

Practica SVN



Entornos de Desarrollo
Zahira Zamora Camacho
Carlos Ignacio Domínguez Merchán
Enero 2013

Índice

Objetivos de la práctica	3
Desarrollo de la práctica	3
1.- PARTE 1	3
1.1.- Creación de un usuario en Assembla	3
1.2.- Creación de un espacio propio	4
2.- PARTE 2: Usar Eclipse con un repositorio	5
3.- PARTE 3: Trabajando con Eclipse-Repositorio	7
3.1.- Crear un nuevo proyecto SVN	7
3.2.- Utilizar el nuevo proyecto SVN	9
4.- PARTE 4: Desarrollo del código de triángulos rectángulos	14

Objetivos de la práctica

Realizar un programa desarrollado en Eclipse utilizando un repositorio. El objetivo principal es familiarizarnos con dicho repositorio y las versiones cuando trabajamos en equipo.

Desarrollo de la práctica

1.- PARTE 1

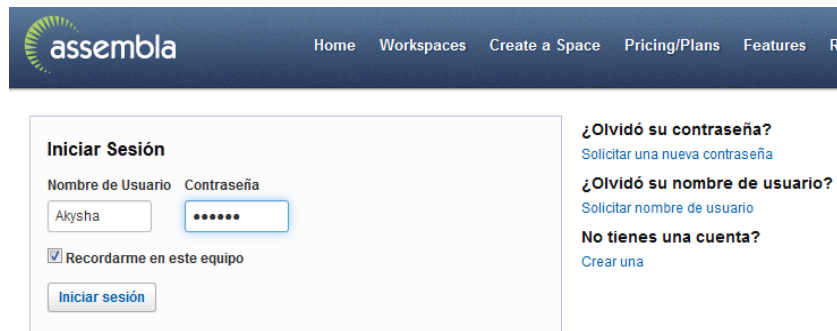
1.1.- Creación de un usuario en Assembla

Para poder utilizar un repositorio hay múltiples páginas donde nos dan gratuitamente un espacio, pero para esta práctica usaremos Assembla. Su dirección en <https://www.assembla.com/home>.

1.- Lo primero que debemos hacer es crearnos un usuario para poder obtener ese espacio y poder utilizar el repositorio que nos asignen. Para ello pinchamos en “login” y nos aparece esta pantalla:

2.- En el apartado “No tienes cuenta?” nos permite crear una nueva cuenta. Pinchamos en “crear una”.

3.- Introducimos nuestros datos y nos registramos. Una vez hecho esto, ya tenemos creada nuestra cuenta y podemos introducir nuestro nombre de usuario y contraseña para autentificarnos.



Iniciar Sesión

Nombre de Usuario: Akysa Contraseña:

☒ Recordarme en este equipo

[Iniciar sesión](#)

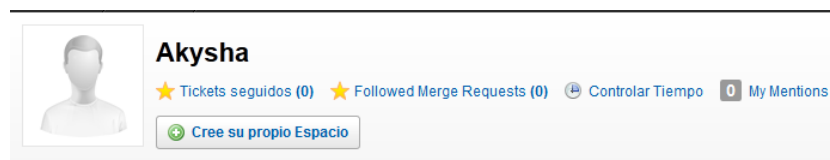
[¿Olvidó su contraseña?](#)
[Solicitar una nueva contraseña](#)

[¿Olvidó su nombre de usuario?](#)
[Solicitar nombre de usuario](#)

[No tienes una cuenta?](#)
[Crear una](#)

1.2.- Creación de un espacio propio

1.- Una vez que estamos autenticados podemos crearnos un espacio con el que poder trabajar nuestras versiones con el equipo. Pinchamos en “Cree su propio Espacio”.



Akysa

★ Tickets seguidos (0) ★ Followed Merge Requests (0) 🕒 Controlar Tiempo 🗉 0 My Mentions

[Cree su propio Espacio](#)

2.- Nos aparece la siguiente pantalla. Elegimos “Cree un Proyecto Público” para que lo pueda ver todo el mundo, aunque no significa que lo pueda modificar.



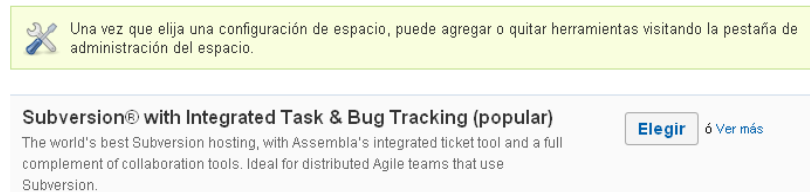
Cree un Proyecto Privado
Los proyectos privados están disponibles sólo para aquellos que hayan sido invitados a un espacio de trabajo. Ningún dato del proyecto está disponible al público.

Cree un Proyecto Público
Los proyectos públicos son visibles a todos. Son ideales para proyectos de código abierto y para comunidades.

[Empezar](#) [Empezar](#)

3.- Nos permite distintas opciones al crearnos el tipo de espacio. Elegimos la primera opción que nos aparece, Subversion.

Elija una configuración estandard



Una vez que elija una configuración de espacio, puede agregar o quitar herramientas visitando la pestaña de administración del espacio.

Subversion® with Integrated Task & Bug Tracking (popular)
The world's best Subversion hosting, with Assembla's integrated ticket tool and a full complement of collaboration tools. Ideal for distributed Agile teams that use Subversion.

[Elegir](#) [ó Ver más](#)

4.- Rellenamos la información que nos solicita. Le asignamos un nombre al espacio y comprobamos la disponibilidad. Si está disponible podemos continuar.

Plan Público Gratuito

1 Elija su configuración 2 Seleccione un Plan 3 Pers

Advertencia: Spaces in Free Public Plans will have all their information visible to other site visitors and search crawlers.
Si necesita un espacio privado, por favor, agréguelo [aplan profesional](#)

Nombre del espacio

Nombre URL - **Disponible**
 ✓ Chequear disponibilidad

Recomendado: Asigne un nombre único de URL para que pueda tener una URL legible para las páginas wiki, trac y subversion. Puede utilizar letras, números o caracteres como '-' y '_'. Use _ cuando quiera un espacio

Descripción

Escriba una descripción corta de su espacio que aparecerá en los resultados de búsqueda y de etiquetas

5.- Elegimos la configuración deseada para la seguridad. En nuestro caso elegimos que sólo se pueda ver y no modificar para el público en general, y permitimos editar a los miembros del equipo. Por último pinchamos a “Cree el Espacio”.

Seguridad

Acceso para no-miembros

Permitir para acceso público.

Elija cómo los usuarios no-miembros podrán acceder a su espacio.
Si elige *Ninguno*, este espacio será visible únicamente para los compañeros de equipo seleccionados.

Acceso a miembros

Permitir para acceso de miembros del equipo.

Elija cómo los miembros de su equipo podrán acceder al espacio.

ó

De este modo ya tenemos el espacio para poder trabajar en equipo y no perder información.

2.- PARTE 2: Usar Eclipse con un repositorio

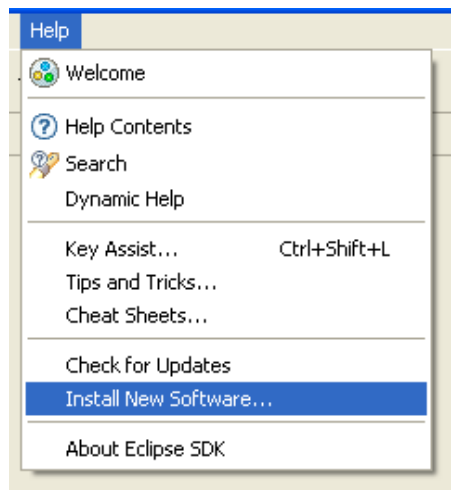
Antes de poder subir y bajar los archivos del repositorio a nuestro IDE, en este caso Eclipse, debemos prepararlo para ello.

1.- Debemos descargarnos estos dos archivos. “Subversive-incubation-0.7.9.I20110819-1700” y “Subversive-connectors-2.2.2.I20110715-1700”. Se pueden descargar directamente desde estos enlaces:

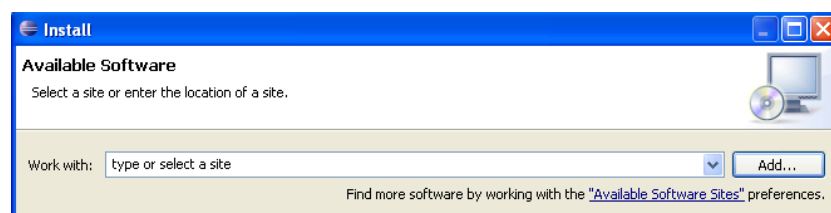
<http://www.eclipse.org/downloads/download.php?file=/technology/subversive/0.7/builds/indigo/Subversive-incubation-0.7.9.I20110819-1700.zip>

<http://community.polarion.com/projects/subversive/download/eclipse/2.0/builds/Subversive-connectors-2.2.2.I20110715-1700.zip>

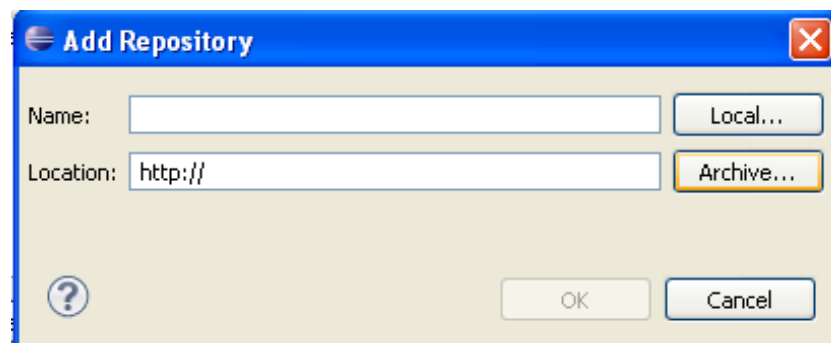
2.- Una vez descargados, tenemos que instalarlos en el Eclipse. Primero instalaremos “Subversive incubation” y después “Subversive connectors”. Para ello abrimos Eclipse y vamos al menú Help → Install New Software...



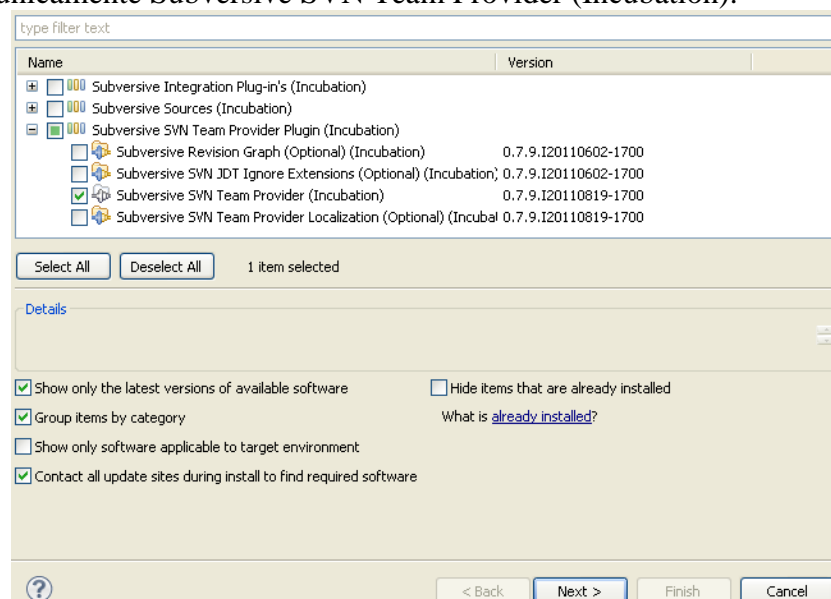
3.- Pinchamos en “add”.



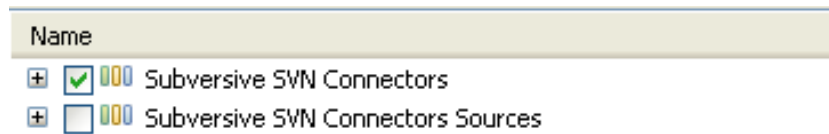
4.- Y en “archive”. Elegimos el archivo donde lo tengamos guardado y comenzamos a la integración del nuevo software.



5.- Elegimos únicamente Subversive SVN Team Provider (Incubation).



6.- Realizamos el mismo proceso para integrar Subversive connectors. Elegimos todos los “Subversive SVN Connectors”.

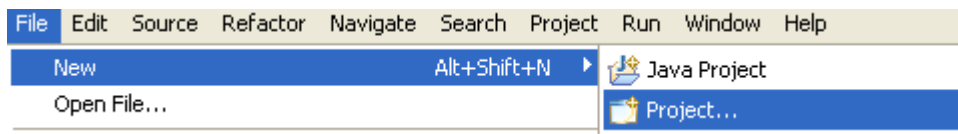


Una vez integrados estos dos software en nuestro Eclipse, ya podemos trabajar desde el Eclipse con nuestro repositorio.

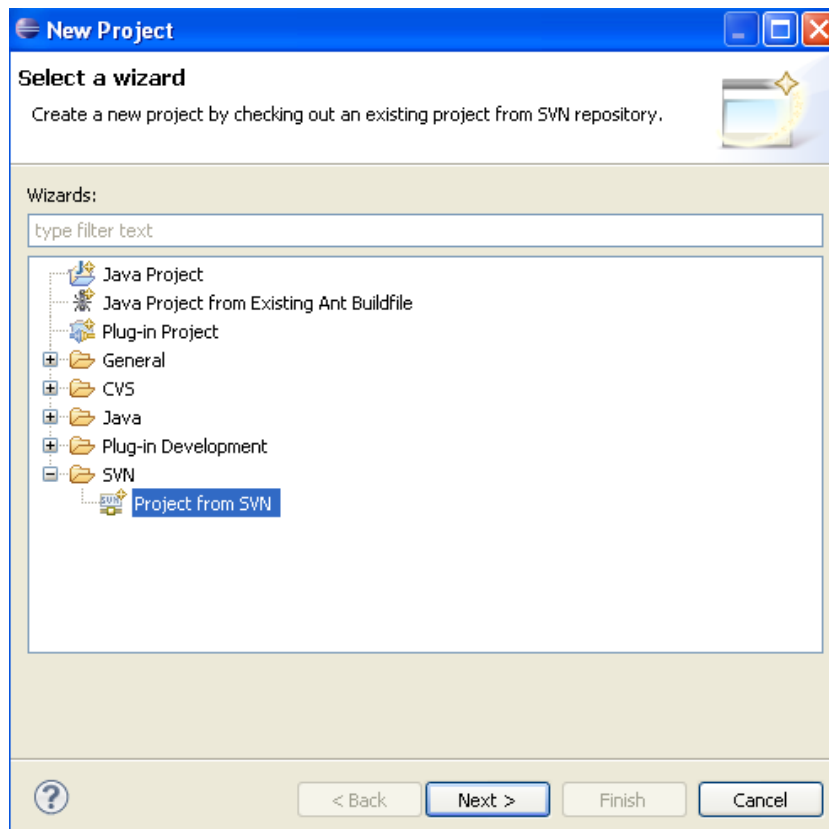
3.- PARTE 3: Trabajando con Eclipse-Repositorio

3.1.- Crear un nuevo proyecto SVN

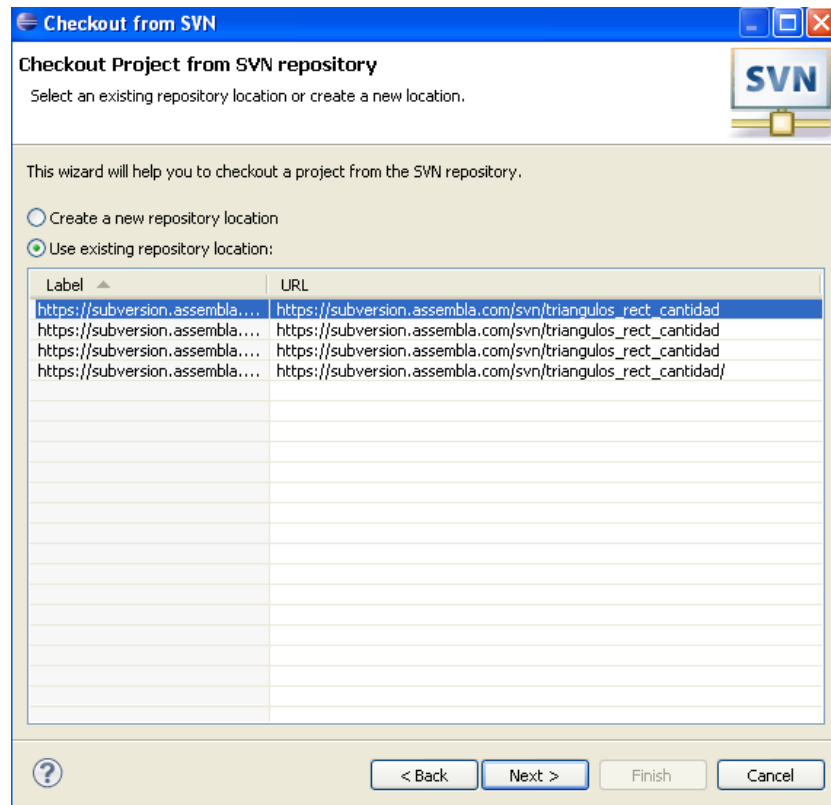
1.- Nos creamos un proyecto SVN. Para ello abrimos Eclipse y vamos a menú File → New → Project...



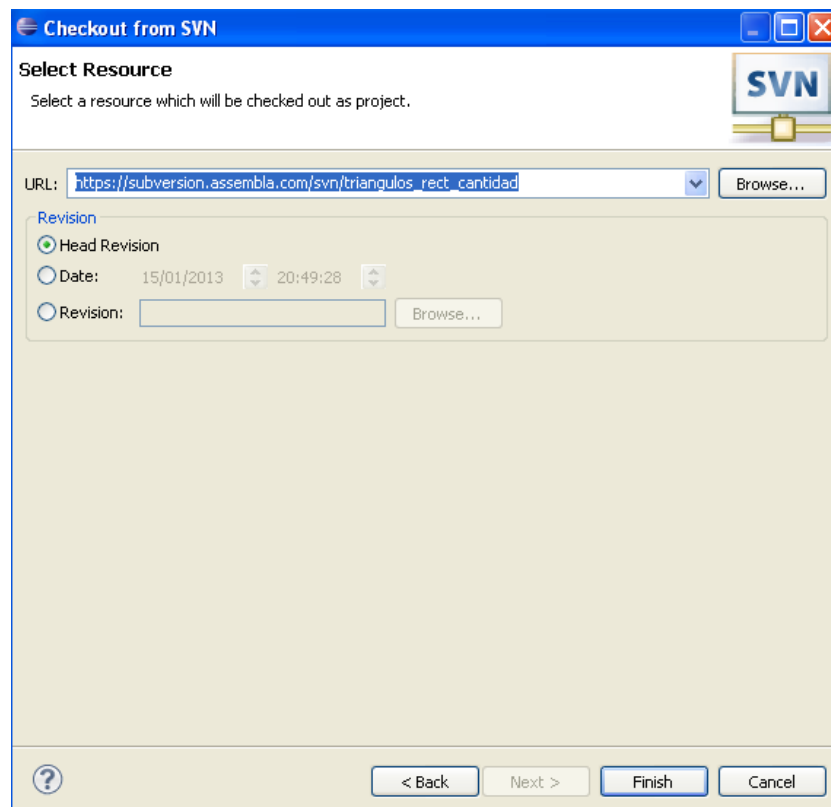
2.- Elegimos SVN – Project from SVN. Pinchamos en Next.



3.- Elegimos la dirección de nuestro repositorio, si no está creada podemos crear una nueva localización. Pinchamos en Next.



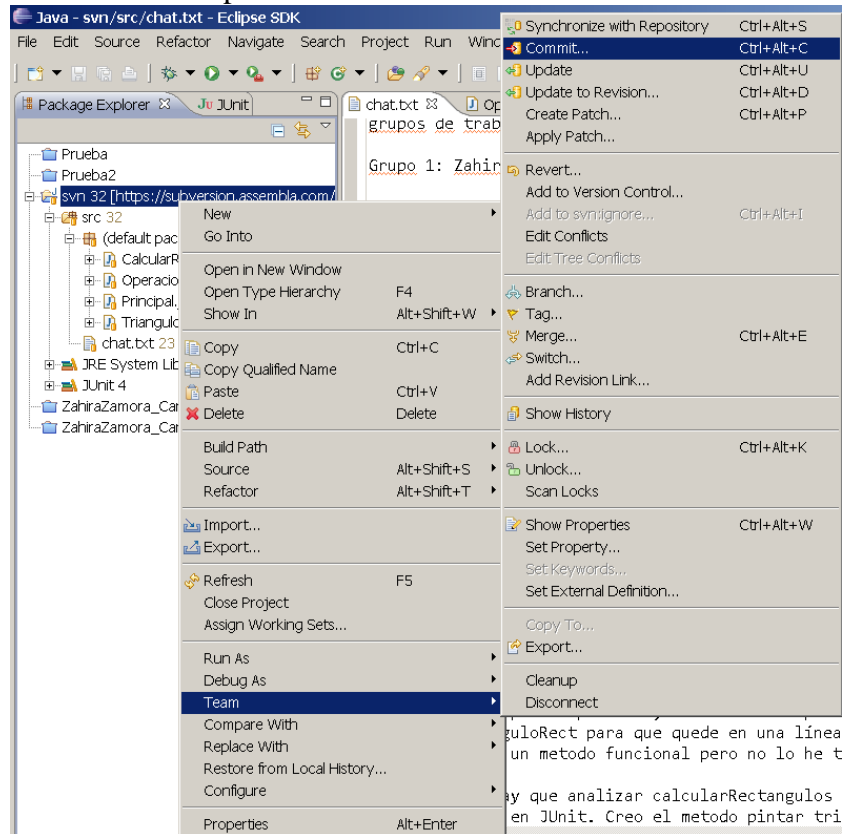
4.- Por último dejamos la opción “Head Revision”, que es para bajarnos lo último del repositorio, la última versión que exista. Pinchamos en Finish.



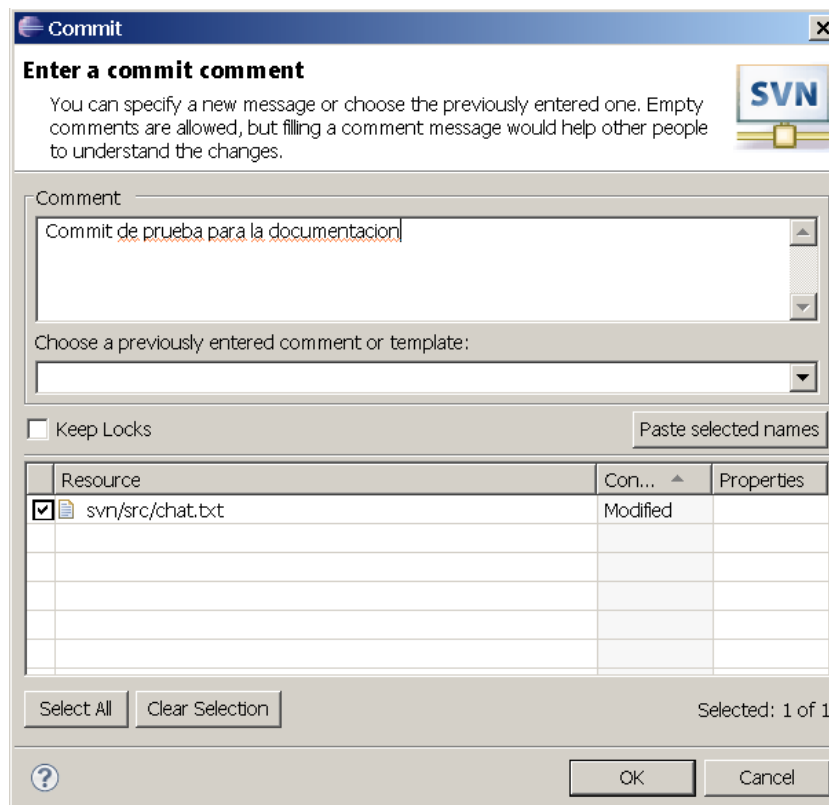
3.2.- Utilizar el nuevo proyecto SVN

Ahora tenemos creado el proyecto y además está conectado con nuestro repositorio. Según vayamos escribiendo nuestro código, cambiándolo, creando nuevas clases...etc deberemos ir subiéndolo al repositorio para que no se pierda nada y para que el resto del gupo pueda utilizar también los cambios efectuados.

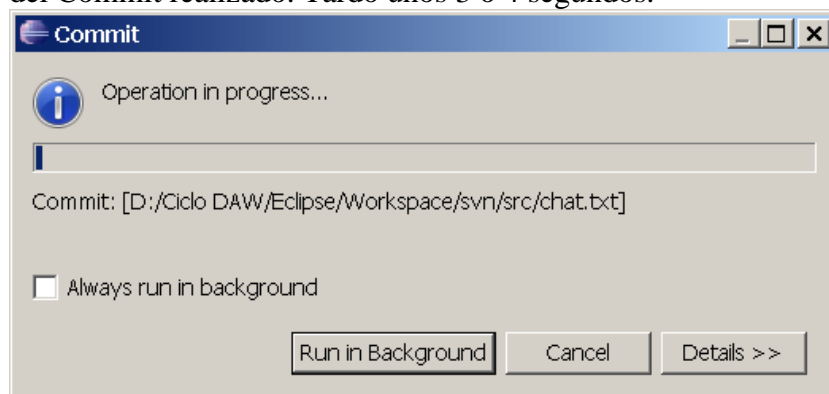
1.1.- Para ello, una vez que queramos subir al repositorio nuestros cambios, haremos un “commit”. Vamos a nuestra carpeta SVN botón derecho → Team → Commit...



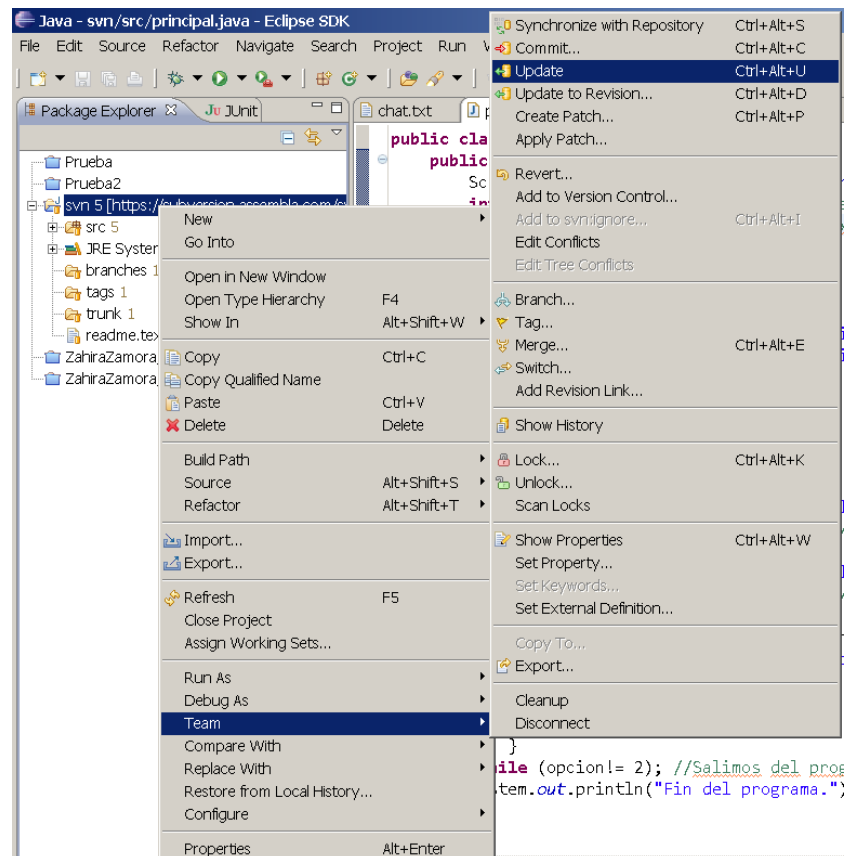
1.2.- Nos pide un comentario para que expliquemos los cambios que hemos efectuado en el proyecto. Así, cuando lo vea otro miembro del equipo, estará mucho mejor informado. Incluso sin hacer Update, se puede ir al repositorio y se puede ver cada commit con el nombre del miembro que lo subió y el comentario añadido.



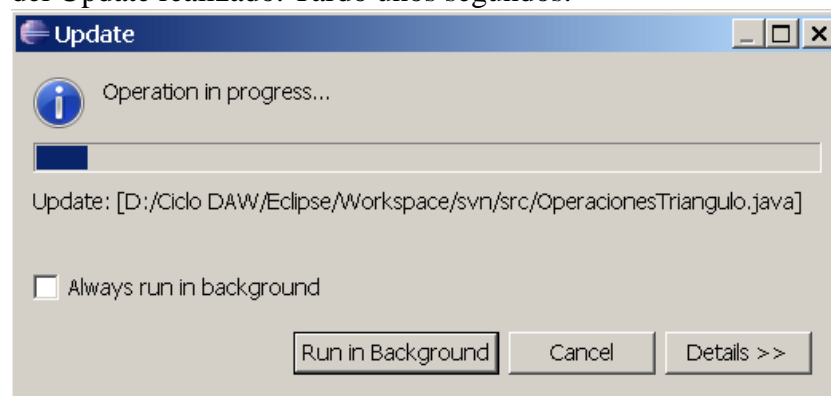
1.3.- Progreso del Commit realizado. Tardó unos 3 ó 4 segundos.



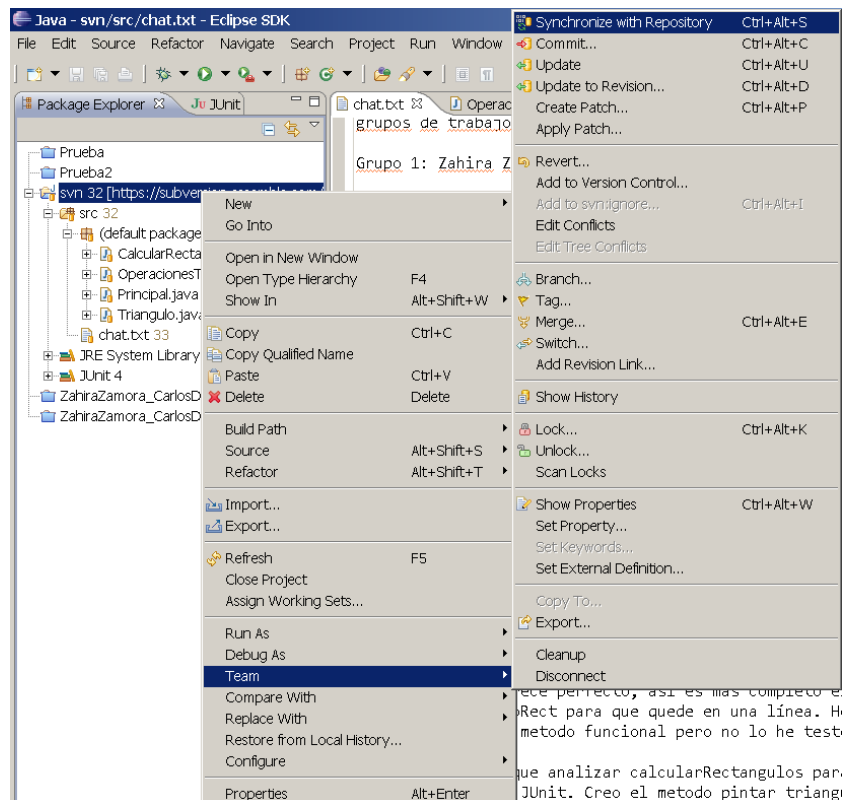
2.1.- Si lo que queremos es bajarnos la última versión que esté en el repositorio porque hay cambios de otro u otros miembros del equipo, haremos un “Update”. Vamos a nuestra carpeta SVN botón derecho → Team → Update...



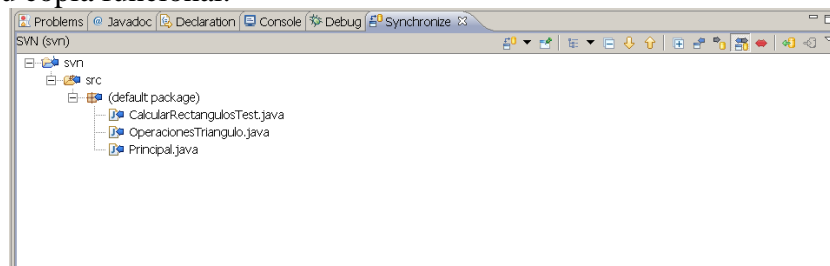
2.2.- Progreso del Update realizado. Tardó unos segundos.



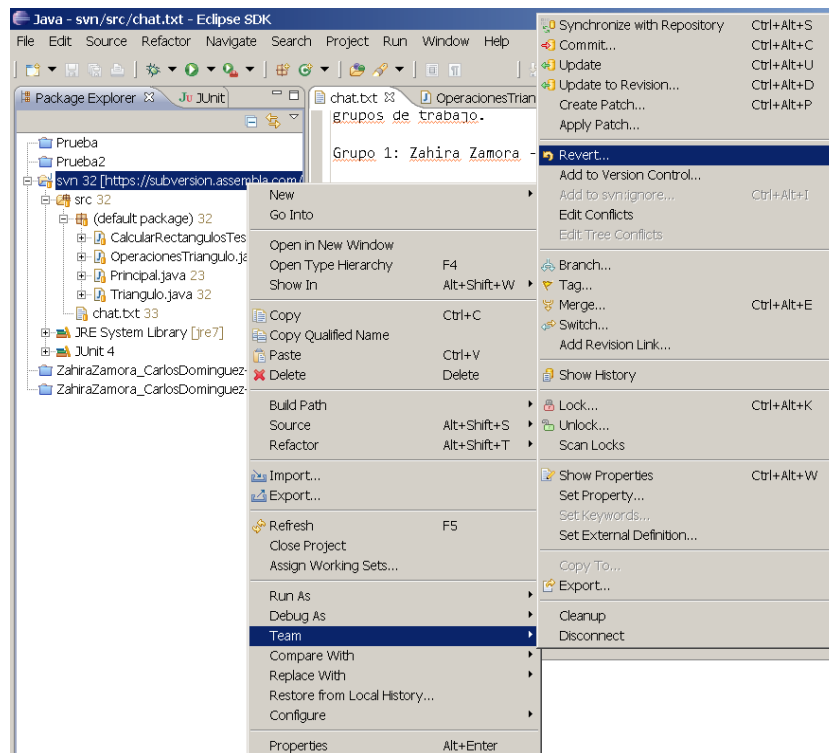
3.1.- Para poder ver si hay cambios en el repositorio que no tienes en tu copia funcional, es decir, que la versión propia ya no coincide con la versión del repositorio y por tanto otros miembros han subido cambios, podemos hacer una sincronización con el repositorio. Vamos a nuestra carpeta SVN botón derecho → Team → Synchronize with Repository...



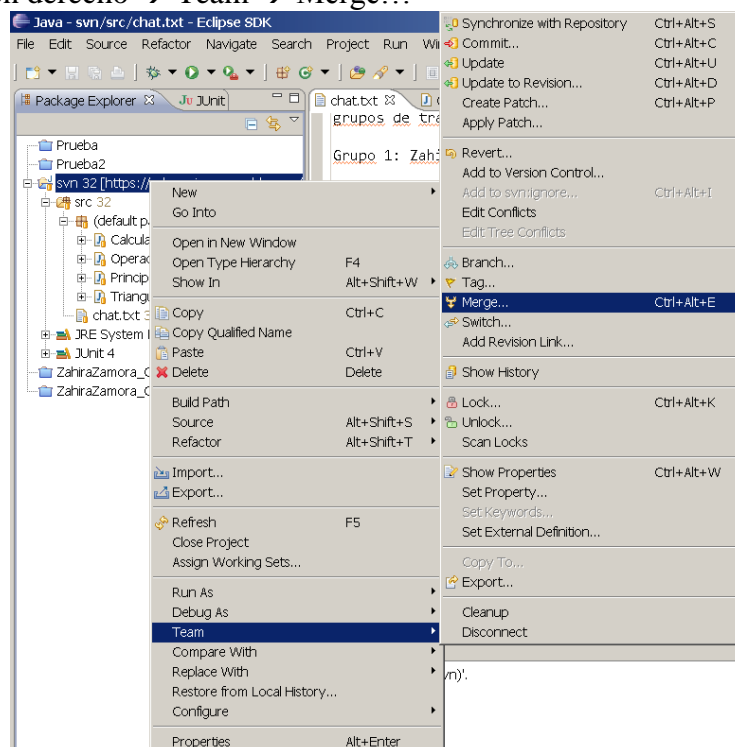
3.2.- En esta ventana nos saldrán los archivos en los que haya diferencias de versiones entre el repositorio y tu copia funcional.



4.- Si nos arrepentimos de los cambios que hemos efectuado y queremos deshacerlos todos podemos hacer un revert. Vamos a nuestra carpeta SVN botón derecho → Team → Revert...



5.- Puede ocurrir que varios miembros del equipo estén cambiando el mismo archivo al mismo tiempo. En este caso hay dos posibilidades. Si los cambios hechos por los miembros son en distintas líneas de código del archivo, o incluso mejor, en distintos métodos, no hay problema, hacemos un “Update” y nos hará un “Merge” (mezcla los dos archivos) automático. Si por el contrario, se coincide en la o las líneas modificadas, nos avisará con un error al intentar hacer el “Update” y tendremos que hacer un “Merge” manual. Vamos a nuestra carpeta SVN botón derecho → Team → Merge...



4.- PARTE 4: Desarrollo del código de triángulos rectángulos

Para no duplicar el código, puesto que están subidos al repositorio, en esta parte de la documentación explicaremos nuestras clases, y la forma en la que hemos resuelto el problema.

Hemos creado las siguientes clases:

- Triangulo: Contiene 3 atributos, que son los lados del triángulo. Están creados el método constructor, los get y los set, y un método para comparar con otro triángulo al que metemos como parámetro y nos dice si son semejantes o no.
- OperacionesTriangulo: Aquí hemos creado únicamente métodos estáticos que calculan distintas cosas de los triángulos.
 - ✓ esTriangulo: Nos devuelve true si el triángulo se puede construir, false en caso contrario.
 - ✓ esTrianguloRect: Nos devuelve true si el triángulo es rectángulo, false en caso contrario.
 - ✓ raizEntera: Nos devuelve true si la raíz del número es un entero, false en caso contrario.
 - ✓ trianguloSemejanteEnArray: Nos devuelve true si en el array existe un triángulo semejante al dado, false en caso contrario.
 - ✓ pintarRectangulos: Escribe por pantalla los triángulos rectángulos.
 - ✓ calcularRectangulos: Calcula los triángulos rectángulos que se pueden crear entre un mínimo y un máximo. Además incluimos otro parámetro booleano, para saber si queremos también calcular los semejantes o no.
- Principal: Contiene tres métodos distintos para pintar por pantalla, pintaMenu, pintaPedirIntervalo y pintaPedirLados usados por el main. También contiene el main que muestra el menú para dar opciones al usuario y ejecuta los métodos necesarios para la ejecución de la opción elegida por el usuario.
- CalcularRectangulosTest: Archivo de J-Unit para crear todos los casos de prueba del programa. Comprobamos los casos en que tiene que dar error (mínimo o máximo menores o iguales que cero y máximo menor o igual que mínimo) y comprobamos los casos cuando existen cero, uno y varios triángulos rectángulos, tanto para el caso en el que se incluyen los semejantes como el caso en el que no se incluyan. Además incluimos un caso concreto para cada opción de las dos mencionadas.

