EBook Gratis

APRENDIZAJE eclipse-plugin

Free unaffiliated eBook created from
Stack Overflow contributors.

#eclipse-plugin

Tabla de contenido

Acerca de 1

Capítulo 1: Empezando con eclipse-plugin 2

Observaciones 2

Examples 2

Instalación o configuración 2

Hola Mundo 4

Creditos 14

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: eclipse-plugin

It is an unofficial and free eclipse-plugin 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-plugin.

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-plugin

Observaciones

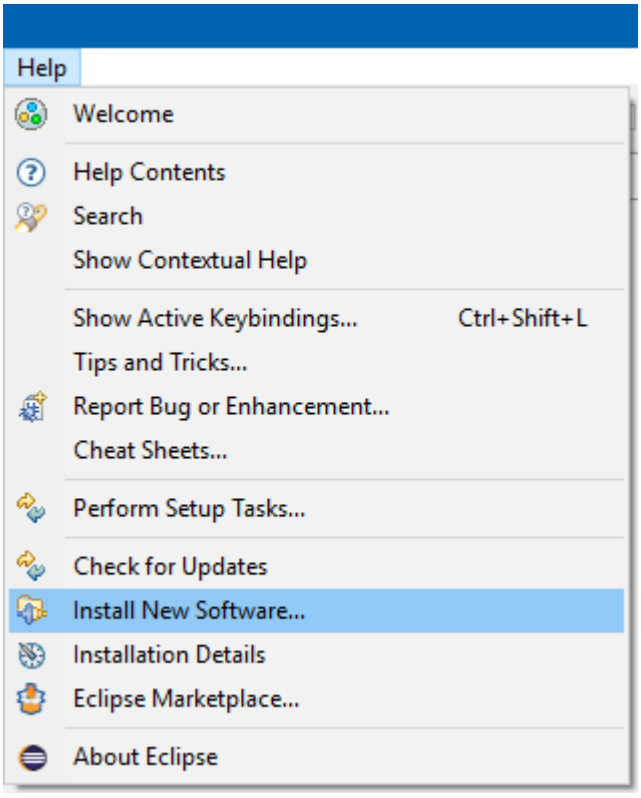
Esta sección proporciona una descripción general de qué es eclipse-plugin y por qué un desarrollador puede querer usarlo.

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

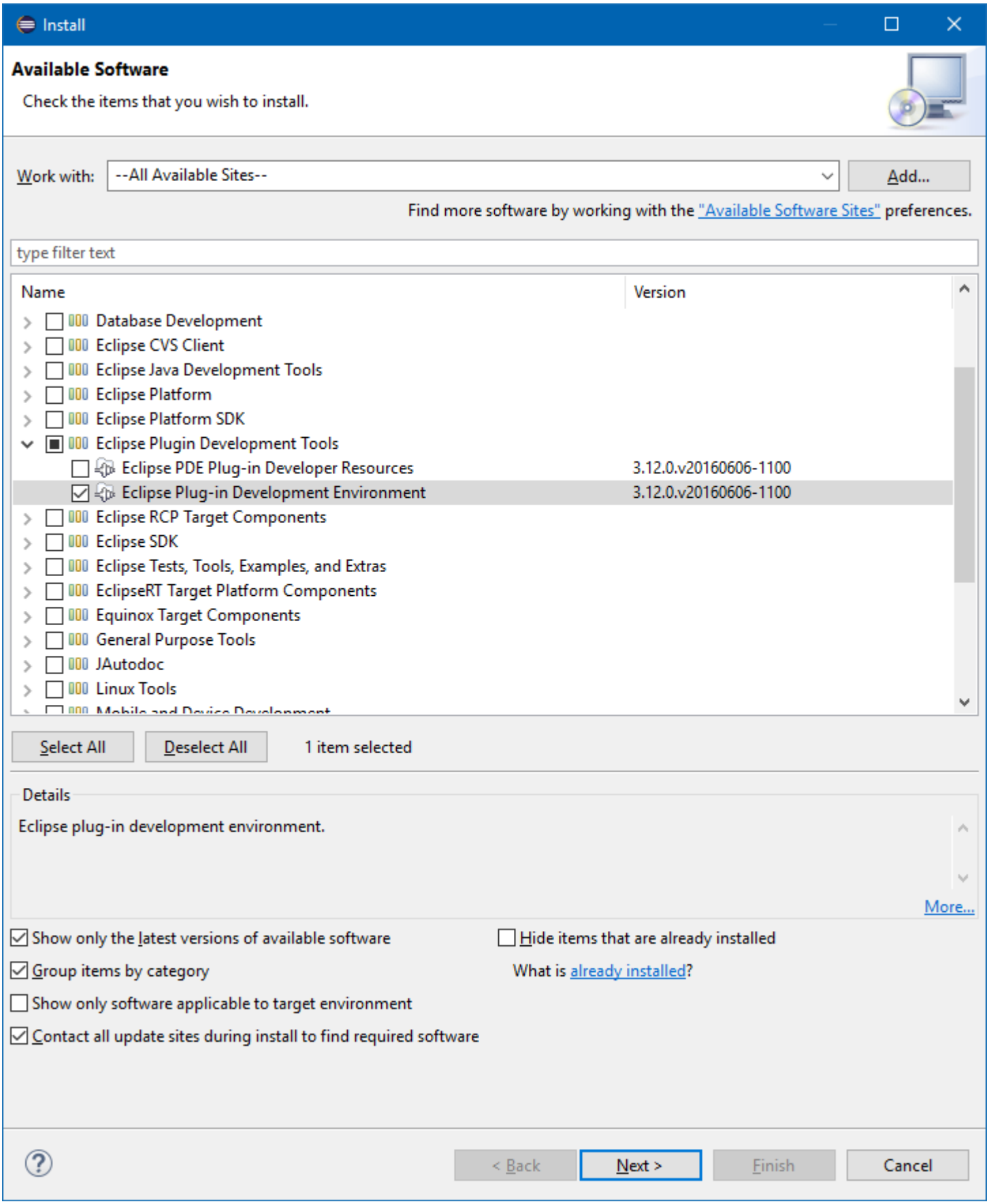
Examples

Instalación o configuración

Suponiendo que tiene Eclipse IDE para desarrolladores de Java instalado, inicie Eclipse, haga clic en " *Ayuda* " -> " *Instalar nuevo software ...* "



Seleccione " *--Todos los sitios disponibles--* " en " *Trabajar con:* " y navegue hasta " *Herramientas de desarrollo de complementos de Eclipse* ". Seleccione " *Eclipse Plug-in Development Environment* " marcando la casilla de verificación delante de él.

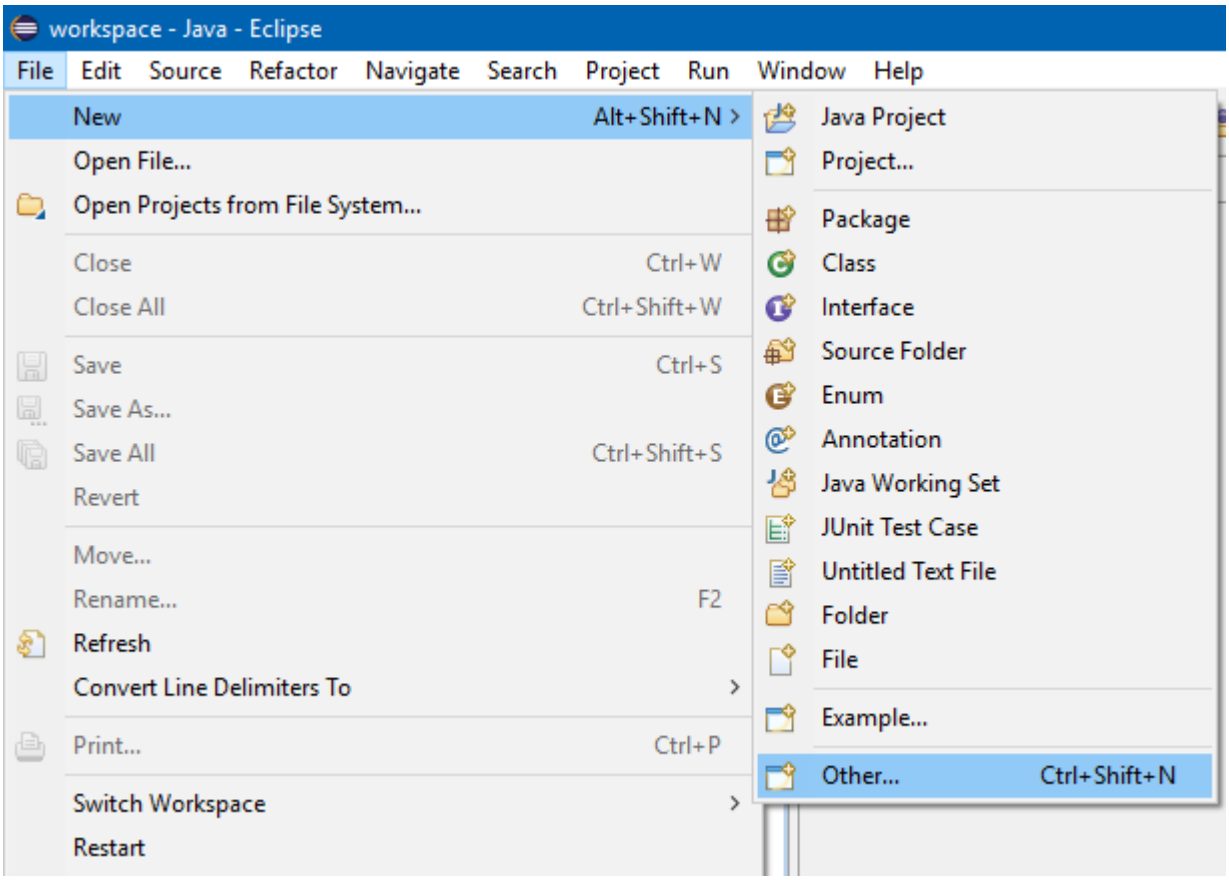


Haga clic en " *Siguiente* " para permitir que Eclipse verifique las dependencias necesarias. Haga clic en " *Siguiente* " de nuevo para iniciar la instalación.

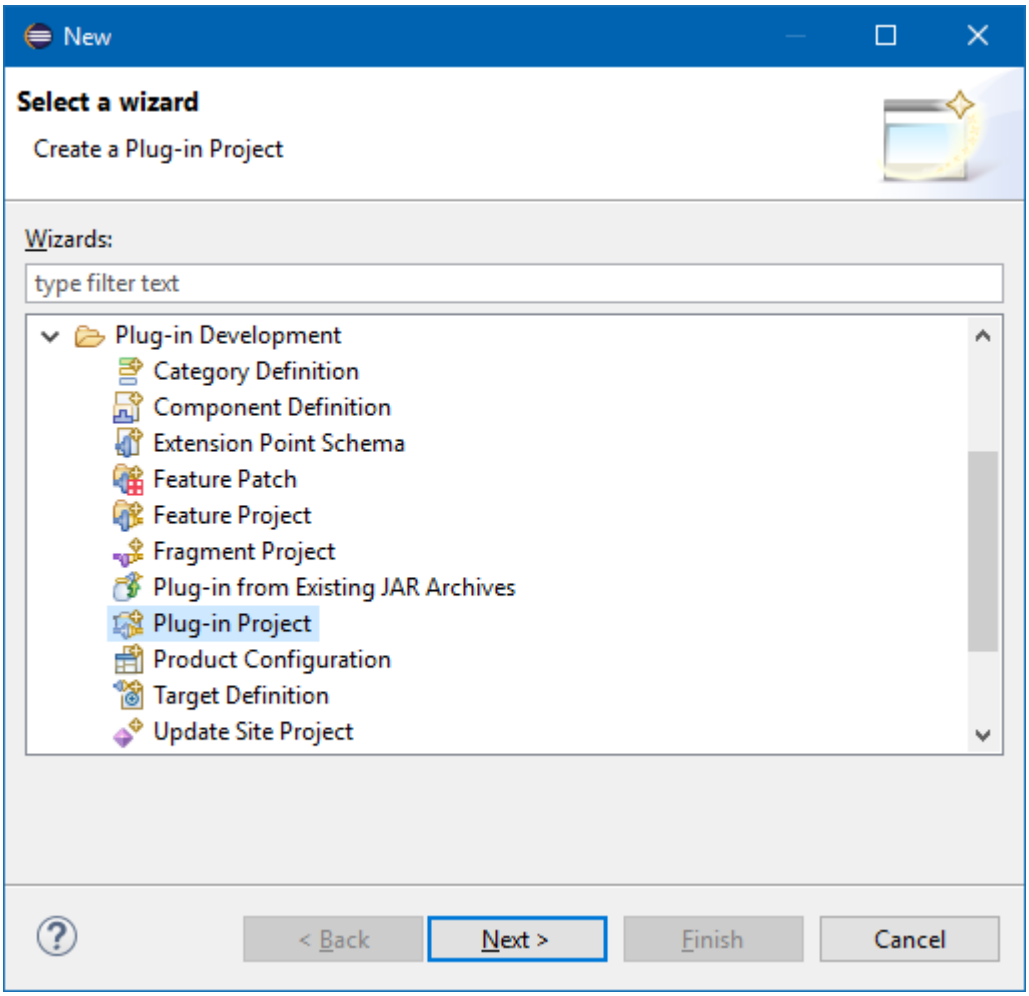
Una vez que haya terminado, reinicie Eclipse.

Hola Mundo

Para crear un complemento de Hello World para Eclipse, haga clic en: *Archivo* ➜ *Nuevo* ➜ *Otro ...*

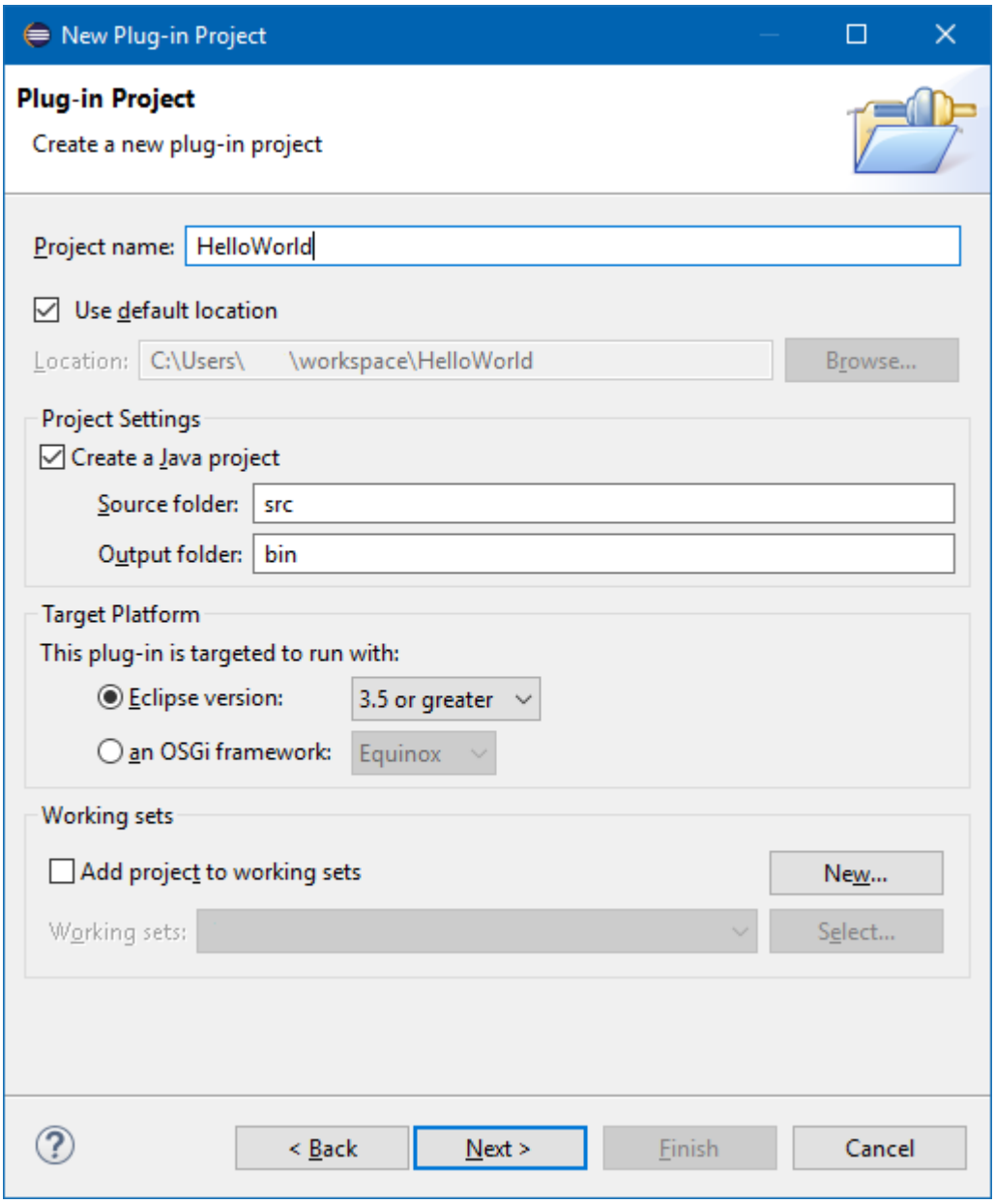


Seleccione *Proyecto de complemento* y haga clic en *Siguiente>*



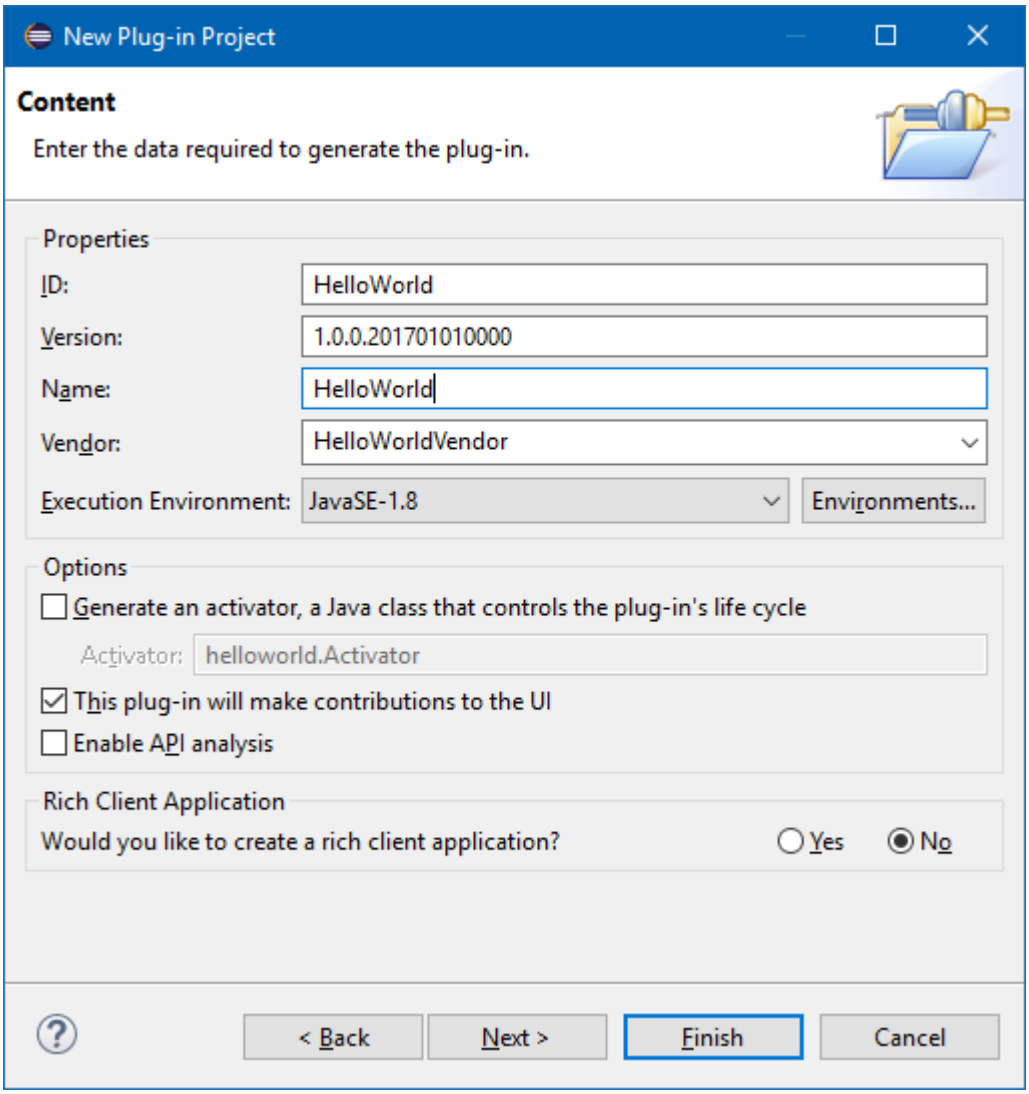
El asistente *Nuevo proyecto de complemento* lo guiará a través de las opciones para crear un nuevo complemento.

Ingrese un nombre de proyecto (como HelloWorld) y haga clic en *Siguiente>*

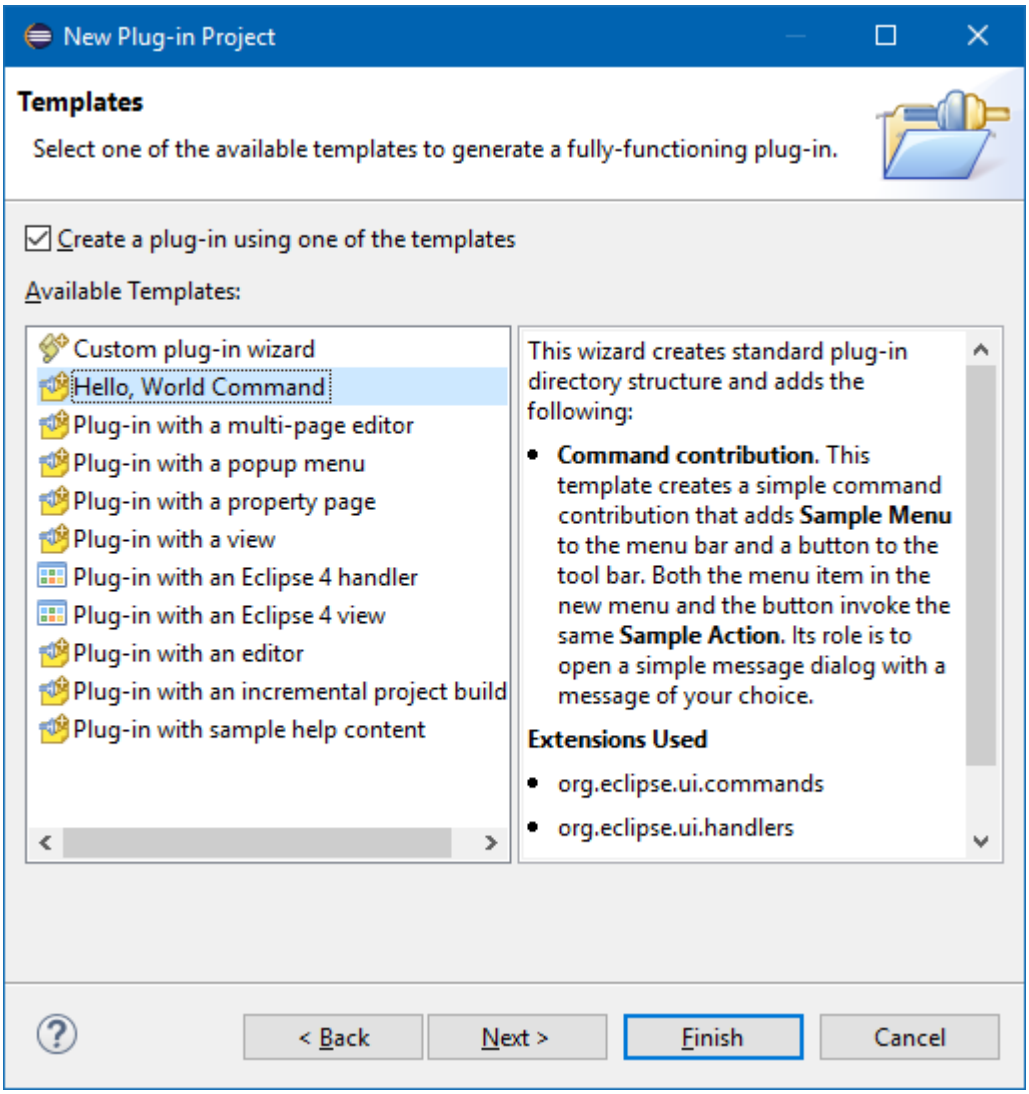


En la página *Contenido* , puede configurar el *ID* , la *versión* , el *nombre* y el *proveedor* del complemento.

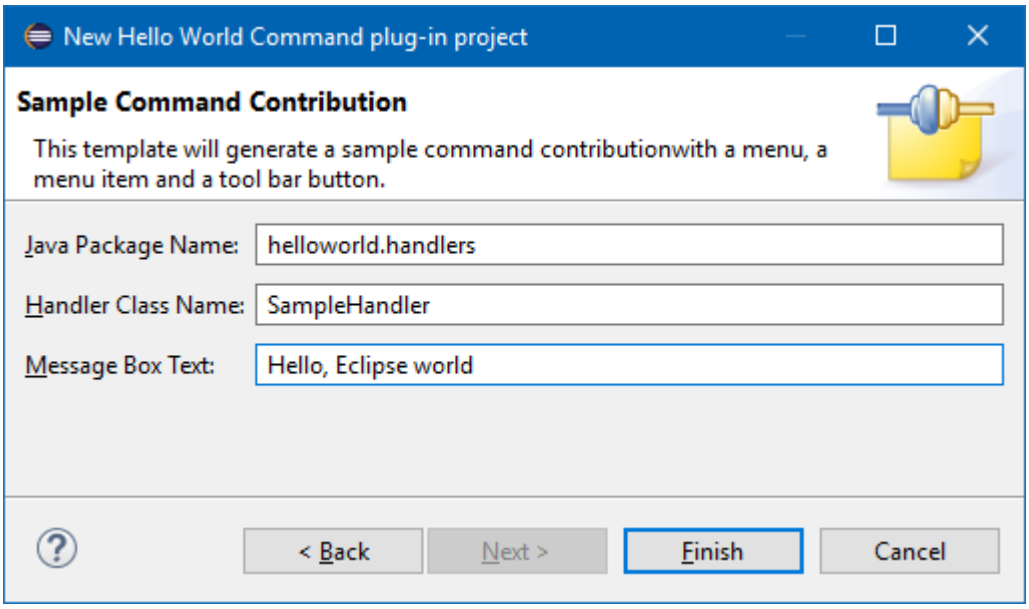
La versión será *1.0.0.qualifier* por defecto. Puede dejar esto como está, pero es mejor cambiarlo por algo significativo. El wiki de eclipse recomienda una sintaxis como *vYYYYMMDD* (año, mes, día).



En la página *Plantillas* , puede elegir crear su complemento desde cualquier plantilla seleccionándolo y haciendo clic en *Siguiente>* . Alternativamente, puede combinar estas plantillas seleccionando el *asistente de complemento personalizado* , o para crear un nuevo complemento sin una plantilla, deseleccionando la casilla de verificación frente a *Crear un complemento usando una de las plantillas* .

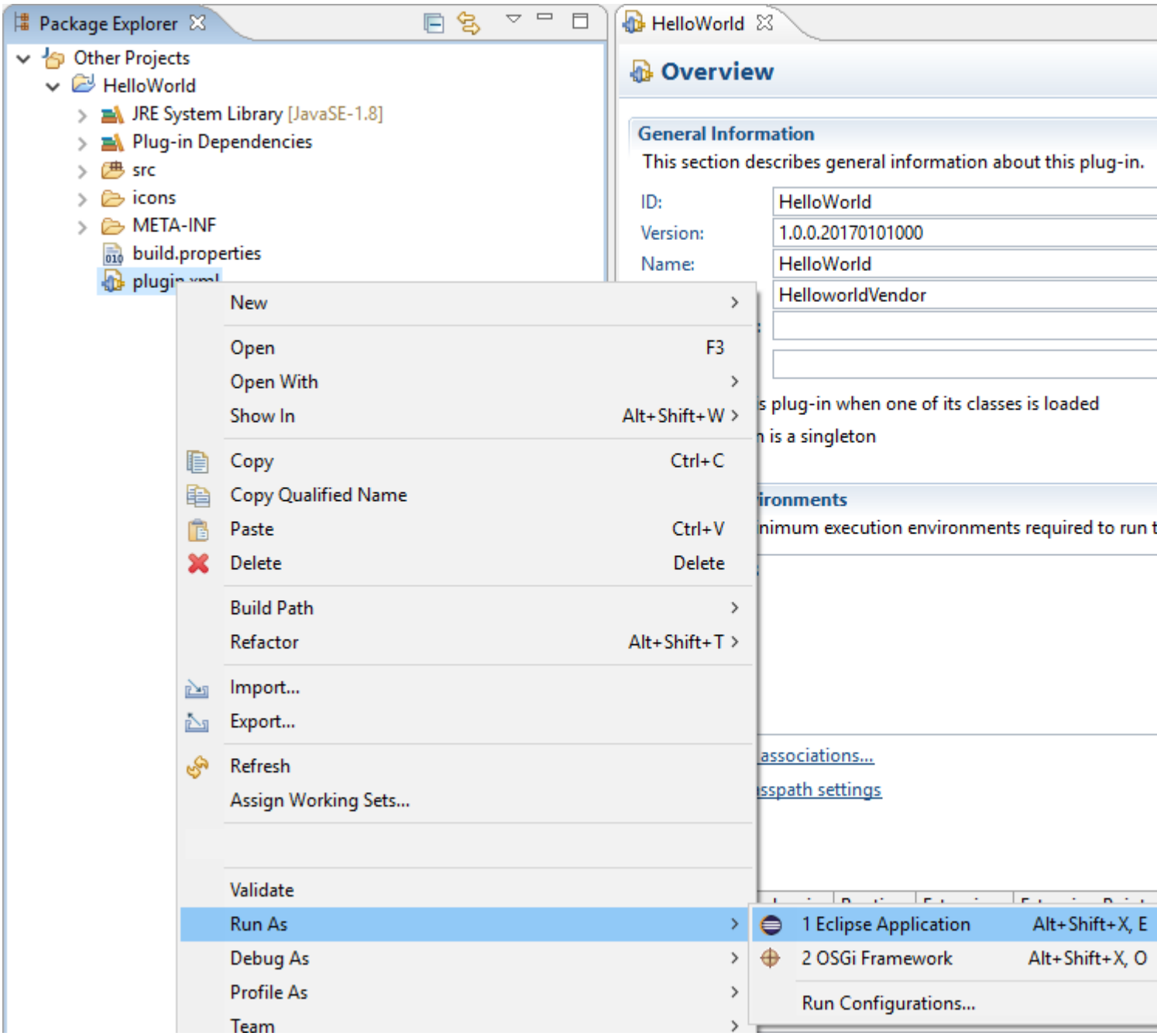


Para la plantilla *Hello, World Command* , hay configuraciones adicionales: el nombre del paquete, el nombre de la clase del manejador y el texto del cuadro de mensaje.



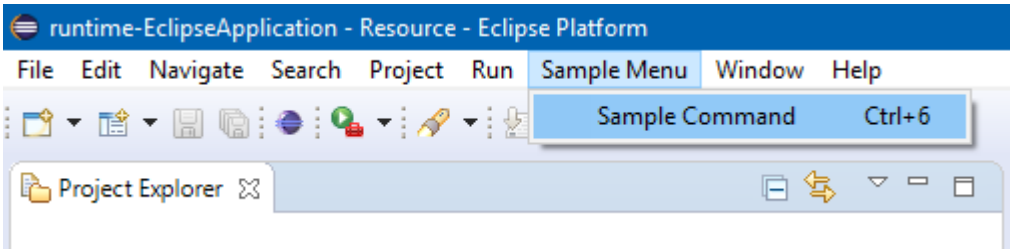
Cuando se crea el complemento, puede ejecutarlo haciendo clic con el botón derecho en el *plugin.xml* ➜ *Ejecutar como* cl *Aplicación Eclipse*

Esto lanzará una nueva instancia de Eclipse (con su propio espacio de trabajo) que tendrá su complemento cargado.



Este complemento de *Hello World* habrá realizado 3 contribuciones a la GUI de Eclipse:

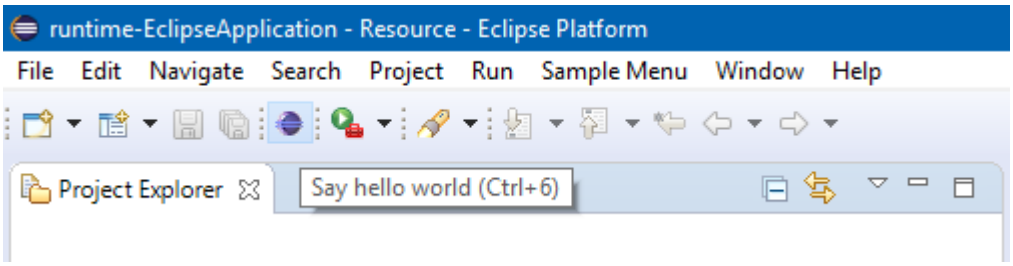
1. Un menú de muestra (con comando de muestra):



Plugin.xml:

```
<extension
      point="org.eclipse.ui.menus">
   <menuContribution
         locationURI="menu:org.eclipse.ui.main.menu?after=additions">
      <menu
            label="Sample Menu"
            mnemonic="M"
            id="HelloWorld.menus.sampleMenu">
         <command
               commandId="HelloWorld.commands.sampleCommand"
               mnemonic="S"
               id="HelloWorld.menus.sampleCommand">
         </command>
      </menu>
   </menuContribution>
</extension>
```

2. Un icono de la barra de herramientas:



Plugin.xml:

```
<extension
      point="org.eclipse.ui.menus">
   <menuContribution
         locationURI="toolbar:org.eclipse.ui.main.toolbar?after=additions">
      <toolbar
            id="HelloWorld.toolbars.sampleToolbar">
         <command
               commandId="HelloWorld.commands.sampleCommand"
               icon="icons/sample.gif"
               tooltip="Say hello world"
               id="HelloWorld.toolbars.sampleCommand">
         </command>
      </toolbar>
   </menuContribution>
</extension>
```

3. Un atajo de tecla (Ctrl + 6)

Plugin.xml:

```
<extension
      point="org.eclipse.ui.bindings">
   <key
         commandId="HelloWorld.commands.sampleCommand"
         contextId="org.eclipse.ui.contexts.window"
         sequence="M1+6"
         schemeId="org.eclipse.ui.defaultAcceleratorConfiguration">
   </key>
```

```
</extension>
```

Al activar cualquiera de estos 3, la clase Handler se ejecutará:

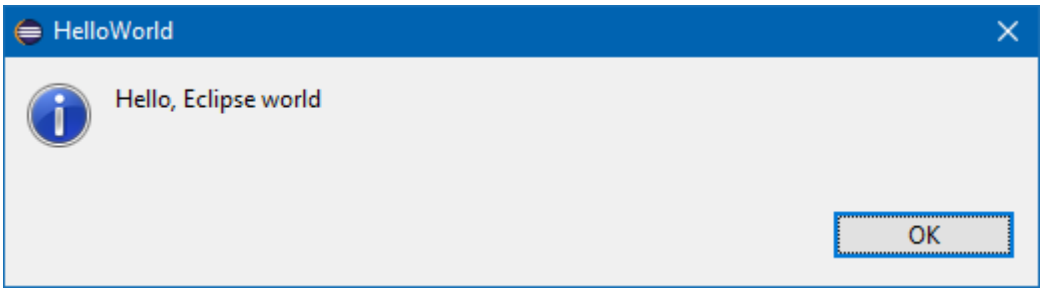
Plugin.xml:

```
<extension
      point="org.eclipse.ui.commands">
   <category
         name="Sample Category"
         id="HelloWorld.commands.category">
   </category>
   <command
         name="Sample Command"
         categoryId="HelloWorld.commands.category"
         id="HelloWorld.commands.sampleCommand">
   </command>
</extension>
<extension
      point="org.eclipse.ui.handlers">
   <handler
         commandId="HelloWorld.commands.sampleCommand"
         class="helloworld.handlers.SampleHandler">
   </handler>
</extension>
```

SampleHandler.java:

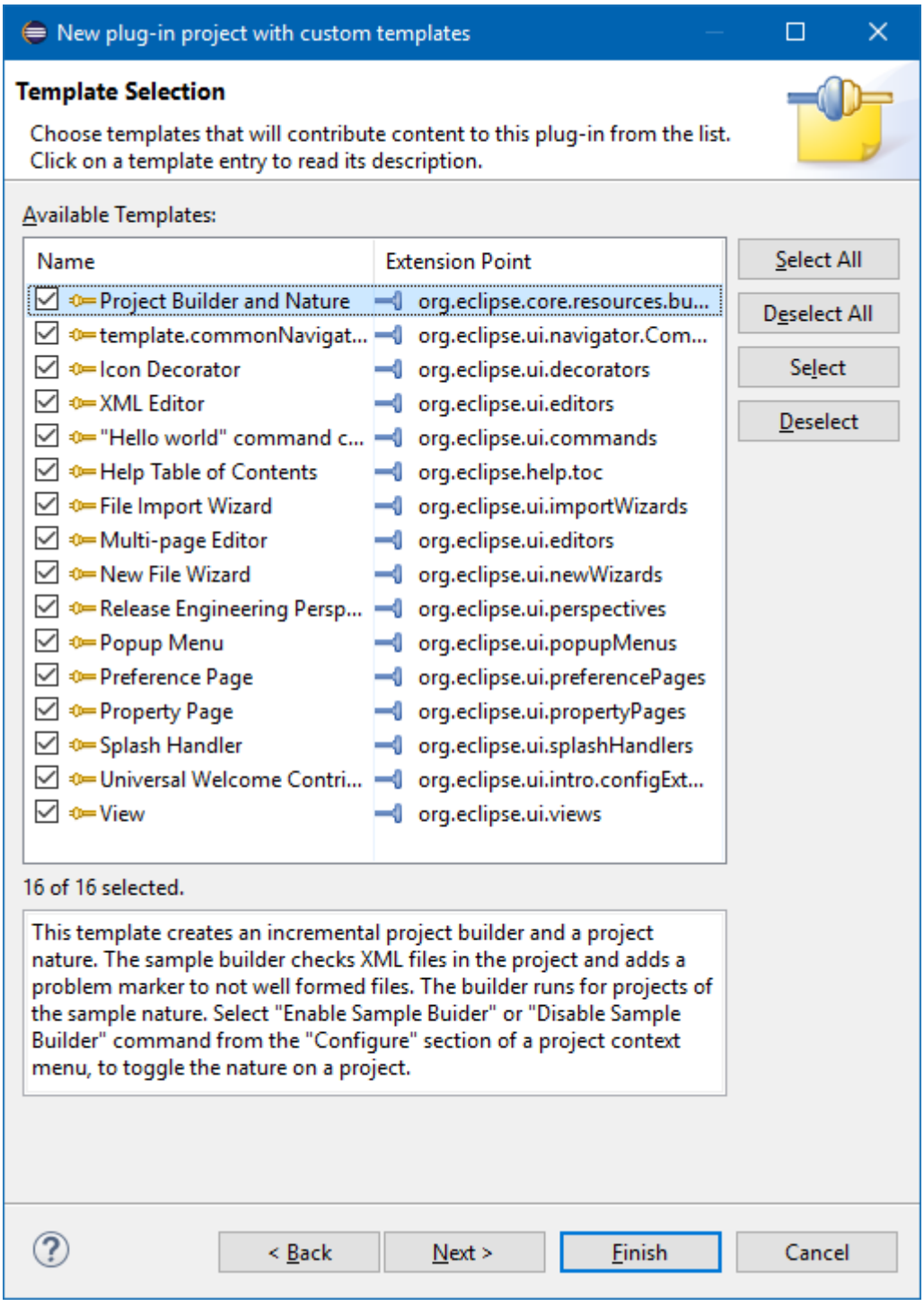
```
package helloworld.handlers;

import org.eclipse.core.commands.AbstractHandler;
import org.eclipse.core.commands.ExecutionEvent;
import org.eclipse.core.commands.ExecutionException;
import org.eclipse.ui.IWorkbenchWindow;
import org.eclipse.ui.handlers.HandlerUtil;
import org.eclipse.jface.dialogs.MessageDialog;

/**
 * Our sample handler extends AbstractHandler, an IHandler base class.
 * @see org.eclipse.core.commands.IHandler
 * @see org.eclipse.core.commands.AbstractHandler
 */
public class SampleHandler extends AbstractHandler {

    @Override
    public Object execute(ExecutionEvent event) throws ExecutionException {
        IWorkbenchWindow window = HandlerUtil.getActiveWorkbenchWindowChecked(event);
        MessageDialog.openInformation(
                window.getShell(),
                "HelloWorld",
                "Hello, Eclipse world");
        return null;
    }
}
```

Cuando se ejecute la clase Handler, MessageBox mostrará:



Esto es todo lo que hace el plug-in Hello World.

Si desea crear un complemento con más funcionalidad, podría haber elegido la plantilla que mejor se adapte a sus necesidades o crear un complemento a través del *asistente de complemento personalizado* para combinar estas plantillas:



Lea Empezando con eclipse-plugin en línea: https://riptutorial.com/es/eclipse-plugin/topic/4982/empezando-con-eclipse-plugin

Creditos

S. No	Capítulos	Contributors
1	Empezando con eclipse-plugin	Community, KompjoeFriek