeh:

[http://en.wikipedia.org/wiki/File:Istanbul\\_grand\\_bazar\\_1.jpg](http://en.wikipedia.org/wiki/File:Istanbul_grand_bazar_1.jpg)