

[Saltar al contenido](#)

d.CodinGames();

Notas sobre algoritmia, programación y enseñanza de la programación

Buscar:

d.CodinGames();

Notas sobre algoritmia, programación y enseñanza de la programación

- **CONCEPTOS BÁSICOS**

Guía básica para usar Netbeans

POR [GABY NIEVA](#) · PUBLICADA 2 NOVIEMBRE, 2015 · ACTUALIZADO 16 MAYO, 2017

Para programar en Java no es necesario utilizar un ambiente de desarrollo específico. Sin embargo para aquellos que están aprendiendo a programar, usar un IDE (Integrated Development Environment) puede ser de gran ayuda.

Eclipse y Netbeans son dos de los IDEs más populares para programar en Java, pero no son los únicos. No tengo nada en particular contra uno u otro IDE, simplemente me gusta Netbeans porque es el IDE que vengo usando desde hace mucho tiempo, pero las críticas son bienvenidas, sobre todo si me ayudan a trabajar mejor y más rápido.

Netbeans, así como Eclipse, se puede usar para programar no sólo en Java, sino también en C++, PHP, HTML y una larga lista de etcétera.

En este post te guiaré paso a paso para crear un programa sencillo en Java usando Netbeans.

NOTA: En este artículo estaré haciendo referencia a la versión más reciente de Netbeans que es la 8.1. Sin embargo, este entorno de

programación se mantiene en constante actualización, así que algunas instrucciones y/o ventanas serán un poco diferentes en las versiones venideras.

Instalación

Antes de instalar Netbeans, debes instalar la máquina virtualmente Java, es decir el [Java Development Kit](#) o **JDK**. Después de descargar el JDK, en Windows haces doble click sobre el archivo para instalar la máquina virtual de Java.

Después descarga Netbeans desde este [link](#). De las opciones mostradas, si no tienes una muy buena conexión a Internet, puedes elegir la primera de ellas (Java SE), la cual te permitirá instalar el paquete básico para trabajar con la edición estándar de Java.

Creando tu primer programa.

La estructura de proyecto.

Netbeans maneja una estructura para trabajar con los programas de java que te permitirán organizar tu trabajo.

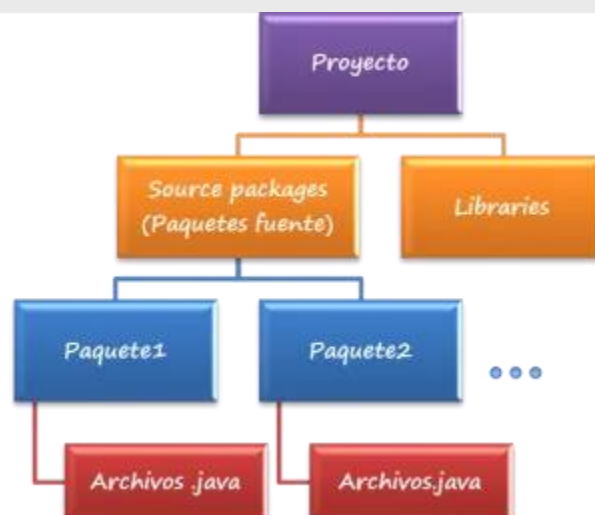


Figura 1. Organización de Proyectos en Netbeans

Como se aprecia en la figura 1, en el nivel más alto de la jerarquía se encuentra el **Proyecto**, el cual agrupa al menos dos carpetas : los paquetes fuente y las bibliotecas o librerías.

Los paquetes fuente o *Source packages* es donde propiamente se guardarán tus archivos de código. Cada paquete agrupará a uno o más archivos .java, pero sólo uno de ellos deberá contener el método *main*. Puedes crear varios paquetes en un proyecto, y a su vez, cada paquete podrá tener su archivo .java con su método main.

Por el momento, y dado que aún no estamos usando Java con el paradigma de Programación Orientada a Objetos, crearás un proyecto con un paquete que contendrá sólo un archivo .java.

Los proyectos se crean en una carpeta con el nombre que le colocaste al crearlo en Netbeans. La carpeta se encuentra en Documentos y se llama *NetBeansProjects*.

Pasos para la creación de un proyecto.

Entonces, para crear tu primer programa, los pasos son los siguientes:

1. En Netbeans, haz click sobre *File*→*New Project...* o presiona las teclas

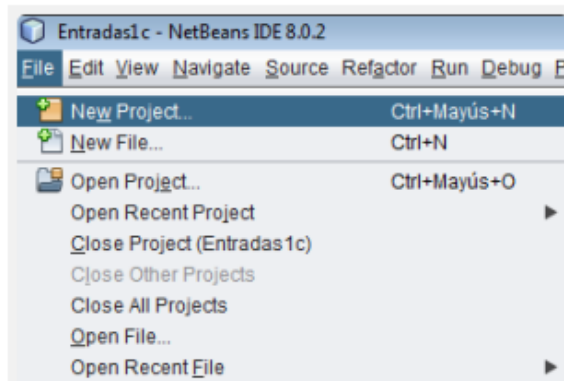


Figura 2. Menú File-> New Project...

2. En el cuadro de diálogo que se abrirá, selecciona el tipo de proyecto a realizar, en este caso en *Categories* selecciona Java y en *Projects: Java Application*. Y presiona el botón *Next*.
3. En el siguiente cuadro de diálogo, escribe el nombre del proyecto (En la figura 3, se indica con el número 1), el cual **DEBE** iniciar con una letra mayúscula y **NO CONTENER** espacios en blanco ni caracteres especiales. (Confía en mí, aunque Netbeans no te haga ninguna indicación, estos detallitos pueden generar que después no puedas compilar tu programa) No realices cambios en las cajas marcadas con los números 2 y 3. Asegúrate de dejar la casilla de verificación *Create Main Class* seleccionada para que Netbeans haga el trabajo de crear el archivo con el método *main*. (Inciso 4). Por último, presiona el botón *Finish*.

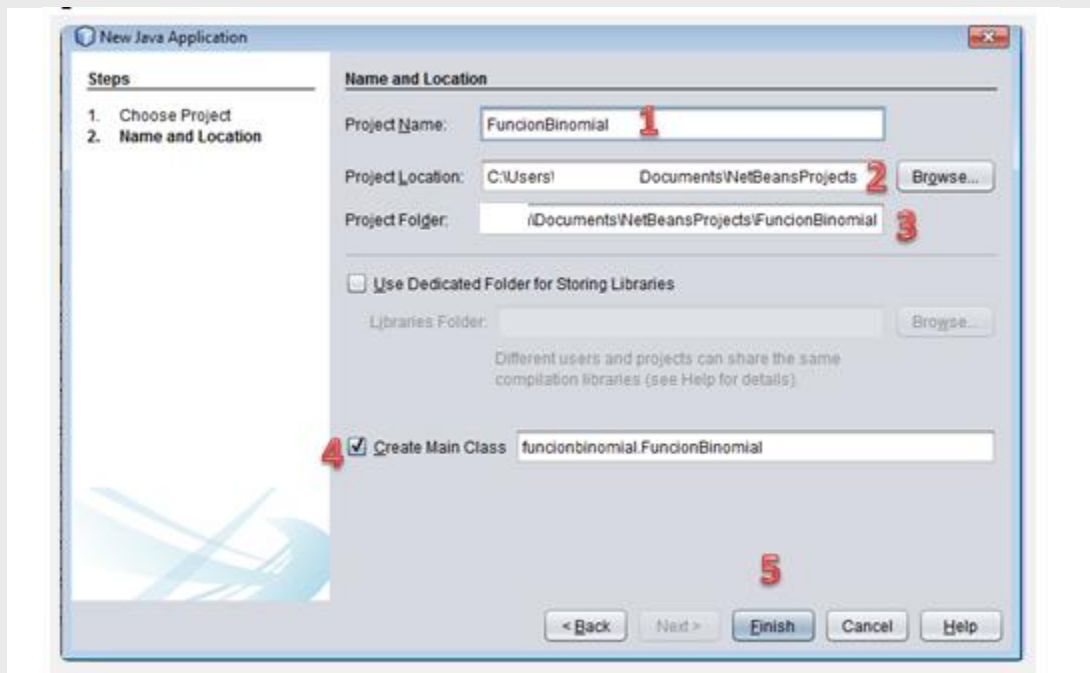


Figura 3. Cuadro de diálogo New Java Application

4. A continuación Netbeans generará la estructura organizacional de tu proyecto, y el archivo principal .java. En la figura 4 puedes ver una guía rápida que te guiará por la interfaz gráfica de éste IDE.

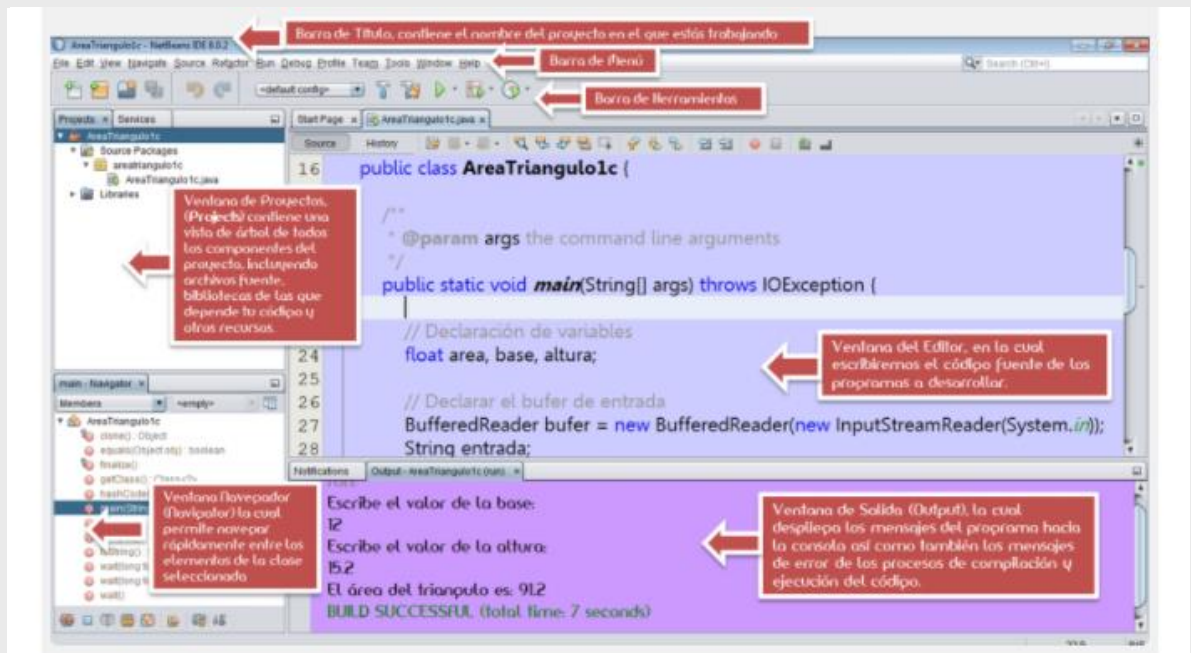


Figura 4. Interfaz gráfica del IDE Netbeans.

Utiliza el editor para capturar el código de tu programa y aprovecha las ayudas que el IDE te ofrecerá conforme vayas introduciendo las instrucciones. Recuerda guardar constantemente tu código y cuando hayas terminado, puedes presionar el botón *Run Project* o la tecla F6. Si tienes errores, se te mostrarán en la ventana *Output* (ver Figura 4), o si todo funciona correctamente, también en esta ventana se mostrará la salida de tu programa.

La barra de herramientas de Netbeans.

En la figura 5 te muestro los íconos que más utilizarás de Netbeans.

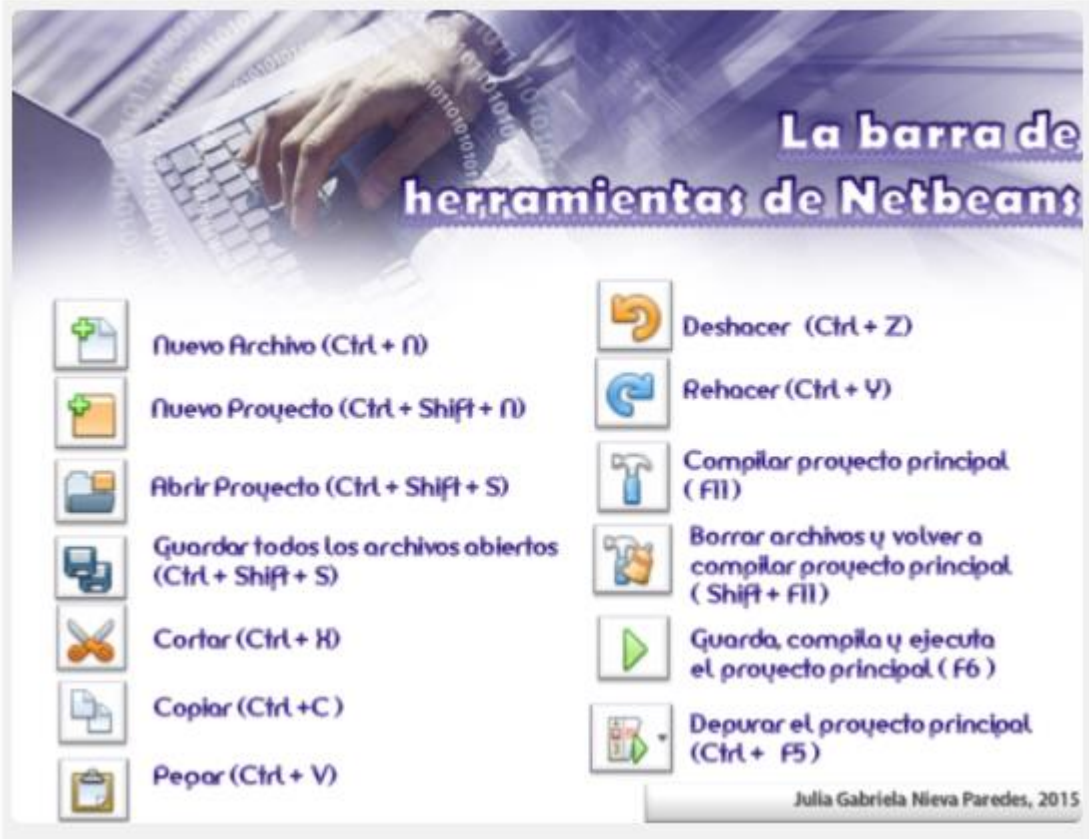


Figura 5. La barra de herramientas de Netbeans

Recibiendo ayuda de Netbeans.

Atajos del teclado.

Los *shortcuts*, *hot keys* o atajos del teclado, son una combinación de teclas que todo software utiliza para ejecutar una acción determinada, en lugar de obligar al usuario a ir por el camino largo: click en el menú adecuado y posteriormente en el comando requerido. Y Netbeans no es la excepción. Consulta aquí ([shortcuts-80](#)) el listado de atajos de teclado que utiliza este IDE. Además de ayudarnos con los comandos del entorno, Netbeans agrega combinaciones de teclas para ayudarnos a escribir código. Por ejemplo, si quieres escribir la línea para imprimir un mensaje de texto en la consola de



salida, puedes escribir `sout` y la tecla tabulador : para que Netbeans la sustituya por `System.out.println(“”);`

Autocompletar.

Creo que lo más genial que tiene Netbeans es la capacidad de completar de forma automática cuando estás usando clases de la API de Java. Sólo ten paciencia y cuidado de no equivocarte con la palabra inicial para que Netbeans reconozca el paquete y la clase que necesitas. Te muestro un ejemplo en la figura 6.

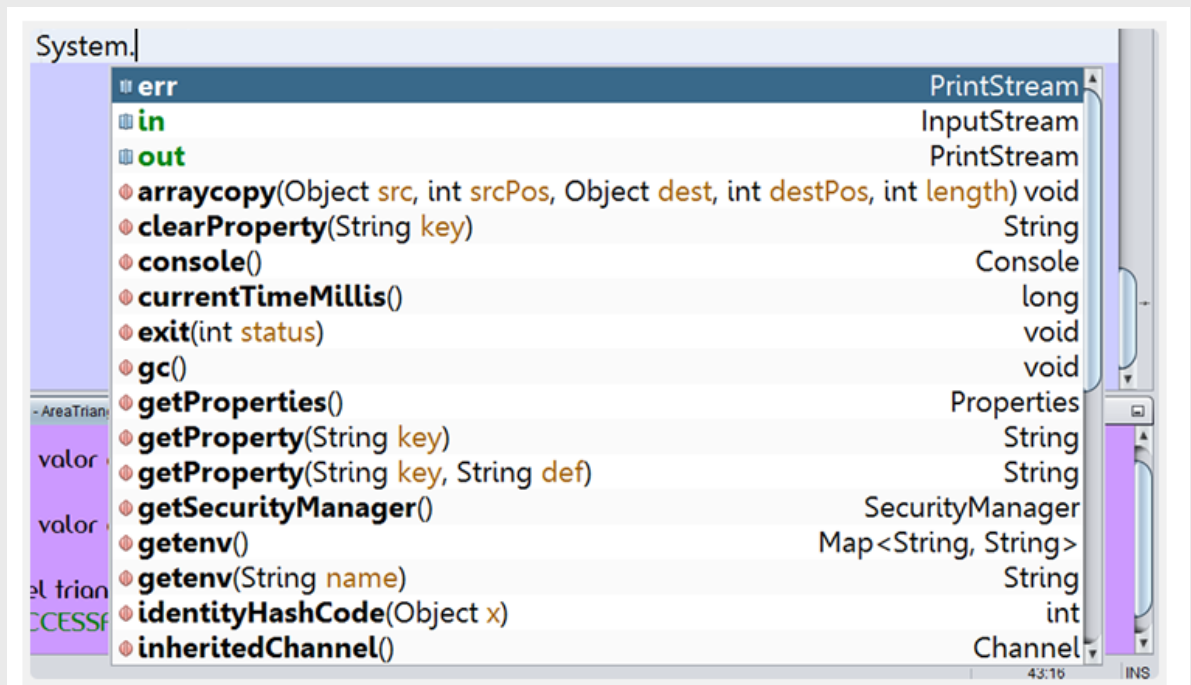


Figura 6. Ejemplo de la característica autocompletar del editor de Netbeans

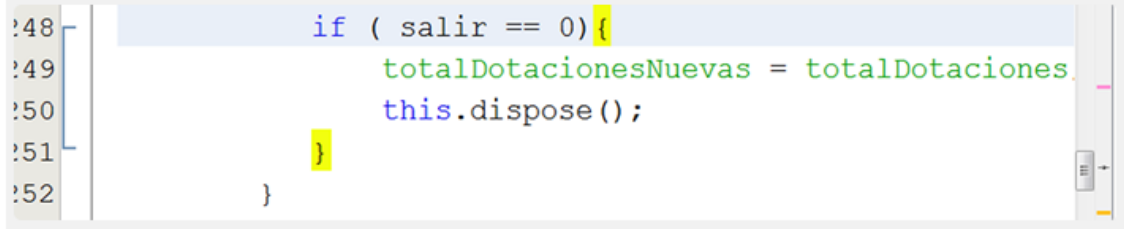
Sistema de colores y estilos de fuentes.

Netbeans te guía para que sepas si estás realizando las cosas bien con todo un sistema de colores y estilos de fuentes . Presta atención, y conforme

vayas editando tus programas te darás cuenta cuando algo no está bien gracias al color o estilo de fuente en que Netbeans va colocando lo que escribes. Por ejemplo, los tipos de datos nativos de Java los coloca en color azul (por lo menos en la configuración estándar). Las cadenas de texto las coloca en naranja. Y así existen otras guías usando colores.

Detección automática de {[(.

Si escribes una llave de apertura { e inmediatamente después presionas “Enter”, Netbeans automáticamente escribirá la llave de cierre. Además, si te es confuso hacer el seguimiento de qué llave abre cuál estructura de control, basta con colocarse en una de ellas, y Netbeans mostrará su respectiva llave de cierre o de apertura, marcándolas en amarillo. Estas dos características también se aplican al uso de paréntesis () y de corchetes []. (Observa la figura 7).

A screenshot of the NetBeans IDE editor window. The code is as follows:

```
:48     if ( salir == 0) {
:49         totalDotacionesNuevas = totalDotaciones
:50         this.dispose();
:51     }
:52 }
```

The opening curly brace on line 48 and the closing curly brace on line 51 are highlighted in yellow. The code is color-coded: 'if' is blue, 'salir == 0' is black, '{' is yellow, 'totalDotacionesNuevas' is green, '=' is black, 'totalDotaciones' is green, '.' is black, 'this.dispose()' is blue, ';' is black, and '}' is yellow.

Figura 7. Ejemplo de detección automática de llaves, corchetes y paréntesis

Detección de errores.

Obviamente la ayuda más importante de Netbeans es indicarte los errores mientras estás escribiendo. Incluso antes de compilar, este IDE te va indicando los errores, y lo hace de diversas formas. En la figura 8 te muestro las múltiples formas en que Netbeans te indica que tienes errores de sintaxis.

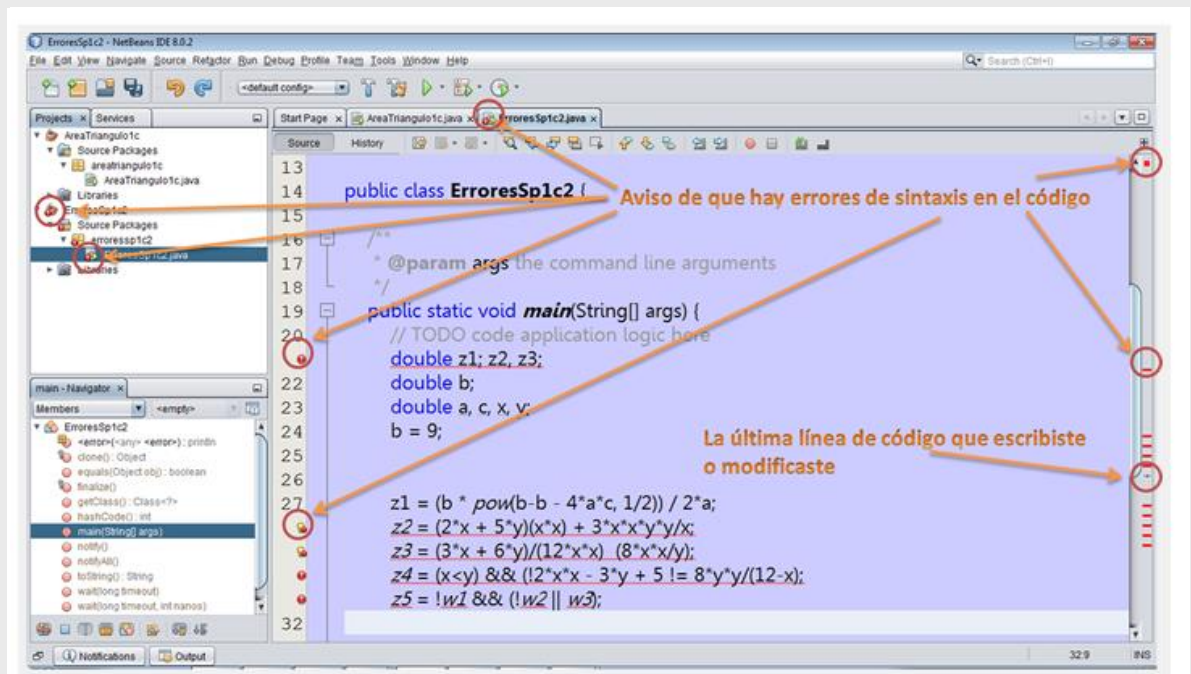


Figura 8. Formas en la que Netbeans detecta los errores de sintaxis.

Mensajes de error.

Aunque Netbeans te indica rápidamente que existe un error, corregirlo quizá no resulte tan fácil, sobre todo porque muchos de estos errores se marcan en la línea en la que el error no se genera. Recuerda que el error probablemente se encuentre una línea arriba. Otra cosa que no ayuda, es que la explicación del error se te muestra en inglés, y como ya mencioné en otros [posts](#) sino eres amig@ de este idioma se te puede complicar el proceso de depuración. En la figura 9 te muestro algunos errores comunes y como puedes solucionarlos. Esta tabla aún no está completa, la hice hace tiempo, para la versión 7, por lo que más adelante la estaré actualizando.

Mensaje de error	Significado	Solución
cannot find symbol	No puede encontrar el símbolo	Una variable no está declarada, o está mal escrita. También puede indicar que un método de una clase se encuentra mal escrito.
incompatible types	Tipos incompatibles	Para que se pueda realizar esa operación, la variable debe ser declarada con otro tipo de dato.
; expected	Se requiere ;	Falta el ; con el que deben terminar todas las instrucciones Java También puede marcar este error cuando alguna expresión aritmética no se ha escrito correctamente
unclosed string literal	Literal de cadena no está cerrada apropiadamente	Verificar que las " encierren adecuadamente una cadena de texto.
package does not exist	Paquete ---- no existe	Netbeans buscó un paquete que no existe dentro de los paquetes estándar de Java. Lo más probable es que una clase se encuentre mal escrita

Figura 9. Algunos errores comunes