

Instalación de Eclipse Galileo y depuradores compatibles con OpenMPI y OpenMP

Programación Distribuida y Paralela



Universidad de Granada



Departamento del LSI

Por:
Daniel Guerrero Martínez
Sergio Rodríguez Lumley

Índice de contenido

Introducción.....	3
1. Instalación de OpenMPI.....	4
2. Instalación de Eclipse Galileo.....	5
Añadir repositorios a Eclipse.....	5
3. (CDT) Obtener Plugin para Eclipse “C Development Tools”.....	8
4. Instalar Parallel Tools Platform (PTP).....	9
5. Resolución de errores.....	13
6. Configuración del entorno.....	14



Introducción

Vamos a instalar un IDE para C y C++ para poder depurar nuestros proyectos con seguridad. El software que instalaremos será el siguiente:

- OpenMPI (Implementación de la interfaz MPI)
- Eclipse Galileo (Versión 3.5.x)
- Plugin para Eclipse Galileo C Development Tools (CDT) 6.0.2
- Plugin para Eclipse Galileo Parallel Tools Platform (PTP) 3.0
- Depurador paralelo basado en gdm, SDM.

Requisitos previos:

- Tener java instalado en la máquina (Por ejemplo: sun-java6-jre y sun-java6-jdk)
- Compilador de C/C++, recomendamos el compilador de GNU GCC 4.4.1 o superior.

Aunque se indicará en cada caso, asumiremos que la instalación de cada elemento será desde Ubuntu (Linux/Unix), con lo que se dispone de un gestor de paquetes y de la herramienta “sudo” como superusuario.

1. Instalación de OpenMPI

OpenMPI es una implementación de la interfaz de paso de mensajes MPI. OpenMPI se caracteriza por su alta eficiencia y prestaciones para la ejecución en entornos distribuidos (clústers de ordenadores).

Se puede instalar descargando la última versión para tu distribución desde su página oficial:

<http://www.open-mpi.org/>

Nosotros vamos a explicar de forma sencilla los paquetes necesarios para su instalación en un sistema Linux/Unix basado en Debian (En concreto, Ubuntu). Los paquetes necesarios son:

openmpi-bin: Programa de ejecución de códigos paralelos (mpirun).

Instala: `openmpi-common libopenmpi1.3`

openssh-client, openssh-server: Programa de comunicación (rutinas de control y presentación) entre procesos.

libopenmpi-dbg: Generador de información de depuración para MPI

libopenmpi-dev: Necesario para el desarrollo de programas basados en MPI (mpicc...)

Comando rápido:

```
sudo apt-get install openmpi-bin openmpi-common openssh-client openssh-server libopenmpi1.3  
libopenmpi-dbg libopenmpi-dev
```

Nota: En sistemas como Ubuntu, el gestor de paquetes se encargará de resolver las dependencias de los paquetes seleccionados, evitando su reinstalación e instalando (si fuera necesario) un compilador de C y C++.

2. Instalación de Eclipse Galileo

Eclipse Galileo es un IDE de código abierto basado en Java. Su desarrollo es abierto y llevada a cabo por la comunidad de Eclipse. En nuestro caso, nos interesa su desarrollo para C/C++ con OpenMPI y OpenMP, además de sus plugins para desarrollo de programas paralelos.

El programa se puede descargar directamente desde su página inicial:

<http://www.eclipse.org>

Existen varios paquetes de eclipse para descargar. Puesto que solo nos interesa su funcionalidad con C y C++, descargaremos el paquete de C y C++. Sin embargo cualquiera nos vale, puesto que instalaremos luego explicaremos cómo instalar su funcionalidad desde plugins.

	Eclipse IDE for Java EE Developers (189 MB) Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn and others. More... Downloads: 126,877	Windows 32bit Mac Carbon 32bit Mac Cocoa 32bit 64bit Linux 32bit 64bit
	Eclipse IDE for Java Developers (92 MB) The essential tools for any Java developer, including a Java IDE, a CVS client, XML Editor and Mylyn. More... Downloads: 82,204	Windows 32bit Mac Carbon 32bit Mac Cocoa 32bit 64bit Linux 32bit 64bit
	Eclipse IDE for C/C++ Developers (79 MB) An IDE for C/C++ developers with Mylyn integration. More... Downloads: 32,503	Windows 32bit Mac Carbon 32bit Mac Cocoa 32bit 64bit Linux 32bit 64bit

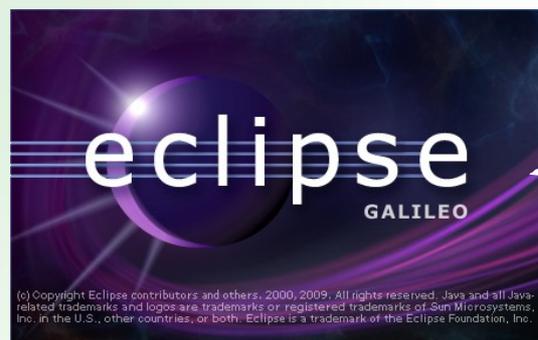
Su instalación desde los repositorios de Ubuntu (Linux/Unix) es tan sencillo como:

```
sudo apt-get install eclipse
```

Añadir repositorios a Eclipse

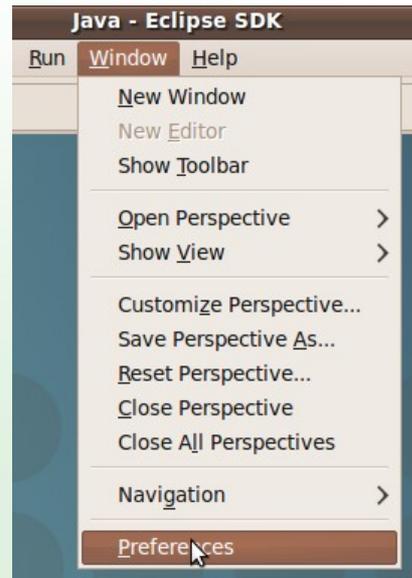
Puesto que es algo que haremos a menudo, vamos a explicar cómo añadir los repositorios principales (oficiales) de Eclipse a su lista de software.

Comenzamos por abrir el programa, debería aparecer una imagen de inicio similar a la mostrada. Nos dará a elegir un espacio de trabajo, seleccionaremos el de “por defecto” puesto que para lo que necesitamos no lo utilizaremos.



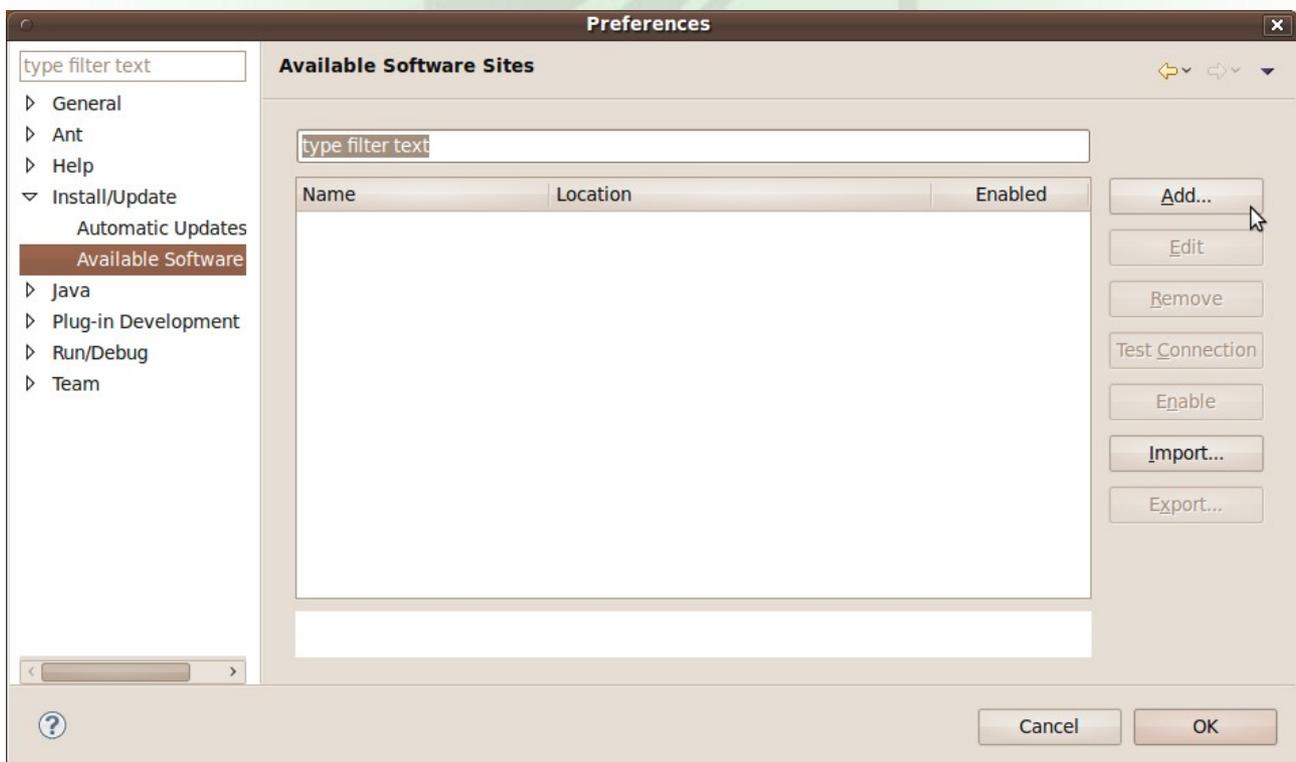
Una vez abierto el programa, seleccionamos abrir el cuadro de preferencias en el menú Ventana → Preferencias.

Window → Preferences



1. Ilustración de Eclipse para ubicación de preferencias

En la ventana de preferencias, buscamos en la barra lateral izquierda Instalación/Actualización (Install/Update) → Software Disponible (Available Software).



Pulsamos el botón de añadir (Add) y añadimos un nombre y la dirección (esta puede ser a través de internet o desde un fichero). En nuestro caso añadiremos:

Nombre (Name):	Repositorio oficial de Eclipse Galileo
Localización (Location):	http://download.eclipse.org/releases/galileo/

Como muestra la siguiente figura:

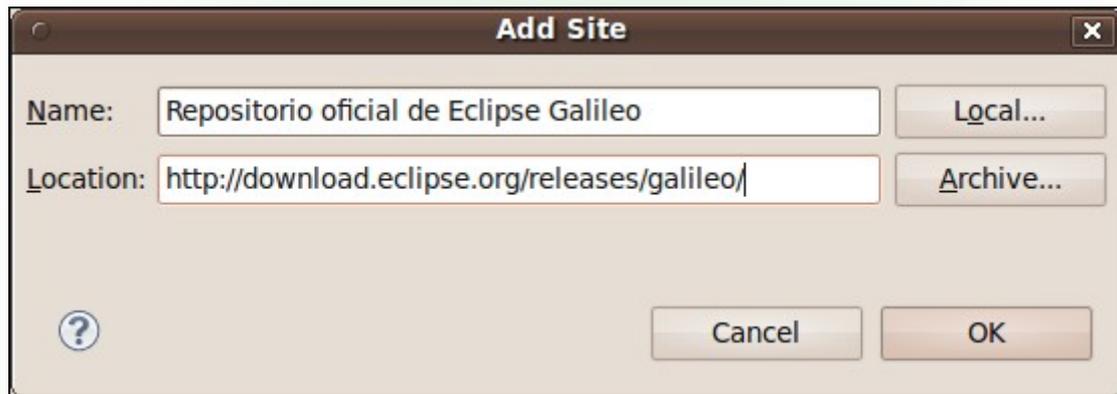


Figura 3. Añadir repositorio a Eclipse

Una vez añadido, pulsamos el botón de Comprobar conexión (Test Connection). Esto nos alertará en caso de algún problema o si todo ha ido bien.

3. (CDT) Obtener Plugin para Eclipse “C Development Tools”

Este plugin es necesario para el desarrollo de C y C++. Aun si hemos descargado el paquete de Eclipse para C y C++, es recomendable instalar este plugin, puesto que así nos aseguramos de utilizar las componentes más actuales. En todo caso, si instalaste el paquete de Eclipse para C y C++ y no quieres instalar este paquete, puedes saltar al siguiente paso, en caso de error, se puede volver aquí en todo momento.

Existen dos formas de obtener este plugin, añadiendo el repositorio online de CDT, o bien descargando el archivo y añadiéndolo localmente a nuestro IDE (Recomendado).

- **Añadir el repositorio online:**

Tal como se explica en este documento, añadimos el repositorio siguiente:

<http://download.eclipse.org/tools/cdt/releases/galileo>

- **Descargando el fichero desde su web oficial (Recomendado):**

- Podemos o bien seleccionar el paquete más nuevo desde el siguiente enlace:
<http://download.eclipse.org/tools/cdt/builds/>
- O seleccionar el paquete recomendado (El primero que aparezca desde arriba) para esta instalación:
<http://download.eclipse.org/tools/cdt/builds/6.0.2/index.html>

Una vez obtenido el paquete, lo añadiremos al repositorio de nuestro IDE de igual forma que hicimos con los repositorios online, solo que seleccionando el fichero descargado desde su ubicación.

Una vez lo tenemos en nuestros repositorios, pasamos a instalar el plugin de desarrollo paralelo (PTP).

4. Instalar Parallel Tools Platform (PTP)

Vamos a efectuar la instalación para ejecutar todos los procesos de forma local (Aunque este plugin es capaz de administrar las hebras remotas, el proceso es mucho más complejo).

1 Instalar el sistema de exploración remoto (Remote System Explorer (RSE) 3.1.1) desde el repositorio oficial de Eclipse Galileo.

1.1 Abre “Ayuda→ Instalar nuevo software”.
“Help → Install New Software”

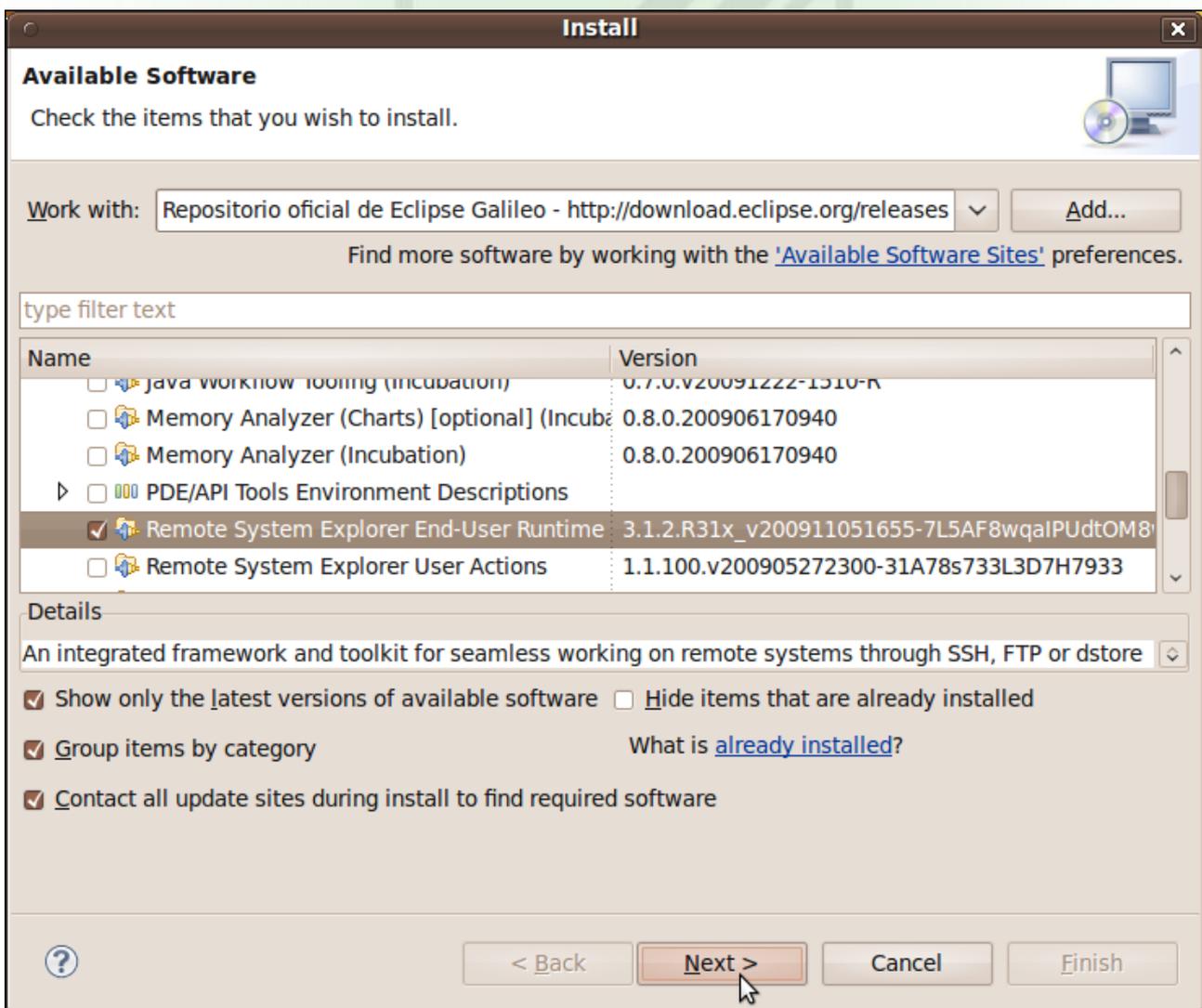
1.2 En la lista de despliegue de “trabajar con” (Work With) seleccione el servidor oficial de Eclipse Galileo (El que añadimos inicialmente).

1.3 Abra “Herramientas de Propósito General”.
“General Purpose Tools”

1.4 Seleccione “Sistema de Exploración Remoto para usuario final” .
“Remote System Explorer End-User Runtime”

1.5 Termine la instalación siguiendo el instalador.

1.5.1 No debería ser necesario reiniciar Eclipse.



Programación Distribuida y Paralela *Figura 4.*

2 Instalamos las herramientas de desarrollo de C/C++ (CDT) 6.0.2 (Desde el archivo descargado).

- 2.1 Abrimos en Eclipse el cuadro de “Ayuda → Instalar Nuevo Software”.
“*Help → Install New Software*”
- 2.2 Hacemos click en “Añadir” (“*Add*”).
- 2.3 En el diálogo de selección de la ruta, seleccionamos el fichero descargado, que tendrá la forma de “cdt-master-xxx.zip”.
- 2.4 Seleccionamos las características que deseamos instalar, o bien seleccionamos la categoría principal para instalarlas todas. En nuestro caso, seleccionamos todas excepto “Mylyn Bridge”, que es una característica con dependencias no cumplidas que no necesitaremos.
- 2.5 Termine la instalación siguiendo la ayuda de instalación.
 - 2.5.1 No debería ser necesario reiniciar Eclipse.

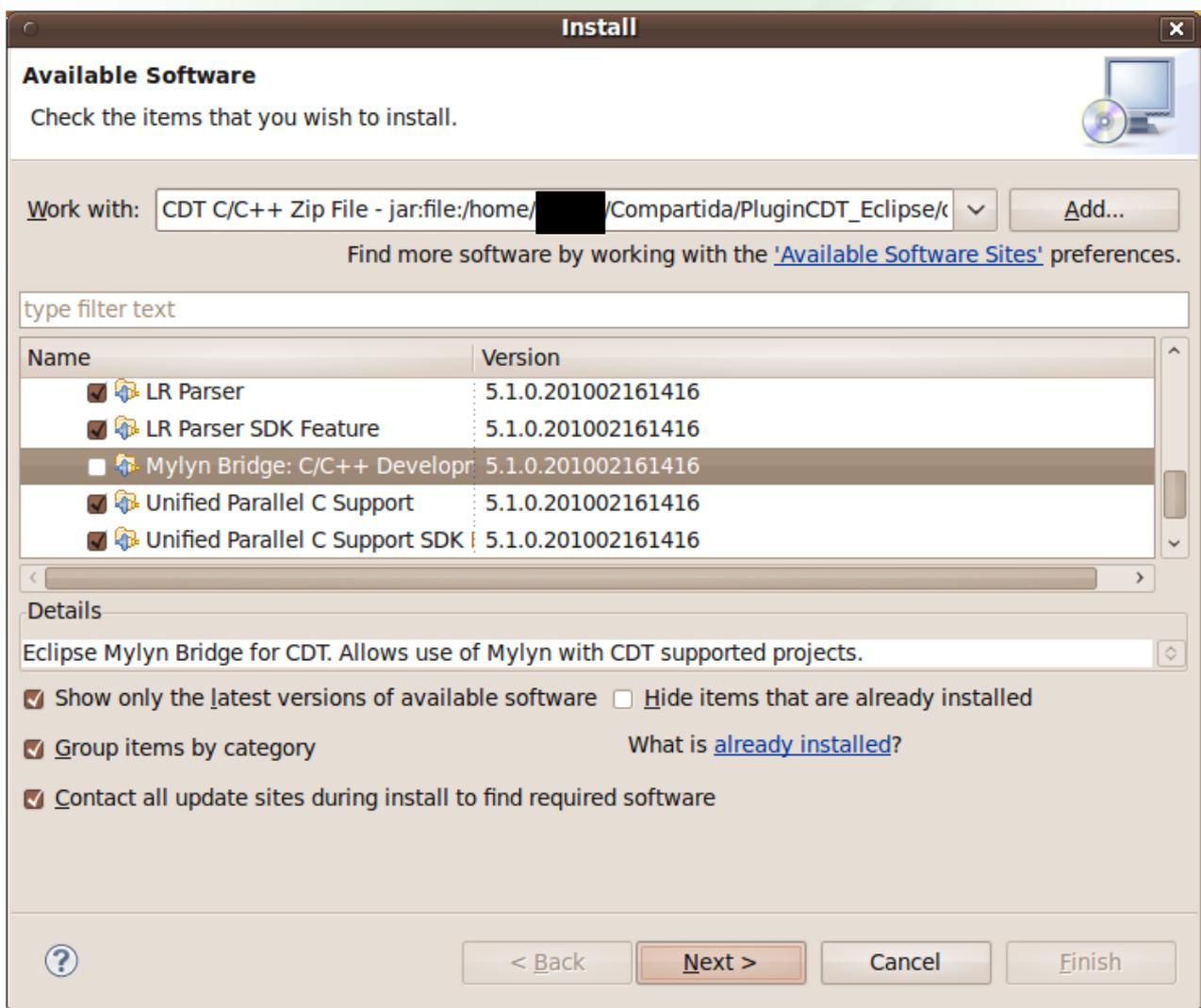


Figura 5.

3 Instalamos PTP 3.0

- 3.1 Abrimos “Ayuda → Instalar Nuevo Software”
“Help → Install New Software”
- 3.2 Hacemos click en “Añadir (Add)” para añadir una nueva dirección de actualización.
- 3.3 Añadimos la localización de la página de actualización de PTP:
<http://download.eclipse.org/tools/ptp/releases/galileo> (Con cualquier nombre)
- 3.4 Seleccionamos las categorías “Parallel Tools Platform” y “Remote Development Tools”.
- 3.5 Seguimos los pasos de instalación.
 - 3.5.1 Debemos reiniciar Eclipse tras la instalación.

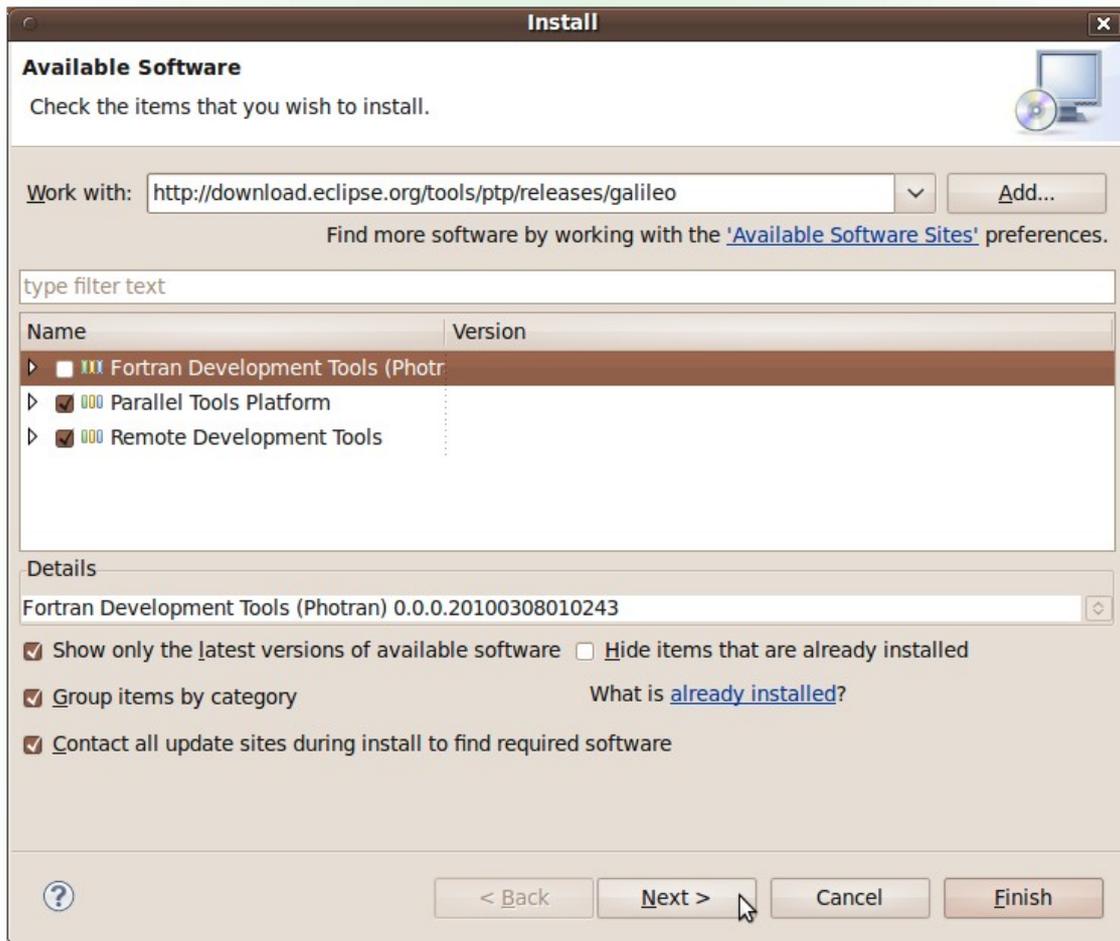


Figura 6.

4 Instalamos el depurador SDM (Scalable Debug Manager).

4.1 Debemos buscar la carpeta de plugins de Eclipse, esta por defecto se encuentra en la carpeta personal de usuario ('home/<usuario>/eclipse/'). Una vez encontrada la carpeta, será necesario buscar dentro la carpeta de plugins. Nos situamos dentro de ella.

4.2 Deberemos buscar una carpeta con el nombre de “**org.eclipse.ptp.<os>.<arch>_3.0.0.AAAAMMDDHHmm**”. El significado de todo esto es el siguiente:

<os> Sistema operativo, en nuestro caso *linux*.
<arch> Arquitectura, *x86* para 32 bits, *x86_x64* para 64 bits y *ppc* para PowerPC.
3.0.0 Es la versión, (No importa si es superior).
AAAAMMDDHHmm Es una serie de números que indican el momento de su compilación, significa A (Año) M (Mes) D (Día) H (Hora) m (minuto).

4.3 Una vez encontremos la carpeta, nos situamos dentro de ella y ejecutamos el comando “sh BUILD”.

sh BUILD

Esto instalará los ejecutables. Comprueba que en la carpeta “bin” se hayan creado los ejecutables (En este caso debería aparecer sdm). Deberemos recordar esta ubicación puesto que se la tenemos que indicar a Eclipse para que coja el ejecutable.

5. Resolución de errores

- *“Cuando llego al paso de instalación de PTP aparecen conflictos que no se pueden resolver”*

Esto puede suceder cuando no se encuentra disponible el soporte correcto para C/C++, aunque hayas descargado la versión explícita de Eclipse para C/C++, es posible que no tenga instalados ciertos componentes imprescindibles para el PTP.

Vuelve al paso de obtener el plugin CDT para Eclipse, si elegiste instalar desde repositorios, descarga el fichero de instalación desde su web oficial y vuelve a instalar las componentes necesarias.



6. Configuración del entorno

La forma más sencilla de configurar el entorno y probar dicha configuración es crear y ejecutar una aplicación de MPI.

1. Abrimos el IDE Eclipse (Tal vez necesitemos utilizar el parámetro `-clean` para forzarlo a reconocer las nuevas funcionalidades y plug-ins).
2. Pasamos a la perspectiva C/C++ (Ventana → Abrir Perspectiva → Otros...)
“C/C++ Perspective (Window → Open Perspective → Other...)”

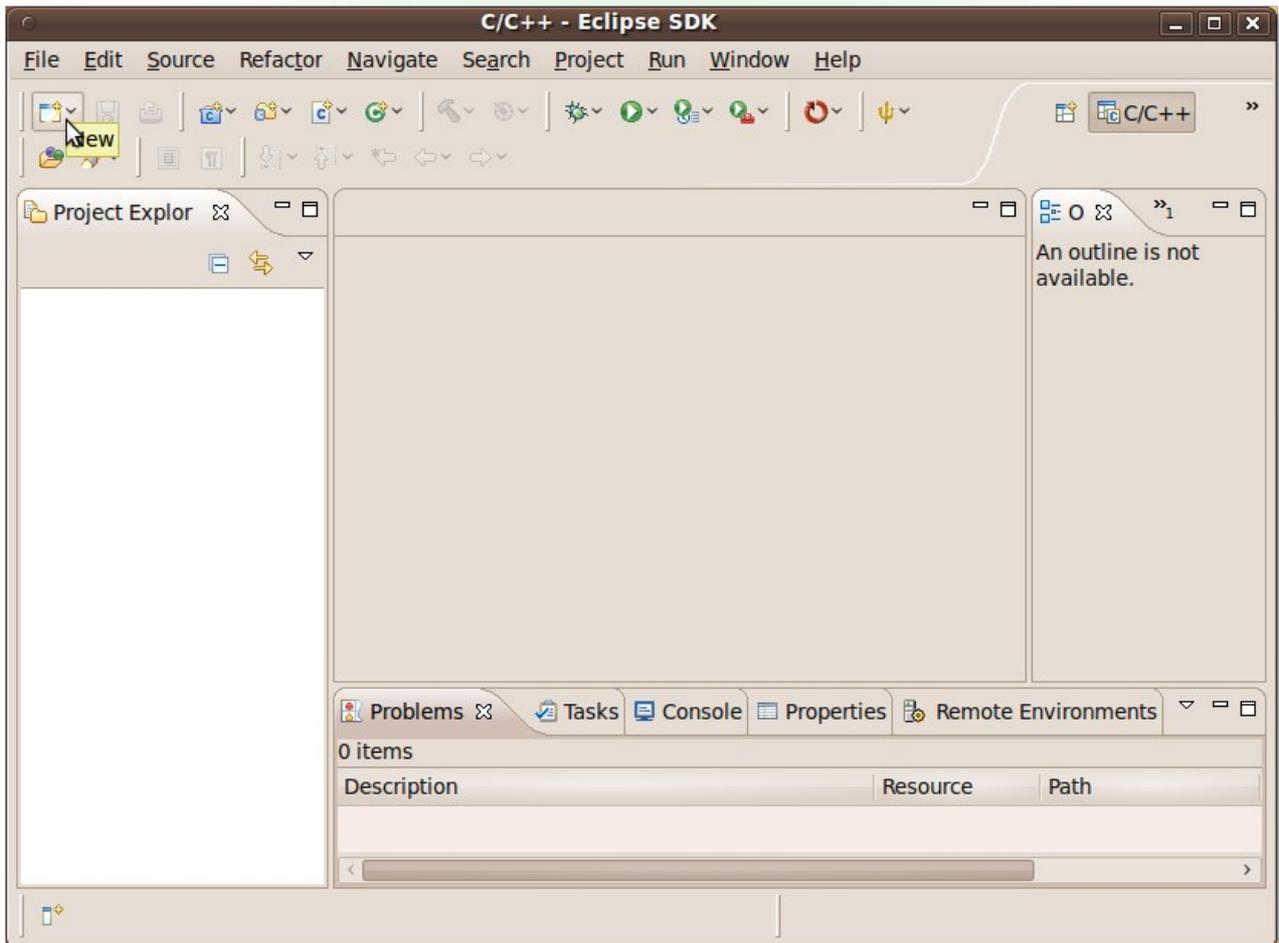
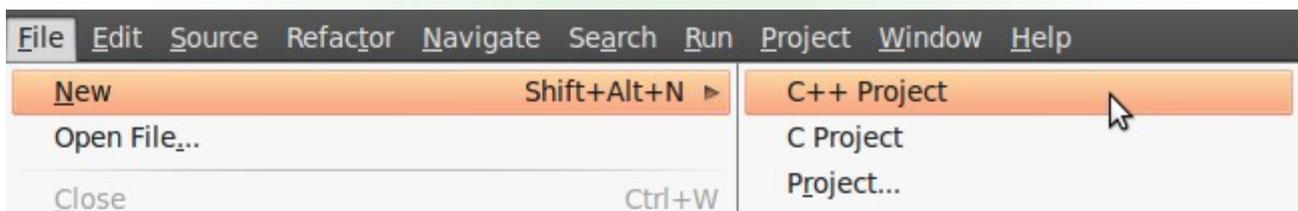


Figura 7. Muestra de la vista en perspectiva C/C++.

3. Creamos un pequeño proyecto de MPI C (Archivo → Nuevo → Proyecto C++)
“(File → New → C++ Project)”



4. Seleccionamos “MPI pi C++ Project”, nos saldrá un cuadro pidiéndonos que configuremos las preferencias de MPI. Aceptamos y las configuramos. Dejaremos todo por defecto e incluiremos la ruta de inclusión de cabeceras de OpenMPI, que será la implementación que utilizaremos. Por defecto será “/usr/include/openmpi”.

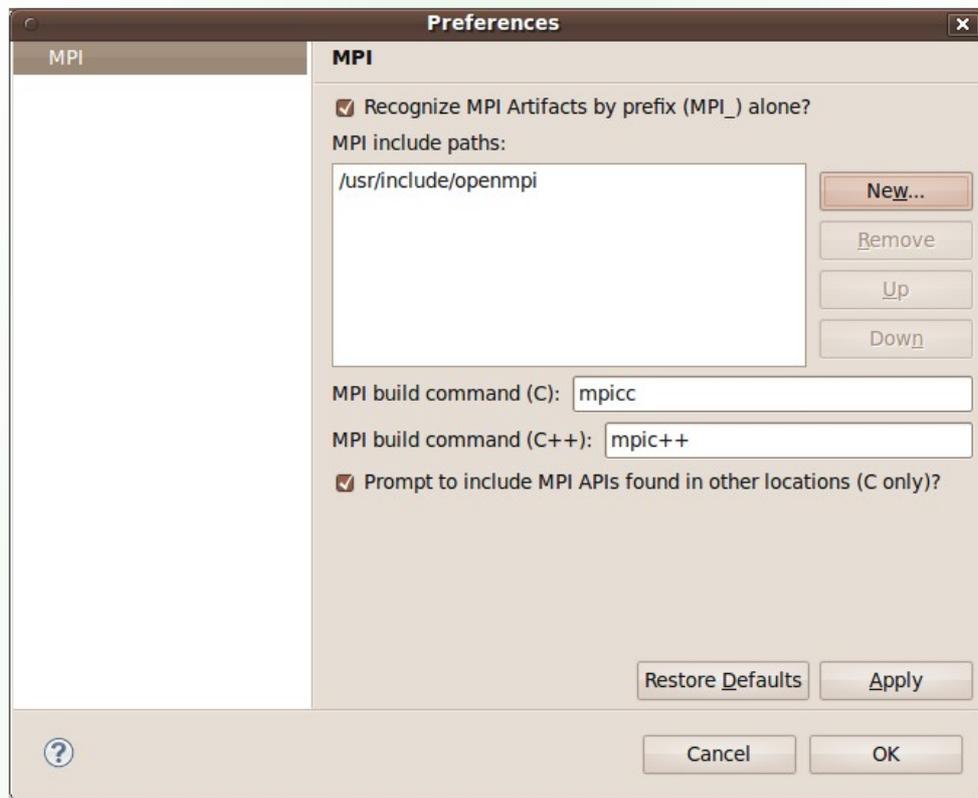


Figura 8. Muestra del resultado de las preferencias de MPI.

5. Una vez configurado, pasamos a crear el proyecto.

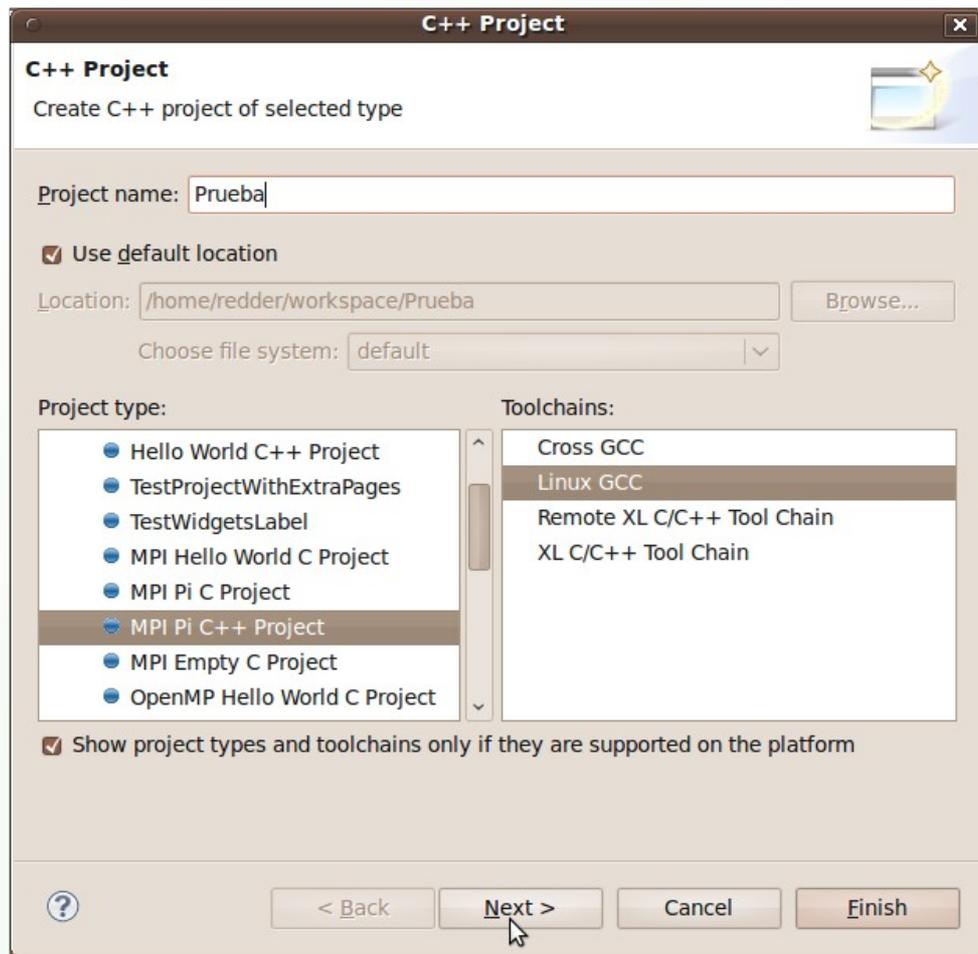


Figura 9. Crear proyecto MPI en C++.

6. Abrimos la perspectiva de ejecución de PTP (PTP Runtime Perspective) (Ventana (Window) → Abrir perspectiva (Open Perspective) → Otros (Other) → “Parallel Runtime”)

7. Cree un gestor de recursos (Resource Manager), con OpenMPI (Dejaremos todos los parámetros por defecto) y lo comenzamos. Debería ver una máquina y uno o más nodos en el cuadro de “Vista de máquinas” (Machine View).

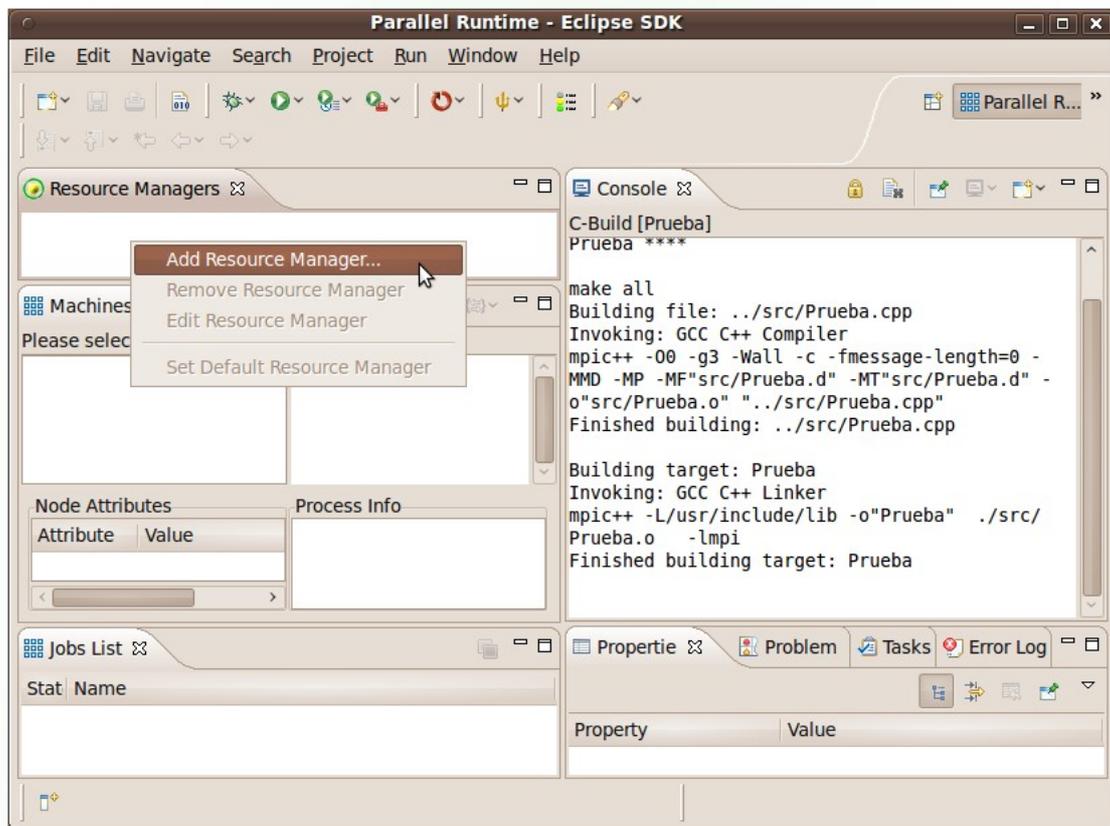


Figura 10. Añadir Gestor de Recursos.

8. Creamos una configuración de ejecución (Launch Configuration) en Ejecutar (Run) → Configuraciones de Ejecución (Run Configurations...). Seleccionamos la configuración de Aplicación Paralela (Parallel Application) y pulsamos sobre el botón de “Nuevo” (Arriba a la izquierda).

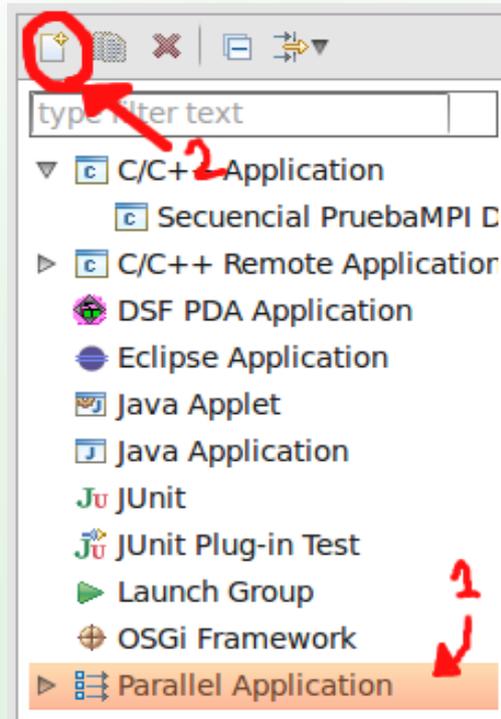


Figura 11. Crear una configuración de ejecución paralela.

1. Rellenamos la información para cada caso particular, en nuestro caso, dejamos todo por defecto, excepto:
 1. En la pestaña de Recursos (Resources) seleccionamos el número de procesos.
 2. En la pestaña de aplicación (Application) seleccionamos nuestro proyecto para “Proyecto Paralelo” (Parallel Project) y el ejecutable para la aplicación.
 3. En la pestaña de depuración, buscamos el depurador **sdm** que construimos anteriormente en la “ruta para el ejecutable de depuración” (“Path to debugger executable”).
9. Ejecutamos la aplicación dándole al botón de “Play”.

Más detalles sobre cómo utilizar PTP en Ayuda (Help) → Contenidos de Ayuda (Help Contents), y click en “Guía de Usuario de PTP”(“Parallel Tools Platform User Guide”).

También podemos encontrar la ayuda de PTP [online \(en inglés\)](#).