

UD 2.2

INSTALACIÓN Y USO DE ENTORNOS IDE

ENTORNOS DE DESARROLLO

PARTE 2 DE 4: INSTALACIÓN, USO Y DEBUGGING DE ECLIPSE (IDE)

Autor: Sergio Badal

Modificado por: Raúl Palao

Fecha: 26-10-2023

Licencia Creative Commons

versión 2.0



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

ÍNDICE DE CONTENIDOS

1. PRESENTACIÓN de eclipse.....	1
1.1 ORÍGENES DE ECLIPSE.....	1
1.2 ¿QUÉ ES ECLIPSE?.....	2
1.3 VERSIONES DE ECLIPSE.....	3
2. INSTALACIÓN DE ECLIPSE.....	4
2.1 INSTALACIÓN Y PRUEBA DE ECLIPSE.....	4
3. Depuración EN ECLIPSE.....	7
3.1 EN QUÉ CONSISTE.....	7
3.2 DEPURACIÓN EN ECLIPSE.....	7
4. BIBLIOGRAFÍA.....	9

1. PRESENTACIÓN DE ECLIPSE

1.1 ORÍGENES DE ECLIPSE

Información de <http://java-white-box.blogspot.com>

Gran parte de la programación de Eclipse fue realizada por IBM antes de que se creara el proyecto Eclipse como tal. **El antecesor de Eclipse fue VisualAge de IBM** y se construyó usando [Smalltalk](#) en un entorno de desarrollo llamado Envy.

Con la aparición de Java en la década de los 90, IBM desarrolló una máquina virtual válida tanto para Smalltalk y Java. La rápida expansión de Java y sus ventajas con miras a una Internet en plena expansión **obligaron a IBM a plantearse el abandono de esta máquina virtual dual y la construcción de una nueva plataforma basada en Java desde el principio**. El producto final resultante fue Eclipse, que ya había costado unos 40 millones de dólares a IBM en el año 2001.


A finales de 2001, IBM y [Borland](#), crearon la **Fundación sin ánimo de lucro Eclipse**, abriéndose así al mundo de código abierto. A este consorcio se han unido progresivamente importantes empresas del desarrollo de software a nivel mundial: Oracle, Rational Software, Red Hat, SuSe, HP, Serena, Ericsson, Novell, entre otras.



Hay dos ausencias significativas:

- **Microsoft** ha sido excluida por su posición de monopolio del mercado
- **Sun Microsystems** cuenta (o contaba en su momento) con su propio IDE y principal competencia de Eclipse: NetBeans.

De hecho, **el nombre de Eclipse fue elegido porque el objetivo era crear un IDE capaz de "eclipsar a Visual Studio" de Microsoft**. Mas allá de eso afirmaron que el juego de palabras ("eclipsar el sol") no era hacia Sun Microsystems (Sun = Sol, en inglés), y se querían asociar con ellos, pero el departamento de marketing de Sun Microsystems se negó a realizar negocios con una organización/empresa con un nombre tan contradictorio a ellos, sumado a que el nombre se encontraba ya utilizado en otros ramos, por ejemplo el dominio del sitio oficial **eclipse.org**, **fue comprado a un equipo de fútbol femenino de Illinois**.

<p>RETO</p> 	<p><i>En la Red hay numerosas páginas que te muestran el historial de “dueños” que ha tenido un dominio concreto. ¿Podrías decirnos en qué año consiguió Eclipse el dominio eclipse.org del mencionado equipo de futbol? ¿Sabrías decirnos cuánto pagó la Fundación Eclipse por ese dominio?</i></p> <p><i>Si lo averiguas, cuéntanoslo en el el FORO DE LA UNIDAD...</i></p>
---	--

Esta plataforma (el IDE), típicamente ha sido usada para desarrollar otros IDE, como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent Azureus y [otras muchas](#).

Existen otras versiones, basadas también en Eclipse, y desarrolladas por terceros, como puede ser el caso de STS (Spring Tool Suite), Amzi! Prolog + Logic Server, Goclipse, MyEclipse, TimeStorm, Aptana Studio, Zend Studio,... La lista de IDEs basados en Eclipse es enorme, con lo cual nos hacemos una idea de la importancia de este.

1.2 ¿QUÉ ES ECLIPSE?

Información de <https://www.ecured.cu>

Eclipse es un entorno de desarrollo integrado (IDE) que contiene un espacio de trabajo base y un sistema de plug-in extensible para personalizar el entorno.

Eclipse está escrito en su mayoría en Java y su uso principal es para desarrollar aplicaciones Java, pero también puede ser usado para desarrollar aplicaciones en otros lenguajes de programación a través de plug-ins, incluyendo Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (incluyendo el marco de trabajo Ruby on Rails), Rust, Scala, y Scheme.

Por ejemplo, aquí tienes un tutorial para trabajar con Python 3:

- <https://www.ics.uci.edu/~pattis/common/handouts/pythoneclipsejava/eclipsepython.html>

También puede usarse para desarrollar documentos con LaTeX (a través de un plug-in TeXlipse) y paquetes para el software Mathematica.

Si bien las funciones **nativas** de Eclipse son más bien de carácter general, las características del programa **se pueden ampliar y mejorar mediante el uso de plug-ins**. Asimismo, a través de estos "plugins" libremente disponibles es posible añadir un sistema de **control de versiones** y muchas funcionalidades más.

Características básicas de Eclipse:

- Dispone de un **editor de texto** con resaltado de sintaxis
- La **compilación** es en tiempo real.
- Tiene **pruebas** unitarias con Junit
- **Refactorización**

1.3 VERSIONES DE ECLIPSE

Las versiones de Eclipse han sido nombradas en base a diferentes cuerpos celestes, más específicamente planetas o satélites naturales como Europa, Ganímedes, Calisto, Galileo y Luna.

Después del lanzamiento de PHOTON en Junio del 2018, todas las versiones posteriores se nombran como formato yyyy-MM!

Version name ↕	Date ↕	Platform version ↕
N/A	21 June 2004	3.0 ^[25]
N/A	28 June 2005	3.1
Callisto	30 June 2006	3.2
Europa	29 June 2007	3.3
Ganymede	25 June 2008	3.4
Galileo	24 June 2009	3.5
Helios	23 June 2010	3.6
Indigo	22 June 2011	3.7
Juno	27 June 2012	3.8 and 4.2 ^[33] [Notes 1]
Kepler	26 June 2013	4.3
Luna	25 June 2014	4.4
Mars	24 June 2015	4.5
Neon	22 June 2016	4.6
Oxygen	28 June 2017	4.7
Photon	27 June 2018	4.8

2018-09	19 September 2018	4.9
2018-12	19 December 2018	4.10
2019-03	20 March 2019	4.11
2019-06	19 June 2019	4.12
2019-09	18 September 2019	4.13
2019-12	18 December 2019	4.14
2020-03	18 March 2020	4.15
2020-06	17 June 2020	4.16
2020-09	16 September 2020	4.17
2020-12	16 December 2020	4.18

Old version Older version, still maintained Latest version

2. INSTALACIÓN DE ECLIPSE

2.1 INSTALACIÓN Y PRUEBA DE ECLIPSE

Una vez tengas JAVA JDK en tu equipo, ve a la página oficial, sección descargas, e instala la versión de **Eclipse** para tu sistema operativo.

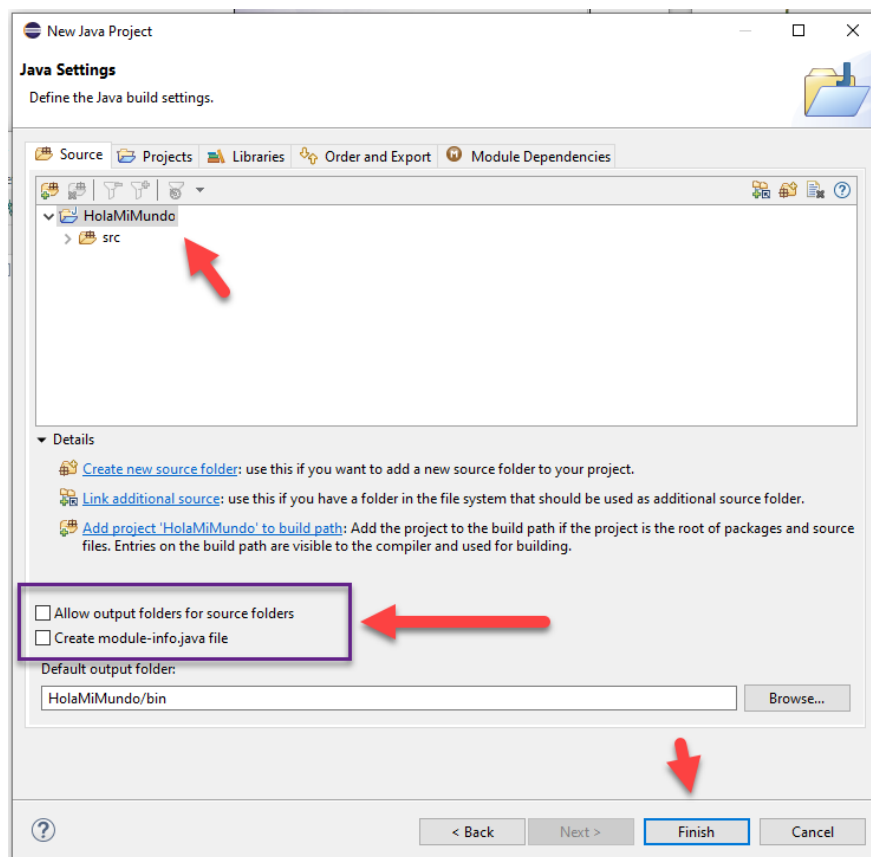
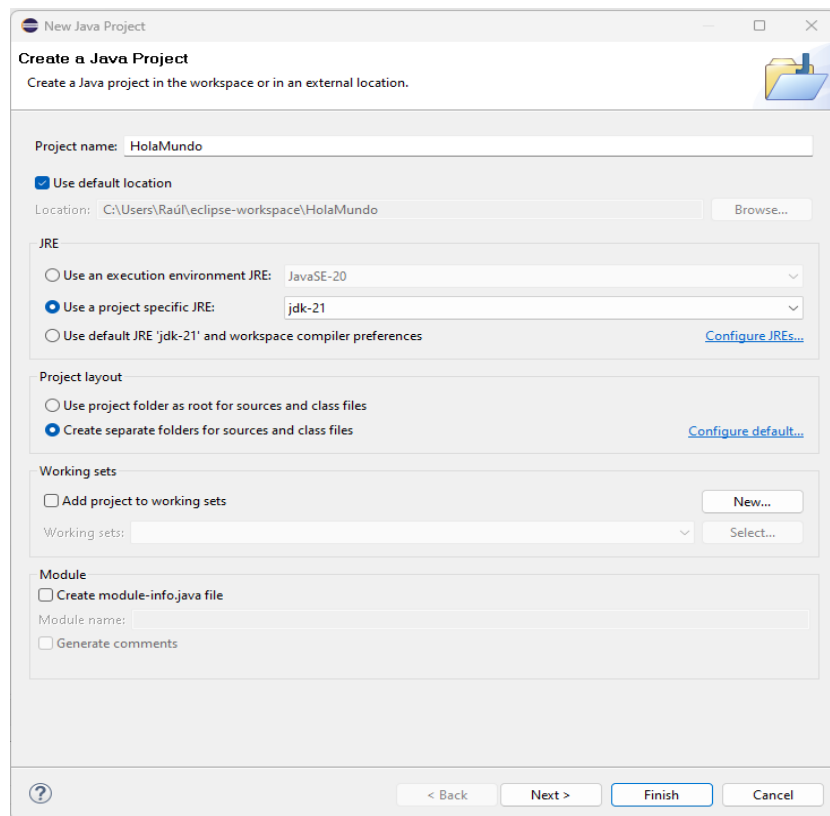
- Página oficial:
 - eclipse.org/downloads

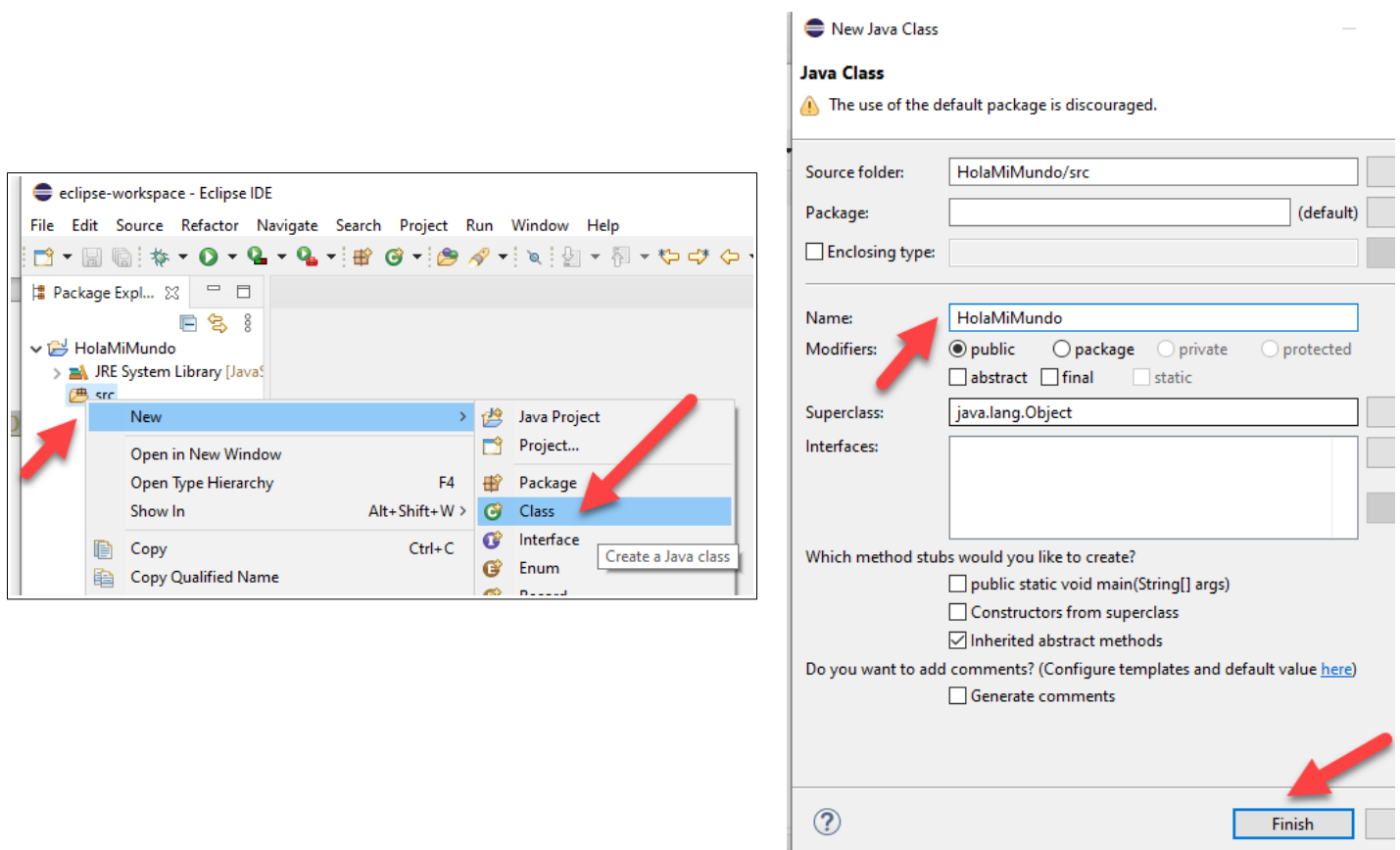
Pasos:

1. Seguiremos este sencillo tutorial:
 - solvetic.com/tutoriales/article/8578-como-instalar-eclipse-ide-windows-10
2. Al instalar te preguntará qué JVM quieres que utilice para ejecutar las aplicaciones Java.



3. Luego, seguiremos los pasos de este tutorial:
 - desarrolloweb.com/articulos/hola-mundo-en-java-con-ide-eclipse.html
4. Que básicamente dice esto:

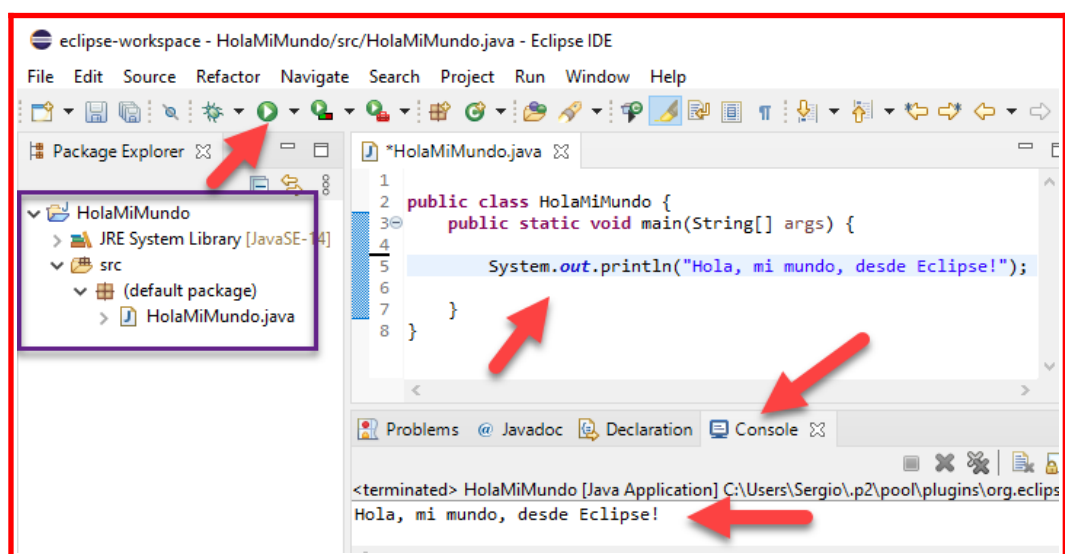




5. Ahora cambia mi mundo por tu nombre en esta frase e imprímela con Java en pantalla:

Hola, **mi mundo**, desde Eclipse!

6. Por último, solo tienes que darle al PLAY para que compile y muestre el resultado en la consola (la parte inferior donde dice "Console").



7. Para cambiarlo a español te **debería** funcionar este otro tutorial:

- programaenlinea.net/cambiar-idioma-eclipse-espanol-primera-parte

3. DEPURACIÓN EN ECLIPSE

3.1 EN QUÉ CONSISTE

La depuración (debug) permite examinar las aplicaciones para buscar errores, ya que posibilita observar las líneas que se van ejecutando, así como los valores que van tomando las variables en cada paso.

Para realizar la depuración de un programa, se debe establecer en primer lugar un punto de interrupción donde debe pararse la ejecución de la aplicación.

3.2 DEPURACIÓN EN ECLIPSE

Esto se consigue de muchas maneras. La más sencilla es cambiar de perspectiva (de Java a Debug).

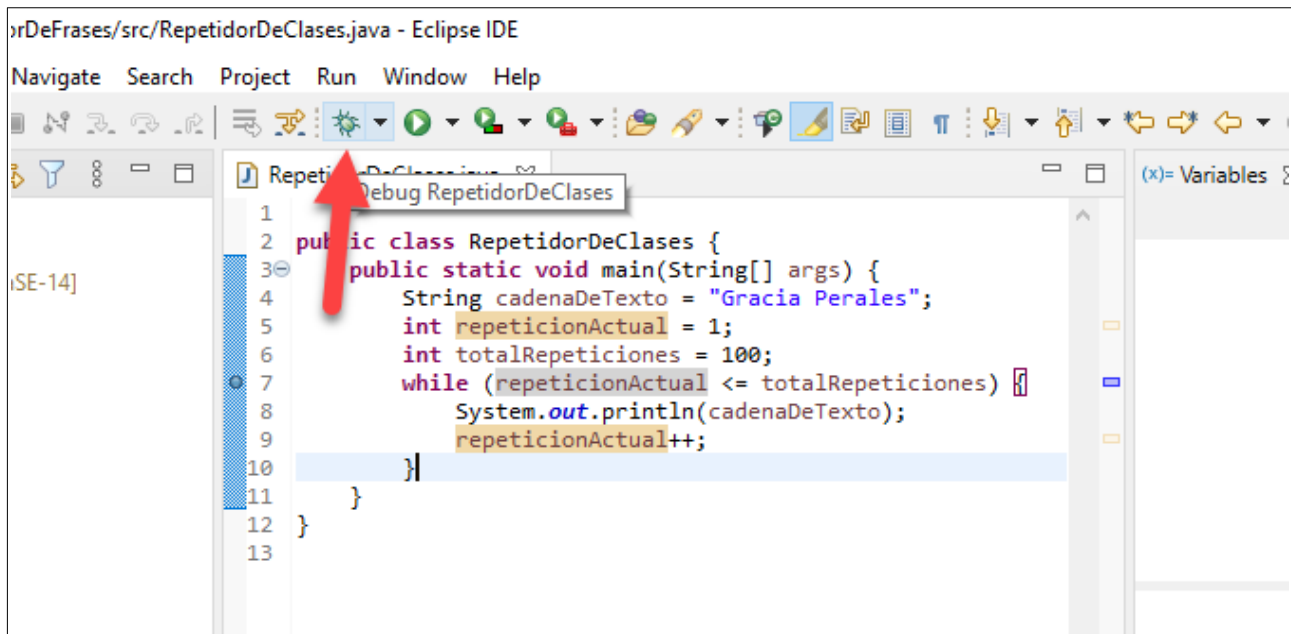
Seguiremos este otro tutorial:

- edu4java.com/es/java/depurar-debug-programa-java-eclipse.html

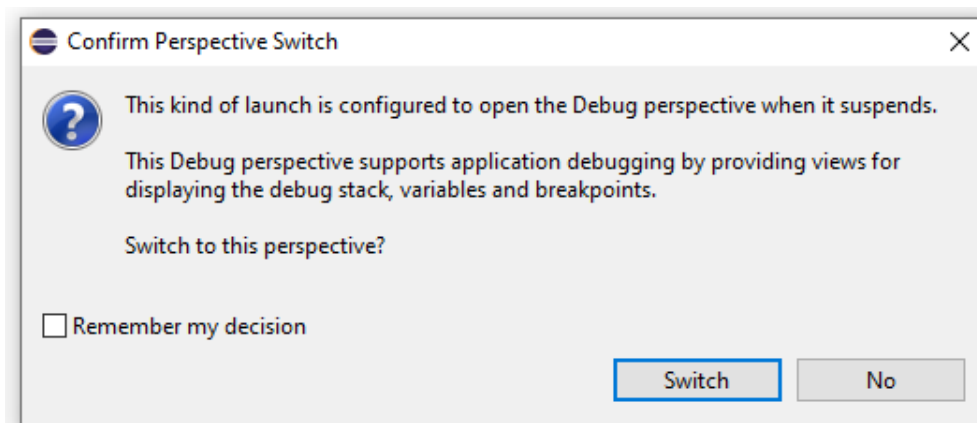
1. Creamos un nuevo proyecto para depurar el siguiente código (el mismo que vimos con NetBeans):

```
public class RepetidorDeClases {  
    public static void main(String[] args) {  
        String cadenaDeTexto = "Gracia Perales";  
        int repeticionActual = 1;  
        int totalRepeticiones = 100;  
        while (repeticionActual <= totalRepeticiones) {  
            System.out.println(cadenaDeTexto);  
            repeticionActual++;  
        }  
    }  
}
```

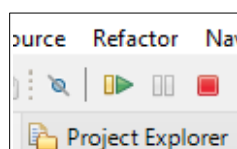
2. Cambiamos a perspectiva de depuración pinchando en el “virus verde”:



3. Si nos sale este mensaje le damos a aceptar:



4. Pulsamos el botón RESUME o pulsamos F8...

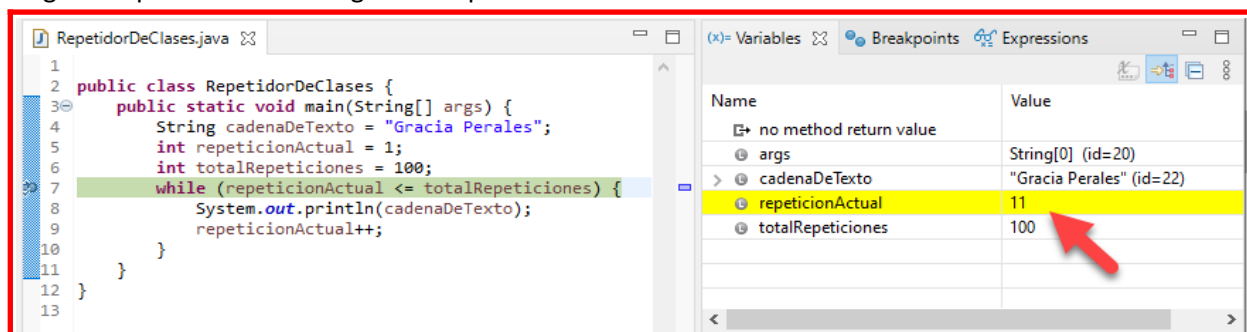


A partir del momento en que se para la ejecución del programa se puede continuar con la ejecución línea a línea utilizando la opción "**Continuar ejecución**":

- Pulsando la tecla F8

Así se va mostrando en verde la línea que se va ejecutando en cada momento, y se van **actualizando** en la ventana lateral los valores que van tomando las variables.

Seguimos pulsando hasta llegar a la repetición número 11.



4. BIBLIOGRAFÍA

- i. Iglesias, C. C. (2020). Entornos de Desarrollo (GRADO SUPERIOR). RA-MA S.A. Editorial y Publicaciones.
- ii. Aldarias, F. (2012): Apuntes de Entornos de Desarrollo, CEEDCV