

**Novática, revista fundada en 1975, es el órgano oficial de expresión y formación continua de ATI (Asociación de Técnicos de Informática)**

ATI es miembro de CEPIS (*Council of European Professional Informatics Societies*) y tiene un acuerdo de colaboración con ACM (*Association for Computing Machinery*). Tiene asimismo acuerdos de vinculación o colaboración con AdaSpain, AI<sup>2</sup> y ASTIC

<http://www.ati.es/novatica/>

**CONSEJO ASesor DE MEDIOS DE COMUNICACION**

Pere Lluís Barbrà, Rafael Fernández Calvo, José Gómez, Manuel Orti Mezquita, Nacho Navarro, Fernando Sanjuán de la Rocha (Presidente), Miquel Sarries, Carlos Sobrino Sánchez, Manuel Solans

**Coordinación Editorial**  
Rafael Fernández Calvo <rfcalvo@ati.es>

**Composición y autoedición**  
Jorge Llácer

**Administración**  
Tomás Brunete, Joan Aguiar, María José Fernández

**SECCIONES TÉCNICAS: COORDINADORES**

- Arquitecturas**  
Antonio Gonzalez Colás (DAC-UPC) <antonio@ac.upc.es>
- Bases de Datos**  
Mario G. Plattini Velthuis (EUI-UCLM) <mpiattini@inf-cr.uclm.es>
- Calidad del Software**  
Juan Carlos Granja (Universidad de Granada) <jcgranja@goliat.ugr.es>
- Derecho y Tecnologías**  
Isabel Hernando Collazos (Fac. Derecho de Donostia, UPV) <ihernando@legaltek.net>
- Enseñanza Universitaria de la Informática**  
Cristóbal Pareja Flores (Dep. Sistemas Informáticos y Programación-UCM) <cpareja@si.ucm.es>
- Euro/Efecto 2000**  
Joaquín Ríos Boutin <jrios@ati.es>
- Informática Gráfica**  
Roberto Vivó (Eurographics, sección española) <rvivo@dsic.upv.es>
- Informática Médica**  
Valentín Masero Vargas (DI-UNEX) <vmasero@umex.es>
- Ingeniería del Software**  
Luís Fernández (PRIS-EI/UEM) <lufern@dpri.esi.uem.es>
- Inteligencia Artificial**  
Federico Barber, Vicente Botti (DSIC-UPV) <fjbotti\_fbarber@dsic.upv.es>
- Interacción Persona-Computador**  
Julio Abascal González (FI-UPV) <julio@si.ehu.es>
- Internet**  
Alonso Álvarez García (TID) <alonso@ati.es>
- Llorenç Pagés Casas (Atlante)** <pages@ati.es>
- Lengua e Informática**  
M. del Carmen Ugarte (IBM) <cugarte@ati.es>
- Lenguajes informáticos**  
Andrés Marín López (Univ. Carlos III) <amarin@it.uc3m.es>
- J. Ángel Velázquez (ESCET-URJC)** <a.velazquez@escet.urjc.es>
- Libertades e Informática**  
Alfonso Escolano (FIR-Univ. de La Laguna) <aescolan@ull.es>
- Lingüística computacional**  
Xavier Gómez Guinovart (Univ. de Vigo) <gomez@uvigo.es>
- Manuel Palomar (Univ. de Alicante)** <mpalomar@dsi.ua.es>
- Profesión informática**  
Rafael Fernández Calvo (ATI) <rfcalvo@ati.es>
- Miquel Sarries Grinyó (Ayto. de Barcelona)** <msarries@ati.es>
- Seguridad**  
Javier Areitio (Redes y Sistemas, Bilbao) <jareitio@orion.deusto.es>
- Sistemas de Tiempo Real**  
Alejandro Alonso, Juan Antonio de la Puente (DIT-UPM) <jaalonso.jpuede@dit.upm.es>
- Software Libre**  
Jesús M. González Barahona, Pedro de las Heras Quirós (GSYC, URJC) <jgibpheras@gsyc.escet.urjc.es>
- Tecnología de Objetos**  
Esperanza Marcos (URJC) <e.marcos@escet.urjc.es>
- Gustavo Rossi (LIFIA-UNLP, Argentina)** <gustavo@sol.info.unpl.edu.ar>
- Tecnologías para la Educación**  
Benita Compostela (F. CC. PP.-UCM) <benita@diad.umet.es>
- Josep Sales Rufí (ESPIRAL)** <jsales@pie.xtec.es>
- Tecnologías y Empresa**  
Pablo Hernández Medrano <phmedrano@terra.es>
- TIC para la Sanidad**  
Valentín Masero Vargas (DI-UNEX) <vmasero@umex.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos. Novática permite la reproducción de todos los artículos, salvo los marcados con © o *copyright*, debiéndose en todo caso citar su procedencia y enviar a Novática un ejemplar de la publicación.

- Coordinación Editorial y Redacción Central (ATI Madrid)**  
Padilla 66, 3º, dcha., 28006 Madrid  
Tf:914029391; fax:913093685 <novatica@ati.es>
- Composición, Edición y Redacción ATI Valencia**  
Palomino 14, 2º, 46003 Valencia  
Tf./fax: 963918531 <secreval@ati.es>
- Administración, Suscripciones y Redacción ATI Cataluña**  
Via Laietana 41, 1º, 08003 Barcelona  
Tf:934125235; fax: 934127713 <secregen@ati.es>
- Redacción ATI Andalucía**  
Isaac Newton, s/n, Ed. Sadiel, Isla Cartuja 41092 Sevilla  
Tf./fax: 954460779 <secreand@ati.es>
- Redacción ATI Aragón**  
Lagasca 9, 3-B, 50006 Zaragoza  
Tf./fax: 976235181 <secreara@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)  
Tf:986581413; fax: 986580162 <secregal@ati.es>
- Publicidad:** Padilla 66, 3º, dcha., 28006 Madrid  
Tf:914029391; fax:913093685 <novatica.publicidad@ati.es>

**Imprenta:** Gráficas Sierra S.L., Atenas, 3, int. bajos, 08006 Barcelona.  
**Depósito Legal:** B 15.154-1975  
**ISBN:** 0211-2124; CODEN NOVACE

**Portada:** Antonio Crespo Foix / © ATI 2001

**SUMARIO**

En resumen: Libertad y madurez 2

NOVIEMBRE - DICIEMBRE 2001

154

**Monografía: «Software Libre/Fuente Abierta: hacia la madurez»**  
(En colaboración con *Informatik/Informatique* y *Upgrade*)  
Coordinada por Joe Ammann, Jesús M. González-Barahona y Pedro de las Heras Quirós

**Presentación: hacia la madurez** 3  
*Joe Ammann, Jesús M. González-Barahona, Pedro de las Heras Quirós*

**Actualidad del software libre** 5  
*Pedro de las Heras Quirós, Jesús M. González-Barahona*

**Eldañó viene de La Haya** 14  
*Richard Stallman*

**Iniciativas europeas sobre el uso de software libre en el Sector Público** 17  
*Juan Jesús Muñoz Esteban*

**Open Source en un gran banco suizo** 22  
*Klaus Bucka-Lassen, Jan Sorensen*

**El proyecto GNU Enterprise: software de aplicación para la empresa** 25  
*Neil Tiffin, Reinhard Müller*

**Contando patatas: el tamaño de Debian 2.2** 30  
*Jesús M. González-Barahona, Miguel A. Ortuño, Pedro de las Heras, José Centeno, Vicente Matellán*

**La crisis del software libre científico** 38  
*David Santo Orcero*

**El proyecto Debian GNU/Linux** 41  
*Javier Fernández-Sanguino Peña*

**Sistemas de ficheros con Journaling en Linux** 45  
*Ricardo Galli*

**Secciones técnicas**

**Ingeniería del Software**

**Un nuevo modelo de evaluación de procesos de software para PYMES a partir de SPICE (ISO/IEC TR-15504-5)** 52  
*Antonia Mas Pichaco, Ángel Igelmo Ganzo, Esperança Amengual Alcover, Gabriel Fontanet Nadal*

**Profesión informática**

**El futuro de la Ingeniería del Software** 57  
*Karol Frühauf*

**Seguridad**

**De mí misma libremente Dios, que del Sircam ya me libro yo (y II)** 59  
*Mª del Carmen Ugarte García*

**Tecnología de Objetos**

**¿Es conveniente la Orientación a Objetos en un primer curso de programación?** 64  
*Jesús J. García Molina*

**Referencias autorizadas** 69

**Sociedad de la Información**

**Programar es crear**

**Ancho de banda en Internet** 72  
*Concurso de Programación ACM 2000: programa E*

**«Fila y asociados»: solución** 73  
*Álvaro Martínez Echevarría*

**Asuntos Interiores**

**Coordinación editorial / Programación de Novática** 76

**Normas de publicación para autores / Socios Institucionales** 77

**Monografía del próximo número: «Gestión del Conocimiento y TIC»**

Klaus Bucka-Lassen, Jan Sorensen  
aragost AG

{<klaus.bucka-lassen, jan.sorensen@aragost.com>}

**Traducción:** Julio Ayesa, Agustín Palomar (Grupo de Lengua e Informática de ATI)

**Resumen:** ¿qué podría inducir a un gran banco suizo a usar un producto **Open Source** (código fuente abierto)? Este artículo usa el ejemplo de **Jakarta Struts** para explicar las ventajas y desventajas del software **Open Source** y remarcar cuáles de éstas son significativas para un proveedor de servicios financieros. En él se describen los problemas que aparecen y que es los que convence a la dirección a usar **Struts** para el desarrollo de su aplicación *Web*.

**Palabras clave:** *Jakarta Struts, Open Source, gran banco suizo, ventajas y desventajas, experiencia.*

## 1. Los comienzos

Al igual que otros «productos de moda», *Open Source* (código fuente abierto) nació en un entorno académico con productos como **Emacs**, **LaTeX**, **Mosaic**, varios compiladores (**C**, **C++**, **Smalltalk**, etc.) y software asociado [GNU, Mosaic]. Y fue adoptado a continuación por usuarios personales.

No hace mucho tiempo, alguien que se hubiera atrevido a comparar **Linux** con **Windows** hubiera causado risa. Hoy, a la vista del impresionante éxito experimentado por este sistema operativo, sus críticos están siendo silenciados. La cuota de mercado de Linux está aumentando constantemente, incluso en el entorno comercial.

Linux ha dejado de ser un producto para idealistas y enemigos de Microsoft y se ha convertido en un serio opositor de Windows. El valor de un sistema operativo es más difícil de establecer que el de una aplicación de software. Un sistema operativo *per se* casi no se usa --sólo las aplicaciones que son desarrolladas para este sistema operativo hacen que sea interesante. El mejor sistema operativo del mundo no tendría éxito sin el software. puesto que requiere una masa crítica de usuarios y desarrolladores. A pesar de esto, Linux ha sido un éxito. Esto demuestra que *Open Source* no es simplemente una moda pasajera. Las razones del éxito de Linux son probablemente las muchas ventajas, inherentes en parte, que el software *Open Source* ofrece en comparación con el software propietario. Esto queda remarcado más abajo en un ejemplo tomado de nuestra propia experiencia.

## 2. Struts

**Struts** es una estructura *Model-View-Controller* (**MVC**) de *Open Source* para aplicaciones *Web* que está construida usando **Java** y **JSP/Servlets** [Struts]. Un gran banco suizo

recientemente decidió usarlo como marco para su desarrollo de aplicaciones *Web* internas .

*Model-View-Controller* es un patrón de diseño ampliamente aceptado que define la construcción lógica, o arquitectura, de una aplicación.

*Model* describe el objeto de negocio y la lógica asociada. *View* refleja su presentación en la pantalla (**GUI**). **MVC** estipula que estos dos elementos sólo pueden comunicarse a través de un *Controller* y, por tanto, están separados uno del otro. Esto significa, por ejemplo, que es relativamente sencillo modificar la vista (*view*) sin que el modelo se vea afectado. La experiencia ha mostrado que esto es un enfoque muy bueno que hace más económico el desarrollo y, en un nivel inferior, el mantenimiento de la aplicación.

**Struts** es una estructura que soporta principalmente y simplifica el desarrollo del *Controller* y la *View*. El desarrollo de la lógica de negocio no se ve influenciado por **Struts**.

**Struts**, junto con otros productos como **Log4J** y **Tomcat**, llega bajo los auspicios del proyecto **Jakarta** promovido por la Fundación de Software **Apache**. Es relativamente nuevo, pues la versión 1.0 fue liberada en Junio de 2001.

Existen varias maneras diferentes de trabajar con **Struts**. En primer lugar, uno puede involucrarse activamente en las discusiones sobre **Struts** como usuario de listas de correo. Se puede informar de fallos y sugerir nuevas características. Pero también uno mismo puede desarrollar nuevo código para **Struts** -- en este caso un comité decide si será incluido en la próxima versión.

### Autores

**Klaus Bucka-Lassen** es Master en Informática por la Universidad de Aarhus (Dinamarca). Ha estado trabajando durante cinco años como desarrollador de **Smalltalk** y **Java** en Suiza y Australia. Actualmente trabaja como arquitecto de sistemas para un proyecto piloto de **Struts** en uno de los principales bancos suizos.

**Jan Sorensen** es Master en Informática por la Universidad de Aarhus (Dinamarca). Ha estado trabajando durante siete años como desarrollador de **C++**, **Smalltalk** y **Java** en Dinamarca, Suiza y los Estados Unidos. "Contribuyente" a **Struts**.

Ambos son co-fundadores y co-propietarios de aragost AG, empresa especializada en la consultoría y el soporte de productos *Open Source* en un entorno orientado a objetos (**Smalltalk** y **Java**). <<http://www.aragost.com/>>

### 3. Struts en funcionamiento

Desde el principio de año, Struts ha sido usado en un proyecto piloto en este gran banco. ¿Qué ventajas se ha obtenido el banco debido al hecho que Struts sea software *Open Source*?

**Seguridad.** El hecho de que el código fuente está disponible públicamente asegura que no ocurre nada que no debería ocurrir. Esto es fundamental para un banco, puesto que no debería haber ningún agujero en la seguridad de una aplicación bancaria. Es casi imposible construir «puertas traseras» en Struts, pues sería claramente visible para cualquiera que mirara el código.

**Robustez y fiabilidad.** Incluso teniendo en cuenta que Struts es un producto totalmente nuevo, da la impresión de ser altamente estable. Los fallos son rápidamente localizados y solucionados puesto que el código fuente está siendo examinado constantemente. Por otra parte, los desarrolladores escriben mejor código si saben que va a ser publicado. Uno de los autores de este artículo identificó recientemente un fallo en Struts. Se refería específicamente a un método que debería estar «a salvo de amenazas» pero no lo estaba. Descubrió el fallo por casualidad mientras leía código fuente, corrigió el fallo y envió una versión corregida. Hasta entonces nadie había encontrado ese problema porque el código sólo se comportaba erróneamente bajo ciertas circunstancias no habituales. Esto significa que fue prácticamente imposible reproducir el fallo haciendo mucho más difícil localizarlo. ¡Así es como los programas *Open Source* llegan a ser más robustos y estables!

**Formación, documentación y soporte.** Se ha escrito ya un considerable número de artículos, recomendaciones, instrucciones y ejercicios en relación con Struts. El soporte mediante las listas de correo funciona de una manera considerablemente más rápida efectiva y amigable que lo que uno puede esperar de la mayoría de los más importantes suministradores de software.

**Facilidad para encontrar fallos.** Localizar fallos en el código propio es un arte (no estamos hablando de fallos en el código Struts). Generalmente se usa un *debugger* para este propósito, con el cual la aplicación puede ser ejecutada línea por línea. Este procedimiento sólo es posible porque Struts es un producto *Open Source*.

**Modificabilidad y Extensibilidad.** El código Struts puede ser modificado, sin embargo, si uno quiere mantener la posibilidad de posteriores migraciones a nuevas versiones de Struts sin grandes problemas, no es recomendable hacer modificaciones permanentes en el código Struts. Sabido esto, pueden ser muy útiles modificaciones temporales, por ejemplo, para registro de datos (*logging*). También pueden ser solventados de esta manera toda clase de fallos durante un corto periodo de tiempo a la espera de las «soluciones oficiales».

**Evaluación.** Como Struts es un producto *Open Source*, se puede probar cualquier clase de situación sin el riesgo de infringir la licencia. Esto significa que se pueden hacer evaluaciones en profundidad. Si Struts fuera un software propietario la decisión

de usarlo o no hubiera debido tomarse antes de que fuera posible probar una versión completa del producto.

**Precio de compra.** Para hablar de todo, el precio de compra también debe ser mencionado aquí. Comparado con las otras ventajas mencionadas anteriormente, el hecho de que Struts esté libre de cargo no es un argumento particularmente significativo en su favor -- en particular no para un gran banco.

### 4. El precio

La evaluación de costes forma parte de cualquier empresa seria. Los costes en un escenario *Open Source* son fáciles de confrontar con los siguientes hechos:

- **Falta de responsabilidad del producto.** Con *Open Source*, nadie es responsable del producto. Por lo tanto, nadie es culpable de ningún daño que pueda surgir como resultado de un producto *Open Source* defectuoso. En este contexto, sin embargo, habría que señalar que los productores de software propietario también afirman que no pueden asumir responsabilidades por ningún daño que pueda surgir del uso de su producto (*disclaimer*, o exención de responsabilidad).

- **No hay un punto de soporte oficial.** Los productores de productos propietarios ofrecen a menudo un servicio de soporte caro. Éste no es el caso habitual con el software *Open Source*. Por eso, es necesario establecer un punto de soporte interno, o hay que obtener soporte de una tercera empresa especializada en el producto *Open Source*. Además, para el banco que se menciona en este artículo (y probablemente para la mayoría de otras grandes empresas similares) las normas y reglas internas para empleados les prohíben en mayor o menor medida tomar parte en foros de discusión en Internet (lo que impide el uso de soporte directo mediante la lista de correo de usuarios de Struts, o al menos lo dificulta).

- **Dificultades para planificar la disponibilidad de nuevas versiones.** Como los desarrolladores disponibles en un proyecto *Open Source* difícilmente pueden ceñirse a una planificación, es difícil, por no decir imposible, establecer un plan de disponibilidad de nuevas versiones. La resolución de errores y la implementación de nuevas prestaciones no puede garantizarse, ni tampoco pueden darse fechas para arreglos (*fixes*). Antes de que esto ocurra, hay que encontrar un voluntario con tiempo y ganas suficientes. Para la mayoría de las aplicaciones, las ventajas parecen merecer la pena (pero a pesar de ello, no es del todo fácil convencer a los directivos de un gran proveedor de servicios financieros de que utilice software *Open Source*). Para conseguir la aprobación para utilizar Struts en un gran banco, hubo que sortear algunos obstáculos, no solamente los que hundan sus raíces en los prejuicios y la incertidumbre al enfrentarse a lo desconocido, sino también los que tienen que ver más con intereses personales.

### 5. Argumentos que convencieron finalmente a los directivos a favor de Struts

- **La credibilidad de la organización que hay detrás de Struts.**

La Apache Software Foundation es reconocida y ya ha llevado al mercado otros varios productos *Open Source* con éxito. El más bien conocido de ellos es el servidor http Apache, el líder del mercado entre los servidores Web. Incluso **Websphere** de IBM se basa en Apache.

- **Referencias.** Como ya se ha mencionado, innumerables artículos y ejercicios sobre Struts están disponibles en la Internet. También está disponible un libro que dedica un capítulo completo a Struts [JSP]. Pero lo que convenció a la dirección más que nada fue el hecho de que IBM canta las alabanzas de Struts en su página oficial [IBM].
- **Entrega definitiva.** Ahora está disponible una entrega definitiva de Struts. Pero un par de meses antes de la decisión, únicamente se disponía de una *pre-release*.
- **Simplicidad e independencia.** Struts es simple y los desarrolladores aprenden a usarlo rápidamente. Struts se construye exclusivamente sobre tecnologías no propietarias, ampliamente probadas y examinadas tales como Servlets, **JSP** y **XML**.
- **Compatibilidad.** Struts podrá ejecutarse en todos los principales servidores de aplicaciones Web incluyendo **Tomcat**, **Websphere** de IBM, **WebLogic** de BEA y **SilverStream**.

En el peor caso, el código de *Open Source* es «tan malo como cualquier código escrito en casa». O por decirlo de otra forma, usted puede descargar Stratus y utilizarlo como punto de inicio para su propio producto; nadie va a obligarle a migrar a nuevas versiones de Struts. Puede enmendar Struts para adaptarse a sus propias necesidades. La alternativa a esto sería reinventar la rueda una vez más, lo que podría significar unos costes considerablemente más altos y un «tiempo de puesta en mercado» más prolongado.

## 6. Conclusión

El software mencionado en este artículo, que se desarrolló usando Struts, ahora está ejecutándose con éxito en un entorno real. Sin Struts, el desarrollo habría tomado mucho más tiempo y la calidad, junto con la mantenibilidad, se habrían resentido. El éxito se debió en parte al hecho de que MVC es un buen patrón de diseño y Struts lo implementa excepcionalmente bien, pero no menos porque Struts es *Open Source* y por lo tanto tiene las ventajas mencionadas anteriormente. A diferencia de lo que podría parecer, no hubo problemas dignos de mención. La curva de aprendizaje adicional fue modesta. Lo que faltaron fueron casos de estudio y «patrones de la mejor práctica» de Struts. Tuvimos que aprender las cosas por el camino difícil.

## 7. El futuro

Según la lista de correo de usuarios de Struts, al menos otro gran banco internacional, Deutsche Bank, ha optado por Struts últimamente. Otros proveedores de servicios financieros, también en Suiza, están evaluando actualmente sus pros y contras.

## 8. Referencias

[Struts] Página principal de Jakarta Struts <<http://jakarta.apache.org/struts/>>

[OSD] Definición de *Open Source* <<http://www.opensource.org/docs/definition.html>>

[GNU] Página principal de GNU <<http://www.gnu.org/>>

[Mosaic] Licencia Mosaic <<http://archive.ncsa.uiuc.edu/SDG/Software/Mosaic/License/LicenseInfo.html>>

[FSF] Fundación Software Libre <<http://www.fsf.org/gnu/the-gnu-project.html>>

[JSP] «JSP Profesional», 2ª Edición; Wrox Press Inc., 29 S. LaSalle St, Suite 520, Chicago, Illinois 60603, USA; **Simon Brown, Robert Burdick, Jayson Falkner, et al.**

[IBM] «Struts, an open-source MVC implementation» <<http://www-106.ibm.com/developerworks/ibm/library/j-struts/>>

[Linux] «The Cathedral and the Bazaar» <<http://tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>>