

NetBeans 3.51

guía rápida

Autor: Jorge Sánchez (www.jorgesanchez.net) año 2004

Manual breve para el manejo del entorno de programación de código abierto NetBeans para la programación Java (<http://www.netbeans.org>)

notas previas

introducción

NetBeans es un entorno de desarrollo o IDE (*integrated development environment*) para todo tipo de tecnologías Java e incluso permite la codificación de programas en C, C++ y otros (aunque está pensado para Java).

Sus funciones son:

- **Editor de código sensible al contenido.** Con soporte para autocompletar el código, coloreado de etiquetas, autotabulación y uso de abreviaturas para varios lenguajes de programación.
- **Soporte para Java, C, C++, XML y lenguajes HTML.**
- **Soporte para JSP, XML, RMI, CORBA, JINI, JDBC y tecnologías Servlet**
- **Incluye CVS (control de versiones) y Ant (compilación avanzada)**
- **Posibilidad de utilizar otras versiones de compiladores, depuradores,...**
- **Creación visual de componentes gráficos**
- **Herramientas con asistentes para facilitar la escritura de código**

requerimientos

Se requiere (al ser una plataforma creada en Java 2) tener instalado el JDK 1,3 o superior.

Sistema operativo	Espacio en disco duro	Procesador y RAM
Windows 2000, Windows XP, Windows NT4 with SP6, Windows 98 2nd edition	125 MB	Pentium III 500 Mhz 256 MB RAM
Red Hat Linux 7.2 Sun Linux 5.0	125 MB	Pentium III 500 Mhz 256 MB RAM
Solaris 8, Solaris 9 operating environments (64-bit, SPARC platform)	125 MB	Ultra™ 10, 333 MHz, 256 MB RAM

instalación del entorno

instalación de la plataforma Java

Es necesario tener instalada la plataforma J2SE v1.3.1 o superior para que NetBeans pueda funcionar.

Para saber si este entorno ha sido instalado, basta colocar el texto `java -version`, en la línea de comandos. El resultado puede ser algo como:

```
java version "1.4.1_02"  
Java(TM) 2 Runtime Environment, Standard Edition (build  
1.4.1_02-b06)  
Java HotSpot(TM) Client VM (build 1.4.1_02-b06, mixed mode)
```

En este caso sí estaría instalada la versión requerida. Si no es así se puede descargar la última plataforma desde la dirección <http://java.sun.com/j2se/1.4.2/download.html>

instalación del entorno IDE NetBeans

Se puede descargar gratuitamente de la dirección www.netbeans.org. O también al descargar el kit de desarrollo J2SE 1.4.2. se puede elegir descargar junto con este IDE. Esta última quizá sea la mejor opción, los pasos serían:

- 1> Ir a <http://java.sun.com/j2se/1.4.2/download.html>
- 2> Hacer clic en el texto **download** que corresponda a nuestro sistema operativo
- 3> Aceptar tras leer las licencias de uso
- 4> Se descargará un archivo en nuestro ordenador (tras avisarnos de ello)
- 5> Ejecutar este archivo y seguir instrucciones. Se instalará el kit de desarrollo de Java J2SE y el IDE NetBeans

arrancar por primera vez el entorno

La primera vez que se arranca NetBeans. Se nos preguntará si poseemos otros entornos compatibles. Si es así podremos hacer que NetBeans utilice (si se desea) la configuración de estos entornos (especialmente Forte for Java o Sun ONE Studio).

Tras estos pasos aparecerá la pantalla inicial del entorno

estructura de las carpetas y directorios

filesystems

En NetBeans, hay un panel dedicado a los archivos y carpetas de las aplicaciones y componentes Java, es el panel de la izquierda titulado **Explorer (filesystems)**. En ese panel deben aparecer los llamados *filesystems*. Un filesystem es una carpeta que

contiene paquetes de Java. Normalmente cada filesystem se refiere a toda una aplicación java (esto incluye todos los paquetes de la aplicación).

Para que una carpeta con archivos de Java pueda ser utilizada por este entorno, debe ser montada (**mount**) en NetBeans. La carpeta montada se considerará el directorio raíz. Si una carpeta interior a esta debe de ser considerada como raíz de otra aplicación, se debe montar por separado.

Para montar una carpeta y que sea considerada un nuevo filesystem por NetBeans:

- 1 > Pulsar el botón derecho sobre el icono **Filesystems** y elegir **Mount-Local Directory** o elegir **File-Mount Filesystem** seleccionar **Local Directory** y pulsar **Next**
- 2 > Seleccionar la carpeta en cuestión y pulsar **Finish**

Se entiende que la carpeta elegida es la que contiene el paquete raíz (*root package*). Si esta carpeta poseyera más carpetas dentro, serán consideradas como paquetes interiores, lo que significa que las clases definidas en esas carpetas sólo podrán ser compiladas si se incluye la instrucción **package** seguida de la ruta.

Por ejemplo si hemos hecho un filesystem de la carpeta **C:\prueba**, y dentro hay una carpeta llamada **utiles** y dentro de esta otra llamada **contadores**, compilar una clase de la carpeta contadores requiere colocar como primera instrucción de la clase: **package utiles.contadores**.

cambiar un filesystem

Si pulsamos el botón derecho sobre el icono Filesystem y elegimos **Customize** podremos cambiar la ruta de la carpeta raíz (esto abriría que hacerlo si hemos movido esa carpeta o si queremos elegir otra carpeta raíz para ese mismo filesystem).

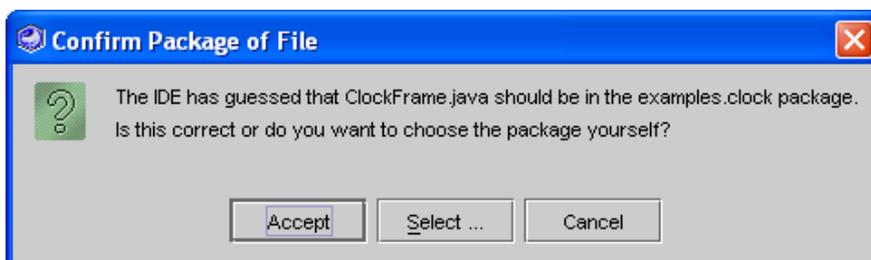
abrir un archivo en un filesystem

Para abrir un archivo de una carpeta del apartado filesystem basta hacerle doble clic

abrir un archivo externo a los filesystems

También se pueden abrir archivos que no pertenecen a filesystems; para ello hay que elegir **File-Open** en los menús y hacer doble clic en el archivo a abrir.

Cuando el archivo abierto con este método esta fuera de cualquier filesystem, entonces se pedirá un paquete apropiado para el archivo y sí es necesario NetBeans montará la carpeta en la que se encuentra el archivo.



El cuadro anterior indica que NetBeans ha encontrado el paquete al que pertenece la aplicación. Pulsando **Accept** estaremos de acuerdo con su selección y entonces se conectará el paquete raíz a la lista de filesystems.

desconectar un filesystem

Si pulsamos el botón derecho sobre un determinado filesystem y elegimos **Unmount filesystem**, entonces se desconecta la carpeta elegida y sus archivos no serán visibles en el IDE a no ser que se vuelva a montar.

compilación y ejecución

Build y compile

Compilar el archivo es preparar el archivo class que corresponde al original java. Hay cuatro formas (las cuatro en el menú **Build**):

- ⦿ **Build.** (F11) Compila el archivo sólo si ha habido cambios desde la última vez.
- ⦿ **Build all.** (Mayúsculas + F11) Lo mismo, pero compila todos los archivos java relacionados.
- ⦿ **Compile.** (F9) Compila el archivo borrando el archivo class anterior.
- ⦿ **Compile all.** (Mayúsculas + F9) Idéntico pero compila además todos los archivos de clases relacionadas

detener la compilación

Si la compilación tarda demasiado tiempo en realizarse, se puede detener haciendo uso de la opción **Stop compile** en el menú **Build** o pulsando la tecla **Alt + Mayúsculas + C**.

Clean

El comando **clean** que se encuentra también en el menú **Build** permite borrar el archivo class relacionado con el actual. Es decir borra el archivo compilado (en el caso de **clean all** borrar todos los compilados relacionados).

errores

Cuando un código posee errores de compilación, en el apartado Output se nos muestra el error, y en el código habrá un aspa de color rojo en cada línea de error. Eso facilita su corrección.

La tecla **F12** permite ir al siguiente error, la tecla **Mayúsculas + F12** permite volver al error anterior

elegir carpeta específica para los archivos class

Los archivos compilados de Java (bytecodes), generan un archivo class. En NetBeans se puede elegir una carpeta específica para esta labor. Para ello:

- 1 > Montar un filesystem para la carpeta en la que se desean almacenar las clases compiladas
- 2 > Elegir **Tool-Options** en la ventana principal
- 3 > Elegir el apartado **Building-Compiler types** y el compilador deseado
- 4 > Elegir en el apartado **Target** el filesystem deseado.

Normalmente no se elige carpeta específica, los archivos **class** y los **java** se almacenarán en la misma carpeta.

la ventana Output

Esa ventana muestra los mensajes del compilador. Se puede recorrer cada mensaje e incluso utilizarla para que nos lleve a las líneas de código relacionadas con el mensaje.

Si se desea que los mensajes de esa ventana se almacenen en un archivo externo, entonces:

- 1 > Elegir **Tool-Options** en la ventana principal
- 2 > Elegir el apartado **IDE Configuration- Look and feel-Output Window**
- 3 > Elegir **True** para la propiedad **Redirect output to file**
- 4 > Elegir la carpeta en la que se almacenarán los mensajes en la propiedad **Redirection directory**

El archivo generado se llamará Output.out.

Desde esta misma zona del cuadro de opciones se pueden configurar los colores y formatos de letra de la ventana **Output**

configuración del compilador

Desde **Tools-Options-Building-Compiler types**, se pueden elegir diversas opciones de compilación. Aparecen además todos los compiladores que puede utilizar NetBeans. Entre ellos están el compilador externo y el interno.

El compilador externo es el compilador `javac` del JDK estándar de Sun. El compilador interno es la versión de `javac` que se utiliza internamente por NetBeans. Las ventajas de utilizar este compilador están en la rapidez de compilado y en el manejo de errores y otras características. Las ventajas de usar el externo están en que se puede elegir exactamente que versión del compilador se utiliza para compilar.

Se pueden añadir compiladores desde **Tools-Options-Building-Compiler types**, pulsando el botón derecho en **Compiler types** y eligiendo **New** y eligiendo después el tipo de compilador. Pulsando el botón derecho sobre un determinado compilador se le puede borrar (**Delete**) o cambiar de nombre (**Rename**).

Para seleccionar el compilador de uso por defecto, hay que ir a **Tool-Options-Editing-Java Sources** y elegir el compilador deseado en la propiedad **Default compiler**

ejecución del código

La tecla **F6** o bien el menú **Build-Execute** son los que realizan esta tarea. Se ejecuta el archivo class previamente compilado. Si el archivo no estaba compilado, se le compila ahora..

configurar argumentos

La función **main** de un archivo Java ejecutable admite argumentos, los cuales se pueden configurar seleccionando el archivo el archivo Java ejecutable en la vista del explorador y eligiendo **Set arguments** en el menú **Build**

configuración del ejecutor

Desde **Tools-Options-Debugging and executing types**, se pueden elegir diversas opciones de ejecución y compilado. Hay varios ejecutores, interno (el que usa internamente el IDE), externo (usando la máquina virtual de Java instalada), de applets, de archivos JAR de servidor J2EE,... Todos se pueden configurar desde sus apartados en **Executor types** dentro de **Tools-Options-Debugging and executing types**

Para seleccionar el ejecutor de uso por defecto, hay que ir a **Tool-Options-Editing-Java Sources** y elegir el ejecutor deseado en la propiedad **Default executor**.

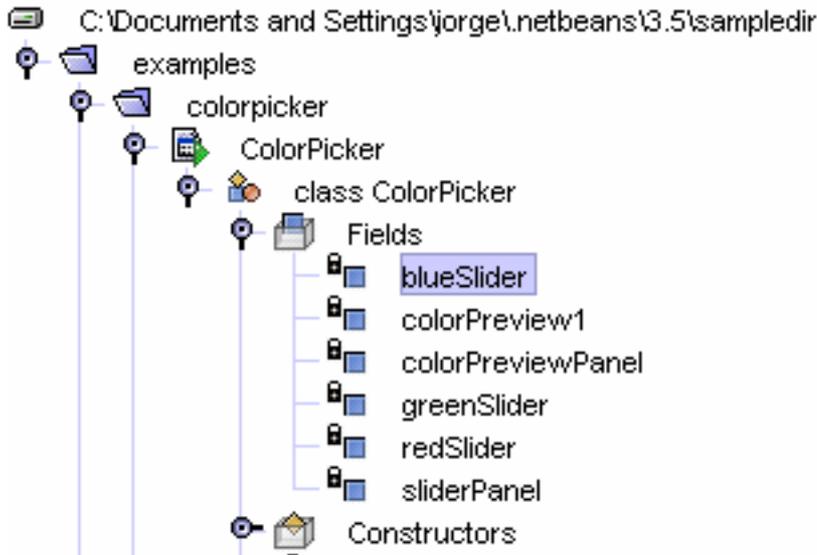
asociar un determinado ejecutor, compilador y depurador a un archivo

Se puede relacionar a un archivo con el compilador, ejecutor y depurador deseado si se selecciona el archivo en el panel de exploración y desde el panel de propiedades se hace clic sobre la pestaña **Execution**

administración de de archivos

el explorador

El panel izquierdo es el explorador (**explorer**) de archivos de NetBeans. Los elementos que poseen el símbolo  son elementos que se pueden expandir. Así podemos expandir los filesystems, para ver sus archivos y carpetas o paquetes interiores. Además podemos expandir los archivos Java para examinar sus elementos.



Elementos expandidos en el panel del explorador. Desde el filesystem que forma la carpeta samplerdir, pasando por las carpetas examples y colorpicker hasta llegar a la aplicación ColorPicker sobre la cual se están mostrando sus elementos.

En la parte de abajo hay tres pequeñas solapas que permiten cambiar la vista que ofrece el explorador. Son:

- ⊙ **Filesystem.** Esta es la solapa que se usa normalmente. Muestra todos los filesystems, carpetas y archivos montados.
- ⊙ **Projects.** Muestra los archivos que forman parte del proyecto actual (si lo hay)
- ⊙ **Running.** Muestra los elementos en proceso de depuración o ejecución

crear nuevos archivos

Todos los archivos en NetBeans se crean a partir de una plantilla (**template**). Éstas proporcionan:

- ⊙ El código básico para el nuevo archivo
- ⊙ Propiedades de ese archivo. Las propiedades incluyen el tipo de compilación, ejecución y depurado dedicado a ese archivo.

De tal forma que al crear un nuevo archivo hay que decidir la finalidad del mismo. Para crear un archivo:

- 1 > Hay que elegir la carpeta en la que deseamos crear el archivo y pulsar el botón secundario del ratón en ella y elegir **New**
- 2 > Seleccionar :
 - ◆ **Folder.** Para crear una carpeta
 - ◆ **Package.** Para crear un paquete.
 - ◆ **Main class.** Para crear una clase principal

- ◆ **Class.** Para crear una clase normal
 - ◆ **JFrame Form.** Para crear un formulario
 - ◆ **All templates.** Para elegir otras plantillas.
- 3> Poner nombre al nuevo archivo

Se puede elegir crear un archivo vacío de código eligiendo la carpeta **classes** y seleccionando la plantilla **Empty** (vacía). Para crear un applet o una aplicación de consola se pueden elegir plantillas en esta misma carpeta de plantillas.

abrir archivos desde el explorador

Un archivo se abre haciéndole doble clic. Ahora bien, al hacerlo NetBeans decide cuál de sus componentes es el más apropiado para mostrar el archivo, puede abrirlo el editor de código, el editor de formularios etc.

En cualquier caso hay tres pestañas en la superior que permiten cambiar de vista sobre el archivo:

- ⊙ **GUI Editing.** Es la forma normal de ver el archivo. A la derecha el código y debajo la ventana de salida (**Output**) desde la que se ven los mensajes del compilador.
- ⊙ **Editing.** Es una forma de ver el archivo en la cual a la izquierda se ve el panel de propiedades y a la derecha el código.
- ⊙ **Debugging.** Ventana del depurador

paquetes (*package*) y carpetas (*folders*)

Los paquetes son la base de la compilación en Java de diversos archivos. NetBeans representa a los paquetes de la misma forma que a las carpetas. De hecho la diferencia es más conceptual, los paquetes estructuran las clases de Java, las carpetas organizan archivos en el sistema operativo.

Se puede crear un paquete pulsando el botón derecho en el icono que lo contendrá y eligiendo **New-Java Package**. En el caso de una carpeta, éstas se crean eligiendo **New-Folder**.

Tanto para borrar carpetas como para borrar paquetes, hay que pulsar el botón derecho en el icono de carpeta a borrar y entonces elegir **Delete**.

propiedades de archivos y carpetas

Al seleccionar cualquier elemento del explorador de archivos de ONE Studio, en el panel inferior se nos mostrarán sus propiedades (**properties**). Éstas, se pueden modificar según nos interese.

En el panel de propiedades también se encuentran los siguientes botones:



Muestra las propiedades en un orden arbitrario



Ordena las propiedades por el nombre



Ordena las propiedades por el tipo



Muestra sólo las propiedades editables

guardar cambios

Basta pulsar `Control+S` o **File-Save**. En este IDE no existe operación **Guardar como (Save as)**, se debe copiar un archivo en su lugar.

cerrar archivos

Cada archivo abierto aparece en una pestaña en el lado derecho. cada pestaña posee el nombre del archivo al que se refiere. En cada nombre aparece una X para poder cerrar el archivo.

Si el nombre del archivo posee un asterisco, con él se indica que el archivo no ha sido guardado. Al cerrar se nos pedirá Guardar (**Save**) o Descartar cambios (**Discard**) o Cancelar el cierre.

copiar y mover archivos

Se pueden mover y copiar archivos entre filesystems o entre paquetes o carpetas distintas. Automáticamente NetBeans modificará las instrucciones de acuerdo con la nueva ubicación. Para ello:

- 1 > Seleccionar el archivo a mover o copiar
- 2 > Pulsar el botón derecho y elegir **Copy** para copiar o **Cut** para mover (también se pueden elegir estos comandos en el menú **Edit**)
- 3 > Seleccionar la nueva ubicación para el archivo
- 4 > Elegir **Paste** en el menú de contexto o en el menú **Edit**

borrar archivos

Se hace como en el explorador de Windows. Se puede seleccionar y pulsar la tecla `Suprimir` o se puede pulsar el botón derecho en él y elegir **Delete**

cambiar el nombre de un archivo o carpeta

Basta pulsar el botón derecho en el archivo o paquete y luego, tras elegir **rename**, elegir el nuevo nombre. También haciendo un clic en el nombre del archivo seleccionado se puede inmediatamente cambiar su nombre.

refrescar la lista de archivos

Si se crean carpetas o archivos desde fuera de NetBeans, éste no refleja los cambios al instante. Por ello, si se tiene la duda sobre si se está mostrando la realidad de los archivos, se puede pulsar el botón secundario del ratón sobre el filesystem o carpeta a actualizar, y elegir **Refresh folder**.

plantillas propias

Cuando se crea un archivo se debe elegir la plantilla sobre la que se basa el archivo. Es muy común desear crear plantillas propias. Para ello:

- 1 > Crear un archivo con el código deseado para la plantilla.
- 2 > Seleccionar el archivo desde el explorador de archivos y tras pulsar el botón secundario del ratón, elegir **Save as template**.
- 3 > Seleccionar la posición de la nueva plantilla

Se podrá utilizar para crear nuevos documentos. Si se usa a menudo aparecerá en el propio menú **New** de creación de documentos. Si no habrá que buscarla desde **New-All templates** en la posición en la que se creó.

modificar plantillas

Para ello:

- 1 > Elegir **Options** en el menú **Tools**
- 2 > Elegir **Source Creation and Management - Templates**.
- 3 > Seleccionar la plantilla a modificar
- 4 > Pulsar el botón secundario en ella y elegir **Open**
- 5 > Realizar los cambios deseados

borrar plantillas

Para ello:

- 1 > Elegir **Options** en el menú **Tools**
- 2 > Elegir **Source Creation and Management - Templates**.
- 3 > Seleccionar la plantilla a modificar
- 4 > Pulsar el botón secundario en ella y elegir **Delete**

otros cambios

Desde ese mismo apartado de **Tools-Options Source Creation and Management - Templates**, se puede:

- ⊙ Cambiar el nombre de la plantilla pulsando el botón derecho y eligiendo **Rename**
- ⊙ Acceder a las propiedades de la plantilla seleccionándola y utilizando el panel de propiedades de la derecha
- ⊙ Acceder a las propiedades de cualquier elemento interno de la plantilla, expandiendo el contenido.

- Copiar la plantilla (**Copy ... Paste**)

plantillas privilegiadas

Cuando se pulsa el botón derecho en una carpeta y se elige **New** se puede entonces elegir una plantilla para crear un documento desde **All templates**. Pero sin embargo aparecen unas plantillas en el mismo menú, son las *privilegiadas*. Para hacer que una plantilla sea privilegiada:

- 1 > Elegir **Options** en el menú **Tools**
- 2 > Elegir **Source Creation and Management - Templates**.
- 3 > Elegir la plantilla que se desea como privilegiada, pulsar en ella el botón secundario de ratón y elegir **Copy**
- 4 > Seleccionar la carpeta de plantillas **Privileged** y pulsando el botón secundario en ella, elegir **Paste-Create link**

edición de código

Source Editor

El **source editor** es el editor de código fuente que utiliza este IDE. En el editor el código aparece de distintos colores según a qué se refiera ese texto. En la línea de estado se indica el número de columna y línea en la que se encuentra el cursor y se apreciará la palabra **INS** si se está en modo de inserción (el texto nuevo se añade sin borrar el que ya existía) o **OVR** si se está en modo de sobrescritura. La tecla `ins` permite cambiar de modo.

Por debajo de la línea de estado hay una pestaña por cada archivo abierto. Se indica en ella el nombre y aparecerá un asterisco si hay código sin guardar.

tabulaciones

Para realizar tabulaciones o sangrados en el código a fin de señalar el código que está inmerso en un bloque, basta con pulsar el tabulador. Todas las líneas siguientes se colocarán automáticamente en esa posición. La tecla `Ctrl+D` permite abandonar la tabulación y colocarse en la tabulación anterior.

Sin embargo el source editor reconoce los elementos de bloque, es decir los signos `{` y `}`, y así cuando se inserta el símbolo de apertura de llave, automáticamente el cursor salta solo al siguiente tabulador. Al cerrar la llave, también el tabulador se coloca sólo en la posición adecuada.

opciones de tabulación

Haciendo uso del menú **Tools** para elegir **Options-Editing-Editor settings-Java Editor**, se pueden seleccionar diversas opciones del editor de código para Java (las opciones para otros tipos de código como XML, HTML, CSS, JSP, etc., se configuran de igual modo). Entre ellas están:

- **Indentation engine.** Permite elegir el motor de tabulaciones automático que usa el editor. Para Java el apropiado es el **Java indentation** porque es el que reconoce las llaves y otros elementos. **Simple indentation** hace tabulaciones más simples en las que el usuario debe pulsar el tabulador para saltar o `Ctrl+D` para abandonar la tabulación. **No indent** elimina la posibilidad de tabular automáticamente.
- **Tab size.** Elige el número de espacios que habrá entre cada salto de tabulación

completar automáticamente el código

Otra de las utilidades fundamentales de este editor, son las facilidades que ofrece para completar el código. Por defecto aparece un menú que nos facilita la escritura de código cada vez que nos disponemos a escribir una propiedad, método, clase, etc.

Esta ayuda aparece automáticamente tras esperar un determinado tiempo que además se puede configurar. También aparece al pulsar `Ctrl+Barra` siempre que se precise de su ayuda

Seleccionar la opción deseada de ese menú de ayuda se puede hacer con un clic de ratón en la opción deseada, llegando hasta la opción usando las teclas de movimiento y pulsando `Intro`, continuar escribiendo hasta que se seleccione la opción y pulsando después `Intro`.

Desde **Tools-Options-Editing-Editor settings-Java Editor** se puede elegir el tiempo que se demorará el menú de ayuda para autocompletar (*Delay of completion window*)

mostrar números de línea

Pulsando el botón derecho en la ventana de código y eligiendo **Margin-Show line numbers** se pueden mostrar y ocultar los números de línea

barra de herramientas del código

En la zona superior hay una barra de herramientas que se puede mostrar y ocultar eligiendo **Margin-Show toolbar** tras pulsar el botón derecho en la ventana de código.

buscar texto concreto desde el código

Esto se realiza desde **Edit-Find** o pulsando `Ctrl+F`. Después se escribe el texto a buscar en el código. Tras buscar el texto en la barra de herramientas del código hay botones que permiten buscar la siguiente o la anterior vez que aparece la palabra buscada en el código. Son:  y .

Además tras buscar se iluminan las palabras que se corresponden con la búsqueda. Esas palabras se pueden volver a apagar con el botón . Ese mismo botón las puede iluminar de nuevo.

reemplazar texto

Se puede también buscar un determinado texto a fin de reemplazarlo con otro. Esto es posible desde **Edit-Replace** o pulsando `Ctrl+H`. Tras ejecutar una de estas dos acciones, entonces basta con poner el texto buscado en el apartado **Find what** y el

nuevo texto en el apartado **Replace with**. El botón **Replace** reemplaza la siguiente vez que aparezca la palabra buscada. El botón **Replace All** reemplaza todas las apariciones de la palabra.

colores y tipos de letra del código

En **Tools-Options-Editing-Editor settings-Java Editor** desde la categoría **Font size** se elige el tamaño de letra del código. En la categoría **Fonts & colors** se permite determinar el coloreado selectivo del código para cada elemento del lenguaje

abreviaturas

Al escribir código es interesante utilizar abreviaturas. NetBeans tiene ya configuradas unas cuantas. Por ejemplo al escribir la abreviatura **sout**, Java cambia la abreviatura por el código **System.out.println(“”);**

Desde **Tools-Options-Editing-Editor settings-Java Editor** en el apartado **abbreviations** aparece la lista de abreviaturas que podemos además cambiar o utilizar para añadir más.

añadir abreviaturas

Desde el apartado **abbreviations** comentado anteriormente. Tras pulsar el botón de los puntos suspensivos, aparece una lista completa de abreviaturas. Pulsando el botón **Add** podremos añadir una nueva.

Tras pulsar el botón hay que colocar la abreviatura en el apartado **Abbreviation** mientras que el apartado **Expansion** se coloca el texto que reemplazará a la abreviatura.

modificar abreviaturas

Basta llegar al apartado **abbreviations**, hacer clic en el botón de puntos suspensivos, elegir la abreviatura y pulsar el botón **Edit**. El cuadro siguiente se manipula como en el apartado anterior.

macros

Son herramientas que facilitan el trabajo común dentro de NetBeans, graban una serie de teclas a las cuales se las asigna un nombre y/o una tecla rápida.

grabar macro

- 1> Pulsar **Ctrl+J, S**, para empezar a grabar
- 2> Escribir lo que se desea grabar en la macro
- 3> Pulsar **Ctrl+J, E** para finalizar la grabación
- 4> Poner nombre a la macro en el apartado **Name**
- 5> Pulsar **Add** y en el cuadro siguiente pulsar la tecla con la que se desee invocar a la macro. Esa tecla ejecutará la macro siempre que se quiera

gestión de macros

En el apartado **Tools-Options-Editing-Editor settings-Java Editor** se puede elegir la propiedad **macros** y al pulsar en el botón de puntos suspensivos, editar o borrar (**remove**) las macros anteriores

javadoc

el entorno javadoc en NetBeans

NetBeans dispone de todo un entorno para crear documentación javadoc

administrador de javadocs

En **Tools-Javadoc manager** se llama al administrador de javadocs, el cual es capaz de montar filesystems que contienen documentos javadoc y gestionar su uso. Lo normal es que haya una carpeta llamada *docs*, dentro de la raíz del paquete que es en la que se genera la documentación.

En la carpeta de documentación el archivo *index.html* será el índice general. Mediante el administrador de archivos se debe montar esta carpeta (**Mount**) para que pueda ser buscada.

generar documentos javadoc

Para ello se selecciona el elemento a documentar y se elige **Tools-Generate javadoc**. Un mensaje indicará en que carpeta se almacenará la documentación. En NetBeans sólo funciona de forma correcta la documentación generada para todo un paquete, carpeta o archivos ZIP y JAR.

Se puede generar también documentación sólo para un archivo, pero no funcionarán bien el resto de funciones.

La documentación se guarda en una carpeta que inicialmente elige el programa, pero que se puede cambiar y elegir cada vez si se configuran adecuadamente las opciones.

opciones de las herramientas javadoc

En **Tools-Options-Code documentation** se puede configurar el comportamiento de los javadocs. Por ejemplo en el apartado **Documentation** si ponemos la propiedad **Ask for destination** a **True** entonces cada vez que generemos un javadoc para un archivo se nos preguntará donde lo queremos crear.

mostrar documentación javadoc de un elemento

Para ello desde el explorador basta seleccionar el elemento sobre el que se quiere revisar y pulsar **Alt+F1** o **Show javadoc**

autocomment

También en **Tools** está la herramienta de autocomentarios que permite colocar comentarios javadoc desde el cuadro en lugar de escribirles a mano.

pasos para la creación de la documentación

En base a los puntos anteriores los pasos más lógicos para crear documentación desde NetBeans serían:

- 1> Crear una carpeta (normalmente **docs**) en el filesystem que se desea documentar
- 2> En **Tools-Options-Code documentation-Documentation** colocar la propiedad **Ask for destination** a **True**
- 3> Seleccionar el filesystem pulsar el botón secundario del ratón en él y elegir **Tools-Generate javadoc**. Se nos preguntará por la carpeta en la que generar la documentación, habrá que elegir la carpeta docs. Tras generar, el índice general lo llevará el archivo **index.html**
- 4> Ir a **Tools-Javadoc manager** y montar (**mount**) la carpeta en la que se encuentra la documentación (carpeta docs)
- 5> Cada vez que se quiera mostrar la documentación de un elemento concreto, colocar el cursor en él y pulsar **Alt+F1**