**M A N U A L**

**D E**

**N E T B E A N S**

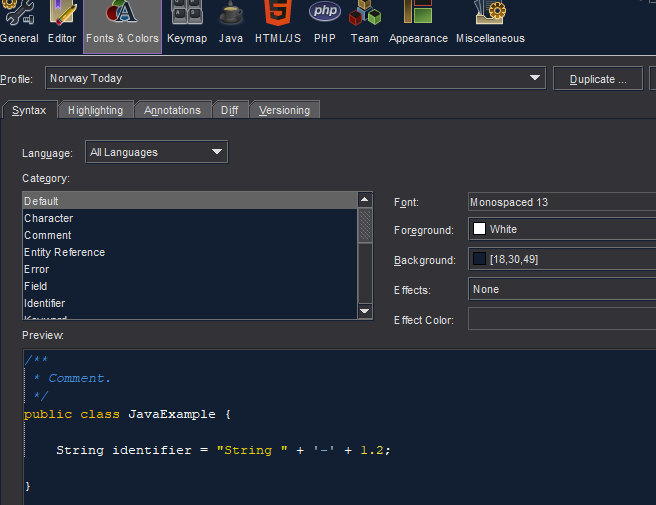
DAVID ESPINOSA GONZALEZ 1 DAW IES PALOMERAS VALLECAS

**I N D I C E :**

1. Personalización del entorno. 3
2. Proceso de utilización del depurador. 6
3. Shortcuts. 12
4. Code templates 13
5. Personalización del entorno.

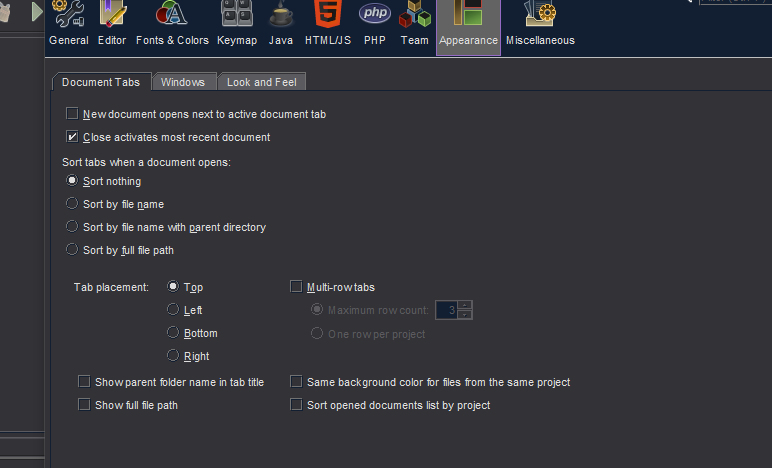
Personalmente me gustan más los tonos más oscuros porque tengo el hábito de trabajar o estudiar mas durante la noche por la tranquilidad que obtengo. Este es el motivo por el que normalmente suelo tener el modo nocturno u oscuro en casi todos los sitios posibles. Por ello, mi personalización del Entorno de desarrollo NetBeans ha sido hacia ese estilo.

Lo primero que hice fue cambiar la fuente, eso se puede hacer si en la barra superior desplegamos el apartado de “Tools” y seleccionamos la ultima opción, que corresponde a “Options”. Se abrirá un desplegable en el que se puede observar una opción que se llama “Fonts & Colors”. La opción que yo he escogido ha sido la del perfil de “Norway Today”.

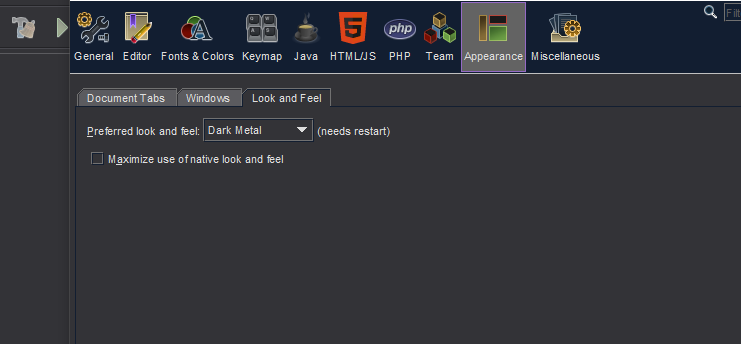


He de dejar claro que esta foto es ya de todo el proceso realizado, al cambiar esta opción solamente estas modificando la fuente y el color de la fuente, no el color o la apariencia del resto del entorno. La apariencia del resto del entorno, la detallare a continuación.

Sin cerrar las opciones, vuelve a desplegar “Tools” y después “Options” si cerraste esa ventana, nos desplazamos hasta la derecha del todo casi. La opción que nos interesa ahora es la de “Appearance”. En esta opción es en la que cambiaremos todo el conjunto del entorno y con la que se puede modificar los colores del entorno. En esta pestaña de la ventana, se puede modificar la posición de las pestañas del entorno, pero la opción que resulta mas llamativa por los cambios visuales en si, es la ultima opción, que se llama “Look and Feel”.



En “Look and Feel”, es donde el entorno dará el cambio más drástico visualmente. Aquí yo tengo la opción marcada de “Dark metal” que es la que me resulta más cómoda y menos molesta si estoy delante de la pantalla durante muchas horas.



Hay muchos temas diferentes en el desplegable. NetBeans es un entorno totalmente adaptable a los gustos visuales del desarrollador/estudiante.

1. Proceso de utilización del depurador.

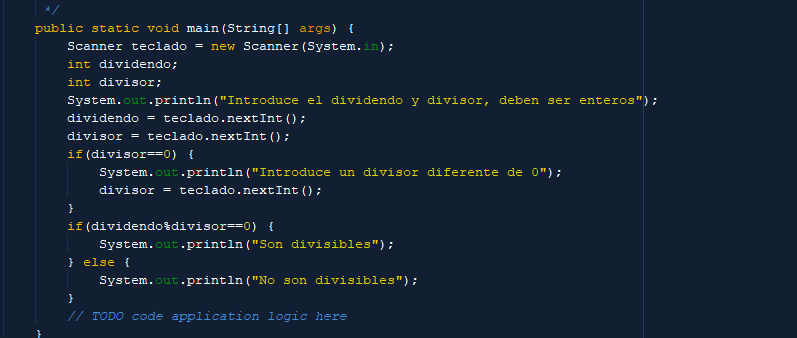
Para mostrar este proceso, el enunciado pide realizar un ejercicio del tema 3 de condicionales en java. En concreto para detallar el proceso de utilización del depurador yo he escogido el ejercicio número 3.

Ej\_3:

Introducir dos números por el teclado, dividendo y divisor. Si dividendo es divisible por divisor, el programa ha de mostrar el mensaje “DIVISIBLES”.

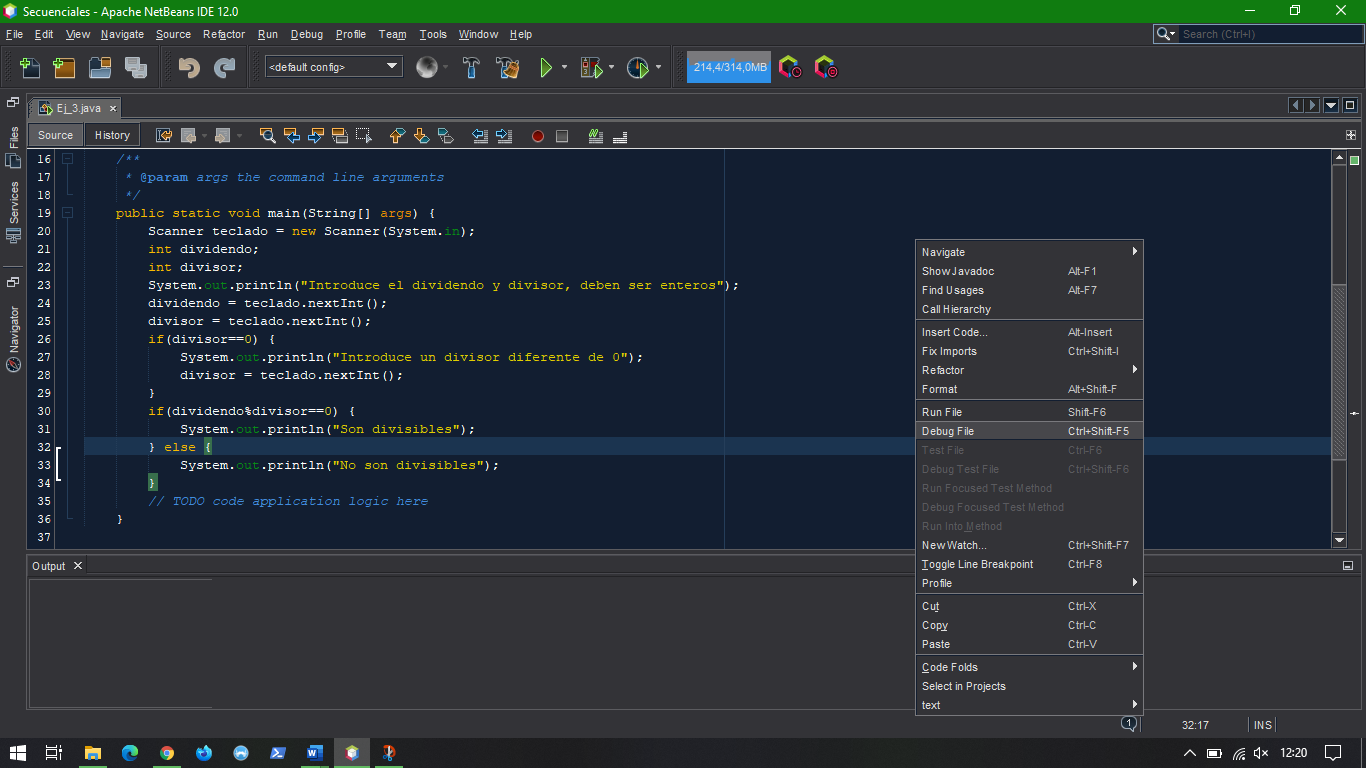
A continuación, se muestra el proceso de utilización del depurador:

Primero se escribe el código en el entorno:



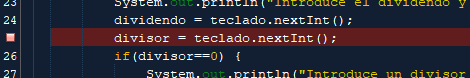
Después se abre el depurador con el shortcut(Ctrl+Shift+F5), con el click derecho y buscando la opción de debug file, o pinchando en el botón superior correspondiente.





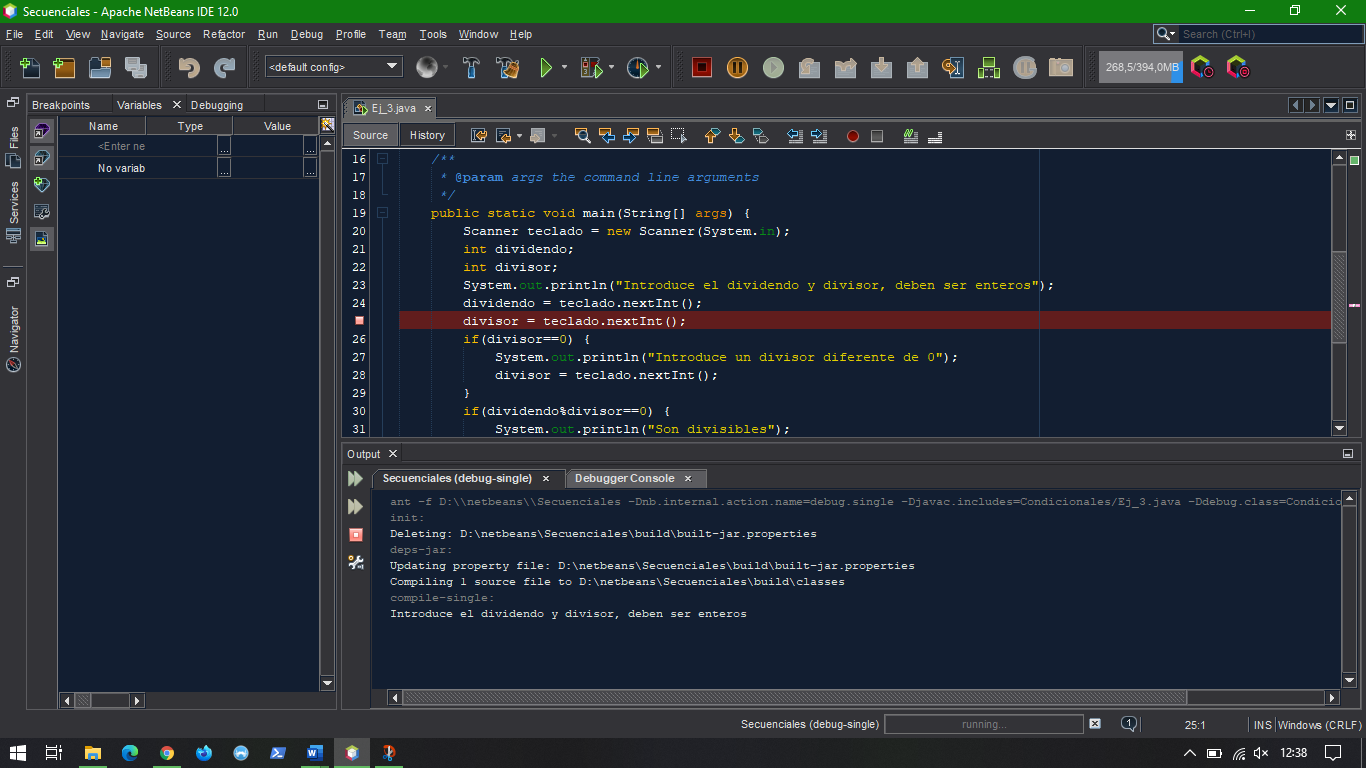
Una vez lanzado el debug file, se abrirán varias pestañas, en mi caso, la pestaña inferior es la que devuelve el resultado del proceso, es el “Output”, y la pestaña que se abre a la izquierda, es la de la que marca los “BreakPoints”, las Variables y el “Debuggin”.

Un “BreakPoint” es una marcación de una línea para poder parar la ejecución del código en ese punto. Sirve para poder ejecutar el código paso a paso y de esta manera poder ver donde está fallando el código, en el caso que algo este mal, o para poder ir observando como el debuggeador recorre el código. Los BreakPoints se marcan pinchando en el numero de la línea en la que queremos hacer la parada. Es importante que se deben de marcar los BreakPoints antes de lanzar el código al debuggeador.

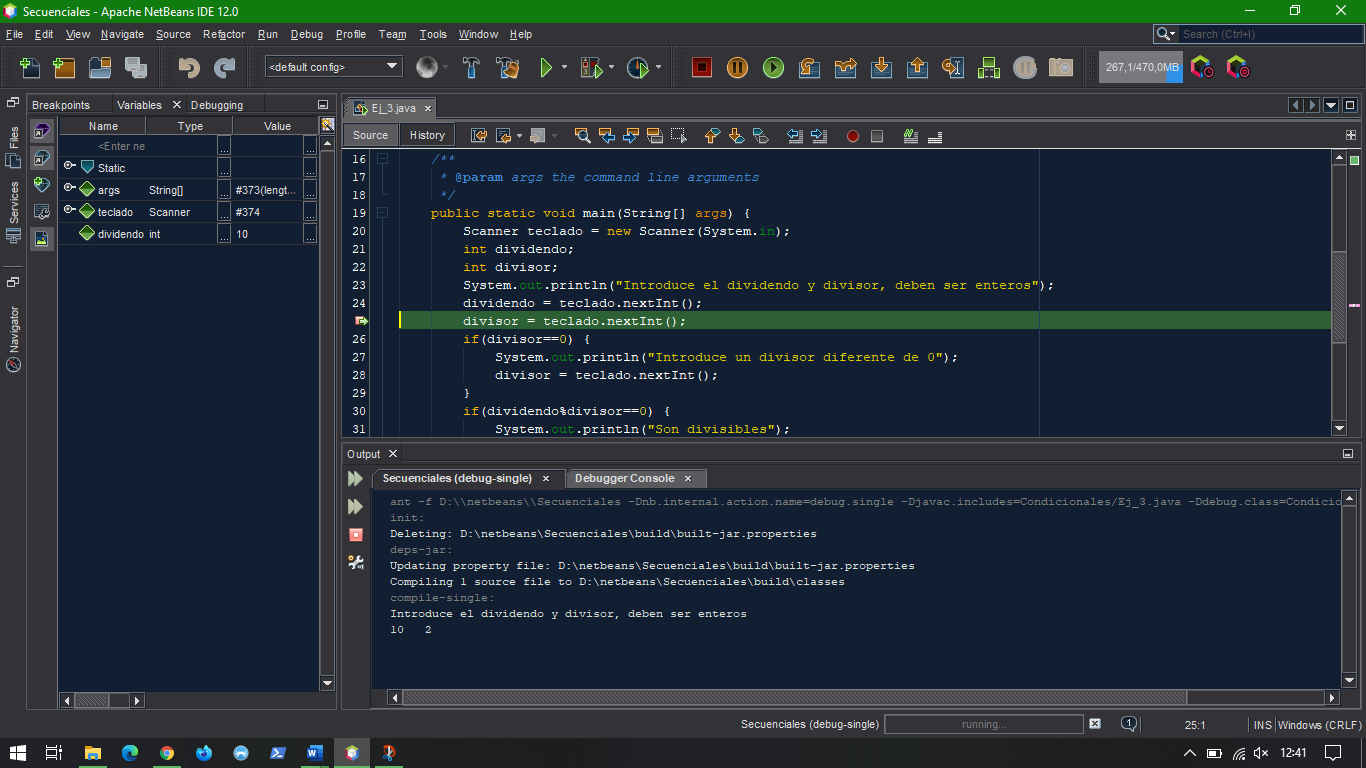


Al pinchar en el número de la línea, se marcará la línea entera en rojo, y aparecerá un cuadrado en el lugar del número.

A continuación, se muestra un pantallazo del entorno con el debuggeador lanzado y un breakpoint marcado para poder ir revisando poco a poco el código.



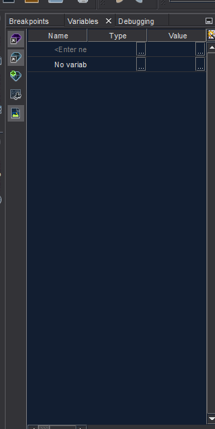
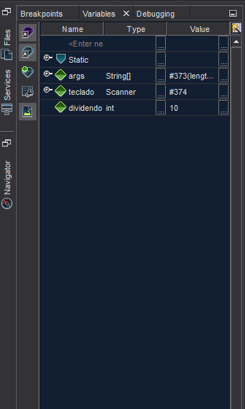
Tal y como se observa, en el output esta pidiendo que introduzcamos las variables. Cuando esto ocurra, la pestaña que se abrió a la izquierda se modificara para mostrarnos las variables que hemos introducido. Es importante señalar que esas variables si sufren modificaciones durante el código también se podrán ver reflejadas en ese apartado.



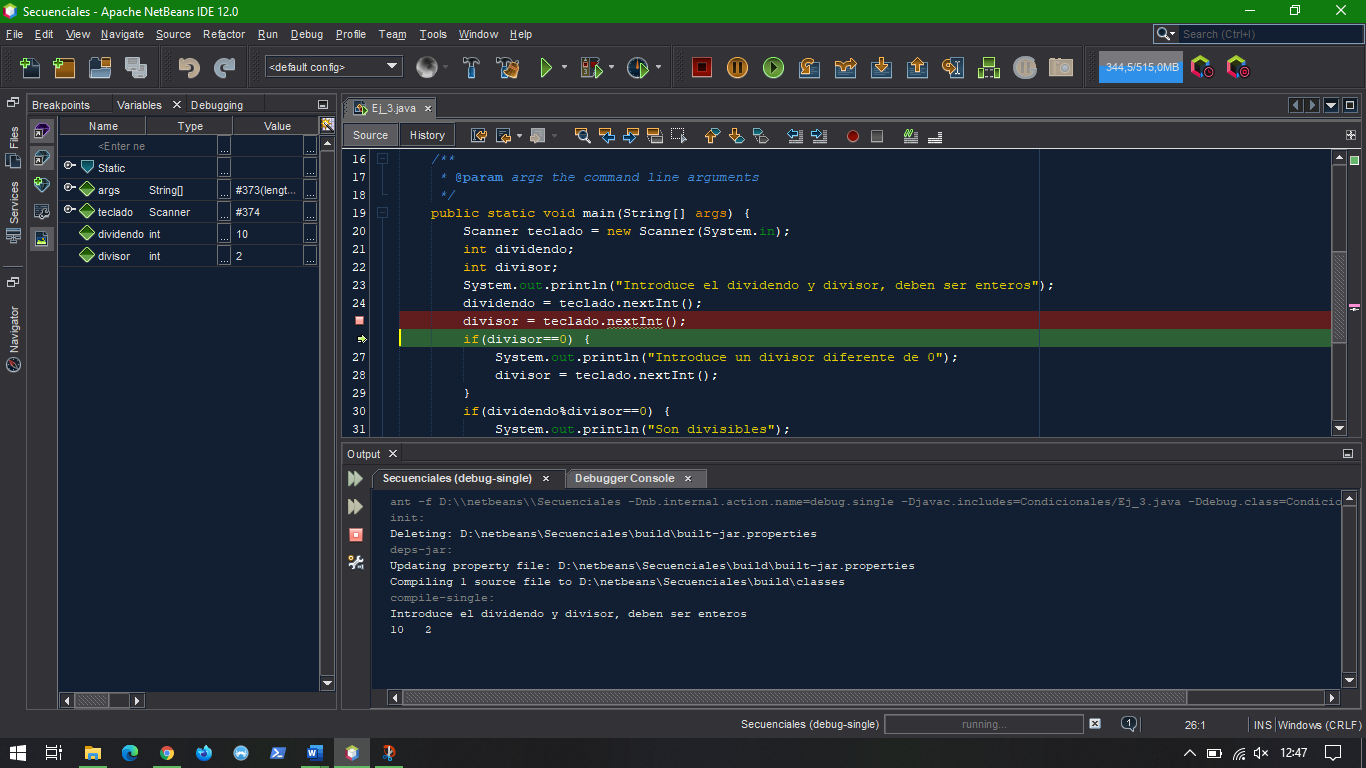
Para avanzar en el código, después de que este llegue al breakpoint, podemos hacerlo mediante el shortcut de F7, o con el botón de la parte superior correspondiente.

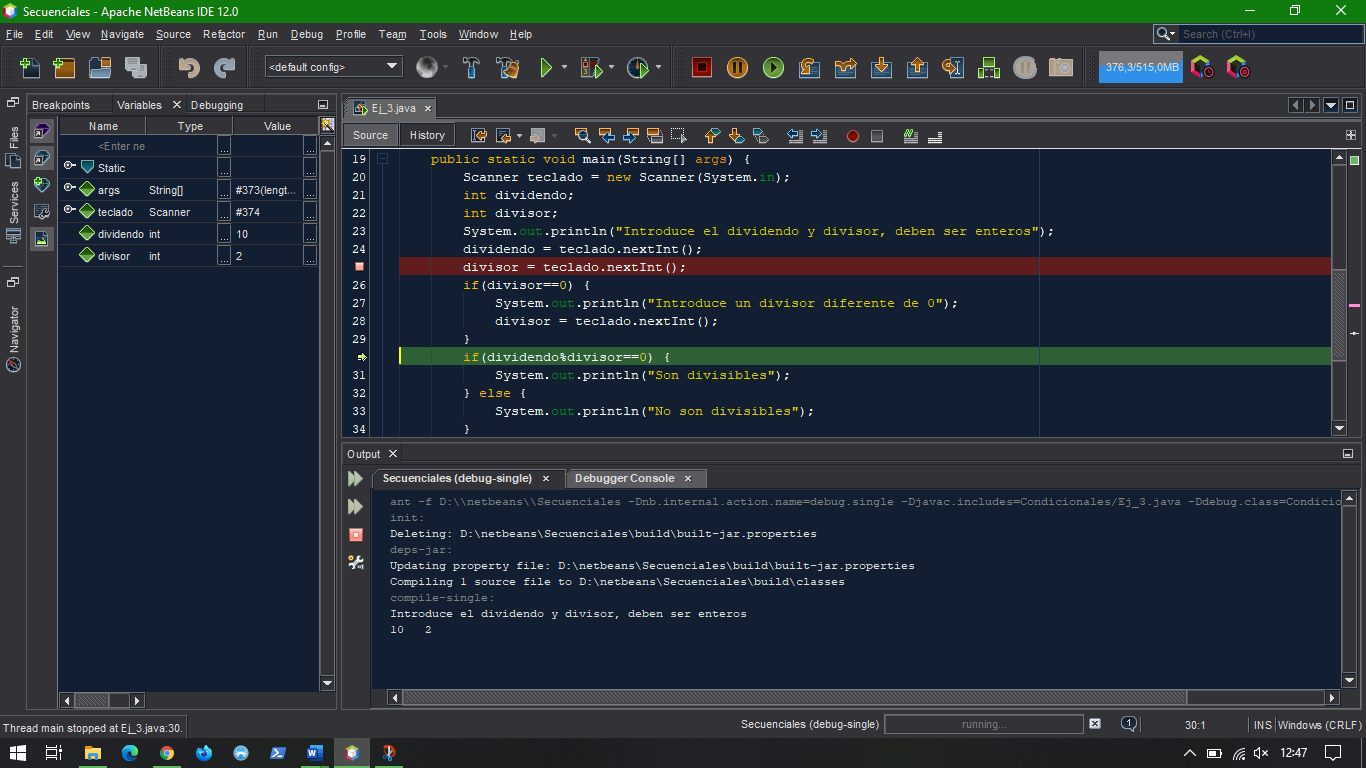


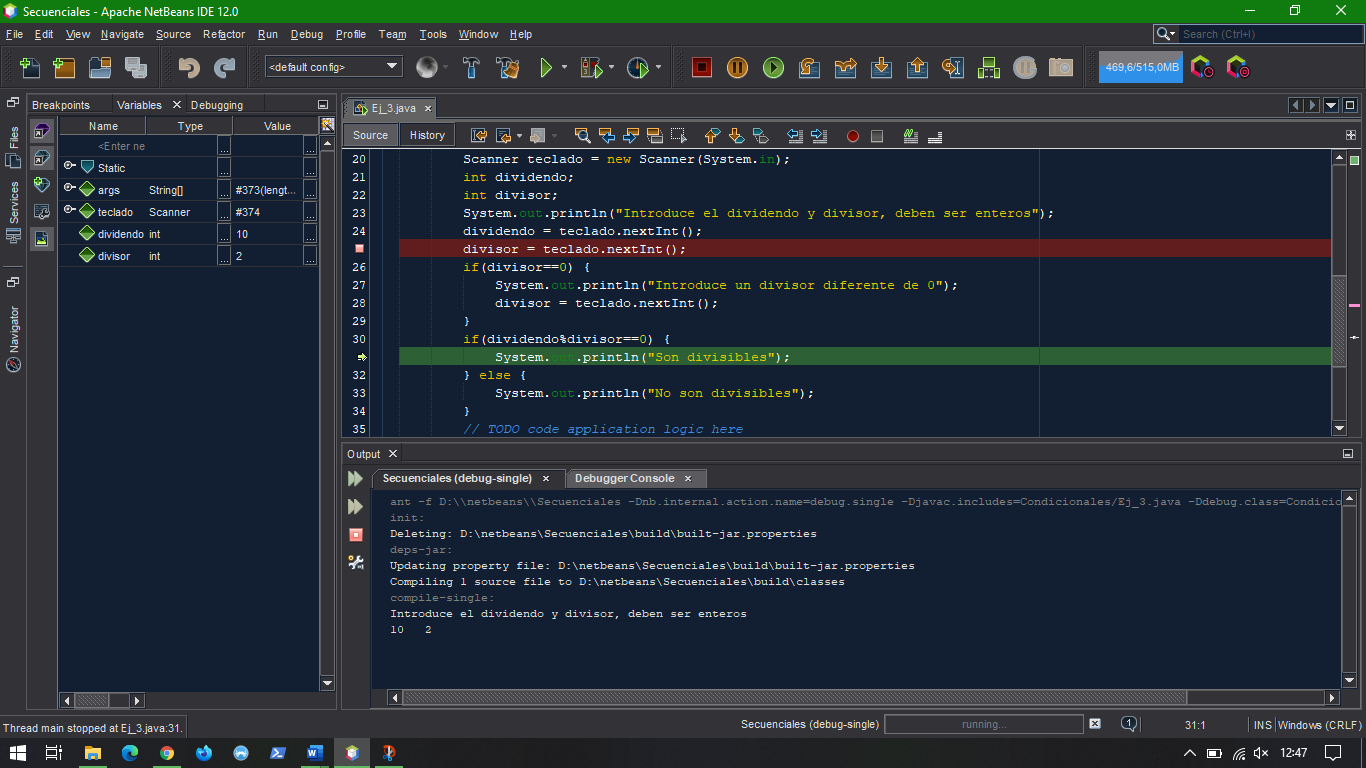
Se puede observar, que ha habido una diferencia entre el primer pantallazo y sus variables y el segundo.

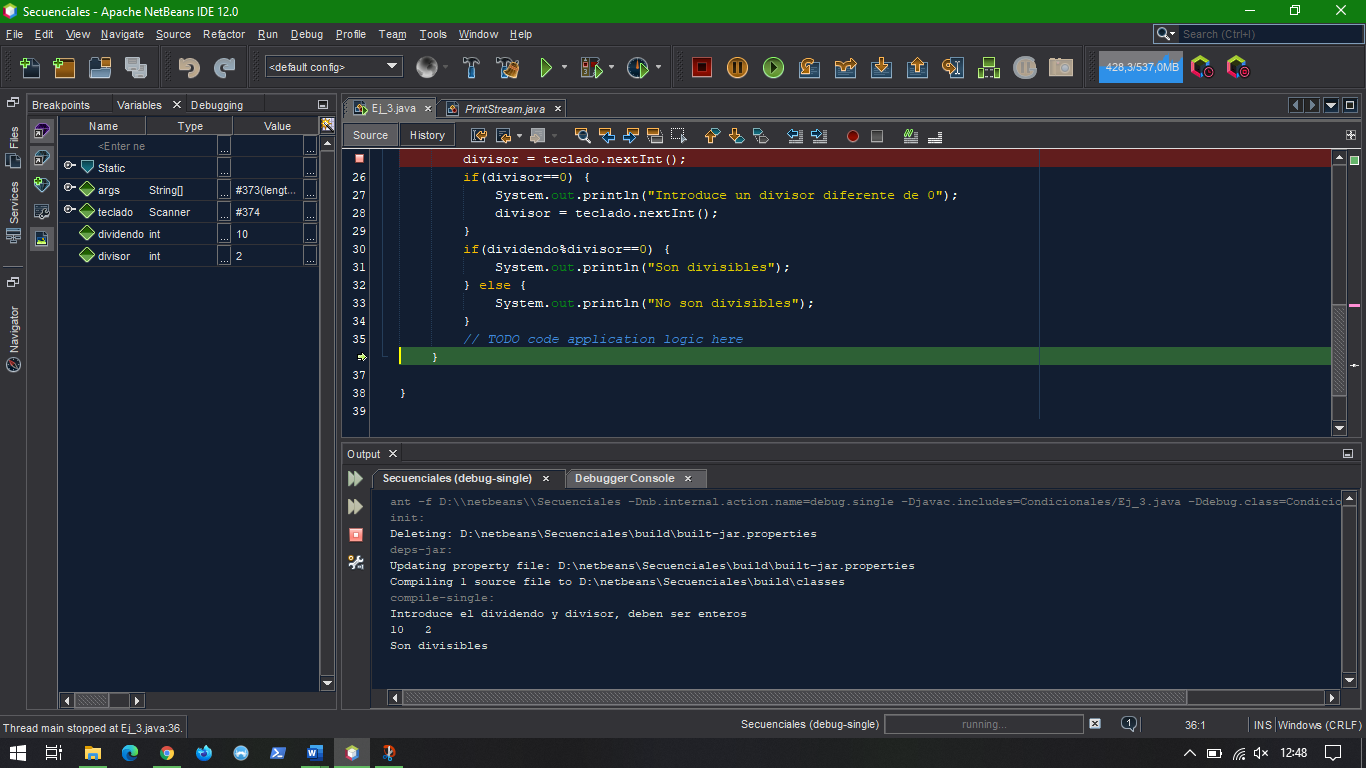


A continuación, muestro una serie de imágenes con el debugeador funcionando:









La línea de color verde representa la línea en el que el debugeador se encuentra en ese momento. Tal y como se puede ver es un método para controlar como se comporta el código, como nuestras variables se ven modificadas, y para la posible corrección de errores.

Es importante señalar que cuando la etapa del debugeador llega a un método de NetBeans, el debugeador nos abrirá una pestaña nueva para ir avanzando en ese método paso a paso. Para evitar eso, puedes saltar introducirte dentro del método con Shift+f8, o presionando el siguiente botón.



1. Shortcuts

Un shortcut es un atajo de teclado que puede servir para acelerar el uso del programa si se conocen bien.

Desde mi punto de vista, los 10 shortcuts que más utilizo son:

* Shit+f6: Este shortcut lo que hace es ejecutar el fichero abierto.
* Ctrl+Shitf+f5: este lo que hace es ejecutar sobre el fichero el debugeador.
* Shift+f5: sirve para detener el debugeador si este está lanzado.
* Ctrl+n: sirve para abrir un fichero nuevo.
* Ctrl+Shift+n: sirve para abrir un proyecto nuevo.
* Ctrl+e: elimina toda la línea de código.
* Ctrl+z: deshace la última acción realizada.
* Ctrl+s: realiza un guardado rápido del fichero.
* Ctrl+Shift+s: realiza un guardado de todos los ficheros que se han modificado.
* Ctrl+Shift+enter: activa o desactiva el modo ventana completa (solo aparece la ventana del código.

1. Java Code Templates

Un java code template, es una abreviación o acrónimo que sirve para acelerar aun mas la escritura del código por que permite que escribiendo 3 palabras se escriba 1 expresión entera.

Un ejemplo de ello seria sout. Cuando escribimos sout, y presionamos el tab, NetBeans nos transforma sout en: System.out.println(“”);

Los 5 java code templates que mas utilizo son:

* Sout: System.out.println(“”);
* Br: break;
* Dowhile: do { } while(condition);
* Ifelse: if (condition) { } else { }
* Bo: boolean

He creado un nuevo code template, para acelerar el uso de este método que últimamente en clase usamos mucho:

scat: Scanner teclado = new Scanner(System.in);

