

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



adictos al trabajo

Autentia
Business solutions
patrocinado por
datos

E-mail: Contraseña:

Deseo registrarme

He olvidado mis datos de acceso

[Inicio](#) [Quiénes somos](#) [Tutoriales](#) [Formación](#) [Comparador de salarios](#) [Nuestro libro](#) [Charlas](#) [Más](#)

Estás en:

[Inicio](#) [Tutoriales](#) EGit un plugin de Eclipse para el sistema de control de versiones distribui...



DESARROLLADO POR:
Alejandro Pérez García

Alejandro es socio fundador de Autentia y nuestro experto en J2EE, Linux y optimización de aplicaciones empresariales.

Ingeniero en Informática y Certified ScrumMaster

Si te gusta lo que ves, puedes contratarle para darte ayuda con soporte experto, impartir **cursos presenciales** en tu empresa o para que **realicemos tus proyectos como factoría** (Madrid). Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Catálogo de servicios
Autentia



Últimas Noticias

Problemas de aspecto en AdictosAITrabajo.com: Refresco de la hoja de estilo.

Comentando el libro: Lucro sucio de Joseph Heath

Nuevo formato de tutoriales podcast con bolígrafo LiveScribe

Estrenamos nuevo aspecto en AdictosAITrabajo

X Charla Autentia - Primeros pasos con Talend

Histórico de NOTICIAS

Últimos Tutoriales del Autor



Fecha de publicación del tutorial: 2010-07-27



Share |

[Regístrate para votar](#)

EGit un plugin de Eclipse para el sistema de control de versiones distribuido Git

Creación: 18-07-2010

Índice de contenidos

1. Introducción
2. Entorno
3. Instalación de EGit
4. Conclusiones
5. Sobre el autor

1. Introducción a Git

En el tutorial <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=git> ya vimos como trabajar en línea de comandos con el sistema de control de versiones distribuido Git. En este tutorial vamos a ver EGit, un plugin de Eclipse para integrar Git en el IDE de forma que ciertas operaciones sean más cómodas.



Git y cómo trabajar con un repositorio de código

Desplegando en Tomcat un proyecto web con Hudson.

Git y cómo trabajar con un repositorio de código distribuido

Introducción a Activiti

Últimas ofertas de empleo

2010-06-25
T. Información -
Analista /
Programador -
BARCELONA.

2. Entorno

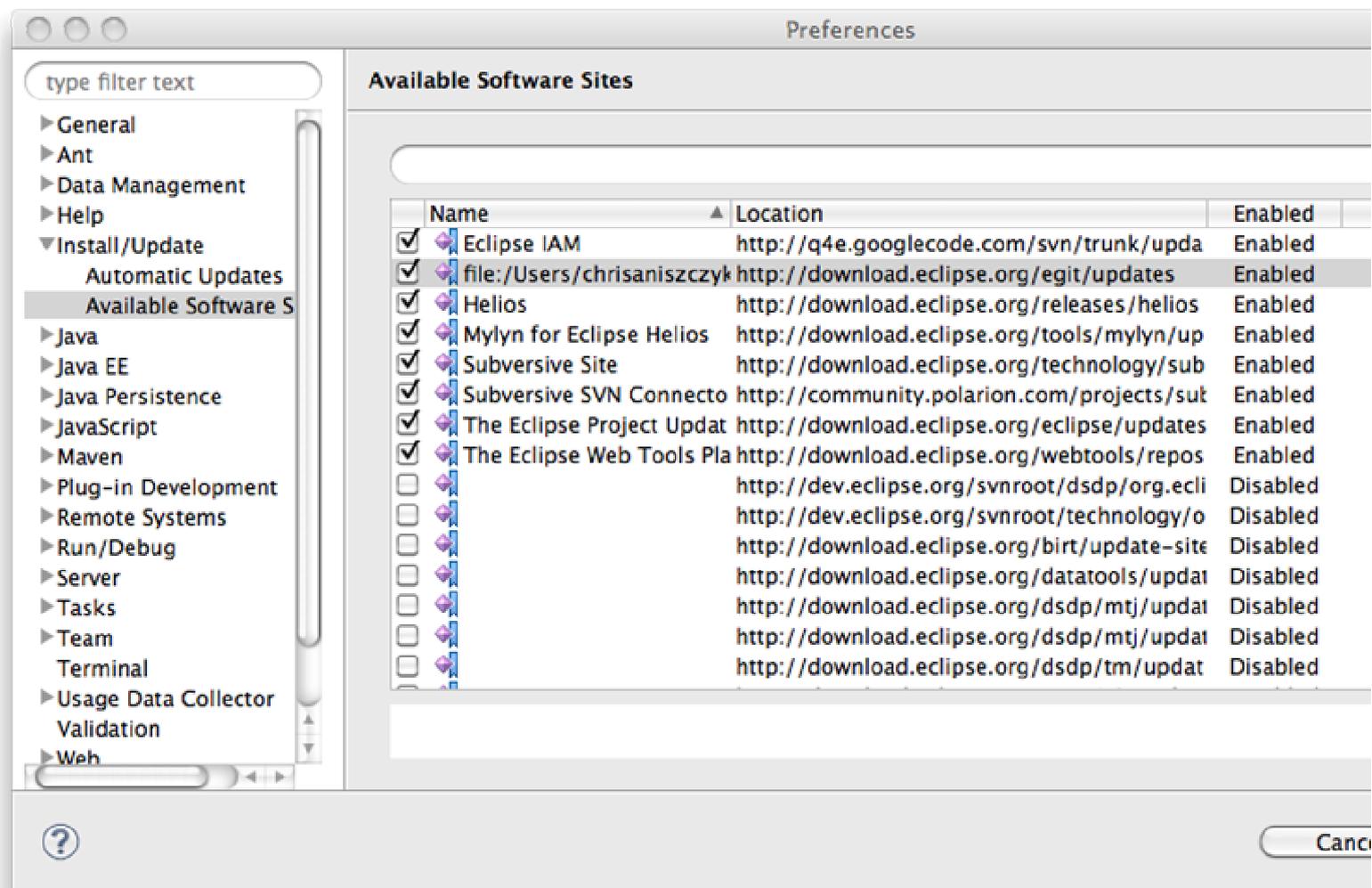
El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 17' (2.93 GHz Intel Core 2 Duo, 4GB DDR3 SDRAM, 128GB Solid State Drive).
- NVIDIA GeForce 9400M + 9600M GT with 512MB
- Sistema Operativo: Mac OS X Snow Leopard 10.6.3
- Eclipse Helios

3. Instalación de EGit

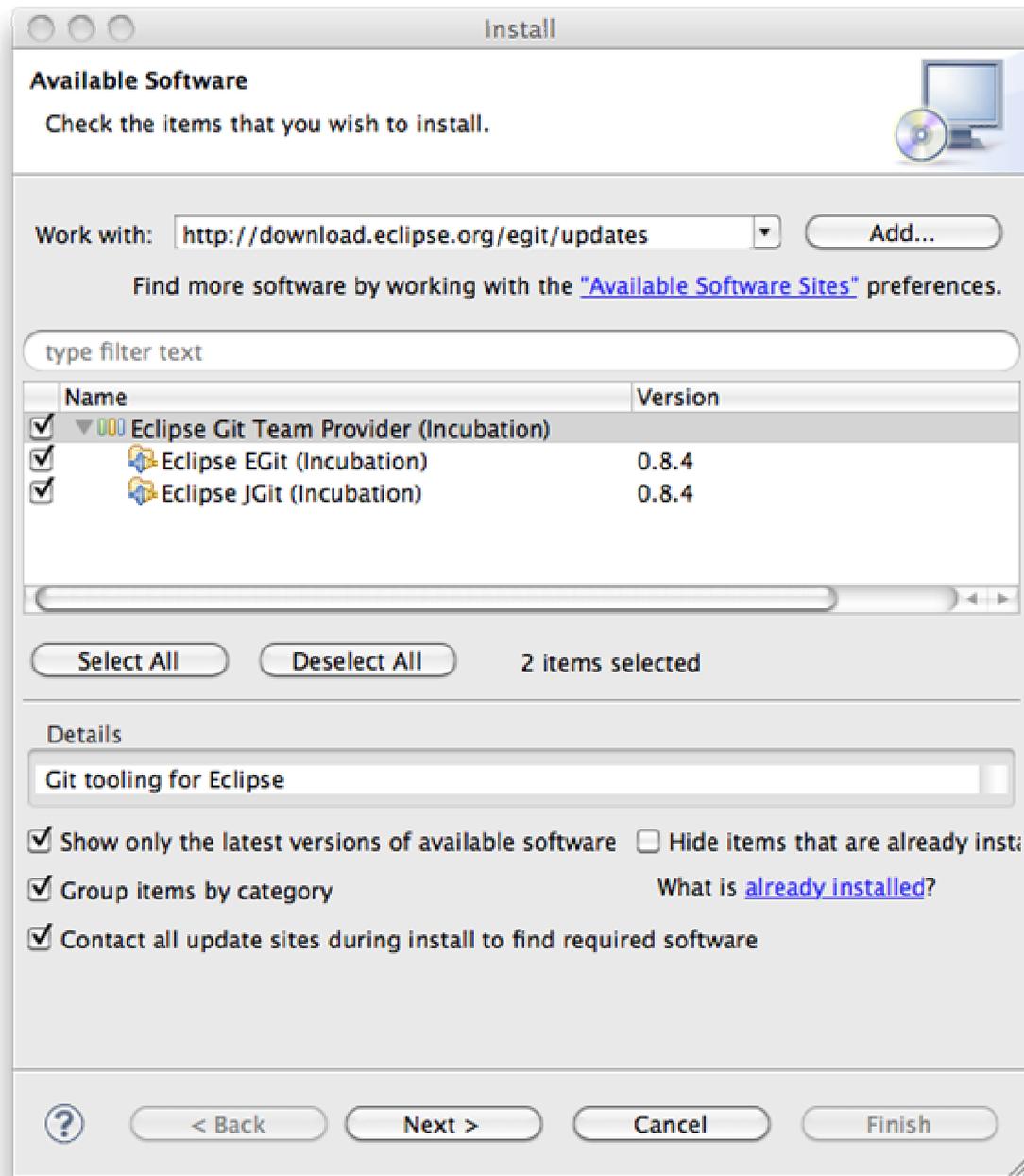
La instalación la vamos a hacer directamente desde el propio Eclipse Helios.

Seleccionamos *Help --> Install New Software...*, y en esta pantalla pinchamos sobre "Available Software Sites" (se encuentra casi arriba a la derecha). Se nos presenta una nueva ventana con una lista de URLs, las seleccionadas son las que Eclipse Helios tiene en cuenta a la hora de instalar o actualizar el sistema. Buscamos una que tiene como *Location* "<http://download.eclipse.org/egit/updates>" y la seleccionamos (Truco: encima de la lista hay un campo de entrada de texto para filtrar el contenido de la lista, si en este campo escribimos "egit" nos resultará más fácil encontrar lo que buscamos).

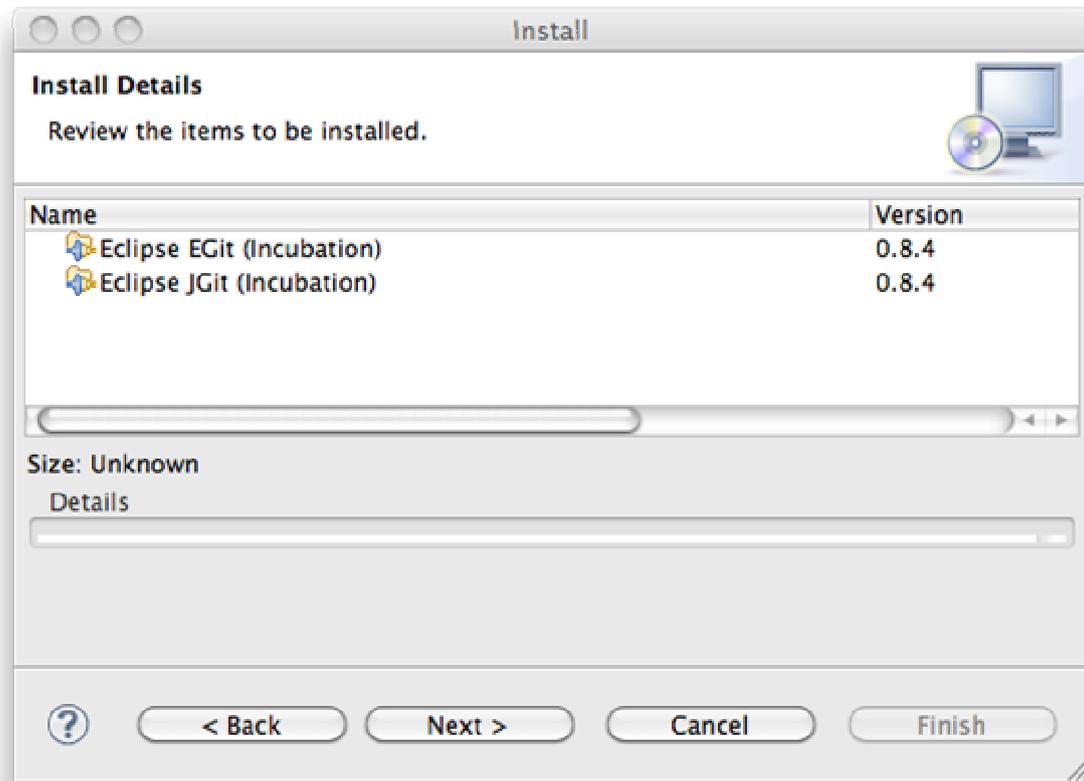


Damos al botón *OK*, y volveremos a la pantalla anterior.

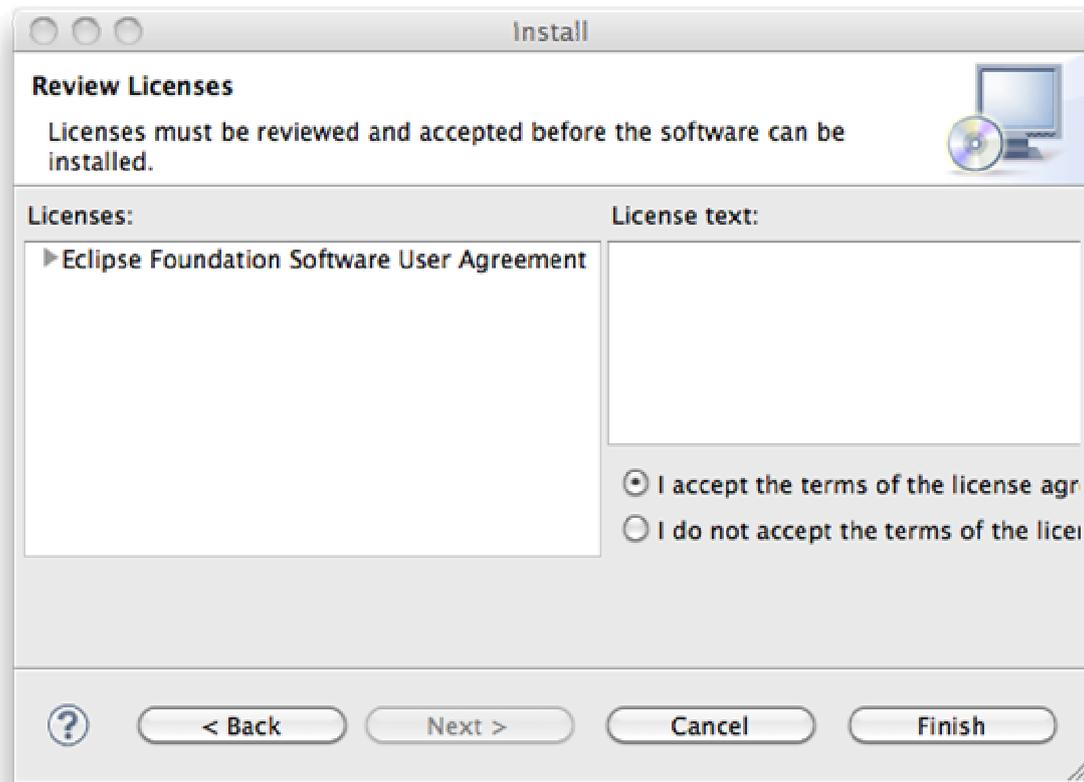
Ahora en la lista desplegable "Work with:" seleccionamos la URL que acabamos de activar, y nos saldrá la lista de software disponible en esta URL.



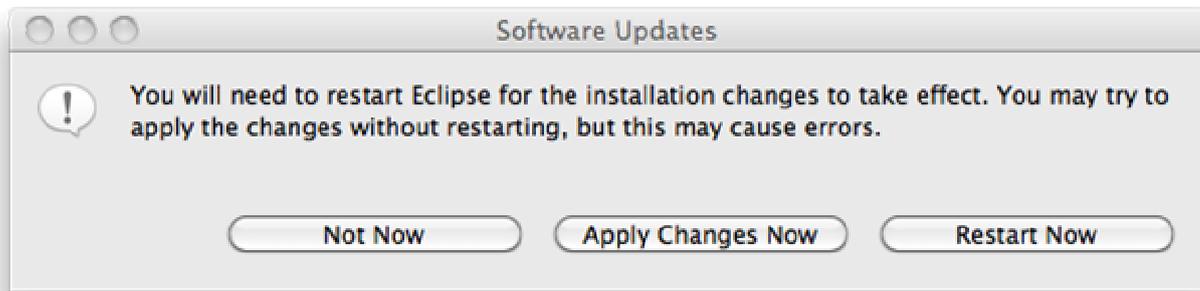
Marcamos todas las opciones (EGit es el plugin de Eclipse y JGit es la librería Java que se encarga de hablar con Git). Damos al botón de *Next >*



Damos otra vez a *Next >*

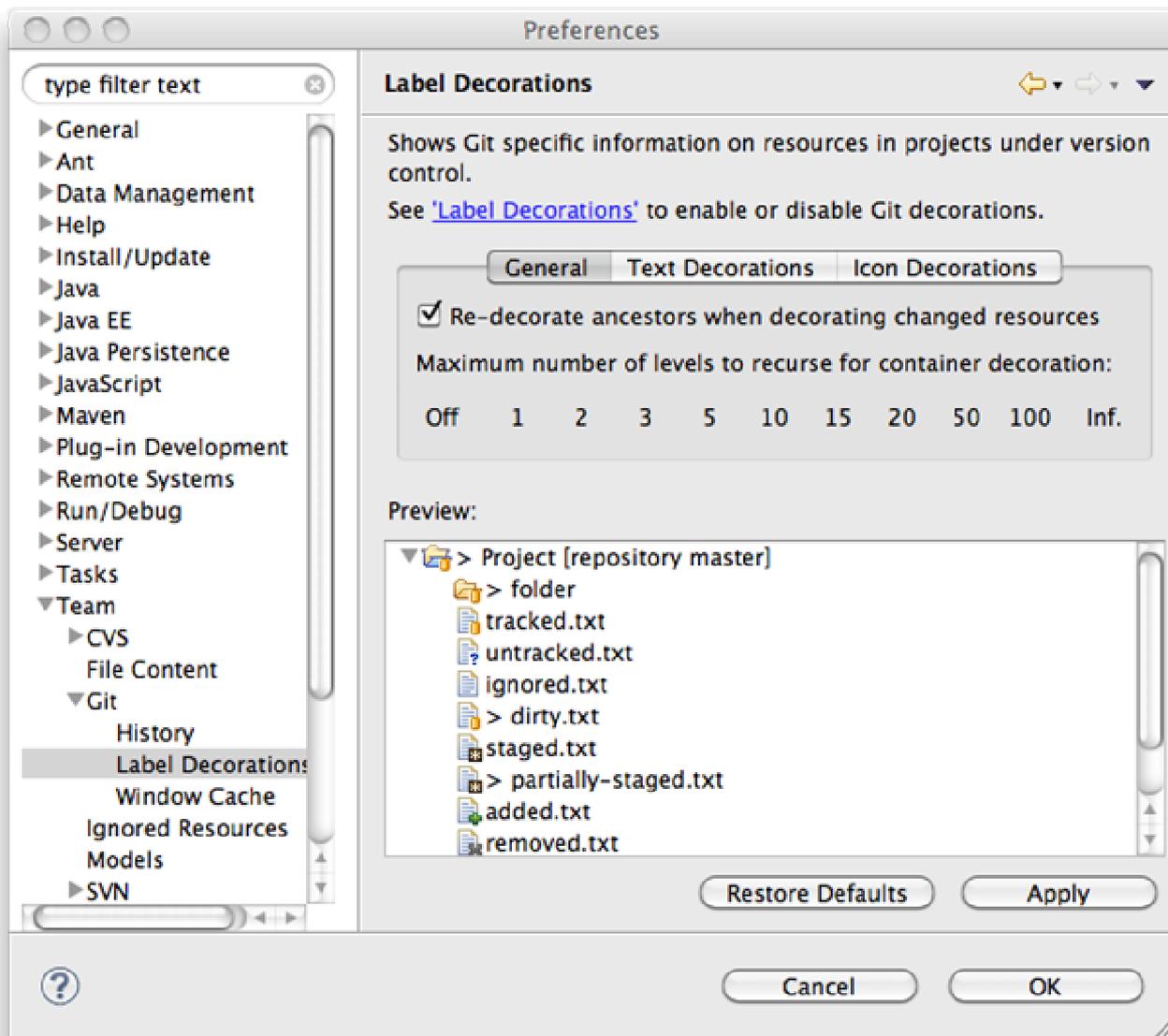


Aceptamos la licencia y damos a *Finish*. En este momento se descargará e instalará el plugin EGit.



Ahora nos recomienda que reiniciemos Eclipse Helios, le hacemos caso y pinchamos sobre "Restart Now".

Una vez reiniciado Eclipse Helios, estamos listos para trabajar. Para comprobar que se ha instalado correctamente EGit podemos ir las preferencias de Eclipse y sobre la rama *Team* deberíamos ver algo como:



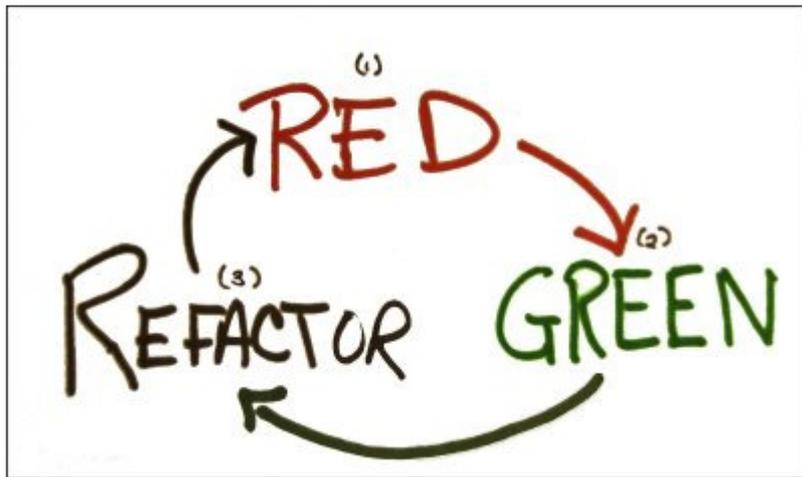
4. Conclusiones

Hemos visto que la instalación del plugin es muy sencilla. Ahora sólo nos queda sacarle partido a EGit y Git.

Por ejemplo, una forma de trabajo que puede ser interesante con este tipo de sistemas de control de versiones distribuidos es:

- Trabajar por tareas siempre es recomendable (Eclipse Mylyn).

- Mientras trabajamos sobre una tarea hacemos TDD (red - green - refactor).



- Cada vez que alcancemos el paso "refactor" hacemos un commit. Este commit es local por lo que es inmediato (no necesitamos conexión de red), y se nos irá quedando guardado todo el histórico de pequeños cambios que vamos haciendo en el proyecto.
- Una vez terminamos la funcionalidad podemos hacer push a un repositorio Git público, de forma que la nueva funcionalidad desarrollada este lista para el resto del equipo.

Os dejo a vosotros la tarea de explorar las capacidades de estos sistemas de control de versiones distribuidos para adecuarlo a vuestro ciclo de desarrollo y conseguir mejorara la calidad del proceso y del código resultante.

5. Sobre el autor

Alejandro Pérez García, Ingeniero en Informática (especialidad de Ingeniería del Software) y Certified ScrumMaster

Socio fundador de Autentia (Formación, Consultoría, Desarrollo de sistemas transaccionales)

<mailto:alejandropg@autentia.com>

Autentia Real Business Solutions S.L. - "Soporte a Desarrollo"

<http://www.autentia.com>

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» **Regístrate** y accede a esta y otras ventajas «

Autor

Mensaje de usuario registrado



Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

