

UD 2.3

INSTALACIÓN Y USO DE ENTORNOS IDE

ENTORNOS DE DESARROLLO 20/21
CFGS DAW

PARTE 3 DE 3: INSTALACIÓN Y USO DE ECLIPSE (IDE)

Autor: Sergio Badal

sergio.badal@ceedcv.es

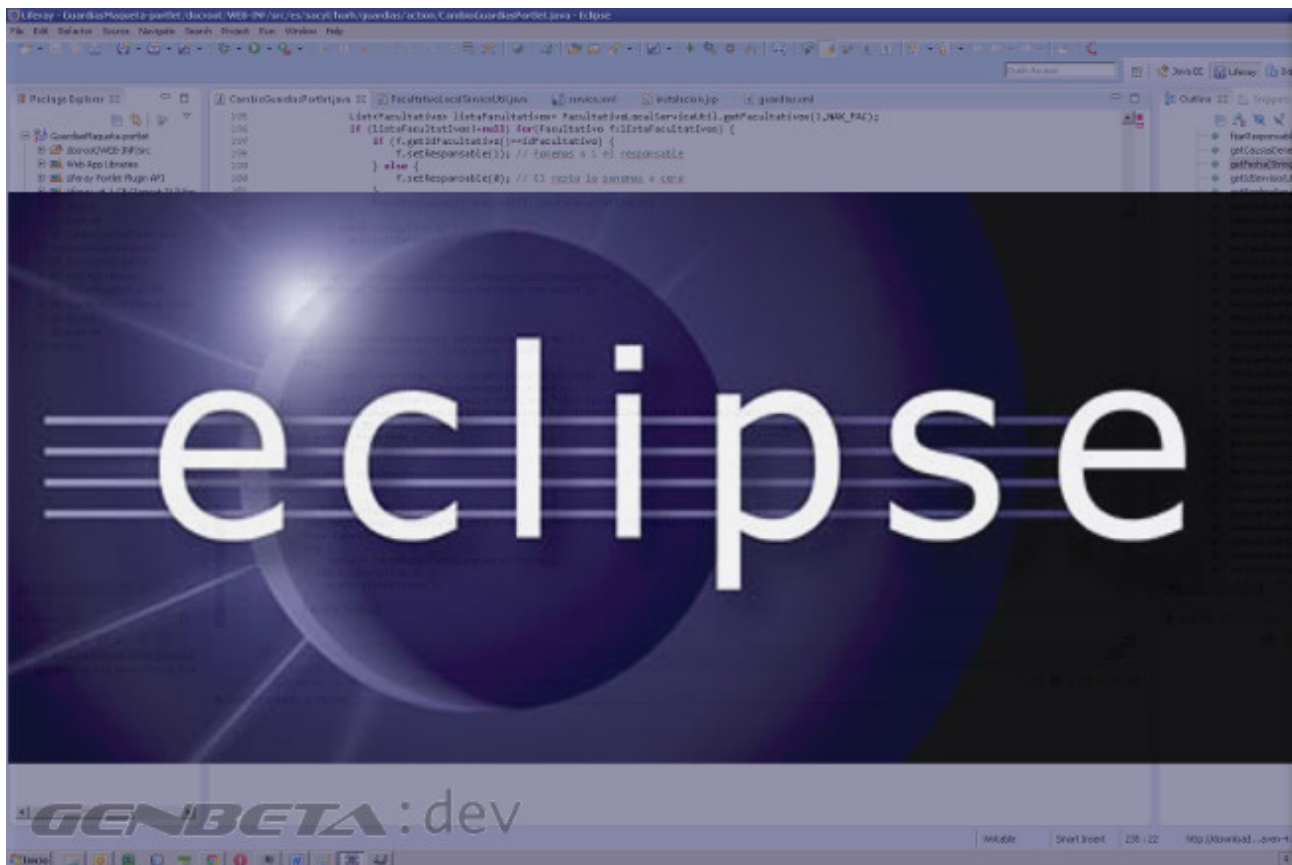
Fecha: 22-10-2020

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
1.4 DIFERENCIAS CON NETBEANS.....	3
2. INSTALACIÓN DE ECLIPSE.....	4
2.1 INSTALACIÓN PREVIA DE JAVA.....	4
2.2 INSTALACIÓN Y PRUEBA DE ECLIPSE.....	5
2.3 ALTERNATIVA AL IDE.....	8
3. Depuración EN ECLIPSE.....	9
3.1 EN QUÉ CONSISTE.....	9
3.2 DEPURACIÓN EN ECLIPSE.....	9
4. BIBLIOGRAFÍA.....	12

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**.

RETO



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 fútbol? ¿Sabrías decirnos cuánto pagó la Fundación Eclipse por ese dominio?

*Si lo averiguas, cuéntanoslo en el el **FORO DE LA UNIDAD**...*

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!

RETO



Hablamos mucho de FUNDACIONES DE SOFTWARE en esta Unidad. ¿Sabrías decirnos diferencias y similitudes entre la Fundación Apache y la Fundación Eclipse? ¿Comparten miembros y filosofía?

*Si lo averiguas, cuéntanoslo en el **FORO DE LA UNIDAD**...*

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

1.4 DIFERENCIAS CON NETBEANS

Información de <https://www.compostela21.com/?goid=77>

Netbeans, comenzó como proyecto de estudiantes en 1996 en Praga y su brazo comercial fue adquirido por Sun Microsystems en 1999. Posteriormente, Sun lo adquirió de forma abierta y luego, cuando Oracle adquirió Sun, Netbeans se entregó a la Fundación Apache.

En **Eclipse**, la plataforma de software se construyó sobre la inspiración de la familia de productos Visual Age de IBM y, aunque visual age es un producto exitoso, no tenía la estructura modular como Netbeans y todo estaba empaquetado como un archivo zip. Más tarde, IBM lo cedió a la Fundación Eclipse.

Ambos IDE son de código abierto y proporcionan un entorno de desarrollo de software razonablemente bueno. Tienen sus propios pros y contras y la selección de las herramientas adecuadas depende del propósito pero la diferencia más significativa es esta que **para un proyecto pequeño, para principiantes, Netbeans es la elección correcta y para un proyecto grande, Eclipse es la elección.**

NetBeans	Eclipse
Fácil de usar, especialmente para principiantes con una interfaz más simple y pueden funcionar sin ningún complemento inicialmente.	Los usuarios se habrán acostumbrado al plugin antes de sumergirse en él.
El rendimiento es estable porque no es necesario instalar complementos.	Más complementos tiene un impacto en el rendimiento.
Más adecuado para el desarrollo web debido a su soporte para HTML5 y PHP.	Más adecuado para manejar grandes proyectos.
Necesita extensión para apoyar el modelado.	Inherentemente admite el modelado.
Requiere menos memoria y el rendimiento es estable.	La falta de memoria ralentiza el rendimiento.
Utiliza el compilador de Java.	Utiliza un compilador fuera de Eclipse y, por lo tanto, la compilación incremental es más rápida.
Licencia bajo licencia pública general y se puede instalar en cualquier máquina que ejecute una máquina virtual Java.	Las licencias se publican bajo el proyecto de licencia pública Eclipse.
Hay complementos limitados.	Hay disponibles variedades de complementos a gran escala para ayudar a los desarrolladores a personalizar el software.
Es fácil de aprender.	Una gran comunidad con abundante conocimiento proporciona un respaldo de respaldo adecuado.
Tiene más componentes modulares para facilitar el desarrollo.	Todos los objetos se vierten en el paquete.

RETO



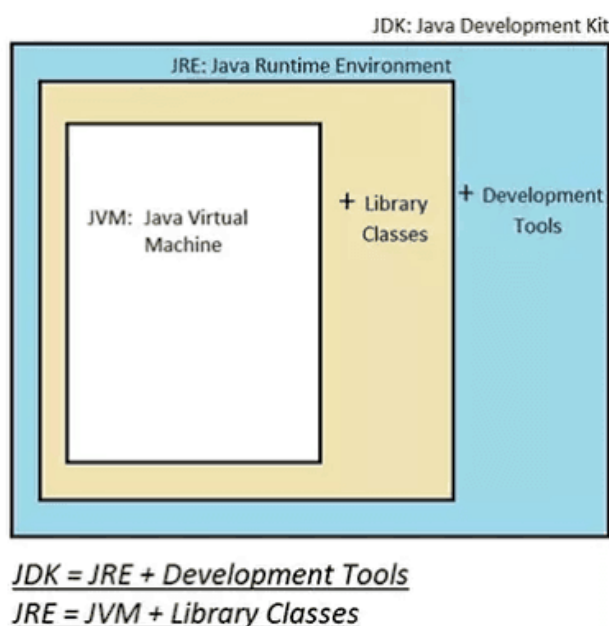
Hablamos mucho de FUNDACIONES DE SOFTWARE en esta Unidad. ¿Sabrías decirnos diferencias y similitudes entre la Fundación Apache y la Fundación Eclipse? ¿Comparten miembros y filosofía?

*Si lo averiguas, cuéntanoslo en el **FORO DE LA UNIDAD**...*

2. INSTALACIÓN DE ECLIPSE

2.1 INSTALACIÓN PREVIA DE JAVA

Antes de instalar este IDE deberás instalar el kit de desarrollo de software (SDK) de JAVA (también llamado JDK) en tu equipo. Sin este “paquete”, no podrás crear aplicaciones en Java.



Importante

Es MUY importante que conozcas las siglas JVM, JRE y JDK. Observa con atención el gráfico anterior para entender qué contiene qué.

Para instalar el JDK en tu equipo, puedes consultar el tutorial que tienes en el Aula Virtual “Cómo instalar JAVA en Windows” o seguir cualquiera de estos enlaces:

- Descarga directa de JAVA JDK SE 8:
 - javadl.oracle.com/webapps/download/AutoDL?BundleId=242990_a4634525489241b9a9e1aa73d9e118e6
- Página oficial:
 - oracle.com/java/technologies/javase/javase-jdk8-downloads.html
- Sencillo tutorial (Windows):
 - youtube.com/watch?v=kPWezAZGPks

2.2 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.

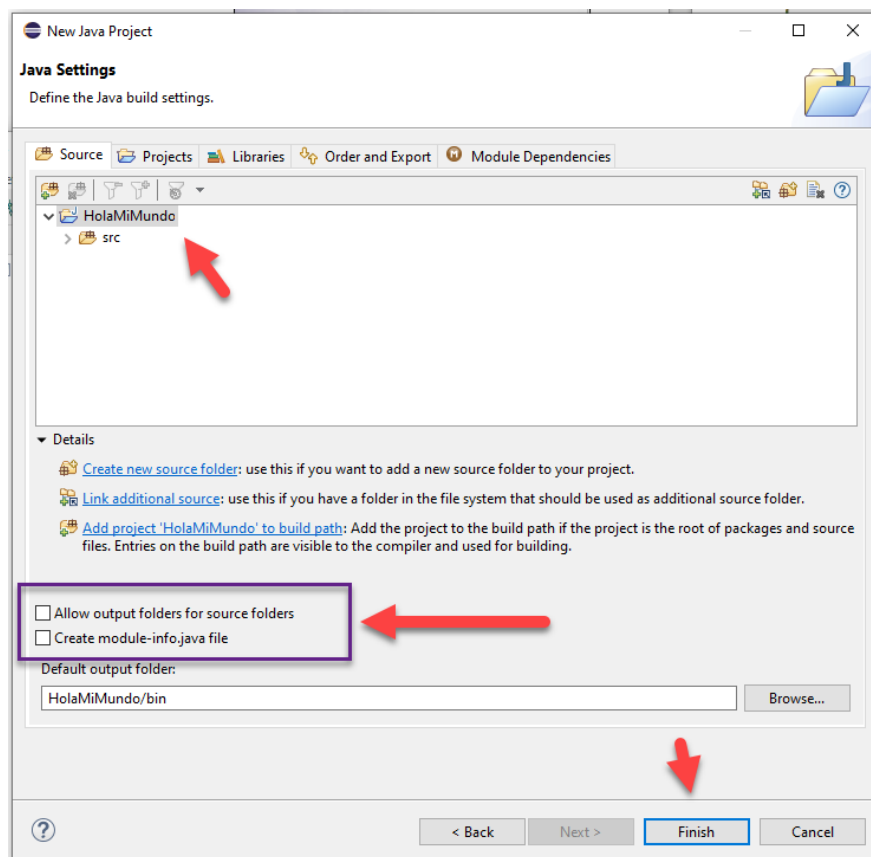
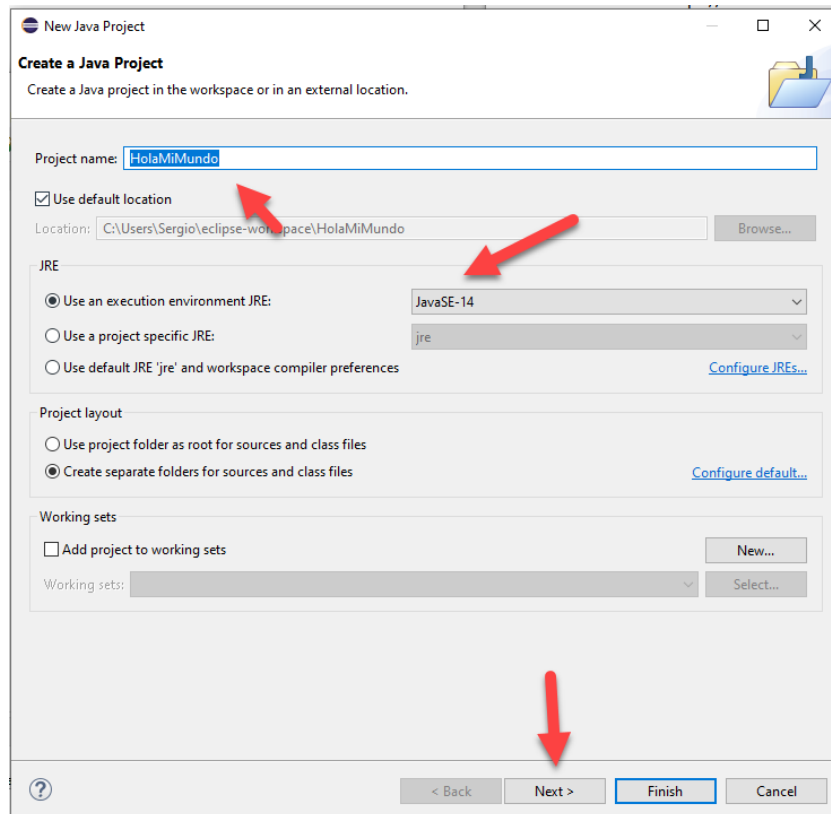
- Página oficial:
 - eclipse.org/downloads

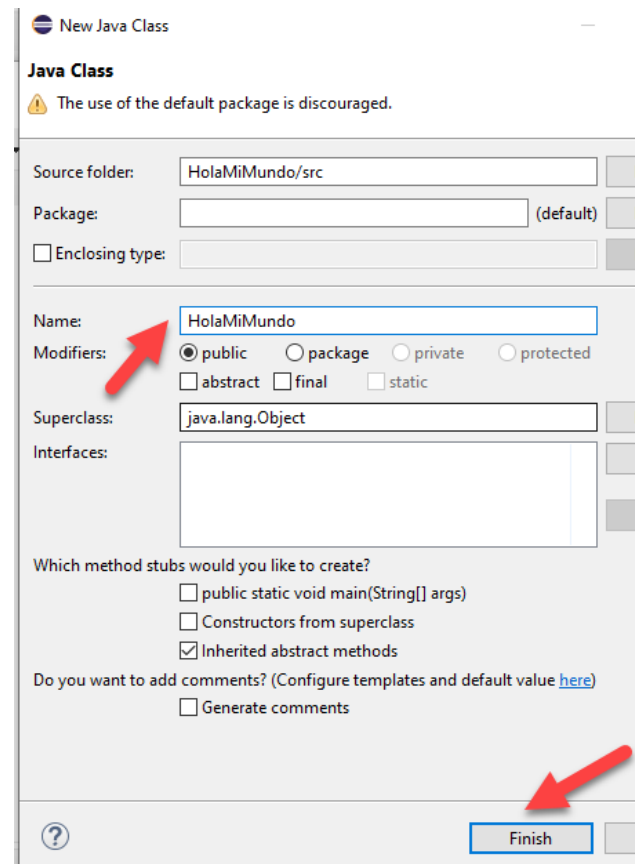
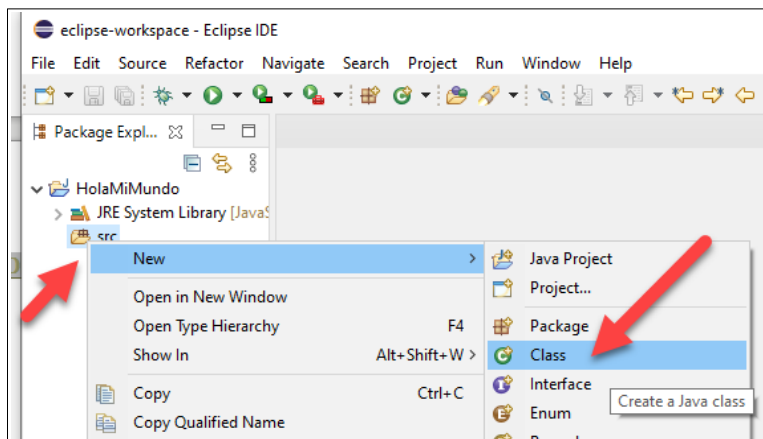
Pasos:

1. Seguiremos este sencillo tutorial (Windows 10):
 - 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 y, si estás instalando la última versión de Eclipse, deberás permitirle descargar una versión superior a la 11.



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

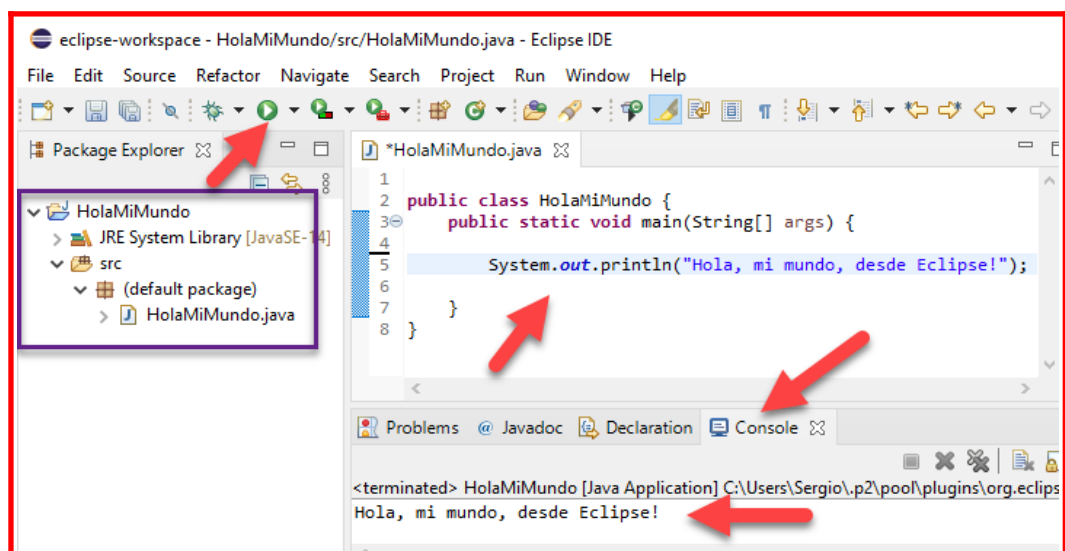




5. Ahora cambia mi mundo por tu nombre en esta frase e imprímela con Java en pantalla:

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

La captura que verás enmarcada en rojo es la que luego tendrás que entregar como parte de la práctica no evaluable para demostrar que has instalado y probado este IDE.

Guarda esa captura y pégala en el documento que entregarás con la práctica.

!!! Recuerda modificarlo para que aparezca TU NOMBRE Y LA MENCIÓN A ECLIPSE !!!

2.3 ALTERNATIVA AL IDE

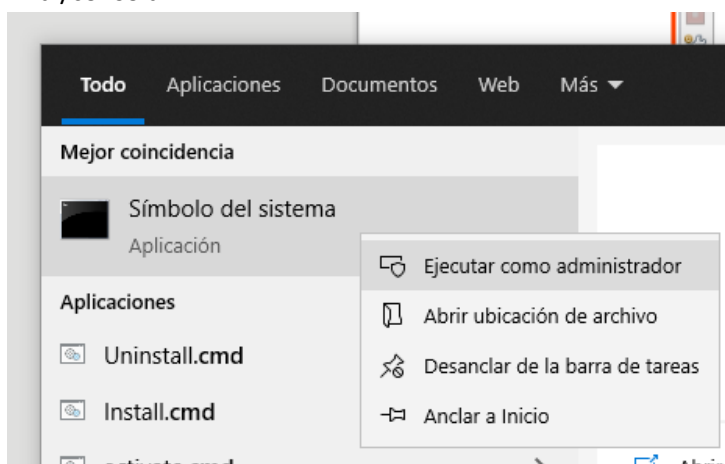
(Repite este proceso si ya lo hiciste con NetBeans pero con la nueva frase)

Recuerda que estás trabajando con un IDE, que no es otra cosa que un “entorno amigable de desarrollo” que intenta hacerte la vida más fácil pero que no es imprescindible para generar un programa.

Concretamente, podríamos afirmar que todo programa (incluso los que tienen miles de líneas de código) se pueden hacer teóricamente desde el bloc de notas de Windows.

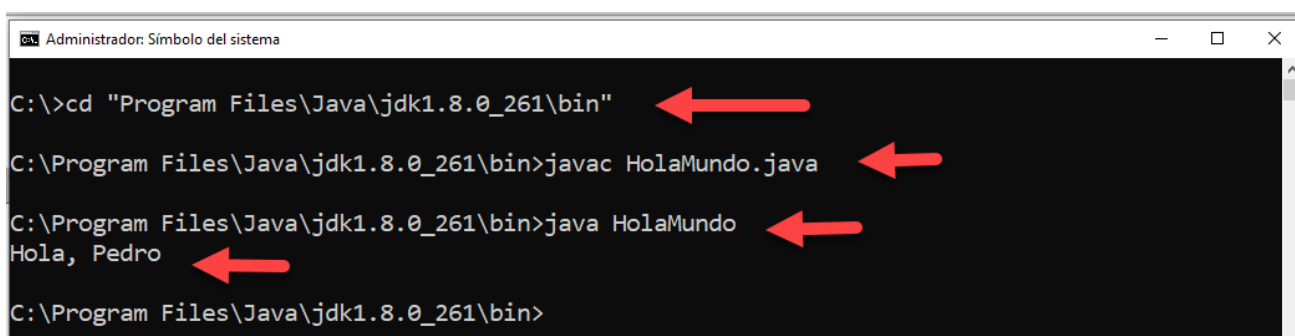
Para muestra, un botón:

1. Abre el bloc de notas y copia todo el código que tienes en NetBeans
2. Cambia la palabra “Pedro” por tu nombre de pila
3. Guarda ese archivo con el nombre **HolaMundo.java** en la carpeta donde tengas instalado JAVA JDK
 - Será algo como: C:\Program Files\Java\jdk-1.8.0_261\bin
 - **OJO, debes seleccionar otras extensiones para que no lo guarde como .txt**
4. Abre una terminal/console:



- En Windows 10 ve a Inicio > cmd (botón derecho “Ejecutar como Administrador”) > cmd > ENTER

- En otras versiones tendrás que buscar el comando “ejecutar”, abrirlo como administrador, teclear el comando cmd y darle a ENTER
- 5. Una vez en la terminal tendrás que entrar en la carpeta de java con la orden cd (change directory):
 - cd C:\Program Files\Java\jdk-1.8.0_261\bin
- 6. Una vez en la carpeta verás algo así, indicándote que estás “dentro de esa carpeta”:
- 7. Ahora ya puedes ejecutar el comando para compilar JAVA “**javac**” y el comando para ejecutarlo “**java**”. Ten presente que primero generamos el archivo y luego lo ejecutamos.



```
Administrador: Símbolo del sistema

C:\>cd "Program Files\Java\jdk1.8.0_261\bin"

C:\Program Files\Java\jdk1.8.0_261\bin>javac HolaMundo.java

C:\Program Files\Java\jdk1.8.0_261\bin>java HolaMundo
Hola, Pedro

C:\Program Files\Java\jdk1.8.0_261\bin>
```

Guarda esa captura y pégala en el documento que entregarás con la práctica.

!!! Recuerda modificarlo para que aparezca TU NOMBRE Y LA MENCIÓN A ECLIPSE !!!

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).

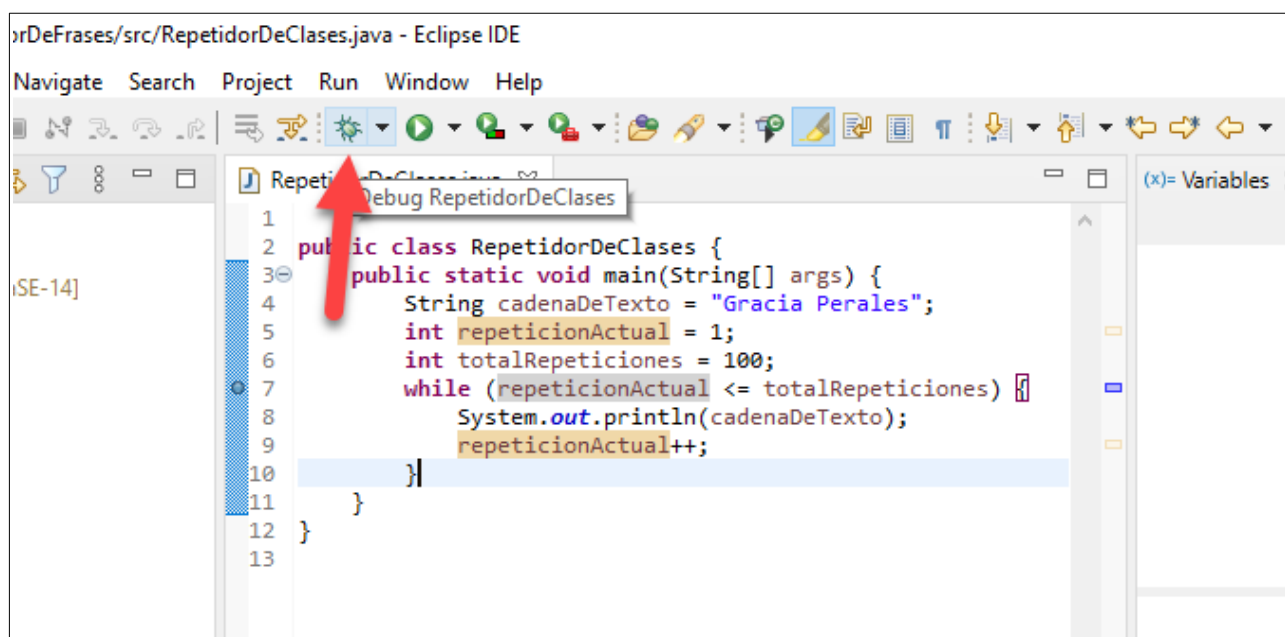
Seguiremos este otro tutorial:

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

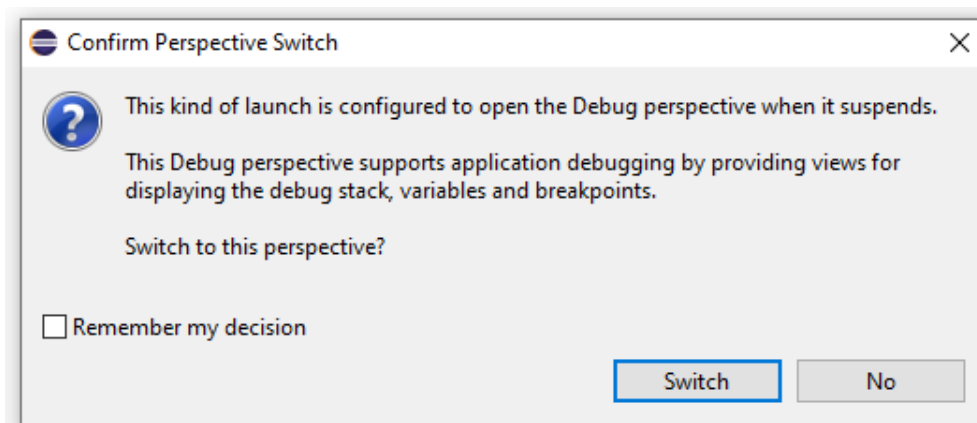
1. Creamos un nuevo proyecto para depurar el siguiente código (el mismo que vimos con NetBeans):

```
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++;
        }
    }
}
```

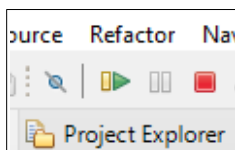
2. Cambiamos a perspectiva de depuración pinchando en el “virus verde”:



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



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

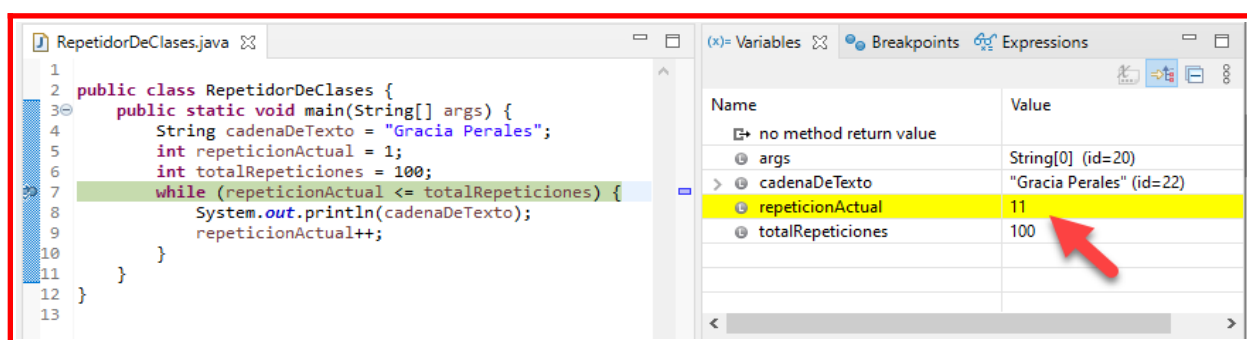


A partir del momento en que se para la ejecución del programa se puede continuar con la ejecución línea a línea utilizando la opción "**Continuar ejecución**":

- 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.



Guarda esa captura y pégala en el documento que entregarás con la práctica.

iii Recuerda modificarlo para que aparezca TUS DOS APELLIDOS !!!

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