



EBook Gratis

APRENDIZAJE eclipse

Free unaffiliated eBook created from
Stack Overflow contributors.

#eclipse

Tabla de contenido

| | |
|--|-----------|
| Acerca de..... | 1 |
| Capítulo 1: Empezando con eclipse..... | 2 |
| Observaciones..... | 2 |
| Versiones..... | 2 |
| Examples..... | 2 |
| Instalación y configuración..... | 3 |
| Instalar Marketplace en Eclipse..... | 4 |
| Útiles atajos de teclado..... | 5 |
| Administrar archivos y proyectos..... | 5 |
| Ventana del editor..... | 5 |
| Navegar en Editor..... | 5 |
| Editar texto..... | 5 |
| Buscar y reemplazar..... | 6 |
| Mueve un bloque de código..... | 6 |
| Creando y ejecutando un programa HelloWorld de Java..... | 6 |
| Crear un nuevo proyecto Java..... | 6 |
| Crear una nueva clase de Java..... | 8 |
| Ejecuta tu clase de Java..... | 10 |
| Importación de proyectos existentes..... | 10 |
| Capítulo 2: Atajos Eclipse..... | 13 |
| Introducción..... | 13 |
| Examples..... | 13 |
| Código de comentario / Uncomment..... | 13 |
| Abrir cuadro de diálogo..... | 13 |
| Para obtener una impresion..... | 13 |
| Generar Getters y Setters..... | 13 |
| Refactor texto resaltado..... | 13 |
| Formato xml..... | 13 |
| Capítulo 3: Cómo funciona la depuración remota de Eclipse detrás de escena..... | 15 |
| Examples..... | 15 |

| | |
|--|-----------|
| ¿Cómo funciona la depuración remota de Eclipse detrás de las escenas?..... | 15 |
| Capítulo 4: Configurando Eclipse..... | 17 |
| Examples..... | 17 |
| Aumento de la memoria máxima del montón para Eclipse..... | 17 |
| Especificando la JVM..... | 17 |
| Cómo configurar el tamaño de fuente de las vistas en Eclipse en Linux..... | 18 |
| Capítulo 5: Configurando Eclipse para C ++..... | 21 |
| Examples..... | 21 |
| Linux + CMake (generador "Unix Makefiles") + Qt (opcional)..... | 21 |
| Qt (opcional)..... | 21 |
| Espacio de trabajo..... | 21 |
| Adjuntar fuentes al proyecto..... | 21 |
| Generador CMake..... | 21 |
| Construir..... | 22 |
| Volver a ejecutar CMake (para volver a generar los makefiles)..... | 22 |
| Capítulo 6: Crear un nuevo espacio de trabajo en Eclipse..... | 23 |
| Examples..... | 23 |
| Cómo crear un espacio de trabajo..... | 23 |
| Capítulo 7: Depuración de programas Java en Eclipse..... | 24 |
| Examples..... | 24 |
| Evaluar expresiones dentro de una sesión de depuración..... | 24 |
| Depuración remota de una aplicación Java..... | 26 |
| Capítulo 8: Depuración remota en Eclipse..... | 28 |
| Examples..... | 28 |
| Configurar la depuración remota de Eclipse para una aplicación..... | 28 |
| Capítulo 9: Procedimiento de despliegue de Tomcat..... | 30 |
| Examples..... | 30 |
| Procedimiento cuando nada más ayuda..... | 30 |
| Creditos..... | 32 |

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [eclipse](#)

It is an unofficial and free eclipse ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official eclipse.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con eclipse

Observaciones

Esta sección proporciona una descripción general de qué es eclipse y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de eclipse, y vincular a los temas relacionados. Dado que la Documentación para eclipse es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

Versiones

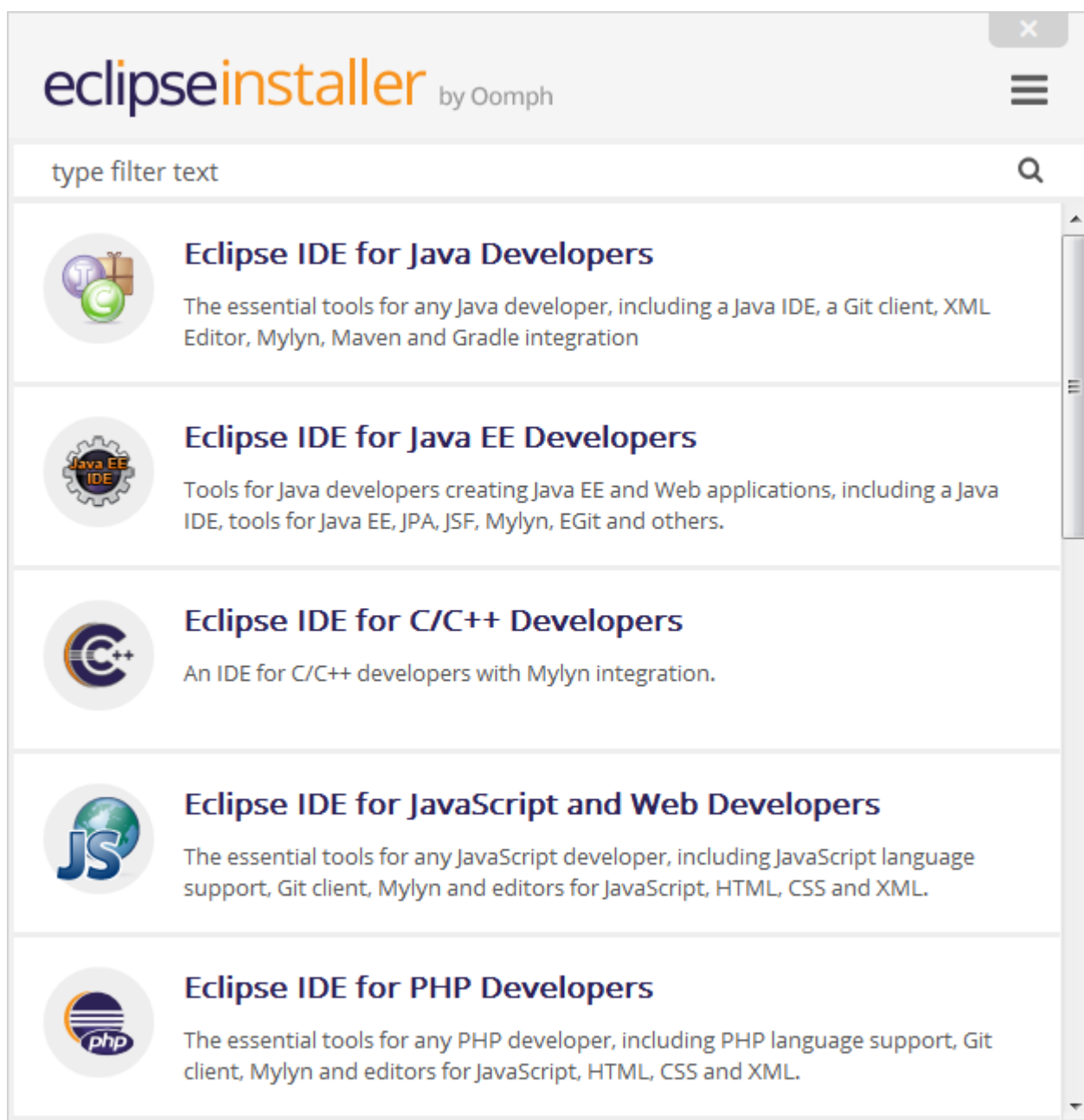
| Versión | Nombre | Fecha de lanzamiento |
|-----------|------------------|----------------------|
| 3.0 | | 2004-06-21 |
| 3.1 | | 2005-06-28 |
| 3.2 | Calisto | 2006-06-30 |
| 3.3 | Europa | 2007-06-29 |
| 3.4 | Ganimedes | 2008-06-25 |
| 3.5 | Galileo | 2009-06-24 |
| 3.6 | Helios | 2010-06-23 |
| 3.7 | Índigo | 2011-06-22 |
| 3.8 y 4.2 | Juno | 2012-06-27 |
| 4.3 | Kepler | 2013-06-26 |
| 4.4 | Luna | 2014-06-25 |
| 4.5 | Marte | 2015-06-24 |
| 4.6 | Neón | 2016-06-22 |
| 4.7 | Oxígeno | 2017-06-28 |
| 4.8 | Fotón (planeado) | 2018-06-01 |

Examples

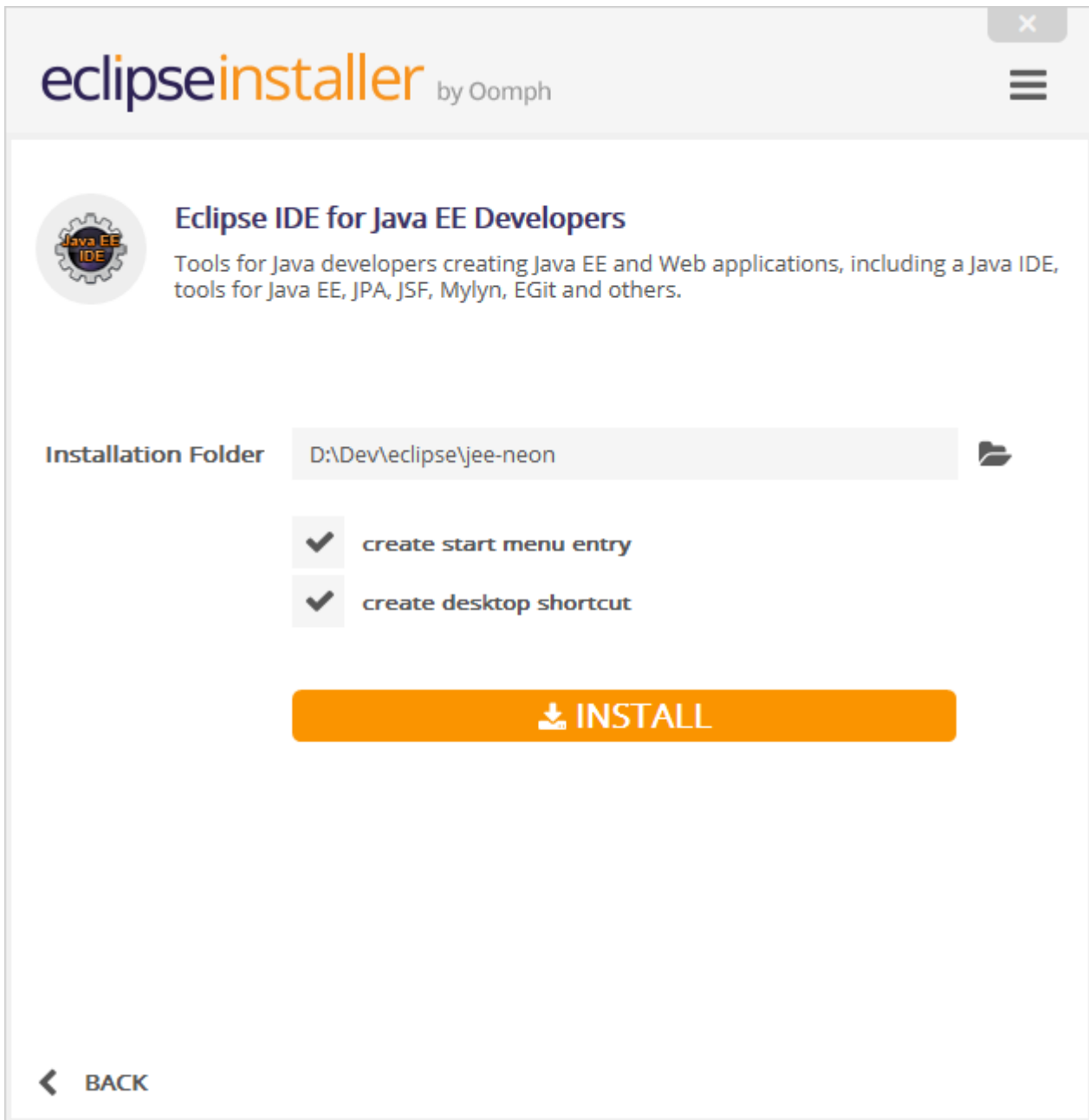
Instalación y configuración

Para instalar Eclipse, vaya a la página web de [descargas de Eclipse](#) , donde generalmente hay un enlace directo para descargar la última versión de Eclipse. Al iniciar Eclipse Mars (versión 4.5), se puede descargar un instalador que lo guía a través del procedimiento de instalación, en lugar de descargar la instalación completa como un archivo comprimido (esta opción aún está disponible). También hay enlaces para descargar paquetes antiguos de Eclipse.

Eclipse viene en varios paquetes diferentes que apuntan a diferentes usuarios como se muestra en la siguiente captura de pantalla del instalador. Por ejemplo, el **IDE de Eclipse para desarrolladores de Java** contiene herramientas básicas que admiten el desarrollo, depuración y creación de aplicaciones Java, así como un soporte básico para el control de versiones, como un complemento que permite la creación de versiones de proyectos utilizando Git, mientras que **Eclipse para desarrolladores de Android** proporciona un entorno para crear aplicaciones de Android.



Una vez que se selecciona un paquete, la página siguiente le permite al usuario seleccionar el directorio de instalación, junto con otras opciones. La siguiente captura de pantalla ilustra el procedimiento en una máquina con Windows.



Al hacer clic en el botón Instalar comenzará la instalación del paquete Eclipse en ese directorio.

Si la máquina ya tiene Java correctamente instalado, Eclipse debería iniciarse bien y generalmente no se requiere configuración o configuración. Sin embargo, es una buena práctica cambiar algunas opciones de configuración para Eclipse, por ejemplo, especificar en qué JVM debe ejecutarse Eclipse, y configurar la memoria mínima y máxima para esa JVM. Para hacerlo, existe un archivo llamado `eclipse.ini` en el directorio de instalación, donde se encuentra esta configuración de inicio. [Esta página](#) contiene detalles sobre cómo configurar Eclipse usando ese archivo.

Instalar Marketplace en Eclipse

Algunas de las versiones clásicas de eclipse no vienen preinstaladas con marketplace, esto puede ser instalado usando los siguientes pasos:

1. Ir a Ayuda → Instalar nuevo software
2. Añadir nuevo repositorio (sitio especificado a continuación)
3. Herramientas de uso general → Cliente de Marketplace
4. Haga clic en Finalizar y ya está.

Sitios de actualización del mercado:

```
Oxygen - http://download.eclipse.org/releases/oxygen/  
Neon - http://download.eclipse.org/releases/neon/  
Mars - http://download.eclipse.org/releases/mars/  
Luna - http://download.eclipse.org/mpc/luna  
Helios - http://download.eclipse.org/releases/helios  
Juno - http://download.eclipse.org/releases/juno/
```

Útiles atajos de teclado

Administrar archivos y proyectos

- **Ctrl + Shift + R** : Abrir recurso (archivo, carpeta o proyecto)
- **Ctrl + Shift + S** : Guardar todos los archivos
- **Ctrl + W** : Cerrar el archivo actual
- **Ctrl + Shift + W** : Cerrar todos los archivos

Ventana del editor

- **F12** : Saltar a la ventana del editor
- **Ctrl + E** : Mostrar lista de editores abiertos. Usa las teclas de flecha y entra para cambiar
- **Ctrl + AvPág / Arriba** : Cambia al siguiente editor / cambia al editor anterior
- **Ctrl + M** : Maximizar o minimizar la ventana actual del editor

Navegar en Editor

- **Ctrl + L** : Ir a la línea
- **Ctrl + Q** : Saltar a la última ubicación editada
- **Ctrl + Shift + P** : con un corchete seleccionado: salta al corchete de apertura o cierre correspondiente
- **Ctrl + Flecha abajo / arriba** : Editor de desplazamiento sin cambiar la posición del cursor

Editar texto

- **Ctrl + D** : Eliminar línea
- **Alt + Shift + Y** : Ajustar el texto seleccionado (ajustar el ancho del texto a la pantalla)
- **Alt + Shift + S** : opciones de menú Open Source
- **Alt + Shift + R** : Refactoriza la palabra resaltada en todos los archivos
- **Ctrl + Alt + Arriba / Ctrl + Alt + Abajo** : Copia las líneas seleccionadas arriba / abajo.

Buscar y reemplazar

- **Ctrl + J** : escriba un término para buscar y luego use Ctrl + J / Ctrl + shift + J para subir / retroceder
- **Ctrl + K / Ctrl + Shift + K** : Ctrl + H luego cerrar la ventana de búsqueda. A continuación, busque la aparición anterior / siguiente del término de búsqueda.

Mueve un bloque de código

- **Mayús + tabulador** : mover a la izquierda
- **Alt + Arriba / Alt + Abajo** : Mueve hacia arriba / abajo.

Creando y ejecutando un programa HelloWorld de Java

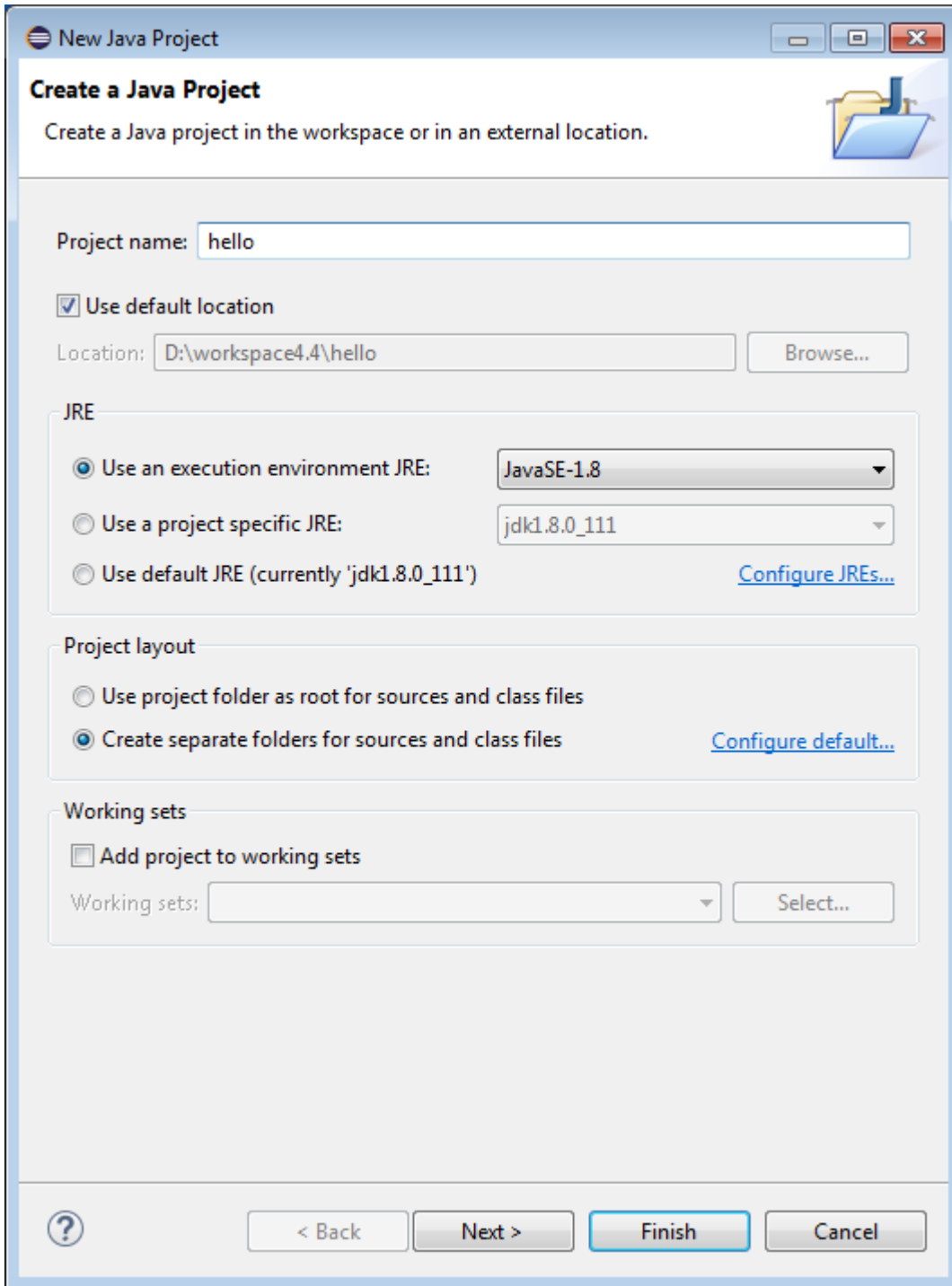
Desde la barra de herramientas abre la Perspectiva de Java.



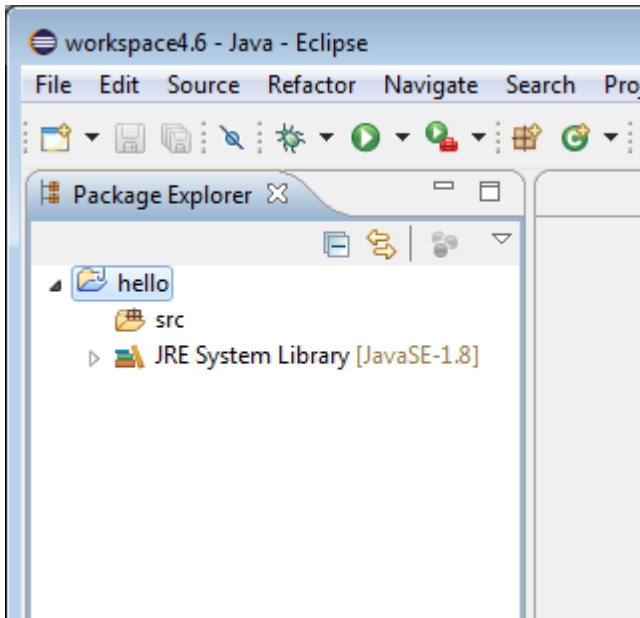
Crear un nuevo proyecto Java

Haga clic con el botón derecho en el Explorador de paquetes y, en el menú, seleccione `New -> Java Project`

En el siguiente cuadro de diálogo, ingrese el nombre del proyecto, luego haga clic en `Finish` .



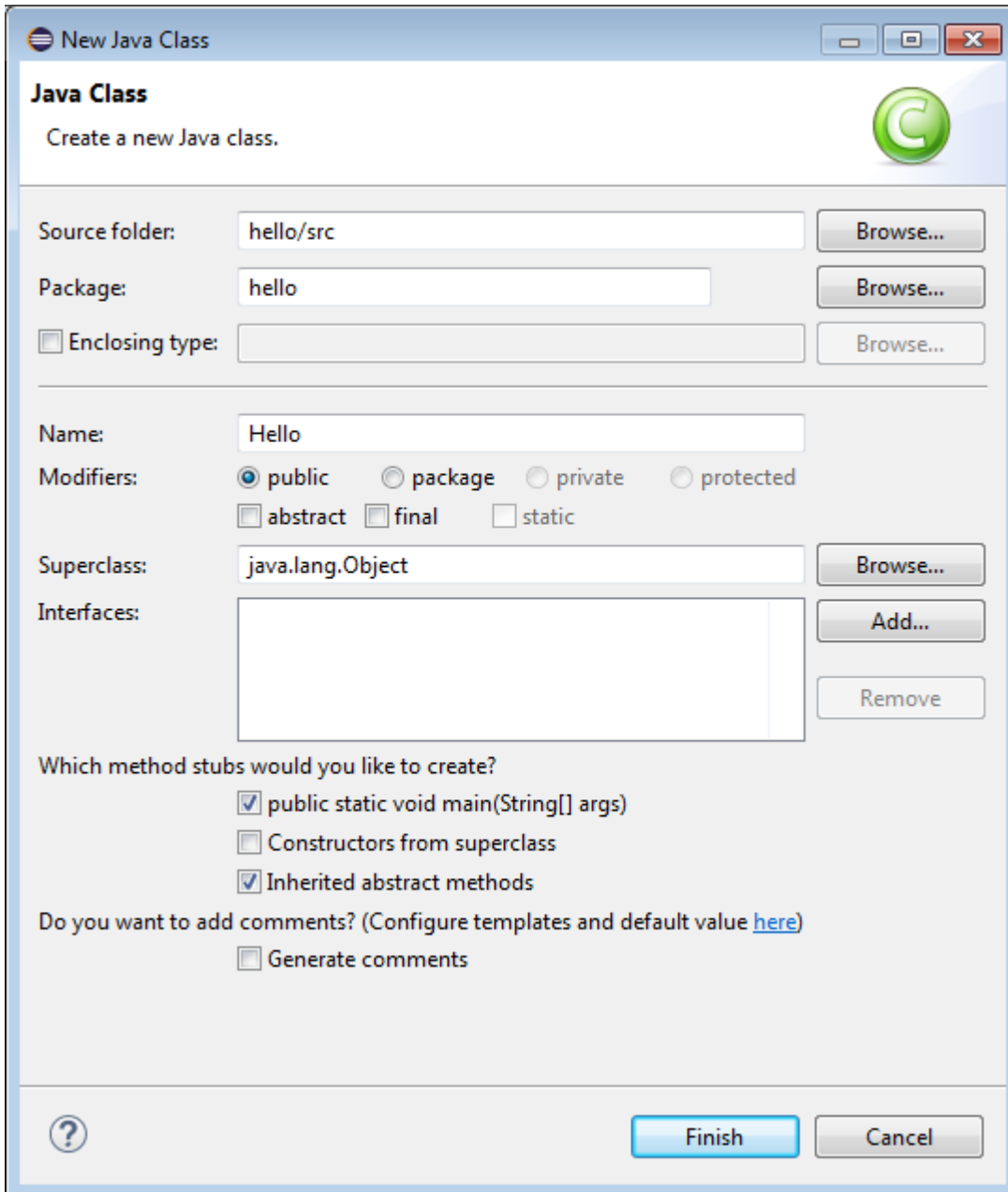
Ahora tienes el nuevo proyecto en tu espacio de trabajo.



Crear una nueva clase de Java

Haga clic con el botón derecho en su proyecto y, en el menú, seleccione `New -> Class` .

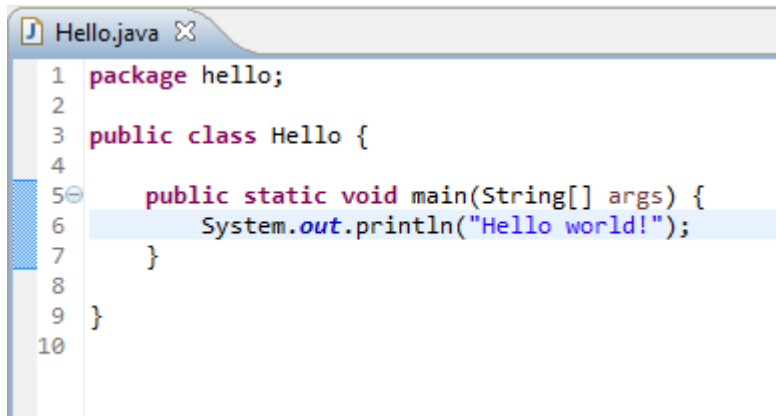
En el cuadro de diálogo en el nombre de la clase (debe comenzar con una letra mayúscula), seleccione también la casilla de verificación `public static void main(String[] args)` , luego haga clic en `Finish` .



Ahora tienes el primer archivo Java en tu proyecto. El editor abrirá automáticamente este nuevo archivo.

```
1 package hello;
2
3 public class Hello {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10 }
11
```

Dentro del método `main` , escriba algún código para imprimir `Hello world!` .

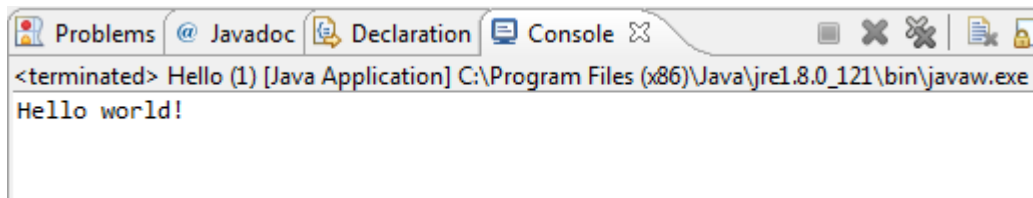


```
1 package hello;
2
3 public class Hello {
4
5     public static void main(String[] args) {
6         System.out.println("Hello world!");
7     }
8
9 }
10
```

Ejecuta tu clase de Java

Haga clic con el botón derecho en su clase de Java y, en el menú, seleccione `Run as -> Java application`.

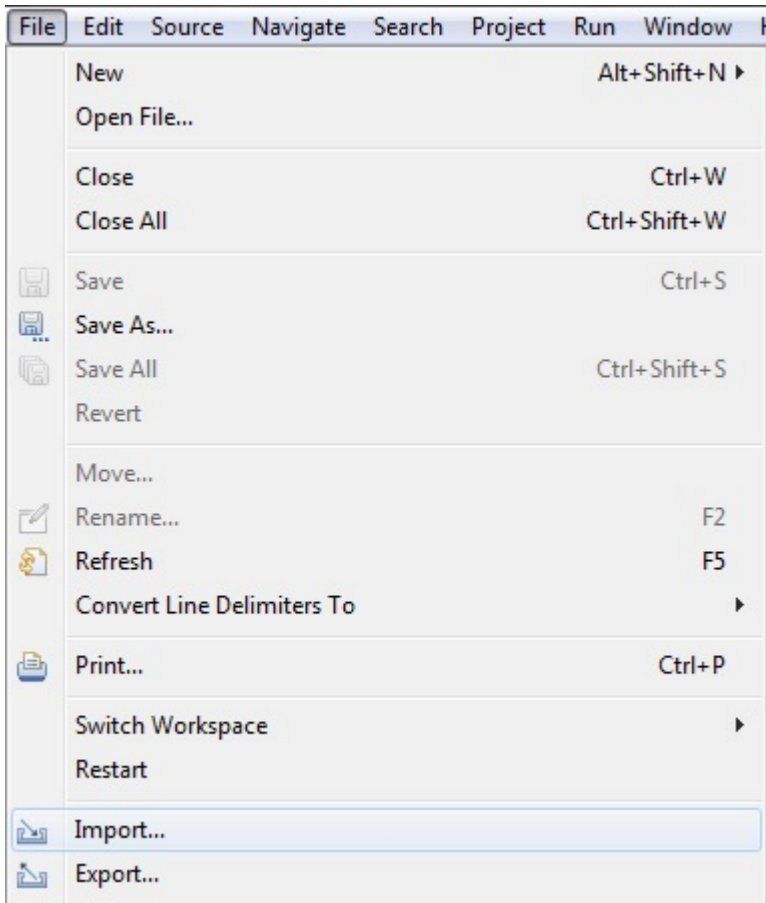
Voilà, verá la salida de su programa Java en la consola.



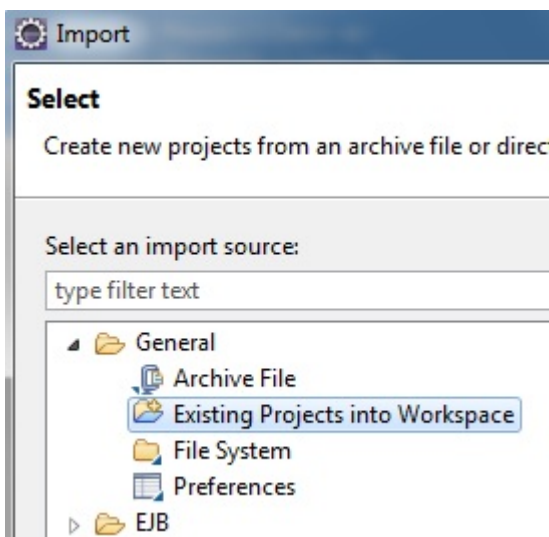
```
<terminated> Hello (1) [Java Application] C:\Program Files (x86)\Java\jre1.8.0_121\bin\javaw.exe
Hello world!
```

Importación de proyectos existentes

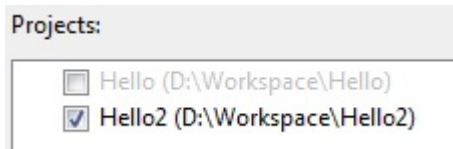
En el menú Archivo, elija la opción 'Importar ...'.



Esto abre el cuadro de diálogo Importar, que solicita el tipo de proyecto / archivo que desea importar. Para un proyecto Java básico, elija 'Proyectos existentes en el área de trabajo' en la carpeta 'General'.

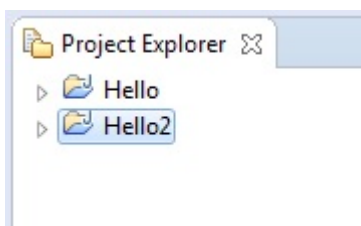


A continuación, seleccione el directorio donde se encuentra el proyecto (s) usando el botón 'Examinar'. Todos los proyectos que se pueden importar a Eclipse se mostrarán en la sección 'Proyectos:'. Si el proyecto ya se ha importado, aún se mostrará, pero la casilla de verificación se desactivará.



También puede importar proyectos directamente desde un archivo comprimido seleccionando 'Seleccionar archivo comprimido' y luego haciendo clic en el botón 'Examinar'.

Una vez que haga clic en 'Finalizar', el proyecto ahora estará visible en su Explorador de proyectos y listo para usar.



Lea Empezando con eclipse en línea: <https://riptutorial.com/es/eclipse/topic/1143/empezando-con-eclipse>

Capítulo 2: Atajos Eclipse

Introducción

Eclipse tiene muchos atajos para hacer tu vida más fácil.

Examples

Código de comentario / Uncomment

Para comentar o descomentar el código, seleccione las líneas y use **Ctrl + Shift + C** o **Ctrl + Shift + /**

Abrir cuadro de diálogo

Para acceder al cuadro de diálogo Abrir recurso, use **Ctrl + Shift + R**. Desde aquí puede comenzar a escribir un nombre de recurso y encontrará todas las coincidencias en el área de trabajo, lo que facilita la búsqueda de un archivo cuando no sabe exactamente dónde está.

Para obtener una impresion

Si quieres `System.out.println ();` pero no quiere escribir todo, solo puede escribir **syso** y presionar **Ctrl + Barra espaciadora** . Escribirá el resto y colocará el cursor entre paréntesis.

Generar Getters y Setters

Eclipse puede generar getters y setters básicos para ti. Haga clic derecho en su archivo de clase y vaya a **Fuente - Generar generadores y definidores** (ALT + MAYÚS + S). Esto abrirá un cuadro de diálogo en el que puede elegir para qué campos desea que se generen captadores y definidores.

Refactor texto resaltado

Cambiar el nombre de una variable o clase suele ser una tarea tediosa, buscando todas las ubicaciones donde se usa. Esto se puede acelerar significativamente al resaltar la palabra, presionar **Alt + Mayús + R** y luego escribir la nueva palabra. Eclipse cambiará automáticamente el nombre de la palabra en cada archivo donde se llame.

Formato xml

Cuando agrega entradas a un xml o copia de otras fuentes, a menudo hay pestañas y espacios desiguales alrededor de las entradas.

Cuando presionas **Ctrl + Shift + F** , alinea fácilmente todo el documento y también quitas las pestañas adicionales. Así, el texto se formatea y, finalmente, se vuelve legible.

Lea Atajos Eclipse en línea: <https://riptutorial.com/es/eclipse/topic/9387/atajos-eclipse>

Capítulo 3: Cómo funciona la depuración remota de Eclipse detrás de escena

Examples

¿Cómo funciona la depuración remota de Eclipse detrás de las escenas?

La depuración de Eclipse comienza con lo que se conoce como Agentes.

La **JVM**, que ejecuta las fuentes `.class` cumplidas, tiene una función que permite inyectar bibliotecas externas (escritas en Java o C++) en la JVM, casi en tiempo de ejecución. Estas bibliotecas externas se denominan Agentes y tienen la capacidad de modificar el contenido de los archivos `.class` que se han ejecutado. Estos agentes tienen acceso a la funcionalidad de la JVM a la que no se puede acceder desde un código Java normal que se ejecuta dentro de la JVM y se pueden usar para hacer cosas interesantes como inyectar y modificar el *código fuente en ejecución*, *la creación de perfiles*, etc. Herramientas como **JRebel** hace uso de Esta pieza de funcionalidad para lograr su magia.

Y para pasar un Agent Lib a una JVM, lo hace a través de los argumentos de inicio, utilizando el

```
agentlib:libname[=options] format.
```

En realidad, estábamos pasando un Agent Lib llamado `jdwp` a la JVM que ejecuta Tomcat. El `jdwp` es una implementación opcional específica de JVM del *JDWP (Java Debug Wire Protocol)* que se utiliza para definir la comunicación entre un depurador y un JVM en ejecución. Su implementación, si está presente, se proporciona como una biblioteca nativa de la JVM como

```
jdwp.so 0 jdwp.dll
```

Entonces ¿Qué es lo que hace?

En términos simples, el agente `jdwp` que pasamos básicamente cumple la función de ser un enlace entre la instancia de JVM que ejecuta una aplicación y un depurador (que puede ubicarse de forma remota o local). Dado que se trata de una biblioteca de agentes, tiene la capacidad de interceptar el código en ejecución, crear un puente entre la JVM y un depurador y aplicar la funcionalidad de un depurador en la JVM. Ya que en la arquitectura JVM, la funcionalidad de depuración no se encuentra dentro de la JVM en sí, sino que se extrae mediante herramientas externas (a las que se hace referencia adecuadamente como depuradores), estas herramientas pueden residir en la máquina local que ejecuta la JVM que se está depurando o ejecutar. Desde la maquina externa. Es esta arquitectura modular desacoplada la que nos permite tener una JVM ejecutándose en una máquina remota y utilizando el JDWP, tener un depurador remoto capaz de comunicarse con ella.

En resumen, así es como funciona el depurador de Eclipse.

Lea [Cómo funciona la depuración remota de Eclipse detrás de escena en línea](https://riptutorial.com/es/eclipse/topic/6247/como-funciona-la-depuracion-remota-de-eclipse-):

<https://riptutorial.com/es/eclipse/topic/6247/como-funciona-la-depuracion-remota-de-eclipse->

Capítulo 4: Configurando Eclipse

Examples

Aumento de la memoria máxima del montón para Eclipse

Para aumentar la cantidad máxima de memoria de pila utilizada en Eclipse, edite el archivo `eclipse.ini` ubicado en el directorio de instalación de Eclipse.

Este archivo especifica las opciones para el inicio de Eclipse, como qué JVM utilizar y las opciones para la JVM. Específicamente, necesita editar el valor de la opción `-Xmx` JVM (o crear uno si no existe).

A continuación se muestra una configuración de ejemplo que establece una memoria de almacenamiento dinámico máxima de 1 GB (1024 m). La línea relevante es `-Xmx1024m` - esto reemplazaría la línea `-Xmx*` existente en su configuración:

```
-startup
plugins/org.eclipse.equinox.launcher_1.3.200.v20160318-1642.jar
--launcher.library
C:/Users/user1/.p2/pool/plugins/org.eclipse.equinox.launcher.win32.win32.x86_64_1.1.400.v20160518-1444
-product
org.eclipse.epp.package.java.product
--launcher.defaultAction
openFile
-showsplash
org.eclipse.platform
--launcher.appendVmargs
-vmargs
-Xms256m
-Xmx1024m
```

Especificando la JVM

Un problema común que encuentran los usuarios de Eclipse está relacionado con la JVM predeterminada del sistema.

Una situación típica es un Windows de 64 bits que tiene instaladas versiones de Java de 32 y 64 bits, y un Eclipse de 32 bits. Si la versión de 64 bits de Java es la predeterminada del sistema, cuando se inicia Eclipse, se muestra un cuadro de diálogo de error.

Especificar la JVM explícitamente en `eclipse.ini` resolverá esto. La entrada `-vm` debe agregar directamente sobre la sección `-vmargs`.

El siguiente ejemplo muestra cómo usar una JVM de 32 bits en un Windows de 64 bits:

```
-startup
plugins/org.eclipse.equinox.launcher_1.3.200.v20160318-1642.jar
```

```
...
-vm
C:/Program Files (x86)/Java/jdk1.7.0_71/bin/javaw.exe
-vmargs
-Xms256m
-Xmx1024m
```

Cómo configurar el tamaño de fuente de las vistas en Eclipse en Linux

Eclipse no le da la posibilidad de cambiar el tamaño de fuente de las vistas como 'Project Explorer' o 'Servers', que se ve feo en Linux ya que Eclipse usa el tamaño de fuente predeterminado (de escritorio). Pero puede editar archivos de configuración específicos para obtener los tamaños de fuente adecuados.

Para arreglar este molesto tamaño de fuente, vaya a

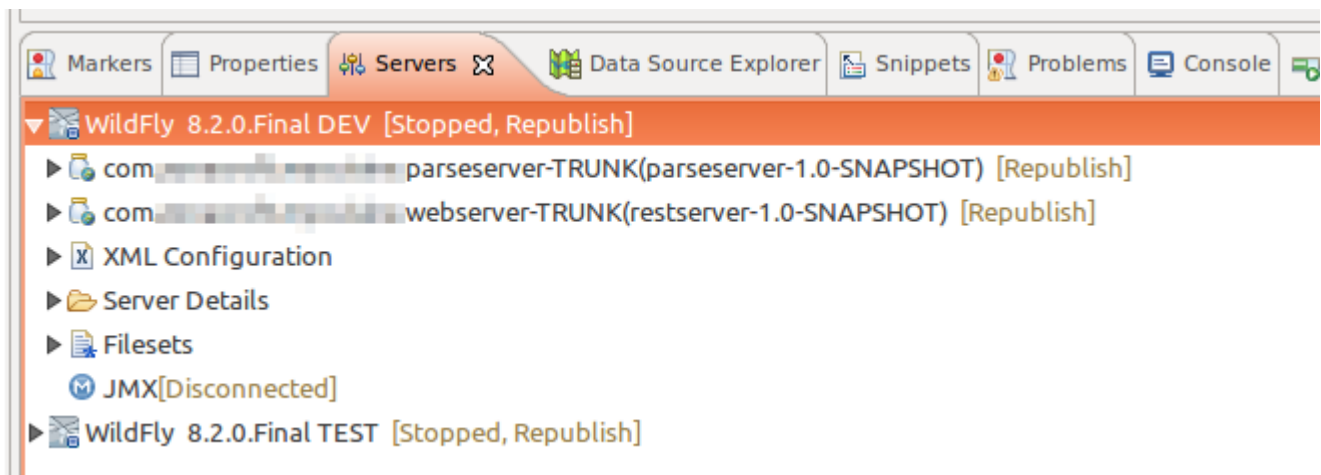
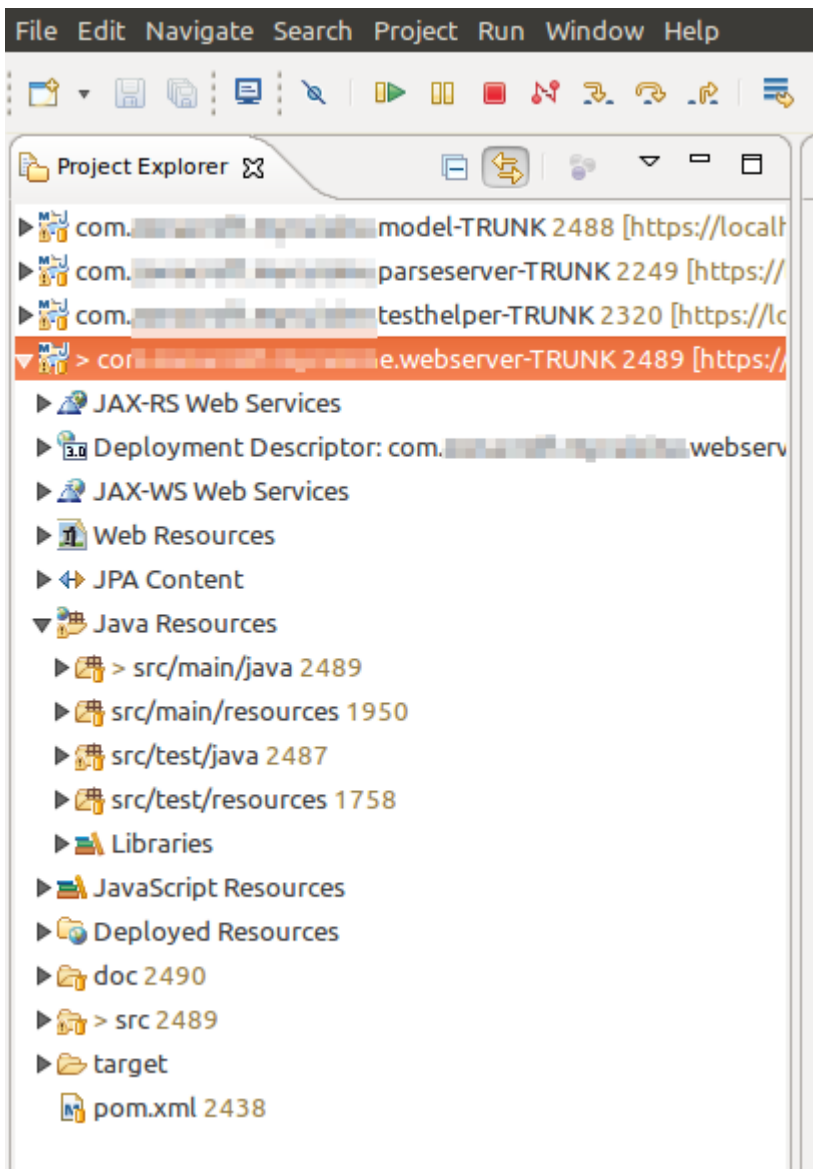
`/[YOUR_INST_DIR]/eclipse/plugins/org.eclipse.ui.themes_[LATEST_INSTALLATION]/css` y agrega este contenido ...

```
.MPart Tree{
  font-family: Sans;
  font-size: 8px;
}
```

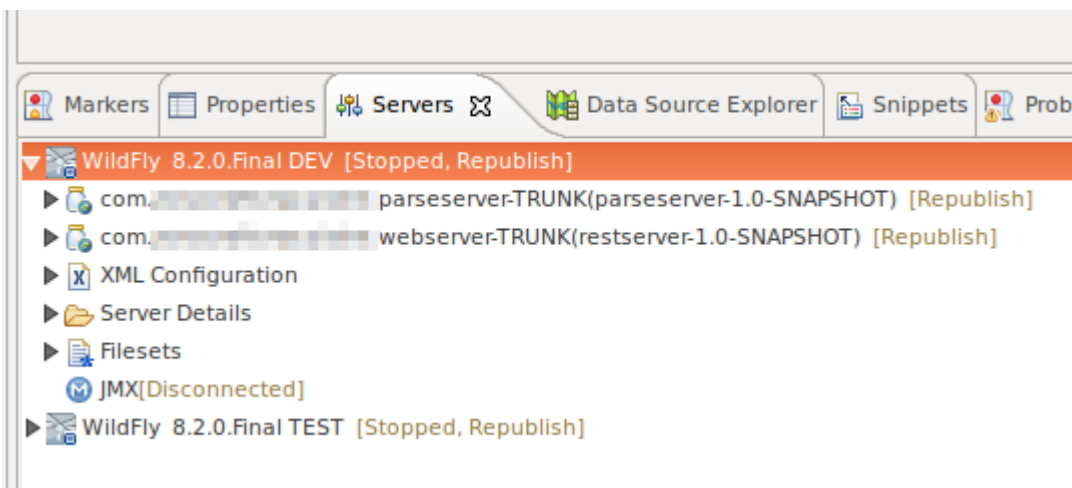
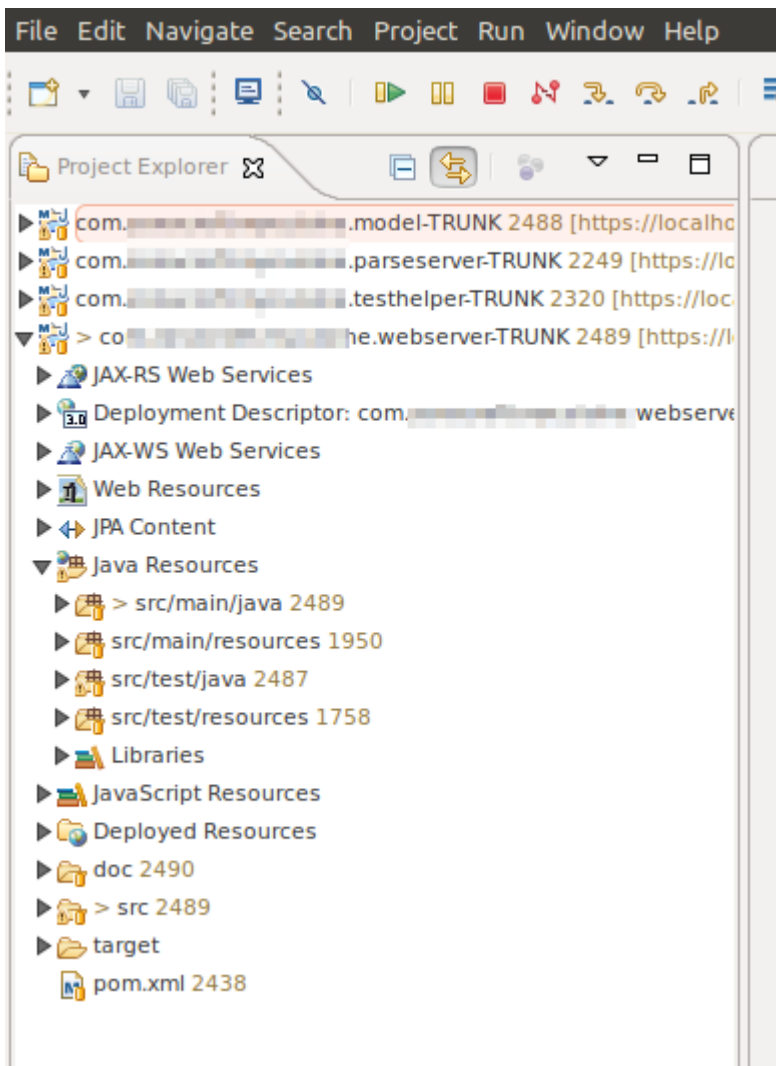
al final de los siguientes archivos:

```
e4_classic_winxp.css
e4_classic_win7.css
```

ANTES DEL CAMBIO



Despues del cambio



Lea Configurando Eclipse en línea: <https://riptutorial.com/es/eclipse/topic/2112/configurando-eclipse>

Capítulo 5: Configurando Eclipse para C ++

Examples

Linux + CMake (generador "Unix Makefiles") + Qt (opcional)

Debería tener un proyecto de proyecto CMake **simple** , y vamos a hacer un espacio de trabajo de Eclipse fuera de él:

```
myproject/  
  .git/  
  CMakeLists.txt  
  src/  
    main.cpp  
workspace/  
  myproject/  
    Release/  
    Debug/
```

Qt (opcional)

- Obtenga el último CDT de Eclipse y luego instale el paquete Qt en él a través de "Ayuda -> Instalar nuevo software".

Espacio de trabajo

- Cree un directorio de "área de trabajo" vacío junto con el directorio de origen de su proyecto CMake.
- Ejecute Eclipse y cambie a ese directorio de "área de trabajo".
- Cree un proyecto en C ++ (para Qt con Eclipse anterior a Neon: cree "Qt Makefile Project" y luego elimine el archivo * .pro, makefile y main.cpp)

Adjuntar fuentes al proyecto

- Vaya a Propiedades del proyecto -> Rutas y símbolos -> Ubicación de origen -> Carpeta de enlaces.
- Marque "Avanzado" y vincule la carpeta de origen del proyecto CMake así:
../.. /myproject/src/ . Funciona porque el espacio de trabajo está justo fuera del directorio del proyecto CMake.

Generador CMake

- Crear carpeta de `Release` en el proyecto.
- Vaya a la vista "Hacer objetivo" (`Ctrl + 3` y luego escriba "Hacer objetivo" si es difícil de encontrar). La vista "Make Target" se ve similar a la vista de proyecto.

- Haga clic derecho en la carpeta "Liberar" y luego haga clic en "Nuevo ...".
 - Desmarque "Igual que el nombre de destino".
 - Desmarque "Usar configuración del constructor".
 - Escriba "Liberar" en el campo "Nombre de destino".
 - Deje "Make target" vacío.
 - Establecer "comando de compilación" a algo como `cmake ../../../../myproject/`.
 - Haga clic en Aceptar.
- Haga doble clic en este destino "Release" que se acaba de crear en la carpeta Release. Eso ejecutará la generación de cmake.

Construir

- Vaya a Propiedades del proyecto y cree una configuración de "Lanzamiento".
- Active la configuración de "Liberación".
- Para la configuración "Liberar", desmarque "Generar Makefiles automáticamente".
- Establecer el directorio de compilación en "Release".
- Habilitar la construcción paralela.

Ahora, puede compilar el proyecto desde Eclipse con un `Ctrl + B` "Build".

Volver a ejecutar CMake (para volver a generar los makefiles)

- Eliminar todo del directorio "Release".
- Ve a la vista "Make Target".
- Haga doble clic en el objetivo "Liberar".

Lea [Configurando Eclipse para C ++ en línea](https://riptutorial.com/es/eclipse/topic/7028/configurando-eclipse-para-c-plusplus):

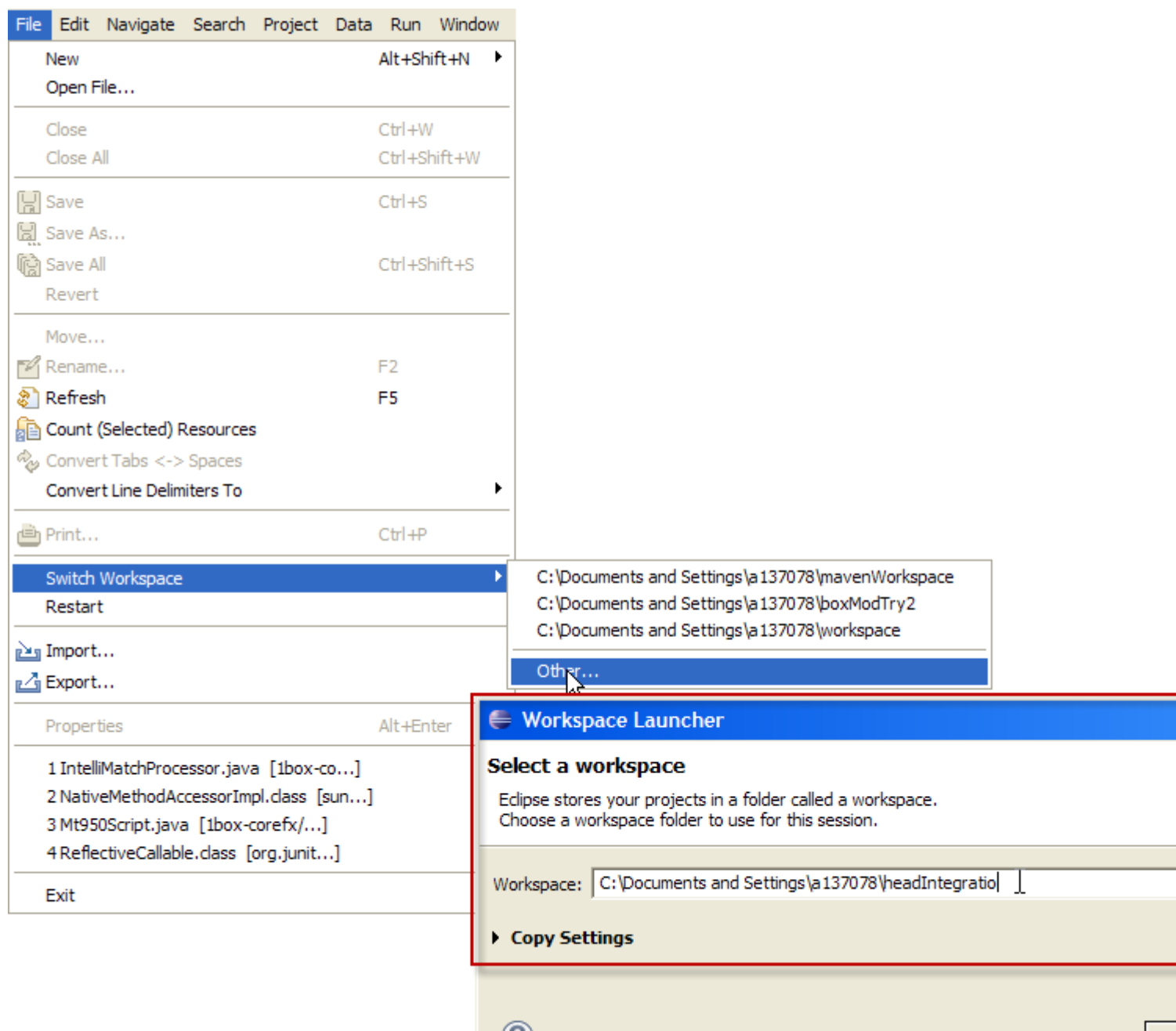
<https://riptutorial.com/es/eclipse/topic/7028/configurando-eclipse-para-c-plusplus>

Capítulo 6: Crear un nuevo espacio de trabajo en Eclipse.

Examples

Cómo crear un espacio de trabajo

Vaya a Archivo -> Cambiar espacio de trabajo -> Otro ... y escriba su nuevo nombre de espacio de trabajo.



Lea [Crear un nuevo espacio de trabajo en Eclipse](https://riptutorial.com/es/eclipse/topic/6345/crear-un-nuevo-espacio-de-trabajo-en-eclipse-). en línea:

<https://riptutorial.com/es/eclipse/topic/6345/crear-un-nuevo-espacio-de-trabajo-en-eclipse->

Capítulo 7: Depuración de programas Java en Eclipse.

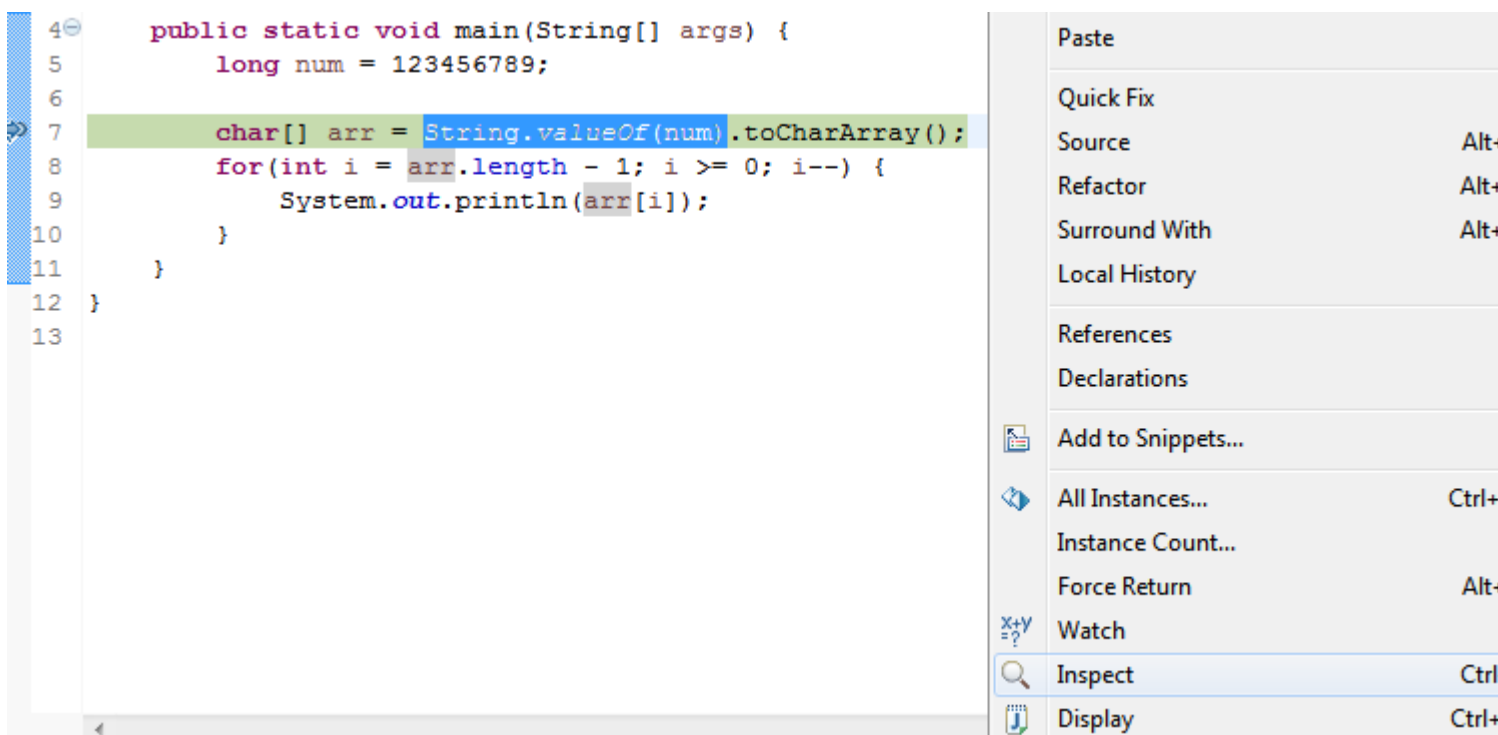
Examples

Evaluar expresiones dentro de una sesión de depuración

Hay varios para evaluar una determinada expresión al depurar una aplicación Java.

1. Inspeccionar manualmente una expresión

Cuando la ejecución del programa se suspende en una determinada línea (ya sea debido a un punto de interrupción o al pasar manualmente por el depurador), puede evaluar manualmente una expresión seleccionando la expresión en el código, luego haciendo clic derecho y seleccionando **Inspeccionar** como se muestra a continuación. captura de pantalla. Alternativamente, haga `Ctrl + Shift + I` después de seleccionar la expresión.



2. Viendo una expresión en la vista de expresiones

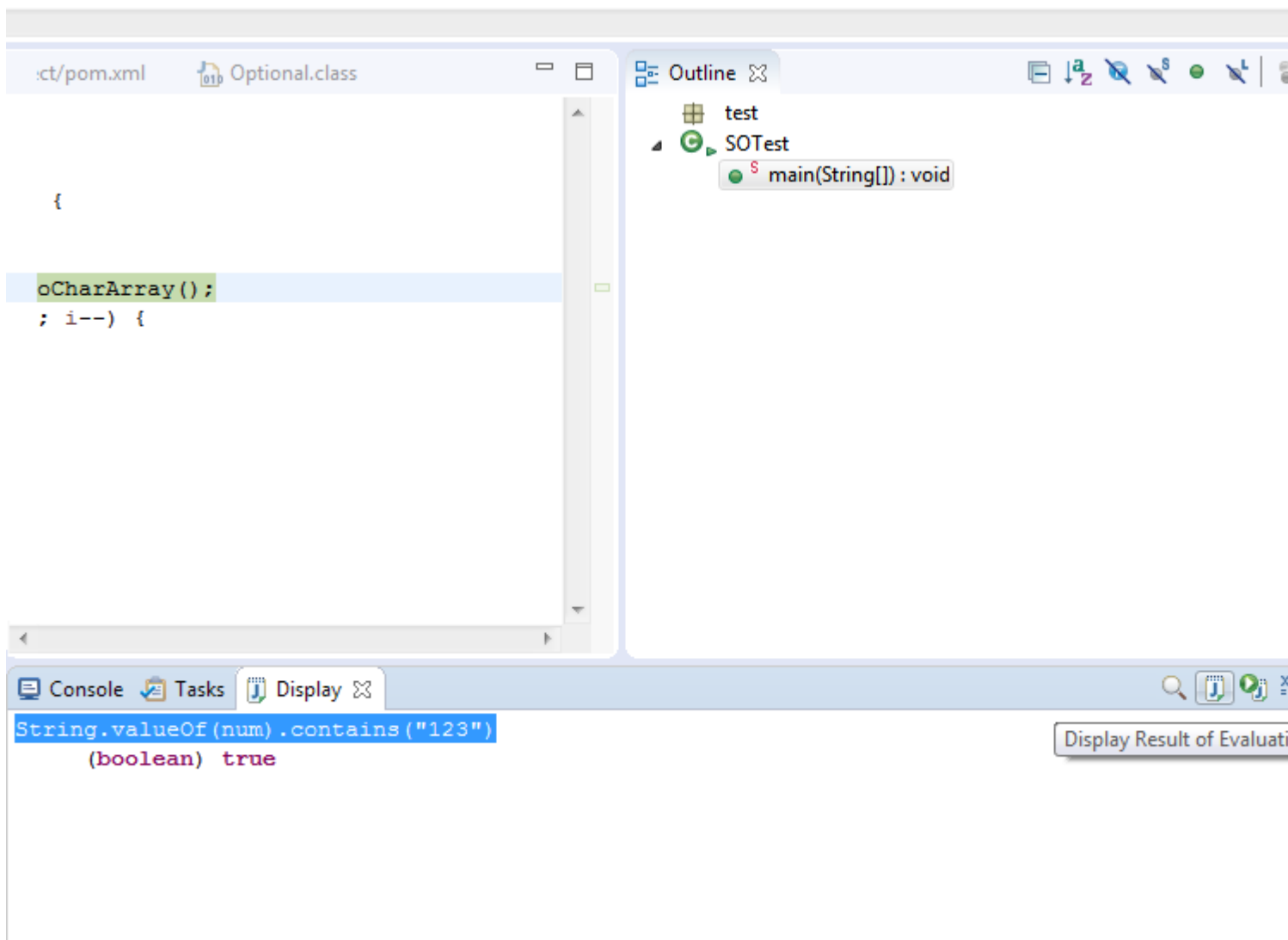
Si desea evaluar continuamente una expresión, por ejemplo, porque está dentro de un bucle, puede verla en la vista Expresiones. De esta manera, su valor se mostrará cada vez que el programa se suspenda en cada iteración del bucle. Para hacer esto, seleccione la expresión deseada, luego haga clic derecho y seleccione **Observar**. Esto abrirá la vista de **Expresiones** y mostrará el valor de la expresión (vea la imagen de abajo). También puede escribir manualmente la expresión en la vista.

| Name | Value |
|----------------------------------|-----------|
| <code>String.valueOf(num)</code> | 123456789 |
| + Add new expression | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

3. Usando la vista Visualizar para evaluar y ejecutar sentencias.

La vista de **Pantalla** le permite escribir sus propias expresiones, declaraciones o cualquier código en general que se evaluaría o ejecutaría en contexto con el código de programa suspendido. Esto puede ser útil si desea evaluar expresiones complejas sin cambiar su original y reiniciar la depuración.

Para abrir la vista de Pantalla, seleccione **Ventana> Mostrar Vista> Pantalla** . Luego escriba su expresión o declaraciones en la vista y seleccione una de las opciones en la barra de herramientas de la vista, por ejemplo, para ejecutar las declaraciones escritas, o muestre el resultado de evaluarlas en la vista como se muestra en la imagen de abajo. El código escrito en la vista de Pantalla también se puede inspeccionar o ejecutar seleccionándolo, luego haciendo clic derecho y seleccionando la acción deseada.



Depuración remota de una aplicación Java

Con el fin de depurar una aplicación Java remota, se debe iniciar con algunos argumentos adicionales para indicar a la JVM que la ejecute en modo de depuración. Esto se hace de la siguiente manera:

```
java -Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=8000 -jar sampleApp.jar
```

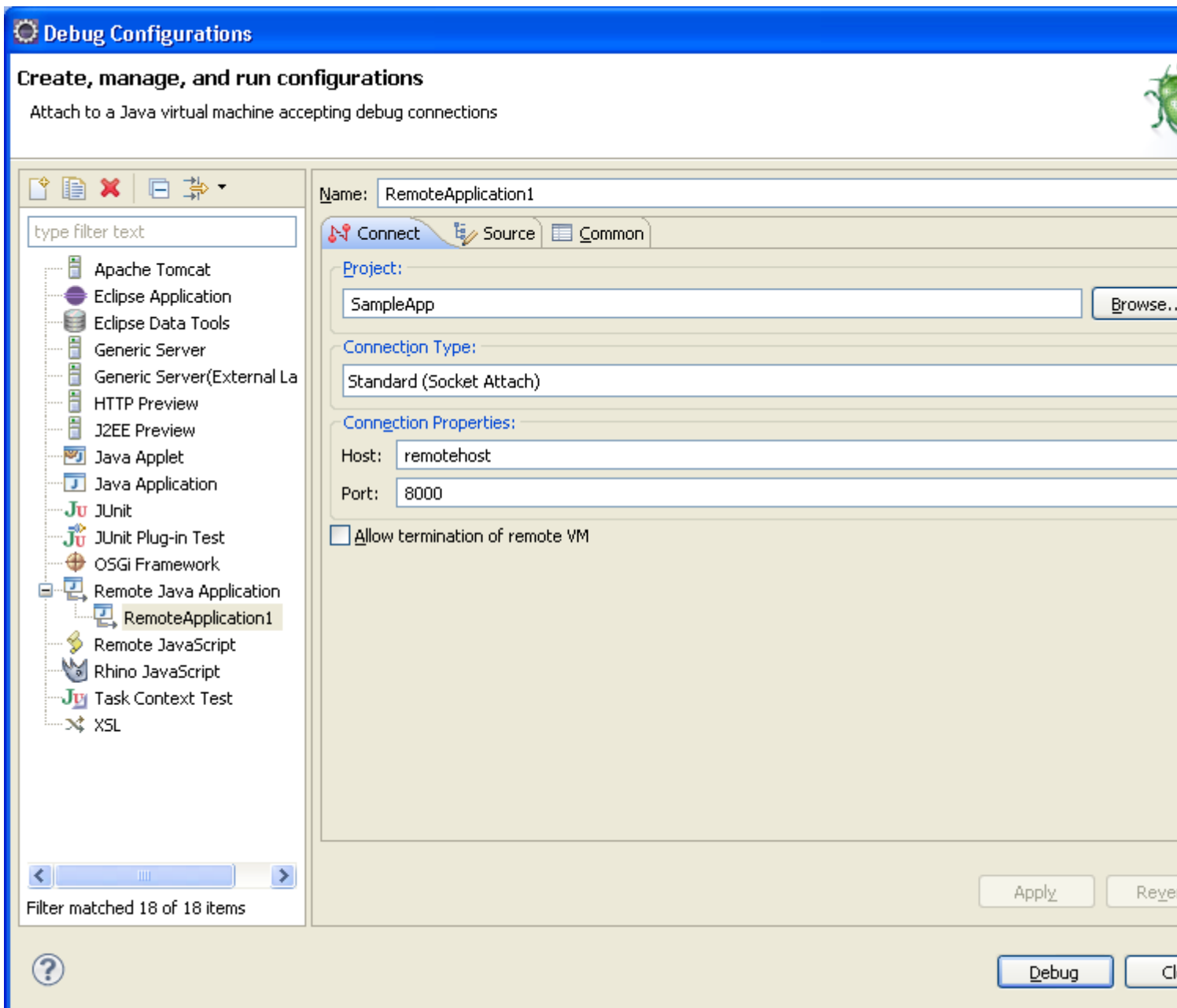
El comando anterior le dice a la JVM que inicie la aplicación sampleApp.jar mientras tiene un socket de servidor escuchando en el puerto 8000 para que un depurador se conecte a él. El argumento de suspensión indica si desea que la JVM espere hasta que un depurador se adjunte al número de puerto antes de que la aplicación se ejecute efectivamente.

Después de iniciar la aplicación remota con los parámetros anteriores, el siguiente paso es conectar su depurador Eclipse a la aplicación remota. Esto se hace de la siguiente manera:

1. Vaya a **Ejecutar > Configuraciones de depuración ...**
2. En el cuadro de diálogo **Configuraciones de depuración** (ver la figura a continuación), seleccione la sección **Aplicación Java remota** y haga clic en **Nueva** acción de

configuración de inicio .

3. Asígnele un nombre a su configuración de depuración, luego seleccione el proyecto que contiene el código de su aplicación remota. Configure el nombre de host o la dirección IP de la máquina remota y el puerto al que se debe conectar el depurador.
4. Haga clic en **Depurar** . El depurador de Eclipse ahora intentará conectarse a la aplicación remota y suspender en los puntos de interrupción establecidos en el código fuente.



Lea Depuración de programas Java en Eclipse. en línea:

<https://riptutorial.com/es/eclipse/topic/4548/depuracion-de-programas-java-en-eclipse->

Capítulo 8: Depuración remota en Eclipse

Examples

Configurar la depuración remota de Eclipse para una aplicación

Los siguientes son los pasos para iniciar un depurador remoto de Eclipse. Esto es útil cuando la aplicación no se inicia desde una instancia de servidor dentro de Eclipse. Esta característica es realmente poderosa y también puede ayudar a depurar el código que reside en el entorno de prueba o producción. Echemos un vistazo a la configuración:

Configuraciones de Eclipse:

1. Haga clic en el botón Ejecutar
2. Seleccione las configuraciones de depuración
3. Seleccione la "Aplicación Java Remota"
4. Nueva configuración
 - a) Nombre: GatewayPortalProject
 - b) Proyecto: GatewayPortal-portlet
 - c) Tipo de conexión: Socket Attach
 - d) Propiedades de conexión:
 - i) localhost ii) 8787

Para JBoss:

`/path/toJboss/jboss-eap-6.1/bin/standalone.conf` **el** `/path/toJboss/jboss-eap-6.1/bin/standalone.conf` en su vm de la siguiente manera: **Descomente la siguiente línea eliminando el #:**

```
JAVA_OPTS="$JAVA_OPTS -agentlib:jdwp=transport=dt_socket,address=8787,server=y,suspend=n"
```

Para Tomcat:

En el archivo **catalina.bat** :

Paso 1:

```
CATALINA_OPTS="-Xdebug -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n"
```

Paso 2:

```
JPDA_OPTS="-agentlib:jdwp=transport=dt_socket,address=8000,server=y,suspend=n"
```

Paso 3: Ejecute Tomcat desde el símbolo del sistema como se muestra a continuación:

```
catalina.sh jpda start
```

Luego debe establecer puntos de interrupción en las clases de Java que desea depurar.

Lea [Depuración remota en Eclipse en línea](#):

<https://riptutorial.com/es/eclipse/topic/3502/depuracion-remota-en-eclipse>

Capítulo 9: Procedimiento de despliegue de Tomcat

Examples

Procedimiento cuando nada más ayuda.

De vez en cuando, las implementaciones frecuentes en tomcat interno comienzan a dar un error constante, sin ninguna causa clara (inicio de escucha o excepción `ClassNotFoundException`). Cuando nada parece curarlo, este procedimiento salva al mundo:

- 1 eliminar la carpeta Servidores
- 2 reiniciar Eclipse
- 3 crear nuevo servidor, agregar proyecto y comenzar

Funciona como el encanto y no es tan largo.

Si por alguna razón esto falla, mi procedimiento largo y original (donde el otro es un atajo que debería hacer lo mismo) está aquí:

- 1 servidor de parada
- 2 proyecto -> limpio
- 3 proyecto de compilación (tuve la compilación automática deshabilitada)
- 4 eliminar servidor
- 5 eliminar la carpeta Servidores
- 6 reiniciar Eclipse
- 7 crear nuevo servidor, agregar proyecto y comenzar.

Con estos siete pasos, los problemas con la implementación nunca salen de su código y control.

Nota:

No necesita otra cosa que actualizar la página si todo va bien. Este procedimiento se realiza una vez por mensaje de error para asegurarse de eliminar el error poco claro, si su código parece correcto y no hizo nada para recibir el error. El error es que contiene la palabra `ClassNotFoundException` o `ListenerStart`, dependiendo del entorno en uso. Tenga en cuenta también que esto no soluciona las `ClassNotFoundExceptions` causadas por bibliotecas faltantes en un proyecto.

Lea Procedimiento de despliegue de Tomcat en línea:

<https://riptutorial.com/es/eclipse/topic/6092/procedimiento-de-despliegue-de-tomcat>

Creditos

| S. No | Capítulos | Contributors |
|-------|--|---|
| 1 | Empezando con eclipse | Aaron Vigal , Ala Eddine JEBALI , Aleksandr M , Ani Menon , Community , Latsuj , manouti , Ray , Thomas Fritsch |
| 2 | Atajos Eclipse | Latsuj , Ray , Srishti Sinha , user7491506 , Yurii COjocari |
| 3 | Cómo funciona la depuración remota de Eclipse detrás de escena | Pritam Banerjee |
| 4 | Configurando Eclipse | Bevor , manouti , mecscsco , romeara |
| 5 | Configurando Eclipse para C ++ | Velkan |
| 6 | Crear un nuevo espacio de trabajo en Eclipse. | rajah9 |
| 7 | Depuración de programas Java en Eclipse. | manouti |
| 8 | Depuración remota en Eclipse | Pritam Banerjee |
| 9 | Procedimiento de despliegue de Tomcat | mico |