

# NetBeans 5.0

FRANCISCO MORERO PEYRONA  
(<http://binfactory.com/peyrona>)

Aunque mantiene su "Look and Feel", Netbeans 5.0 es prácticamente una reedición completa de uno de los primeros IDE para Java escrito en Java.

## Introducción

Si todo ha salido bien, esta publicación coincidirá con una nueva versión mayor del entorno de desarrollo oficial para Java: NetBeans 5.0, la cual está prevista para diciembre de 2005 o enero de 2006.

Esta versión es mucho más que un conjunto de nuevas herramientas y corrección de errores, ya que supone casi una nueva reedición del producto, habiéndose modificado grandes porciones del código que internamente se encarga de los principales procesos de NetBeans. Sin embargo, y por suerte para los usuarios de versiones anteriores de este entorno, se ha mantenido el mismo aspecto y modus operandi, lo que en Java llamamos "Look and Feel".

Aun cuando en este artículo principalmente nos centraremos en el IDE, también cubriremos otras herramientas como el Profiler (en el momento de escribir este artículo se encuentra en fase M9, aunque para cuando usted lo lea debería estar finalizado) o el Mobility Pack.

## Orígenes de NetBeans

Históricamente NetBeans es uno de los primeros IDEs que aparecieron para Java y, si no recuerdo mal, el primero de importancia que estaba escrito íntegramente en Java. También ostenta el honor de ser uno de los dos que aún existe desde aquellos años (el otro es JBuilder).

Recuerdo que en aquellos tiempos anteriores a NetBeans disponíamos de Visual Café de Symantec, JBuilder de Borland, Visual Age for Java de IBM y poco más. Todos ellos escritos en C (no fue hasta la versión 3.0 de JBuilder que Borland decidió reescribirlo en Java).

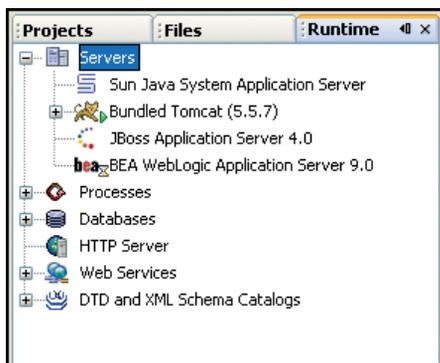


Figura 1. Manejando servidores.

Sin embargo, una compañía checa estaba a punto de dar el gran salto con un IDE poco conocido (si se compara con los que he nombrado antes), llamado NetBeans.

Sun adquirió la compañía checa y designó a NetBeans como el IDE oficial para desarrollos Java, lo hizo gratuito y abrió su código. Sun sabía que, si quería popularizar su joven plataforma y lenguaje, necesitaba disponer de una herramienta de desarrollo que fuese gratuita y, lógicamente, ésta tenía que estar desarrollada utilizando el propio Java para demostrar la capacidad y las posibilidades de la plataforma.

Fue por aquellos años cuando descubrí este IDE: me gustó mucho pero, honestamente, creo que en esos días no estaba a la altura de los IDEs comerciales; así que aconsejé a la empresa para la que trabajaba seguir desarrollando con Visual Age for Java (que era la herramienta que utilizábamos en aquellos años).

De entonces a ahora el panorama ha cambiado radicalmente: NetBeans 5.0 es sin lugar a dudas uno de los dos o tres mejores IDEs que existen para Java. Cada nueva versión ha supuesto un salto en la escala evolutiva, constituyendo esta versión, y gracias a herramientas como Matisse, su nuevo profiler o a su mejorado debugger, más que una evolución toda una revolución.

## IDE: Aspectos generales

Normalmente las nuevas versiones de un producto, especialmente cuando éste ha existido desde hace tanto tiempo, se reducen a tres cosas: corregir bugs, introducir

algunas mejoras en las herramientas existentes y quizás añadir alguna herramienta nueva.

Sin embargo, una de las cosas que más me gustó cuando Sun me invitó a visitar al equipo de NetBeans en Praga fue que ellos tienen muy claro que están dispuestos a rehacer, desde los cimientos si hace falta, una parte o todo NetBeans si fuese necesario: todo para conseguir que este IDE esté a la altura de lo que los usuarios esperan. Y no lo dicen repitiendo como loros una consigna de Sun, lo sienten de ese modo. Este orgullo por el trabajo bien hecho, además de admirable, es también lo que hace que después de tantos años NetBeans siga siendo uno de los mejores IDEs para desarrollo en Java.

Así, en esta versión se han rehecho un buen montón de cosas "under the hood" (debajo del capó); a veces se ha invertido un buen montón de horas en algo que el usuario no va a poder ver, y que solo algunos podrán o sabrán apreciar. Algunos ejemplos son el acceso a CVS, que se ha rehecho completamente por dentro y por fuera, o el acceso a bases de datos, que por ahora solo se ha hecho por dentro, y en el que he tenido el orgullo de trabajar.

A estas alturas NetBeans, además de un producto maduro, es inmensamente grande, lo que hace inabordable realizar un repaso de todas sus características, ni si quiera enumerativamente. Por eso, voy a centrarme en aquellos aspectos que considero más destacables o que creo que son los más utilizados por los programadores. Por razones de espacio, resulta de todo punto de vista imposible hacer un repaso, aun pasando de puntillas, ni siquiera de las principales áreas de NetBeans 5.0.

Por ello, el resto del presente artículo puede considerarse como dividido en dos grandes áreas: empezaré por hacer un repaso breve de aquellos aspectos más comunes a todos los IDEs, para examinar posteriormente con más detalles otras herramientas que componen este entorno de desarrollo.

## Nuevo modelo de plugins

Mientras que Eclipse siempre ha estado

basado en un sistema de plugins muy simple y muy bien documentado, NetBeans, aun siendo código abierto, no se ha distinguido por facilitar la creación de extensiones por parte de terceros.

En Praga pregunté a uno de los jefes del proyecto y me dijo que la idea de ellos había sido llegar a acuerdos con terceros para la creación de extensiones (plugins) de gran calidad y bien integradas con el IDE.

Me pareció parcialmente lógica su respuesta y así se lo hice saber, aunque también le dije que en mi opinión se equivocaban: se rió y me dijo que estaban terminando un sistema de plugins mucho más simple y con un wizard dentro de NetBeans para crearlos, al estilo de Eclipse; corregir es de sabios.

Por lo que, dicho esto, quien lo desee ahora puede lanzarse a escribir plugins para NetBeans de un modo fácil y accesible.

## Debugger

Siempre he tenido la sensación de que un buen debugger quizás sea la parte fundamental de un entorno de desarrollo.

El de NetBeans siempre ha sido más que aceptable. Sin embargo, con esta versión se le ha dotado de ciertas mejoras que lo llevan a eso que los estadounidenses llaman el "state of the art" de los debuggers.

Entre esta mejoras, quisiera destacar:

- Mejor integración con el editor: el establecimiento de los breakpoints (puntos de ruptura) y sus propiedades es accesible ahora desde el propio editor.
- Se ha mejorado la inspección de variables, especialmente en lo que se refiere a la visualización de arrays largos.
- Se permite la evaluación de expresiones sobre la marcha ("on the fly"), así como la invocación de métodos.
- Se ha incorporado un debugger para Ant.

## CVS (Concurrent Version System)

Como ya he comentado, el módulo de interacción con CVS ha sido rehecho completamente.

Ahora NetBeans reconoce los directorios de trabajo de CVS y lista en la ventana de "Versioning" los ficheros que han cambiado. Y, puesto que las acciones que más se realizan contra un servidor CVS son las de "commit" y "update" de ficheros, éstas se encuentran ahora disponibles (para facilitar su acceso) desde: "Proyectos" -> "Ficheros

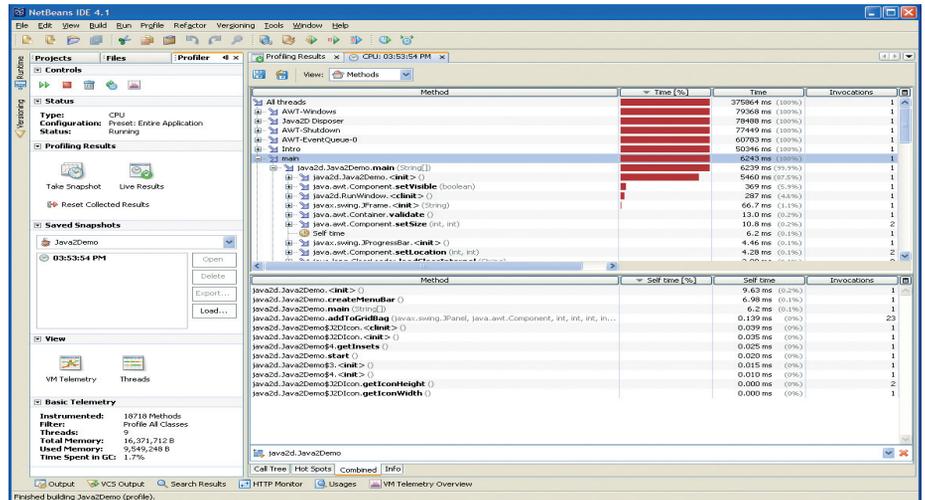


Figura 2. Profiler mostrando el uso de CPU.

y Favoritos", además de estarlo desde la ventana de "Versioning".

Se han añadido ciertas marcas y colores para mostrar la situación o estado actual de: ficheros, carpetas, paquetes y proyectos. Y el comando "Diff" permite, de un modo sencillo, mostrar las diferencias de estas entidades entre local y remoto de un golpe.

## Mejoras en el editor

El editor de código, aun siendo excelente en la versión anterior, sufría de ciertas carencias si se le compara con el de Eclipse. En esta versión se han tomado algunas ideas de Eclipse y se han incorporado funcionalidades nuevas y originales.

A continuación expongo solo algunas de ellas:

- Se ha mejorado la herramienta de completar código:
  - Plantillas definidas por el usuario.
  - Generación de "getters" y "setters".
  - Clases anónimas.
  - Algunas otras mejoras menores.
- Se ha mejorado sustancialmente el refactorizado ("refactoring").
- En el editor de código:
  - Sugerencias que ayudan en la generación de código.
  - Barra lateral con marcas de errores y warnings (al estilo del de Eclipse).
  - Rodar un trozo de código seleccionado con un "try-catch".
- Editor SQL mejorado con coloreo de sintaxis.
- Navegador XML con indentación automática mejorada y con colapsadores de estructuras de código ("code folding").

## Web Frameworks

Se ha mejorado sustancialmente la facilidad de incorporación y manejo de JavaServer Faces (JSF) y Struts, que puede activarse desde el nuevo "New Project Wizard" indicando que deseamos crear una aplicación web.

Desde este asistente (wizard) ahora también se permite escoger plantillas para la creación de JSF Managed Beans, Struts Actions, y Struts Form Beans.

Y, lo mejor de todo, si tenemos aplicaciones web realizadas con versiones anteriores, ahora con NetBeans 5.0 podemos añadirles JSFs y Struts. Basta con ir al nuevo panel "Frameworks" en la caja de diálogo de "Propiedades del Proyecto" ("Project Properties").

## Web Services

Los servicios web están de moda en las TI. Y aun cuando NetBeans daba soporte a esta tecnología en la versión anterior, en esta nueva este soporte ha sido mejorado permitiendo, entre otras cosas, crear clientes de servicios web en aplicaciones J2SE.

Todas las librerías necesarias para crear y desplegar un cliente de servicio web se suministran con el IDE, lo que facilita ambas tareas.

Se permite, así mismo, invocar operaciones de servicios web directamente desde una página JSP, lo que evita la fastidiosa tarea de creación de un Servlet para este único propósito.

También se ha añadido un nuevo wizard que permite crear el fichero WSDL directamente en el IDE y utilizarlo para generar nuestros ficheros de servicios web.

Como ya he dicho, estas que acabo de rela-

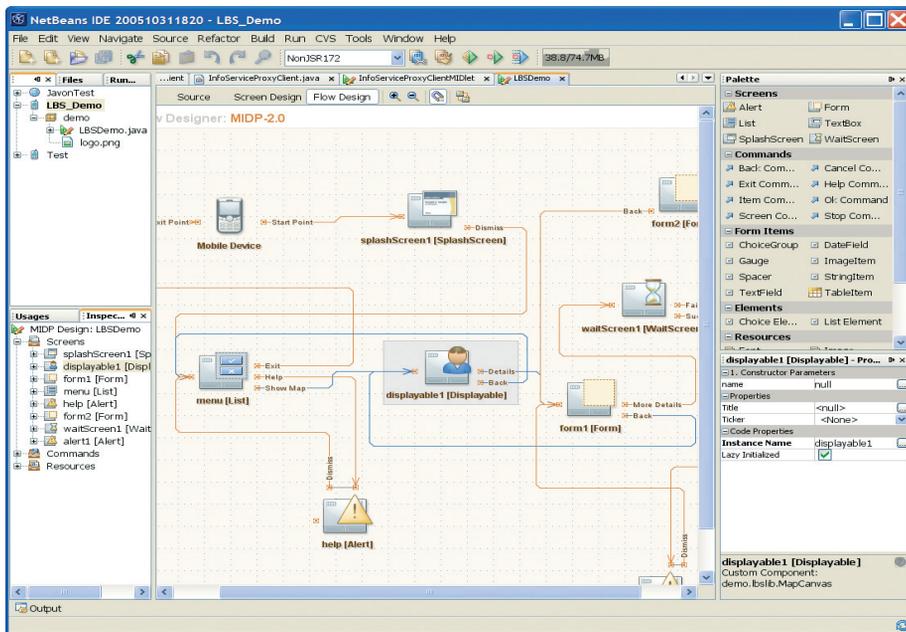


Figura 3. Ejemplo de uso de Graph Library y Mobility Pack.

tar son algunas de las muchas mejoras disponibles en esta nueva versión de NetBeans.

Las he tratado suficientemente porque deseo centrarme en otras herramientas del IDE a las que me gustaría darles un trato especial.

No es que las que vienen a continuación sean mejores, ni más útiles, pero sí que son las que me han parecido más novedosas. Esas que le dan a NetBeans 5.0 un toque diferencial con respecto al resto de los entornos de desarrollo.

## IDE: Matisse

Aun cuando quizás sea ésta la parte que más me ha impresionado de la versión 5.0, es la más difícil de describir. El problema radica en que hay que "probarlo" para apreciarlo.

Matisse es un diseñador de GUIs (del inglés "Interfaz Gráfica de Usuario"), con el que podemos diseñar visualmente pantallas (ventanas y diálogos) utilizando cualquier combinación de "artefactos" Swing, ya sean estándares (pertenecientes a la Java Foundation Classes, JFC) o creados por terceros, para nuestras aplicaciones.

Hasta ahora, y sobre todo para los programadores de aplicaciones de escritorio, uno de los mayores problemas de Swing han sido los Layouts: salvo que se quiera hacer una entrada de datos trivial, siempre hay que recurrir a un buen número de JPanels

dentro de JPanels, cada uno con distintos Layouts y, por desgracia, casi siempre recurriendo al más complicado de todos: el "GridbagLayout".

Se ha escrito mucho sobre qué y cómo mejorar esta situación, pero lo cierto es que hasta ahora nadie había aportado una solución que, ni de lejos, llegara a la excelencia de Matisse que, aun sin ser perfecta, está a mucha distancia de cualquier otra. Para ello, el equipo de Swing para NetBeans (no el de Swing de Sun, como cabría esperar) ha creado un nuevo Layout Manager, llamado coloquialmente "NaturalLayout Manager", y sobre él ha construido Matisse, una herramienta de composición de GUIs basada en el paradigma "Free Design" (Diseño Libre), el cual permite distribuir los componentes libremente sobre el contenedor, normalmente un "JPanel", y proporciona líneas guía para la alineación y el espaciado óptimo de los componentes.

Matisse resulta muy intuitivo, fácil de aprender por ser muy natural, y a veces sorprendentemente inteligente a la hora de anticiparse o darnos pistas sobre lo que estamos haciendo.

Por último quisiera ofrecer una nota técnica sobre Matisse: este nuevo Layout Manager no está incluido en el JRE 1.5, por lo que hay que incluir su "jar" correspondiente con nuestras aplicaciones; al menos hasta que aparezca la siguiente versión de Java (nombre clave actual Mustang), que tiene previsto incluirlo de serie.

## IDE: Developer Collaboration

Esta herramienta, quizás por lo breve de su existencia, no ha tenido demasiada repercusión en la comunidad NetBeans y sin embargo, a mi parecer, es una de las más útiles para desarrollo en equipos dispersos, es decir, cuando los participantes no se encuentra en la misma habitación.

Puede imaginar esta herramienta como una combinación de aplicación para chat, de escritorio remoto y de pizarra colaborativa. Todo ello sobre el entorno de NetBeans, permitiendo que varias personas realicen acciones en el IDE, intercambien opiniones y alteren el código mientras que el resto de los asistentes a la sesión ven qué hacen ellos y qué están haciendo los demás, y todo esto en tiempo real.

Como se explica en su página, los objetivos del proyecto pueden resumirse en los siguientes puntos:

- Desarrollar un protocolo de mensajería XML independiente del cliente y proporcionar una implementación open source en Java de dicho protocolo.
- Proporcionar un marco de trabajo rico, flexible y seguro para construir aplicaciones Java usando la colaboración.
- Distribuir implementaciones de proveedores de protocolos de comunicaciones de más bajo nivel (XMPP, JXTA, SIP, etc.), así como módulos de características de colaboración de alto nivel dentro del marco de trabajo.

Y como una imagen, un vídeo en este caso, vale más que mil palabras, le recomiendo que vea la demostración Flash de esta herramienta, que está disponible desde la página del producto (véase al final de este artículo bajo "Referencias"). A mí me impresionó, especialmente por su potencia y facilidad de uso.

## Herramienta: Analizador (Profiler)

Nunca he entendido por qué el debugger se considera parte integral de un IDE y no ocurre lo mismo con los analizadores de código en tiempo de ejecución (profilers): para mí es casi tan importante disponer de un buen profiler como de un buen debugger.

En el caso de NetBeans 5.0, podemos afirmar con rotundidad que dispone de uno de los mejores profilers (quizás el mejor) que existen para la plataforma



Java. Esta herramienta, basada en los resultados de un proyecto de investigación de Sun Laboratories llamado JFluid, supone un verdadero avance en este campo.

Tradicionalmente, uno de los problemas más comunes en estas herramientas ha sido la gran demanda de recursos (tiempo de micro y memoria RAM) que consumen. Sin embargo, JFluid utiliza "Dynamic Bytecode Instrumentation" para disminuir drásticamente esta sobrecarga, permitiendo de este modo obtener resultados en situaciones donde resultaría imposible, o casi, utilizar este tipo de herramientas de análisis.

Básicamente esta tecnología se basa en una funcionalidad de la JVM que permite a herramientas externas "inyectar" y "extraer" bytecodes de un programa mientras que éste se está ejecutando. Estos bytecodes adicionales no cambian la semántica de la aplicación en ejecución, sino que se encargan de emitir eventos, tales como la entrada o salida de métodos, reserva de memoria, etc. Estos eventos son recogidos por las herramientas, recabando información de lo que está sucediendo durante la ejecución y poder así mostrar al programador la información que necesita para optimizar su programa (si se ha quedado de piedra, no se preocupe, nos ha pasado a casi todos la primera vez que leímos que los ingenieros de Sun hacen este tipo de cosas).

Esta tecnología permite, además, ser activada y desactivada durante la ejecución de la aplicación que estemos estudiando, así como aplicarla solo a uno o varios módulos concretos de la aplicación, aquellos objeto de interés, sin que recabe información del resto de los módulos, mejorando de este modo aún más su rendimiento. Así mismo, todos los parámetros de análisis (CPU, memoria, threads, call-tree, etc.) pueden ser modificados sin detener ni reiniciar la aplicación, lo que se consigue utilizando la ya nombrada tecnología de "Dynamic Bytecode Instrumentation".

Concluyendo: el nuevo profiler para NetBeans 5.0 es muy completo y proporciona toda la información necesaria para poder detectar y corregir los cuellos de botella y, en general, los problemas de ejecución de nuestras aplicaciones.

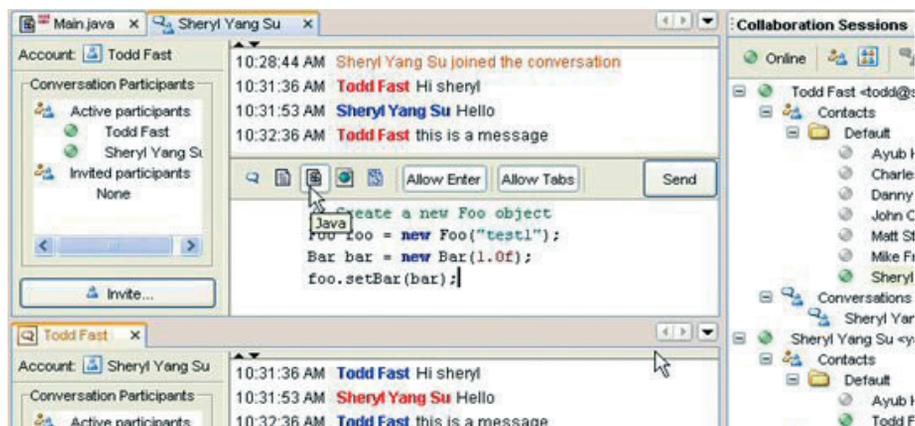


Figura 4. La herramienta Collaboration.

Por el tipo de trabajo que realizo cotidianamente y mis necesidades esta herramienta forma, junto con Matisse, aquello que más valoro de esta nueva versión, por la cantidad de horas de trabajo que me van a ahorrar en mi quehacer diario.

En el momento de realizar esta presentación, aún no está disponible la versión definitiva del profiler, pero es de esperar que aparezca junto con la versión 5.0 de NetBeans. Tampoco está claro si estará integrado con el IDE o será una herramienta independiente que se añade al IDE (un plugin), siendo esta segunda opción la más probable al día de hoy.

### Otras herramientas

Veamos ahora otras herramientas que, no por ser menos conocidas o menos utilizadas por el público en general, son menos importantes.

#### NetBeans Mobility Pack

Con la versión 4 de NetBeans apareció esta utilidad que permite desarrollar aplicaciones para la versión JME (Java Micro Edition) de la plataforma, que tiene entre los teléfonos móviles su mayor cuota de aceptación.

Esta herramienta contiene utilidades que facilitan enormemente todas las fases del desarrollo, pruebas y deployment (distribución) de una aplicación JME. Quizás su componente más llamativo sea el Visual Mobile Designer, compuesto por: diseñador de flujo de la aplicación, diseñador de pantallas y editor de acciones que genera código Java. Resulta muy fácil de manejar al ser totalmente visual y permitir el uso de muchas de las acciones vía "drag & drop" (arrastrar y soltar).

#### NetBeans Graph Library

El módulo Mobility Pack está basado en una herramienta llamada NetBeans Graph Library (una pieza de software del estilo del conocido JGraph).

Aun cuando Mobility Pack se encontraba dentro de la versión anterior de NetBeans, no ha sido hasta la 5.0 que disponemos de un API para poder utilizarlo. Así, y desde ahora, podemos crear nuestro propio diseñador gráfico con esta herramienta "drag & drop" (arrastrar y soltar) que permite la visualización y edición de estructuras de nodos.

#### Herramientas de terceros

Por otro lado, es reseñable el cada vez mayor número de empresas que desarrollan software que se integra con los IDEs Java, siendo NetBeans y Eclipse los más soportados por terceros.

En este área, y hablando ahora de aplicaciones comerciales, quisiera citar la suite de herramientas AppPerfect, desarrolladas por la empresa californiana del mismo nombre. Este conjunto de herramientas permite, entre otras cosas: analizar, probar, afinar y monitorizar aplicaciones J2SE y J2EE, tanto de escritorio (desktop) como aplicaciones web.

El AppPerfect Test Studio se compone de los siguientes módulos: AppPerfect Unit Tester, AppPerfect Code Analyzer, AppPerfect Java Profiler, AppPerfect Functional Tester y AppPerfect Load Tester.

#### NetBeans (Rich Client) Platform

Por último, no quisiera finalizar sin dedicar unas líneas a esta herramienta, quizás no muy conocida, pero que puede resultar tremendamente útil en algunos

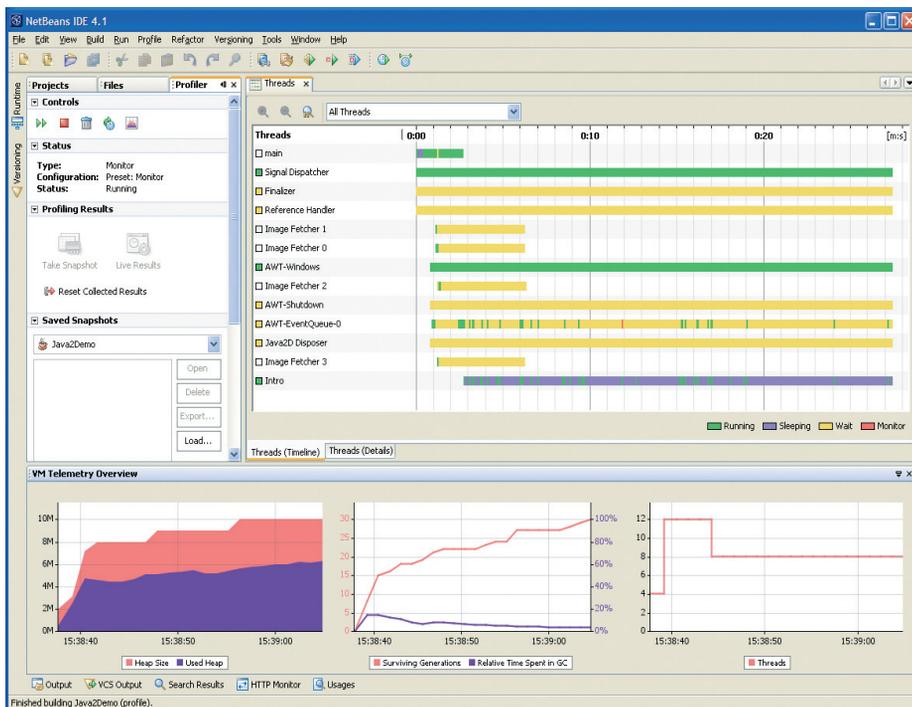


Figura 5. Profiler mostrando la telemetría de la JVM.

desarrollos. En este sentido, NetBeans funciona como Eclipse y también, en menor medida, como JBuilder: el propio entorno de trabajo (paneles arrastrables, menús, configuraciones, árboles, editores, etc.) forman un conjunto de herramientas con un API propio para utilizarlo. Si necesitásemos crear una aplicación con una funcionalidad parecida al área de trabajo de NetBeans (imaginemos que deseamos, por ejemplo, desarrollar un diseñador para crear páginas DHTML), nos podemos ahorrar un innumerable número de horas de trabajo utilizando el propio NetBeans como punto de partida pudiendo, de este modo, centrarnos en los aspectos propios de nuestra aplicación.

## Rendimiento

No quisiera terminar sin comentar la agradable sorpresa que me ha producido ver las mejoras en el rendimiento (performance) del entorno de trabajo, así como la razonablemente reducida cantidad de recursos que consume. No puedo comparar a NetBeans 5.0 Beta con la que yo he trabajado ("NB 5.0 development" que es casi para uso interno de Sun) porque estas versiones de estado temprano de desarrollo no están optimizadas en absoluto. Tampoco

puedo compararla con la 4.x porque hace ya mucho que no la utilizo y no me acuerdo. Pero sí puedo decir que el viejo argumento de que NetBeans es bastante más lento y más pesado que Eclipse ya no es válido, al menos en mi entorno de trabajo: Linux Debian/Ubuntu 5.10 sobre un AMD 3600+ y 1Gb RAM.

En este entorno, mientras que Eclipse al arrancar se me va fácilmente a los 70 Mb de RAM, NetBeans comienza consumiendo 40 Mb, y esta distancia suele mantenerse para el mismo proyecto ejecutado en ambos entornos, lo que supone una nada despreciable diferencia de aproximadamente el 57%.

De todos modos, que nadie considere esto como un test de consumo de memoria: nada más lejos ni de mi intención ni de la realidad.

En cuanto a la velocidad de renderización de los componentes visuales, también he observado una gran mejora; claro que esto muy posiblemente sea debido no solo a las optimizaciones realizadas en el código de NetBeans, sino también a las incesantes mejoras de rendimiento que Sun aplica en cada nueva versión de su máquina virtual (JVM).

Y si finalmente cuajan todas las optimizaciones de las que estamos teniendo noticia en estos meses, correr NetBeans

con la próxima versión de Java, de nombre clave Mustang, va a ser verdaderamente impresionante.

## Conclusiones

Hasta aquí nuestra breve introducción a NetBeans 5.0. Todo lo expuesto no es más que un mero rascar la superficie de un gran bloque de hielo. Seguro que muchos lectores habrán echado en falta esta o aquella otra característica que consideran digna de ser reseñada, de igual modo que a otros les habrá parecido que me extendí en exceso sobre algo que no tiene demasiada importancia. A todos les pido disculpas y les ruego que entiendan que no es tarea fácil condensar una herramienta de esta envergadura en tan poco espacio.

Lo que sí espero haber expuesto con claridad es que estamos ante un producto de una potencia y una fiabilidad difícilmente imaginables en el entorno Java hace solo cinco años.

Incluso si usted no desarrolla software o si no programa en Java, échele un vistazo, merece la pena: yo al menos me divierto "jugando" con un trabajo tan bien hecho como este. 

## Referencias

- NetBeans website oficial: <http://www.netbeans.org>
- Completa introducción a NB 5.0 (Flash en inglés): <http://www.javaloobby.org/eps/netbeans5/>
- IDE: <http://www.netbeans.org/community/releases/50/index.html>
- Matisse flash demo: <http://www.netbeans.org/files/documents/4/475/matisse.html>
- JFluid (este proyecto se ha "graduado" pasando a llamarse NetBeans Profiler): <http://profiler.netbeans.org/index.html>
- Graph Library: <http://graph.netbeans.org/>
- Nuevo soporte para CVS (Flash en inglés, incluido en la introducción completa en Flash): <http://www.netbeans.org/kb/articles/flash-cvs-announce.html>
- NetBeans (Rich Client) Platform: <http://www.netbeans.org/products/platform/>, <http://www.netbeans.org/products/platform/howitworks.html>
- Mobility Pack: <http://www.netbeans.org/products/mobility/>
- Collaboration Project: <http://collab.netbeans.org/>
- AppPerfect suite: <http://www.appperfect.com/>