

**Novática**, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática). **Novática** edita también **UPGRADE**, revista digital de **CEPIS** (Council of European Professional Informatics Societies), en lengua inglesa, y es miembro fundador de **UPENET** (**UPGRADE** European Network)

<<http://www.ati.es/novatica/>>  
<<http://www.upgrade-cepis.org/>>

**ATI** es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies) y es representante de España en **IFIP** (International Federation for Information Processing); tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AIZ** y **ASTIC**.

**Consejo Editorial**

Antoni Carbonell Noguera, Juan Manuel Cueva Lovelle, Juan Antonio Esteban Iriarte Francisco, Julio Crespo, Celestino Martín Alonso, Josep Molas i Bertrán, Olga Pallás Codina, Fernando Píera Gómez (Presidente del Consejo), Ramón Puigjaner Trepal, Miquel Sàrries Griño, Asunción Yturbe Herranz

**Coordinación Editorial**

Rafael Fernández Calvo <rfcalvo@ati.es>

**Composición y autoedición**

Jorge Lázcan Gil de Ramales

**Traducciones**

Grupo de Lengua e Informática de ATI <<http://www.ati.es/gl/lengua-informatica/>>

**Administración**

Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

**Secciones Técnicas: Coordinadores**

**Administración Pública electrónica**

Gumersindo García Arribas, Francisco López Crespo (MAP) <gumersindo.garcia@map.es>, <flc@ati.es>

**Arquitecturas**

Jordi Tubella Murgadas (DAC-UPC) <jordit@ac.upc.es>  
Victor Viñals Yufiera (Univ. de Zaragoza) <victor@unizar.es>

**Auditoría eITIC**

Marina Tourinho Trullitro, Manuel Palao García-Suelto (ASIA) <marinatourinho@marinatourino.com>, <manuel@palao.com>

**Bases de datos**

Coral Calero Muñoz, Mario G. Plattini Velthuis (Escuela Superior de Informática, UCLM) <Coral.Calero@uclm.es>, <mplattini@inf-cr.uclm.es>

**Derecho e Incentivos**

Isabel Hernández Colados (Fac. Derecho de Donostia, UPV) <ihernando@legaltek.net>  
Elena Davara Fernández Marcos (Davara & Davara) <edavara@davara.com>

**Enseñanza Universitaria de la Informática**

Joaquín Ezepeleta Mateo (CPS-UZAR) <ezepeleta@posta.unizar.es>  
Cristóbal Pareja Flores (DSP-UCM) <cpareja@sip.ucm.es>

**Gestión del Conocimiento**

Juan Baiget Solé (Cap Gemini Ernst & Young) <juan.baiget@ati.es>

**Informática y Filosofía**

José Corco Jorinúa (UC) <jcorco@unica.edu>  
Esperanza Marcos Martínez (ESCEIT-URJC) <cuca@escet.urjc.es>

**Informática Gráfica**

Miguel Chover Solís (Universitat Jaume I de Castellón) <chover@lsi.uji.es>  
Roberto Vivó Hernández (Eurographics, sección española) <rvivo@dsic.upv.es>

**Ingeniería del Software**

Javier Dolado Cosín (DLS-UPV) <dolado@si.ehu.es>  
Luis Fernández Sanz (PDS-EJ-UEM) <lfern@dpriis.esi.uem.es>

**Inteligencia Artificial**

Federico Barber Sanchis, Vicente Botti Navarro (DSIC-UPV) <fvotti\_barber@dsic.upv.es>

**Interacción Persona-Computador**

Julio Abascal González (FI-UPV) <julio@si.ehu.es>  
Jesus Lorés Vidal (Univ. de Lleida) <jesus@eup.udl.es>

**Internet**

Alonso Álvarez García (TID) <alonso@ati.es>  
Llucen Pagés Casas (Indra) <pages@ati.es>

**Lengua e Informática**

M. del Carmen Ugarte García (IBM) <cuarte@ati.es>

**Lenguajes Informáticos**

Andrés Martín López (Univ. Carlos III) <amarin@it.uc3m.es>  
J. Angel Velázquez Huibide (ESCEIT-URJC) <a.velazquez@escet.urjc.es>

**Librerías e Informática**

Alonso Escolano (FIR-Univ. de La Laguna) <aescolano@ull.es>  
Xavier Gómez Guinovart (Univ. de Vigo) <xgg@uvigo.es>

**Lingüística computacional**

Manuel Palomar (Univ. de Alicante) <mpalomar@disi.ua.es>

**Mundo estudiantil**

Adolfo Vázquez Rodríguez (Rama de Estudiantes del IEEE-UCM) <a.vazquez@ieee.org>

**Profesión Informática**

Rafael Fernández Calvo (ATI) <rfcalvo@ati.es>  
Miquel Sàrries Griño (Ayto. de Barcelona) <msarries@ati.es>

**Redes y servicios telemáticos**

Luis Gujardo Colombia (DCOM-UPV) <lgujardo@doom.upv.es>  
Jesús Solís Pareta (DAC-UPC) <pareta@ac.upc.es>

**Seguridad**

Javier Arellito Bertolin (Univ. de Deusto) <jarellito@eside.deusto.es>  
Javier López Muñoz (ETS Informática-UMA) <jlm@lcc.uma.es>

**Sistemas de Tiempo Real**

Alejandro Alonso Muñoz, Juan Antonio de la Puente Añaró (DIT-UPM) <aalonso.igunte@di.upm.es>

**Software Libre**

Jesus M. González Barahona, Pedro de las Heras Quirós (GSYC-URJC) <jmgh.phas@gsyc.esct.urjc.es>

**Teconología de Objetos**

Jesus García Molina (DIS-UM) <jmolina@correo.um.es>  
Gustavo Rossi (LIFIA-UNLP, Argentina) <gustavo@sol.info.unlp.edu.ar>

**Teconologías para la Educación**

Juan Manuel Dodero Barro (UC3M) <ddero@inf.uc3m.es>

**Teconologías y Empresa**

Pablo Hernández Medrano (Bluemart) <pablohm@bluemart.biz>

**TIC para la Sanidad**

Valentín Masero Vargas (DI-UNEX) <vmasero@unex.es>

**TIC y Turismo**

Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga) <aguayo.guevara@cc.uma.es>

**Coordinación Editorial, Redacción Central y Redacción ATI Madrid**

Padilla 66, 3º, dcha., 28006 Madrid  
Tfn. 914029391; fax 913093685 <novatica@ati.es>

**Composición, Edición y Redacción ATI Valencia**

Av. del Reino de Valencia 23, 46005 Valencia  
Tfn./fax 963300392 <secretgati@ati.es>

**Administración y Redacción ATI Cataluña**

Ciudad de Granada 131, 08018 Barcelona  
Tfn. 934125225; fax 934127113 <secretgen@ati.es>

**Redacción ATI Andalucía**

Isaac Newton, s/n, Ed. Sadiel, Isla Cartuja 41092 Sevilla, Tfn./fax 954460779 <secretand@ati.es>

**Redacción ATI Aragón**

Lagasca 9, 3-B, 50006 Zaragoza, Tfn./fax 976235181 <secretara@ati.es>

**Redacción ATI Asturias-Cantabria**

<gp-astucant@ati.es>  
**Redacción ATI Castilla-La Mancha** <gp-clmancha@ati.es>

**Redacción ATI Galicia**

Recinto Ferial s/n, 36540 Silleda (Pontevedra)  
Tfn. 986581413; fax 986580162 <secretgati@ati.es>

**Suscripción y Ventas**

<<http://www.ati.es/novatica/interes.html>>, o en ATI Cataluña o ATI Madrid

**Publicidad**

Padilla 66, 3º, dcha., 28006 Madrid  
Tfn. 914029391; fax 913093685 <novatica.publicidad@ati.es>

**Imprenta**

Derra S.A., Juan de Austria 66, 08005 Barcelona  
**Dedición legal:** B 13.134-1975 -- ISSN: 0211-2124; CODEN NOVACE

**Portada:** Antonio Crespo Foix / © ATI 2005  
**Diseño:** Fernando Agrieta / © ATI 2005

<b>editorial</b>	<b>&gt; 02</b>
<b>Las patentes de Software: el gran revolcón</b>	
<b>Andalucía: otra Ley de Colegios excluyente e inoperante</b>	
<b>en resumen</b>	<b>&gt; 02</b>
<b>El Software Libre en el diván</b>	
<i>Rafael Fernández Calvo</i>	
<b>monografía</b>	
<b>El Software Libre como objeto de estudio</b>	
(En colaboración con <b>UPGRADE</b> y con la cooperación del proyecto europeo CALIBRE)	
Editores invitados: <i>Jesús M. González Barahona, Stefan Koch</i>	
<b>Presentación. El Software Libre al microscopio</b>	<b>&gt; 03</b>
<i>Jesús M. González Barahona, Stefan Koch</i>	
<b>CALIBRE en la cresta de la ola europea del Software de Código Abierto</b>	<b>&gt; 05</b>
<i>Andrea Deverell, Par Agerfalk</i>	
<b>¿Será el movimiento del Software Libre el nuevo escalón en el modelo de organización de la producción en el sector TI?</b>	<b>&gt; 06</b>
<i>Nicolas Jullien</i>	
<b>Debian 3.1 (Sarge) como caso de estudio de medición del Software Libre: resultados preliminares</b>	<b>&gt; 11</b>
<i>Juan José Amor Iglesias, Jesús M. González Barahona, Gregorio Robles Martínez, Israel Herráiz Taberero</i>	
<b>El análisis institucional aplicado al estudio del Software Libre como "bien comunal"</b>	<b>&gt; 15</b>
<i>Charles M. Schweik</i>	
<b>Sobre proyectos de Software Libre / Código Abierto de "puerta cerrada": enseñanzas del enfoque de selección de desarrolladores para Firefox de Mozilla</b>	<b>&gt; 23</b>
<i>Sandeep Krishnamurthy</i>	
<b>Agilidad y desarrollo de Software Libre</b>	<b>&gt; 27</b>
<i>Alberto Sillitti, Giancarlo Succi</i>	
<b>secciones técnicas</b>	
<b>Enseñanza Universitaria de la Informática</b>	
<b>Propuesta de objetivos formativos para el primer curso de las Ingenierías Informáticas y de algunas estrategias docentes para conseguirlos</b>	<b>&gt; 31</b>
<i>Fermín Sánchez Carracedo, Ricard Gavalà Mestre</i>	
<b>Gestión del Conocimiento</b>	
<b>Escenarios del conocimiento: conocimiento orgánico e inorgánico</b>	<b>&gt; 36</b>
<i>Juan Baiget Solé</i>	
<b>Ingeniería del Software</b>	
<b>La gestión de la diversidad de procesos por los informáticos: reflexiones</b>	<b>&gt; 38</b>
<i>Daniilo Caivano, Corrado Aaron Visaggio</i>	
<b>Aspectos pragmáticos en el Desarrollo por el Usuario Final</b>	<b>&gt; 45</b>
<i>José Antonio Macías Iglesias</i>	
<b>Lenguajes informáticos</b>	
<b>Mono: mucho más que una implementación libre de .Net</b>	<b>&gt; 48</b>
<i>Jordi Mas i Hernández</i>	
<b>Lingüística computacional</b>	
<b>Procesamiento y aplicaciones de los corpus paralelos</b>	<b>&gt; 50</b>
<i>Xavier Gómez Guinovart</i>	
<b>Redes y servicios telemáticos</b>	
<b>Extensión del servicio A/V Streaming de CORBA. Modelo empírico basado en el control de tráfico</b>	<b>&gt; 55</b>
<i>Antonio Javier García Sánchez, Felipe García Sanchez, Pablo Pavón Mariño, Joan García Haro</i>	
<b>Encaminamiento inter-dominio con calidad de servicio basado en Overlay Entities distribuidas y QBGP</b>	<b>&gt; 61</b>
<i>Marcelo Yannuzzi, Alexandre Fonte, Xavier Masip Bruin, Edmundo Monteiro, Sergi Sánchez López, Marilía Curado, Jordi Domingo Pascual</i>	
<b>Referencias autorizadas</b>	<b>&gt; 68</b>
<b>sociedad de la información</b>	
<b>Programar es crear</b>	
<b>Un evento que mejora cada año: el Concurso Universitario de Programación de la Comunidad Autónoma de Madrid (CUPCAM)</b>	<b>&gt; 74</b>
<i>Adolfo Vázquez Rodríguez</i>	
<b>Dominó Solitario (CUPCAM 2005, problema A)</b>	<b>&gt; 75</b>
<i>Antonio Fernández Anta</i>	
<b>asuntos interiores</b>	
<b>Coordinación editorial / Programación de Novática</b>	<b>&gt; 76</b>
<b>Normas de publicación para autores / Socios Institucionales</b>	<b>&gt; 77</b>

**Monografía del próximo número:**  
**"Estandarización y normalización en Seguridad"**

Juan José Amor Iglesias, Jesús M. González Barahona, Gregorio Robles Martínez, Israel Herráiz Tabernero

Grupo de Sistemas y Comunicaciones (GSYC), Universidad Rey Juan Carlos (Móstoles, Madrid)

<{jjamor,jgb,grex,herraz}@gsyc.escet.urjc.es>

## 1. Introducción

El 6 de junio de 2005, el proyecto Debian anunció el lanzamiento oficial de la versión 3.1 de Debian GNU/Linux, conocida como "Sarge", después de casi tres años de desarrollo [6]. La distribución Debian la prepara el proyecto Debian, un grupo de casi 1.400 voluntarios (también conocidos como desarrolladores) cuyas tareas principales son la adaptación y empaquetado de los programas incluidos en la distribución [11]. Los desarrolladores de Debian empaquetan los programas que obtienen de sus autores originales (lo que en la jerga se conoce como *upstream*), asegurándose de que se integra bien con el resto del sistema Debian. Por esta razón, hay una serie de reglas que debe cumplir todo paquete y que están detalladas en el manual de Políticas de Debian [5].

Debian 3.1 incluye la mayoría de los paquetes de software libre disponibles en el momento de su publicación. Sólo en la distribución principal, compuesta únicamente de Software Libre, SL, (de acuerdo con las reglas del SL en Debian), aparecen más de 8.600 paquetes fuente. La distribución completa incluye casi 15.300 paquetes binarios, que los usuarios pueden instalar fácilmente a partir de varias fuentes de distribución, incluyendo Internet.

En este artículo hemos analizado este sistema, mostrando su tamaño, y comparándolo con otros sistemas GNU/Linux contemporáneos<sup>1</sup>. Decidimos escribir este artículo como una actualización de aquél otro titulado *Contando Patatas* [8], y *Measuring Woody* [1], que se hicieron para las versiones anteriores del sistema Debian.

La estructura de este artículo es ésta: en la siguiente sección presentaremos brevemente los métodos usados para recolectar los datos mostrados aquí; después veremos los resultados de medir Debian 3.1 (incluyendo cuentas totales, por lenguajes de programación, los mayores paquetes, etc); en la sección 3 se comentan brevemente los resultados y su interpretación, y se presentan comparaciones con Red Hat Linux y otros sistemas operativos, sean libres o privativos; finalmente, mostraremos nuestras conclusiones y las referencias citadas.

## 2. Recolección de los datos

En este trabajo sólo hemos considerado la sección principal (**main**), que es la más im-

# Debian 3.1 (Sarge) como caso de estudio de medición de Software Libre: resultados preliminares

© El presente artículo está protegido por derechos de autor según la licencia *Creative Commons Attribution-ShareAlike License*, que se puede consultar en <<http://creativecommons.org/licenses/by-sa/2.0/>>

**Resumen:** *el sistema operativo Debian es una de las distribuciones GNU/Linux más populares y no sólo por la cantidad de usuarios finales que tiene sino también porque se utiliza mucho como base para crear otros sistemas. Al margen de su popularidad, es también una de las mayores compilaciones de Software Libre realizadas, por lo que es un buen punto de partida para analizar el estado actual del Software Libre (a veces conocido como Open Source). Este trabajo es un estudio preliminar acerca de la nueva versión de Debian (la 3.1, conocida también como Sarge), que ha sido publicada oficialmente hace pocas semanas. Mostraremos el tamaño de Debian en número de líneas de código fuente (cerca de los 230 millones), qué lenguajes de programación se utilizan, cuál es el tamaño de los paquetes que integran la distribución y qué coste estimado tiene, mediante la aplicación de técnicas clásicas de estimación de costes. Esto último nos puede dar una idea del coste que tendría crear Debian 3.1 desde cero (alrededor de ocho mil millones de dólares).*

**Palabras clave:** *COCOMO, Debian, Ingeniería del Software Libre, líneas de código, Linux, Software Libre.*

### Autores

**Juan José Amor Iglesias** es Licenciado en Informática por la Universidad Politécnica de Madrid y actualmente realiza estudios de doctorado en la Universidad Rey Juan Carlos, de Madrid. Desde 1995 ha colaborado en diversas organizaciones relacionadas con el Software Libre: ha sido cofundador de LuCAS, el portal de documentación libre más conocido e Hispalinux, y colabora habitualmente con Barrapunto.com.

**Jesús M. González Barahona** es profesor e investigador en la Universidad Rey Juan Carlos de Madrid. Empezó a trabajar en la divulgación del Software Libre al principio de los 90. Desde entonces, ha realizado numerosas actividades en este área, como la organización de seminarios y cursos y la participación en diversos grupos de trabajo en Software Libre. Actualmente, colabora con diversos proyectos de Software Libre (como Debian), con asociaciones para la defensa y divulgación del Software Libre, y es consultor en temas relacionados con la adopción del Software Libre en varias empresas. Sus intereses investigadores se centran en la Ingeniería del Software Libre, en particular en las medidas cuantitativas sobre el desarrollo del Software Libre y las herramientas distribuidas utilizadas. Es socio de ATI y coordinador de la sección de Software Libre de *Novática*.

**Gregorio Robles Martínez** es estudiante de Doctorado en la Universidad Rey Juan Carlos de Madrid. Su actividad investigadora principal se centra en la ingeniería del Software Libre, específicamente en la adquisición de conocimiento sobre el desarrollo del Software Libre a partir de datos cuantitativos. Ha estado involucrado en el proyecto FLOSS y actualmente participa en la acción coordinada CALIBRE y el proyecto FLOSSWorld, todos ellos patrocinados por la Comisión Europea, programa IST.

**Israel Herráiz Tabernero** es Ingeniero Industrial y actualmente realiza estudios de Doctorado en Informática en la Universidad Rey Juan Carlos. Descubrió el Software Libre en 2000 y desde entonces ha desarrollado algunas utilidades para la Ingeniería Química.

portante y, sobre todo, la más grande en cualquier versión de Debian. Está compuesta únicamente de SL (de acuerdo con las reglas del software libre de Debian [7]). En Debian hay otras secciones, como **non-free** y **contrib**, que no vamos a tratar aquí.

La técnica usada para recolectar los datos es la siguiente: en primer lugar, se obtienen los paquetes fuente del sistema a partir de los archivos públicos disponibles en Internet, como [archive.debian.org](http://archive.debian.org) <<ftp://archive.debian.org>>. En Debian podemos conseguir los paquetes fuente o los binarios. En este estudio hemos considerado los paquetes fuente, mientras que los usuarios suelen obtener los binarios para instalarlos directamente en un ordenador. Cada paquete fuen-

te en Debian puede servir para generar uno o más paquetes binarios.

Como segundo paso, hemos analizado los paquetes, con la ayuda de la herramienta **SLOCCount**<sup>2</sup> [12]. La cuenta de líneas de código es tan sólo una estimación, debido a ciertas características de la herramienta (fundamentada en heurísticas para la identificación del código fuente y su lenguaje de programación) y el criterio que se ha seguido para la selección de los paquetes a contar [8].

El último paso comprende la identificación y eliminación de los paquetes que aparecen varias veces aunque en diferentes versiones (por ejemplo, esto es lo que ocurre con el

Lenguaje	Líneas de código fuente (SLOC)	%
C	130.847.000	57
C++	38.602.000	16,8
Shell	20.763.000	9
LISP	6.919.000	3
Perl	6.415.000	2,8
Python	4.129.000	1,8
Java	3.679.000	1,6
FORTRAN	2.724.000	1,2
PHP	2.144.000	0,93
Pascal	1.423.000	0,62
Ada	1.401.000	0,61
<b>TOTALES</b>	<b>229.496.000</b>	<b>100</b>

Tabla 1. Cuentas de líneas de código fuente de Debian 3.1 por lenguaje de programación.

compilador GCC), con el fin de no contar varias veces el mismo código fuente. Esto puede hacer que nuestra estimación quede por debajo del valor real (pues las versiones más antiguas no estarán necesariamente completamente contenidas en las más modernas). En ocasiones, sin embargo, hemos estimado tantas diferencias entre las diferentes versiones, que hemos dejado ambas. Por ejemplo, hemos contabilizado emacs y xemacs, o gcc y gnat (compilador de Ada de GNU). El resultado final de este paso es un conjunto de informes y análisis estadísticos, tomando los datos generados en pasos anteriores y analizándolos desde diferentes puntos de vista. Estos resultados son los que presentamos a continuación.

**3. Resultados de las medidas sobre Debian 3.1**

En total, siguiendo la metodología descrita, hemos contado 229.496.000 líneas de código fuente en Debian 3.1. En las siguientes secciones se presentan resultados clasificados (con todos los números de manera aproximada; véase [4] para mayor precisión).

**3.1. Lenguajes de programación**

Las cuentas de líneas de código fuente (SLOC, *Source Lines Of Code*), en número y porcentaje, clasificadas por lenguaje de programación, son, de manera muy aproximada, las que aparecen en la **tabla 1**.

Por debajo del 0,5% hay otros lenguajes como Objective C (0,37%), ML (0,31%), Yacc (0,29%), Ruby (0,26%), C# (0,23%) o Lex (0,10%). Por debajo del 0,1% encontramos algunos lenguajes más.

El gráfico de tarta de **figura 1** muestra la importancia de unos lenguajes de programación sobre otros. Casi todos los paquetes están escritos en lenguaje C, pero destaca también la abundancia del C++, siendo el lenguaje más usado en algunos de los paquetes (como OpenOffice.org o Mozilla).

El siguiente lenguaje en importancia es el de Shell, principalmente por su uso en guiones de configuración y otras tareas auxiliares de muchos paquetes. Sorprendentemente, destaca la importancia del lenguaje LISP, aunque se explica por ser el lenguaje principal de algunos paquetes grandes, como emacs, y su uso en menor medida en muchos otros. Aunque no se muestra en estos resultados, hay una tendencia a la pérdida de importancia de C frente a otros lenguajes; mientras que se observa un crecimiento muy importante de otros lenguajes como Java, PHP y Python.

**3.2. Los paquetes más grandes**

A continuación daremos una lista de los paquetes más grandes (todos ellos tienen más de 2 millones de líneas de código), ordenada por tamaño, del sistema Debian 3.1. Para cada paquete daremos su nombre, su versión, el número de líneas de código fuente totales (SLOCs), la proporción de lenguajes usados y una pequeña descripción del propósito del paquete.

■ OpenOffice.org (1.1.3): 5.181.000 SLOC. C++ supone 3.547.000 SLOC. C supone 1.040.000 SLOC. Además, se encuentra código escrito en 15 lenguajes distintos, tanto lenguajes usados para guiones (scripts), como Shell, TCL, Python y AWK; como otros lenguajes (Pascal, Java, Objective-C, List, etc). Openoffice.Org es un completo paquete ofimático.

■ Linux kernel (2.6.8): 4.043.000 SLOC. C supone 3.794.000 SLOC, mientras que el resto lo componen *makefiles*, ensamblador y scripts en varios lenguajes. Linux es el núcleo del sistema operativo Debian 3.1.

■ NVU (N-View) (0.80): 2.480.000 SLOC. Casi todo está escrito en C++, con 798.000 SLOC. Hay otros lenguajes, sobre todo para guiones. N-View es un sistema de autoría web, que rivaliza con sistemas privativos como Microsoft FrontPage.

■ Mozilla (1.7.7): 2.437.000 SLOC. Casi todo está escrito en C++, con más de 1.567.000 SLOC, aunque tiene una parte importante de C, con 789.000 SLOC. Mozilla es un conjunto de programas para el usuario de Internet (navegador web, cliente de correo, etc).

■ GCC-3.4 (3.4.3): 2.422.000 SLOC. C supone 1.031.000 SLOC, mientras que Ada comprende 485.000 SLOC y C++, 244.000 SLOC. El resto de los lenguajes tienen una presencia anecdótica. GCC es la popular colección de compiladores de GNU.

■ XFS-XTT (1.4.1): 2.347.000 SLOC. Casi todo está escrito en lenguaje C, con 2.193.000 SLOC. Es un servidor de tipos de letra X-TrueType.

■ XFree86 (4.3.0): 2.316.000 SLOC. Principalmente está escrito en C, con 2.177.000 SLOC. Es una realización del sistema X Window, que incluye servidor y programas básicos.

■ VNC4 (4.0): 2.055.000 SLOC. VNC4 es un sistema de acceso remoto a la consola, hecho fundamentalmente en lenguaje C, con 1.920.000 SLOC.

■ Insight (6.1): 1.690.000 SLOC, principalmente escrito en C (1.445.000 SLOC). Insight es un depurador gráfico basado en GDB.

■ kFreeBSD5-source (5.3): 1.630.000 SLOC. Se trata del código fuente del núcleo del sistema operativo FreeBSD 5.3, que es la base del futuro sistema Debian GNU/kFreeBSD.

■ Obsérvese que esta clasificación es dependiente de las decisiones tomadas por los desarrolladores a la hora de dividir el software ofimático.

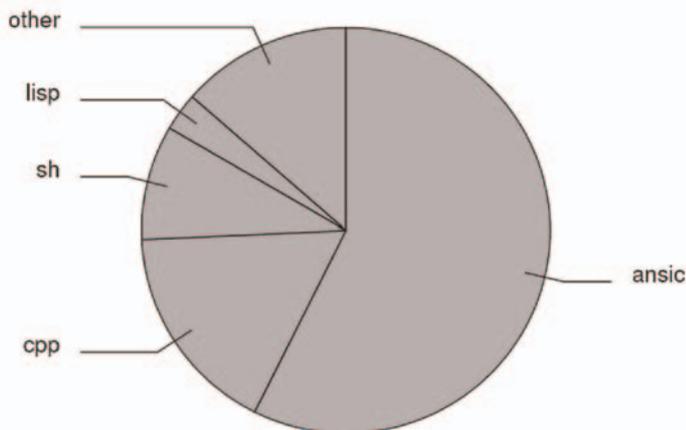


Figura 1. Distribución en líneas de código para los lenguajes más usados en Debian 3.1.



Debian 3.1 incluye la mayoría de los paquetes de Software Libre disponibles

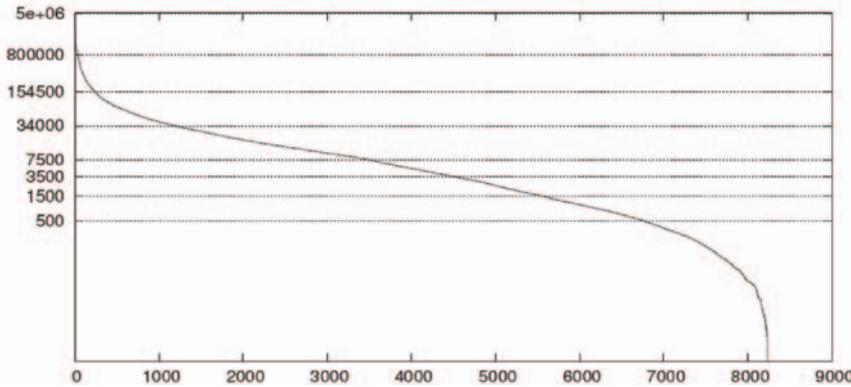


Figura 2. Tamaños de paquete en Debian 3.1 (el número de líneas, SLOC, se representa en escala logarítmica).

ware en paquetes. Por ejemplo, si todas las extensiones de emacs hubiesen sido empaquetadas juntas, posiblemente estaría en la lista de los diez paquetes más grandes. De todas formas, el paquete fuente Debian normalmente coincide con lo que el autor original considera como paquete, que se basa normalmente en principios de modularización del software.

La figura 2 muestra los tamaños de todos los paquete de Debian 3.1. Hemos observado en el estudio de las diferentes versiones de Debian publicadas, desde la versión 2.0 (publicada en 1998) hasta la 3.0 (de 2002), que el tamaño medio de los paquetes se mantiene alrededor de las 23.000 líneas de código (véase [10]). En el caso de Debian 3.1 este tamaño medio se ha incrementado a las 26,600 líneas. El motivo de este incremento no está claro, pero puede deberse a que el número de paquetes crece a un ritmo mayor que el de desarrolladores, lo que hace difícil ese equilibrio.

**3.3. Estimaciones de coste y esfuerzo**

El modelo COCOMO (COntstructive COSt MOdel) [2] nos sirve para dar una estimación aproximada del esfuerzo económico y humano requerido para generar un programa de cierto tamaño. Como entrada, toma el tamaño en líneas de código fuente (SLOC). Dado que estas estimaciones están pensadas para los procesos ‘clásicos’ de generación de software y para proyectos grandes, los resultados obtenidos deben interpretarse siempre con mucha precaución. En cualquier caso, utilizaremos el modelo básico de COCOMO para dar una estimación del coste basada en el tamaño.

Utilizando el número de líneas de código medidas en Debian, los resultados que ofrece el modelo COCOMO básico son los siguientes:

- Número de líneas de código (SLOC): 229.495.824
- Esfuerzo estimado: 714,440.52 meses-hombre (59,536.71 años-hombre). Según la fórmula:  $2.4 * (KSLOC ^ 1.05)$
- Tiempo estimado de desarrollo: 105.84 meses (8.82 años). Según la fórmula:  $2.5 * (Effort ^ 0.38)$
- Coste económico estimado: 8,043,000,000 dólares americanos.

Para llegar a estos resultados, cada proyecto ha sido estimado como si se desarrollase independientemente de los demás (lo cual será cierto en casi todos los casos). Para calcular el coste estimado, se han utilizado los salarios medios de los programadores a tiempo completo en el año 2000, de acuerdo con [3], que era de 56.286 dólares al año, y un factor de sobrecarga de 2.4 (para más detalle sobre este factor y otros detalles del modelo de estimación, véase [13]).

Sistema operativo	Líneas de código fuente (SLOC)
Microsoft Windows 3.1 (abril 1992)	3.000.000
Sun Solaris (octubre 1998)	7.500.000
Microsoft Windows 95 (agosto 1995)	15.000.000
Red Hat Linux 6.2 (marzo 2000)	17.000.000
Microsoft Windows 2000 (febrero 2000)	29.000.000
Red Hat Linux 7.1 (abril 2001)	30.000.000
Microsoft Windows XP (2002)	40.000.000
Red Hat Linux 8.0 (septiembre 2002)	50.000.000
Fedora Core 4 (versión previa; mayo 2005)	76.000.000
Debian 3.0 (julio 2002)	105.000.000
Debian 3.1 (junio 2005)	229.500.000

Tabla 2. Comparación de tamaño de diversos sistemas operativos.

## 4. Comparación con otros sistemas

Para darle más contexto a todos estos números, en la **tabla 2** se muestran los tamaños estimados de otros sistemas operativos. Estos tamaños han sido obtenidos de diversas fuentes, mostradas en [10], y se proporcionan en líneas de código (siempre como aproximaciones).

Casi todos estos números son estimaciones (de hecho, lo son para todos los casos, excepto para Red Hat Linux, Fedora Core y Debian) y no están detallados, de manera que es difícil saber qué se consideró como línea de código fuente en todos esos casos (por ejemplo, si se contaron los comentarios o no). Sin embargo, para el objetivo del artículo proporciona una visión suficientemente amplia.

Nótese también que, mientras que Red Hat y Debian incluyen muchas aplicaciones, tanto Microsoft como Sun suelen incluir muchas menos (y en cualquier caso, son aplicaciones más sencillas).

Si se contabilizasen las herramientas que usualmente acompañan a un sistema de Microsoft o Sun completamente instalado, los tamaños serían bastante mayores. Sin embargo, también es cierto que esas aplicaciones son desarrolladas por terceros, y no sería comparable con el caso de Debian, en el que su equipo de desarrolladores es responsable de todo el conjunto.

Podemos ver que las distribuciones de Linux en general y Debian 3.1 en particular, forman algunos de los mayores paquetes de software mantenidos por un equipo de desarrolladores.

## 5. Conclusiones y trabajo futuro

Debian es uno de los mayores sistemas software del mundo, probablemente el mayor. Su tamaño ha crecido versión tras versión, siendo la versión 3.1 el doble de grande que la 3.0. Los lenguajes principales utilizados para desarrollar los paquetes incluidos en Debian son, desde varias versiones, C y C++. El conjunto de C, C++ y Shell representa más del 75% de todo el código disponible en Debian. El número de paquetes continúa creciendo y se duplica aproximadamente cada dos años.

El sistema Debian GNU/Linux, mantenido por un grupo de voluntarios distribuidos por todo el mundo, muestra en primer lugar una buena salud y una tendencia de continuo crecimiento. Independientemente de su enorme tamaño, continúa entregando versiones estables.

Sin embargo, hay algunos aspectos que cuestionan la sostenibilidad del proyecto en el futuro. Por ejemplo, el tamaño medio nos da

una evidencia de comportamiento inestable, probablemente debido a un crecimiento de los paquetes más rápido que el número de mantenedores de Debian. Por otro lado, no debemos obviar que se ha tardado casi tres años en publicar una nueva versión estable y que la fecha de publicación ha sido retrasada varias veces.

En lo referente a otros sistemas software, no hay muchos estudios detallados en tamaño de sistemas operativos modernos. De ellos, el más destacado es el trabajo de David A. Wheeler, que midió el tamaño de Red Hat 6.2 y Red Hat 7.1. Otras referencias nos proporcionan sumas totales del tamaño de algunos sistemas de Sun y Microsoft, pero no tienen más detalle que el mencionado. En cualquier caso, Debian supera, de largo, en tamaño, a todas las demás, aunque esta comparación debe considerarse con cuidado.

Para finalizar, es importante destacar el hecho de que este artículo ofrece estimaciones de un estudio preliminar (ya que la versión no estaba totalmente finalizada cuando se hizo). Sin embargo, creemos que es lo suficientemente precisa para llevarnos a algunas conclusiones y comparar Debian 3.1 con otros sistemas software.

## Agradecimientos

Este trabajo ha sido financiado en parte por la Comisión Europea, dentro del CALIBRE CA, programa IST, número de contrato 004337; por la Universidad Rey Juan Carlos con el proyecto PPR-2004-42 y por la CICyT dentro del proyecto TIN2004-07296.

## Referencias

- [1] Juan José Amor, Gregorio Robles, Jesús M. González-Barahona. *Measuring Woody: The size of Debian 3.0* (pendiente de publicación). Estará disponible en <http://people.debian.org/~jgb/debian-counting/>.
- [2] Barry W. Boehm. *Software Engineering Economics*, Prentice Hall, 1981.
- [3] Computer World. *Salary Survey 2000*. <http://www.computerworld.com/cwi/careers/surveysandreports>.
- [4] Jesús M. González Barahona, Gregorio Robles, Juan José Amor. *Debian Counting*. <http://libresoft.urjc.es/debian-counting/>.
- [5] Debian Project. *Debian Policy Manual*. <http://www.debian.org/doc/debian-policy/>.
- [6] Debian Project. *Debian GNU/Linux 3.1 released (June 6th 2005)*. <http://www.debian.org/News/2005/20050606>.
- [7] Debian Project. *Debian Free Software Guidelines (part of the Debian Social Contract)*. [http://www.debian.org/social\\_contract](http://www.debian.org/social_contract).
- [8] Jesús M. González Barahona, Miguel A. Ortuño Pérez, Pedro de las Heras Quirós, José Centeno González, Vicente Matellán Olivera. *Contando patatas: el tamaño de Debian 2.2./ Counting potatoes: The size of Debian 2.2*. *Novática*, nº 151 (nov.-dic. 2001), <http://www.ati.es/novatica/2001/154/154-30.pdf> (en español); *UPGRADE*, vol. 2, issue 6, December 2001, <http://upgrade-cepis.org/issues/2001/6/up2-6Gonzalez.pdf> (en inglés).
- [9] Jesús M. González-Barahona, Gregorio Robles, Miguel Ortuño-Pérez, Luis Rodero-Merino, José Centeno-González, Vicente Matellán-Olivera, Eva Castro-Barbero, Pedro de-las-Heras-Quirós. *Anatomy of two GNU/Linux distributions*. Capítulo del libro «Free/Open Source Software Development» editado por Stefan Koch y publicado por Idea Group, Inc., 2004.
- [10] Gregorio Robles, Jesús M. González-Barahona, Luis López, Juan José Amor. *Toy Story: an analysis of the evolution of Debian GNU/Linux*, noviembre 2004 (pendiente de publicación). Borrador disponible <http://libresoft.urjc.es/debian-counting/>.
- [11] Gregorio Robles, Jesús M. González-Barahona, Martin Michlmayr. *Evolution of Volunteer Participation in Libre Software Projects: Evidence from Debian*, julio 2005, Proceedings of the First International Conference on Open Source Systems. Genova, Italy, pp. 100-107. <http://gsync.escet.urjc.es/~grex/volunteers-robles-jgb-martin.pdf>.
- [12] David Wheeler. *SLOccount*. <http://www.dwheeler.com/sloccount/>.
- [13] David A. Wheeler. *More Than a Gigabuck: Estimating GNU/Linux's Size*. <http://www.dwheeler.com/sloc>.

## Notas

- 1 Los sistemas GNU/Linux se conocen también como 'distribuciones'.
- 2 Concretamente hemos utilizado la versión 2.26, que es capaz de reconocer 27 lenguajes de programación.