

U02.- Eclipse. Instalación y Uso

Instalación Eclipse

Para instalar Eclipse bastará con descargarse el archivo comprimido y descomprimirlo en el disco duro.

Accederemos a la web de Eclipse (<http://www.eclipse.org/downloads/>) . Nos descargamos la versión Eclipse IDE for Java Developers de 32 o 64 bits en función de nuestro equipo.

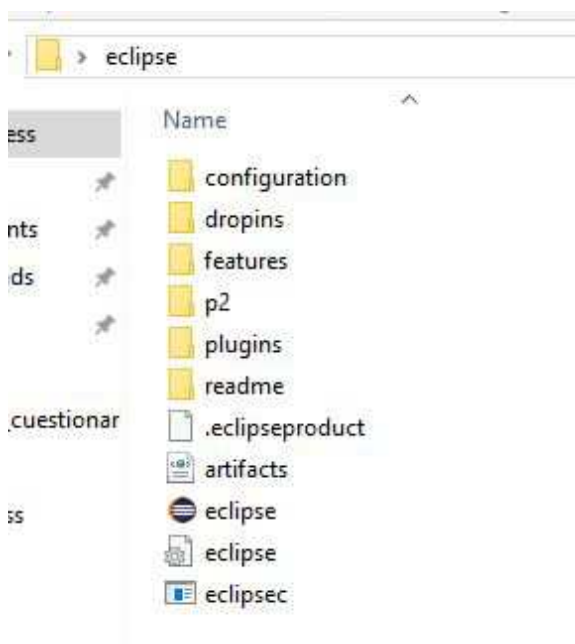
<http://www.eclipse.org/downloads/eclipse-packages/>

Seleccionamos:

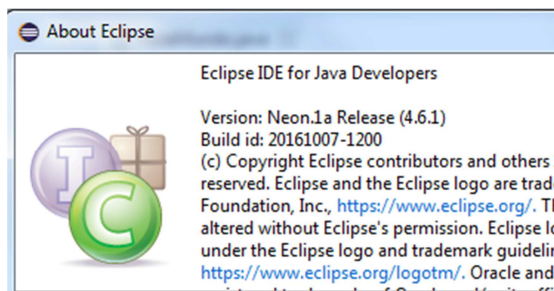


Una vez descargado el fichero bastará con descomprimir la carpeta eclipse en la ubicación que deseemos.

Nombre	Tamaño	Comprimido	Tipo	Modificado	CRC32
..			Folder		
eclipse			Folder	24/09/2015 6:36	

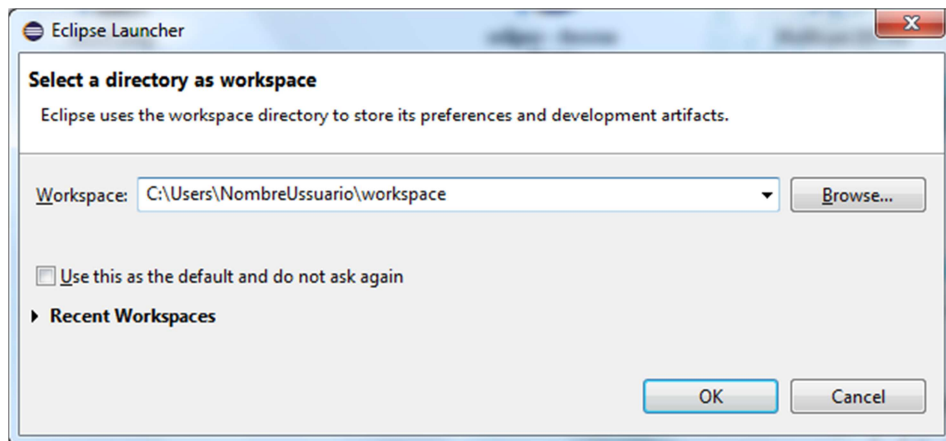


Instalamos la siguiente versión:



Tras esto ejecutamos el archivo eclipse.exe.

Aparecerá un cuadro de diálogo en el que pedirá seleccionar el workspace de trabajo. Eclipse almacena los proyectos en espacios de trabajo llamados workspaces. Se puede crear en ese momento la carpeta para el workspace o seleccionar una ya creada en el equipo.

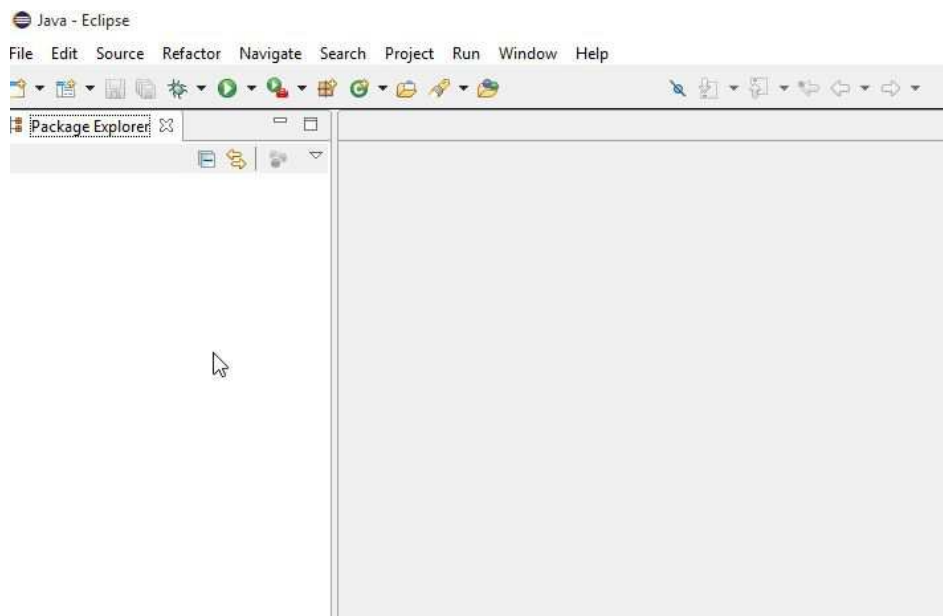


En mi caso voy a dejar a ruta por defecto. Pulsamos ok.

Nos carga Eclipse con la ventana de bienvenida y una serie de opciones para ver tutoriales, o para ver lo nuevo, o ejemplo. En nuestro caso seleccionaremos workbench (en la parte superior derecha)

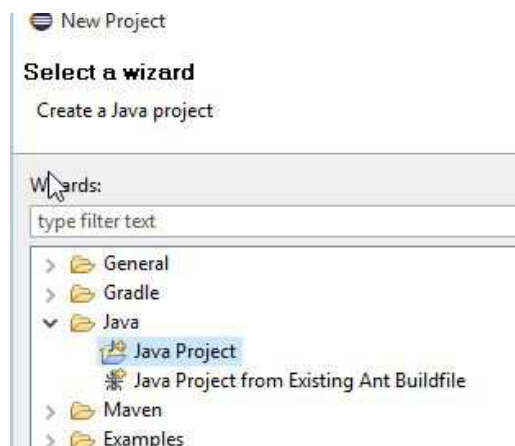


A continuación aparece la ventana principal de Eclipse.



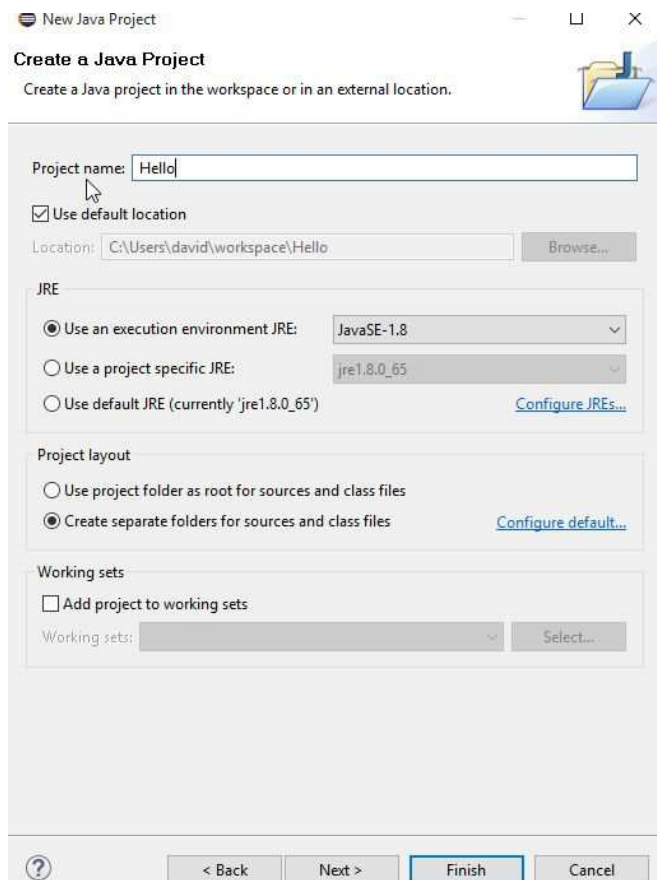
Mi primer programa “hola Eclipse”

Para crear nuestro primer proyecto. Hacemos "File / New / Project...". Nos sale una ventana donde seleccionamos "Java project".



Ahora encontramos una ventana de configuración del proyecto

1. Asignamos un nombre al proyecto. Hello
2. Dejamos marcada "Use default location", para que use la ruta de nuestro workspace configurado y Eclipse se encargará de hacer un subdirectorio para este nuevo proyecto.
3. JRE puedes dejar la que tienes por defecto en Eclipse.
4. En project layout puedes dejar también la configuración que te sale "créate separate folders..."
5. Pulsamos "Finish"



Una vez tenemos creada la estructura del proyecto, vamos a crear la clase para poder añadir el código de programación (al igual que hacíamos con Netbeans).

Seleccionamos el proyecto pulsamos el botón derecho New Class , nos aparece ventana donde debemos introducir los datos de la clase y pulsar finish. Lo rellenamos según la imagen, fijaos que es muy similar a los datos que rellenamos en NetBeans al crear una clase.

Finalmente pulsamos finish

Java Class
Create a new Java class.

Source folder: Hello/src Browse...

Package: helloeclipse Browse...

☐ Enclosing type: Browse...

Name: Helloeclipse

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?
☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

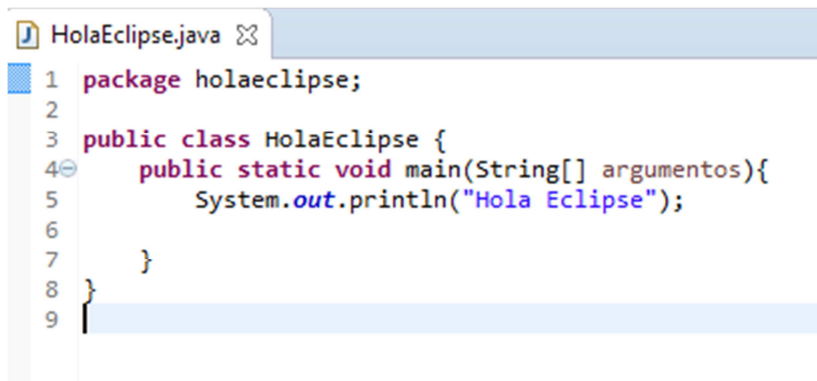
Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

Finish Cancel

Nos crea la estructura de la clase

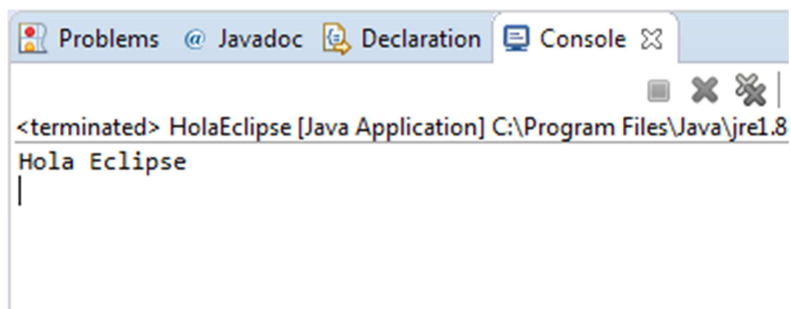
```
Helloeclipse.java
1 package helloeclipse;
2
3 public class Helloeclipse
4
5 }
6
```

Ahora nos restará añadir el código de la clase principal (main)
La clase quedará de la siguiente forma



```
1 package holaclipse;
2
3 public class HolaEclipse {
4     public static void main(String[] argumentos){
5         System.out.println("Hola Eclipse");
6     }
7 }
8
9
```

Tras esto podemos guardar y mediante el menú Run>run probar el funcionamiento.

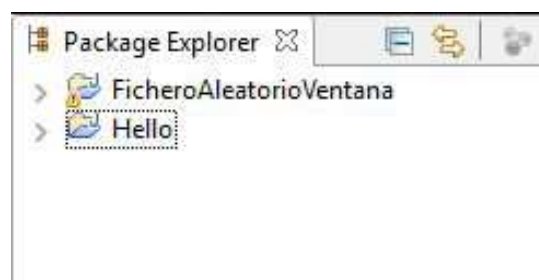


```
<terminated> HolaEclipse [Java Application] C:\Program Files\Java\jre1.8
Hola Eclipse
```

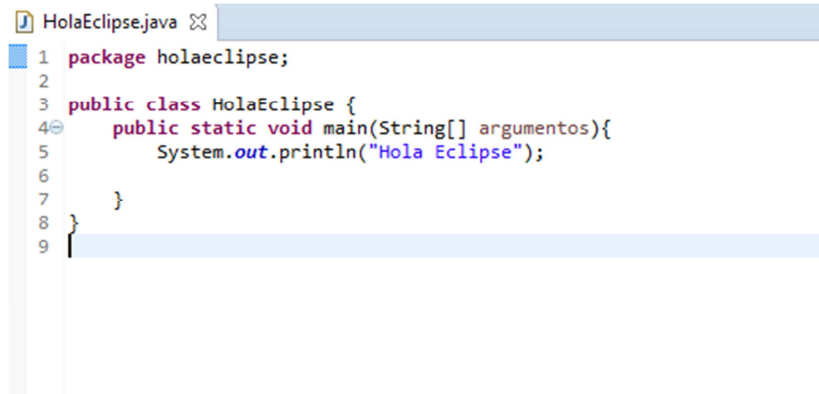
Recorriendo Eclipse

En la ventana principal de trabajo de Eclipse podremos distinguir los siguientes bloques

- Package Explorer: zona de proyectos, desde aquí se podrá navegar por todo el workspace, por los proyectos y los elementos que los componen, carpetas, archivos, librerías, etc...



- Zona de edición: desde aquí se escribirá el código de los programas. Observa que el texto aparece resaltado para identificar la sintaxis y palabras del lenguaje. Si se escribe una instrucción errónea el editor marcará la línea y ofrecerá posibles soluciones. Además al escribir código muestra propuestas de plantilla

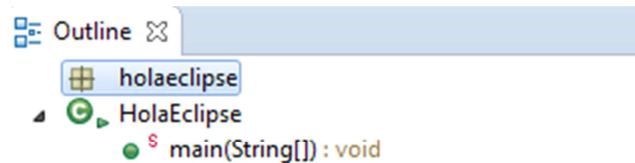


```

1 package holaclipse;
2
3 public class HolaEclipse {
4     public static void main(String[] argumentos){
5         System.out.println("Hola Eclipse");
6     }
7 }
8
9

```

- Outline En esta zona se muestra el esquema de la clase que se edita en ese momento, permite acceder de forma más rápida a sus métodos y atributos.

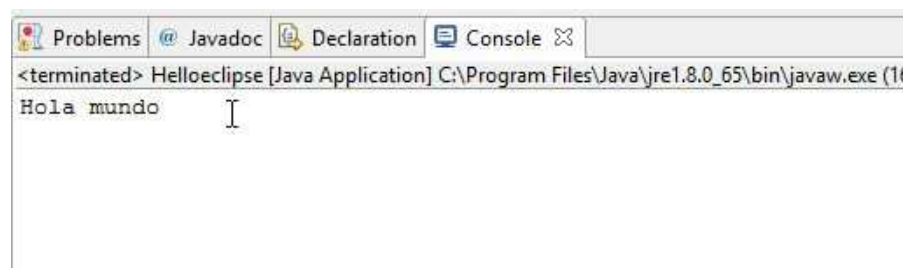


```

Outline
├── holaclipse
│   └── HolaEclipse
│       └── main(String[]) : void

```

- Consola Java: En esta zona se muestra el resultado de la ejecución de los programas, los mensajes de salida por consola, y los errores de ejecución.



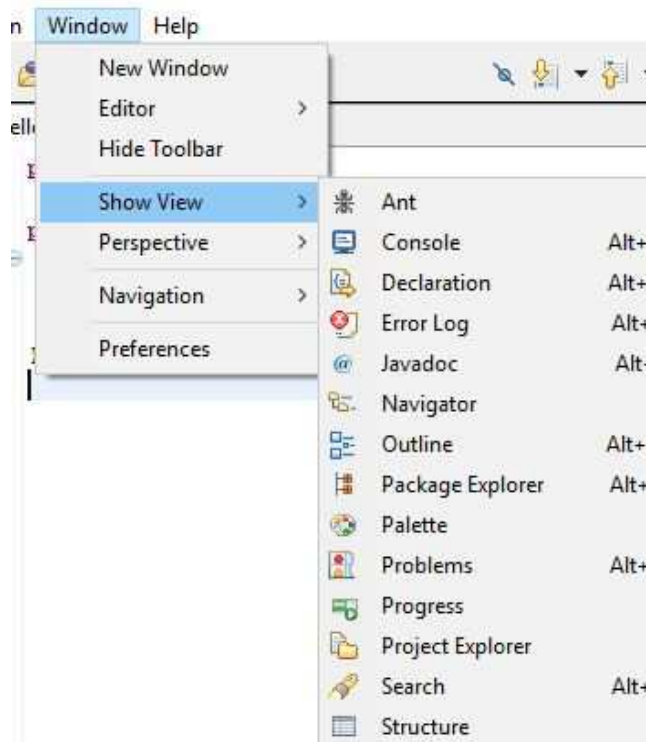
```

Problems  Javadoc  Declaration  Console
<terminated> Helloeclipse [Java Application] C:\Program Files\Java\jre1.8.0_65\bin\javaw.exe (11)
Hola mundo

```

La visualización se podrá modificar en función del proyecto a realizar y los plugins instalados.

Se podrán mostrar otras vistas dentro de Eclipse desde el menú Windows/Show view.



Practicando

1. Radio. Realiza un programa en Eclipse que pida el radio y calcule el área y longitud de la circunferencia. Según las fórmulas:

Longitud de la circunferencia = $2 \cdot \pi \cdot \text{Radio}$

Area de la circunferencia = $\pi \cdot \text{Radio}^2$

```
Introduce radio de la circunferencia:
2
Longitud de la circunferencia -> 12.566370614359172
Area de la circunferencia -> 12.566370614359172
```

2. Convertemetros. Programa que pase una velocidad en Km/h a m/s. La velocidad se lee por teclado.

```
Introduce velocidad en Km/h:
20
20.0 Km/h -> 5.555555555555555 m/s
```


Eclipse. Instalar el Entorno Gráfico

Instalando plugins

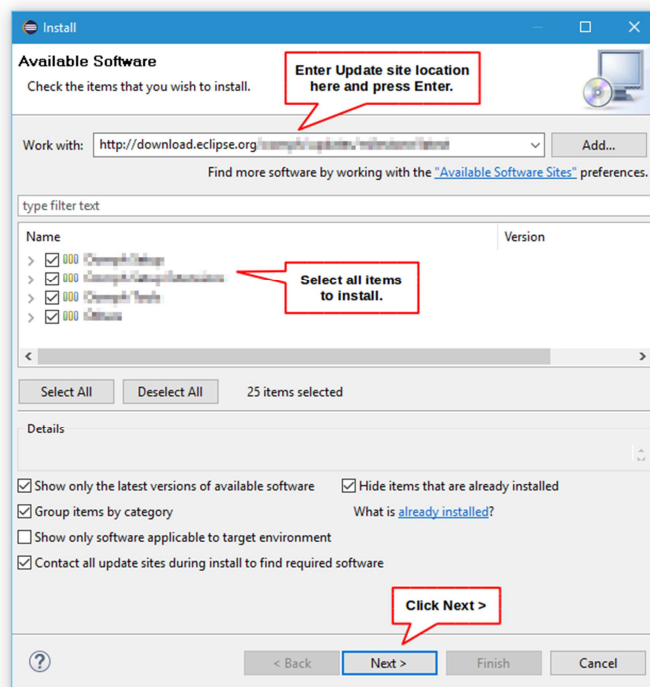
Un plugin es un programa que añade funcionalidades específicas a otro programa que hace de programa principal. Su presencia es muy habitual en los navegadores web, en sistemas de reproducción de música y audio, en CMS y en herramientas de desarrollo como Eclipse.

Los plugins no son actualizaciones, sino propiedades añadidas a programas originales.

Plugin WindowBuilder

Eclipse no dispone de una interfaz gráfica de usuario para crear aplicaciones con ventanas, para ello será necesario instalar plugins. Uno de los plugins más populares es el WindowBuilder, desarrollado por Google y que nos permite desarrollar de forma rápida y cómoda la GUI de las aplicaciones Java.

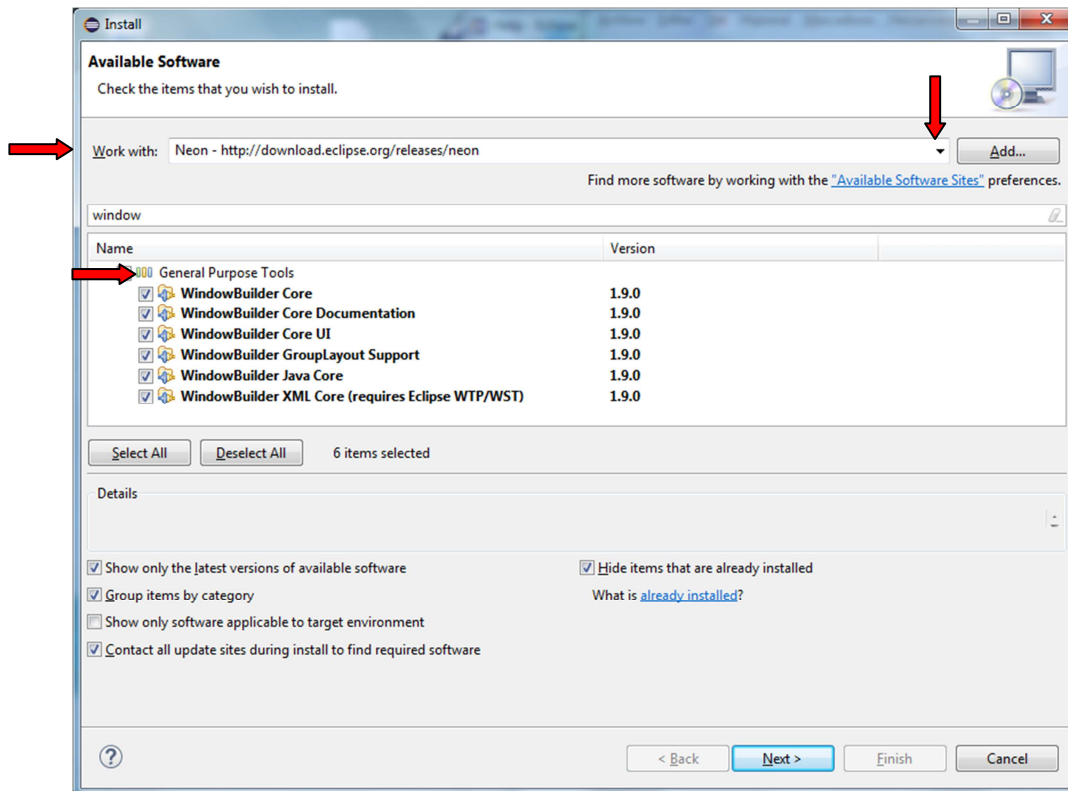
Para instalarlo debes acceder al Menu -> Help -> Install new software



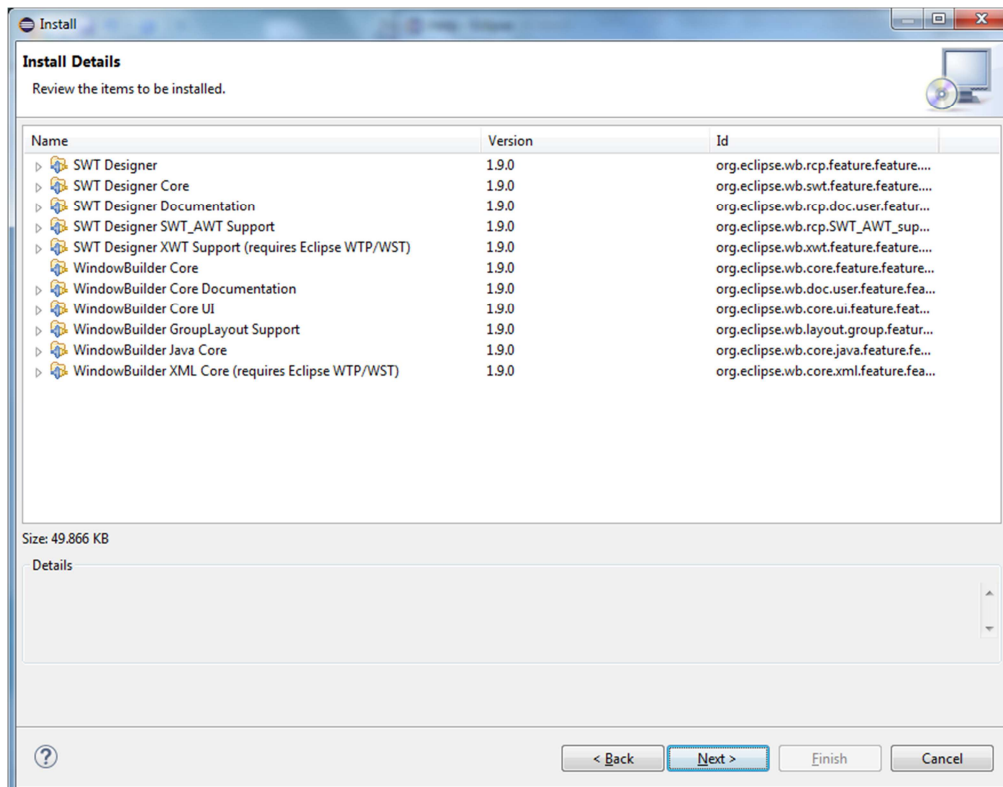
Debemos instalar los siguientes componentes:

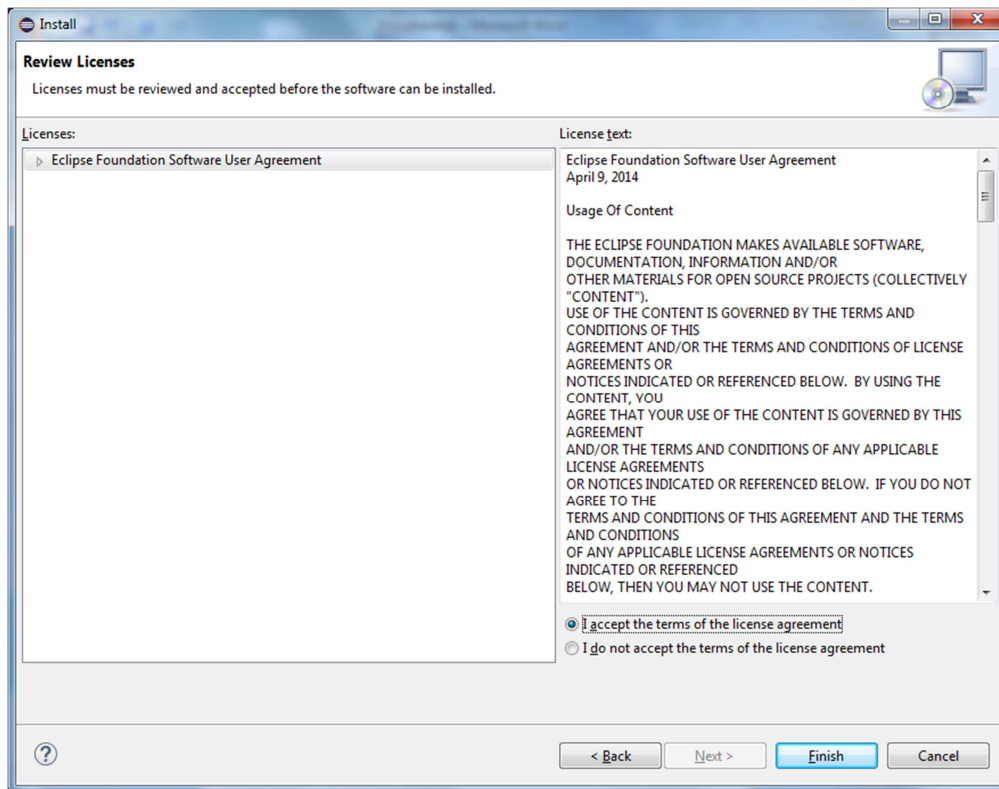
SWT Designer
SWT Designer Core
SWT Designer Documentation
SWT Designer SWT_AWT Support.
SWT Designer XWT Support

WindowBuilder Core
WindowBuilder Core Documentation
WindowBuilder Core UI
WindowBuilder GroupLayout Support
WindowBuilder Java Core
WindowBuilder XML Core



Así pues:





Y ya lo tenemos instalado!

En otro documento vemos una introducción a su uso

Nuevas Variables de entorno

Si al ejecutar Eclipse nos aparece una ventana de error que nos dice:

A Java Runtime Environment (JRE) or Java Development Kit (JDK) must be available in order to run Eclipse. No Java virtual machine was found after searching the following locations: C:\Users...\eclipse\rebin\javaw.exe javaw.exe in your current PATH

Debemos crear una nueva variable de entorno

- Nombre de la variable: JAVA_HOME
- Valor de la variable: Ruta donde está instalado el JDK de java, por defecto está en C:\Program Files\Java\jdk

Abrir la ventana de "Propiedades del sistema", donde pulsaremos el botón "Variables de entorno...". En la nueva ventana que se llama justo "Variables de entorno" pulsamos sobre el botón "Nueva...". En el recuadro de "Nueva variable del sistema" introducimos lo siguiente:

