Guía de desarrollo, Anexo 10.00

Guía Técnica de uso, estructura y nombrado en SVN

Oficina de Calidad

GERENCIA INFORMÁTICA
JOSEFA VALCÁRCEL, 44
28027-MADRID

Índice General

1 INTRODUCCIÓN 5

1.1 OBJETIVO 5
1.2 AUDIENCIA 5
1.3 GLOSARIO 5
1.3.1 Términos 5
1.3.2 Acrónimos 6

2 ESTRUCTURA DE CARPETAS EN SUBVERSION 7

2.1 ESTRUCTURA DE SVN PARA APLICACIONES YA EXISTENTES 7
2.2 ESTRUCTURA DE SVN PARA APLICACIONES NUEVAS 11
2.3 ESTRUCTURA DE SVN PARA LA DOCUMENTACIÓN 12

3 GESTIÓN DE LA CONFIGURACIÓN 17

3.1 CONVENCIÓN DE VERSIONADO 17
3.2 NOMENCLATURA DE NOMBRADO DE ETIQUETAS (TAGS) 18
3.3 NOMENCLATURA DE NOMBRADO DE RAMAS (BRANCHES) 22

4 CICLO DE VIDA DE DESARROLLO CON SUBVERSION 23

4.1 DESARROLLO CORRECTIVO 23
4.2 DESARROLLO EVOLUTIVO 24
4.3 DESARROLLO EVOLUTIVO Y CORRECTIVO 25

5 UTILIZACIÓN CLIENTES SVN 26

5.1 TORTOISESVN 26
5.1.1 Instalación del cliente TortoiseSVN 26
5.1.2 Obtener repositorio con TortoiseSVN 26
5.1.3 Añadir ficheros/carpetas al repositorio con TortoiseSVN 27
5.1.4 Actualización de archivos en el repositorio con SVN 29
5.1.5 Bloqueo de Ficheros en SVN 30
5.1.6 Creación de una rama/etiqueta con TortoiseSVN 32
5.1.7 Fusión de ramas con SVN 40
5.1.8 Renombrado de Ficheros/Carpetas en TortoiseSVN 43
5.1.9 Reubicación de Ficheros/Carpetas en TortoiseSVN 44
5.2 PLUGIN DE SUBVERSION PARA ECLIPSE 46
5.2.1 Instalación del Plugin 46
5.2.2 Utilización del Plugin 49
5.2.3 Importar un proyecto SVN existente 50
5.2.4 Importar un proyecto SVN existente con selección de carpetas. 51
5.2.5 Añadir un proyecto de Eclipse a SVN 54
5.2.6 Interfaz hacia SVN desde Eclipse 56

Índice de Ilustraciones y Tablas

Ilustración 1: Ejemplo de estructura en SVN con subsistemas 9
Ilustración 2: Estructura en SVN sin subsistemas 10
Ilustración 3. Nomenclatura de etiquetas 19
Ilustración 4. Nombrado de versión nueva no retrocompatible 20
Ilustración 5. Cambio de funcionalidad o nueva funcionalidad retrocompatible 20
Ilustración 6. Error de código 21
Ilustración 7. Despliegue asociado al entorno de pre 21
Ilustración 8. Versionado paralelo de código y documentación 22
Ilustración 9. Nombrado de ramas 22
Ilustración 10. Desarrollo correctivo 24
Ilustración 11. Desarrollo evolutivo 25
Ilustración 12. Desarrollo evolutivo y correctivo 25
Ilustración 13: Checkout con TortoiseSVN 27
Ilustración 14: Añadir carpeta con TortoiseSVN 28
Ilustración 15: Cambio de icono. 28
Ilustración 16: Confirmar carpeta 29
Ilustración 17: Estado de fichero modificado en Tortoise SVN 29
Ilustración 18: Actualización de ficheros en Tortoise SVN 30
Ilustración 19: Fichero Actualizado en Tortoise SVN 30
Ilustración 20: Bloqueo de ficheros en Tortoise SVN 31
Ilustración 21: Desbloqueo de ficheros en Tortoise SVN 32
Ilustración 22: Creación de una etiqueta 33
Ilustración 23: Creación de carpeta de etiqueta 34
Ilustración 24: Carpeta destino etiqueta 35
Ilustración 25: incluir carpeta a etiquetar 36
Ilustración 26: Actualización de la carpeta 37
Ilustración 27: Creación de una etiqueta 38
Ilustración 28: Creación de carpeta de rama 39
Ilustración 29: Carpeta destino de la rama 40
Ilustración 30: Fusión de ramas 41
Ilustración 31: Inclusión de la versión 42
Ilustración 32: Selección del fichero a Renombrar 44
Ilustración 33: Selección de la carpeta a reubicar 44
Ilustración 34: Menú Contextual Mover Carpeta 45
Ilustración 35: Confirmación en la nueva ubicación 45
Ilustración 36: Confirmación en la antigua ubicación 46
Ilustración 37: Configuración del Proxy en RSA 47
Ilustración 38: Instalación Subclipse 48
Ilustración 39: Selección de módulos 49
Ilustración 40: Selección de la dirección del repositorio 50
Ilustración 41: Realizar el checkout del proyecto 51
Ilustración 42: Selección de la dirección del repositorio 52
Ilustración 43: Selección de las carpetas/módulos 53

Ilustración 44: Realizar el checkout de los proyectos 54
Ilustración 45: Selección del repositorio donde subir el proyecto 55
Ilustración 46: Selección de archivos e introducción de comentario 55
Ilustración 47: Menú de opciones de SVN desde Eclipse 56

Tabla 1. Estructura del repositorio del proyecto para la documentación del proyecto 13
Tabla 2. Estructura del repositorio del proyecto para la documentación de la aplicación 15
Tabla 3. Convención de versionado 18

1 Introducción

1.1 Objetivo

Subversion (SVN) es un sistema de control de versiones libre y de código fuente abierto. Se trata de un sistema de gestión de la configuración que gestionará en DGT las fuentes de las aplicaciones y la documentación del proyecto, facilitando el control de las modificaciones que se van realizando a lo largo de todo el ciclo de vida del producto y del proyecto.

El objetivo de este documento es describir la estructura de SVN y la nomenclatura que deben seguir todos los proyectos de construcción de aplicaciones software de la Dirección General de Tráfico (DGT).

1.2 Audiencia

Este documento está dirigido a todas las personas que colaboren en labores relacionadas con la gestión, análisis, diseño, desarrollo, auditoría, pruebas, implantación y explotación de los sistemas de información de la gerencia de informática de la Dirección General de Tráfico.

1.3 Glosario

1.3.1 Términos

En este apartado se incluyen, organizados por orden alfabético, los términos que se utilizan en el propio documento para facilitar su lectura y comprensión.

- **Aplicación**: Software diseñado para cumplir con unas necesidades específicas de un usuario. Por ejemplo, software de navegación, proceso de control, etc

1.3.2 Acrónimos

En este apartado se incluyen, organizados por orden alfabético, los acrónimos que se utilizan en el propio documento para facilitar su lectura y comprensión.

- **BBDD**: Bases de Datos
- **DGT**: Dirección General de Tráfico
- **GI:** Gerencia de Informática
- **IC:** Integración Continua
- **SVN:** Subversion, herramienta de control de versiones
- **TI:** Tecnologías de la Información

2 Estructura de carpetas en Subversion

La estructura que se ha definido para el SVN, sistema de gestión de la configuración que gestionará las fuentes de las aplicaciones y la documentación del proyecto de la DGT, tiene como objetivo dar soporte a los diferentes ciclos de vida existentes en el desarrollo de aplicaciones software, entre los que se identifican: el ciclo de vida de la gestión del proyecto, el ciclo de vida de la documentación y el ciclo de vida del software. Es por ello que tanto la gestión del proyecto, como la documentación y código de la aplicación contarán con un directorio o repositorio (dependiendo de la aplicación) independientes.

Para las aplicaciones ya existentes en la DGT, con acrónimo dado de alta en el SVN, se ha definido una estructura basada en directorios manteniendo la ruta existente en el SVN, mientras que para los proyectos nuevos, es decir, para aquellos cuyo acrónimo no existe en el SVN de la DGT, se ha definido una estructura de SVN basada en repositorios. A continuación se describen con detalle cada una de las estructuras definidas.

2.1 Estructura de SVN para aplicaciones ya existentes

Para las aplicaciones ya dadas de alta en la DGT, la estructura de SVN definida requiere de una reorganización de los directorios existentes y la creación de nuevos directorios. A continuación se describen cada uno de los directorios que formarán parte de la estructura del SVN:

- **acro-proyxx**, (siendo xx un número entero que comenzará en 01) que contará a su vez con la estructura: 0.lineasbases (se explica más adelante), branches, tags y trunk. Este directorio será el específico para la gestión del proyecto, además podrá incluir la documentación global de las aplicaciones en el que caso de que cuente con subsistemas.
- **acro-[subsistema]-dev,** que contará a su vez con la estructura: 0.lineasbases (se explica más adelante), branches (donde se encontraran las diferentes líneas de desarrollo, con su nomenclatura correspondiente, al margen de la línea principal de desarrollo que se encuentra en trunk), tags (donde se encontraran las etiquetas con las versiones de

código de la aplicación) y trunk (se corresponde con se utiliza para alojar la línea base del desarrollo). Este es el directorio específico para el código de la aplicación, en el caso de que existieran subsistemas habrá un directorio de este tipo por cada uno de ellos.

- **acro-[subsistema]-doc,** que contará a su vez con la estructura: 0.lineasbases (se explica más adelante), tags (donde se encontraran las etiquetas con las versiones de documentación de la aplicación) y trunk (se corresponde con se utiliza para alojar la línea base del desarrollo. Este es el directorio específico para la documentación técnica de la aplicación cuyo ciclo de vida es independiente del código; en el caso de que existieran subsistemas habrá un directorio de este tipo por cada uno de ellos, a excepción del subsistema de base de datos para el que no existiría este directorio.
- **acro-bbdd,** que contará a su vez con la estructura: 0.lineasbases (se explica más adelante), tags (donde se encontraran las etiquetas con las versiones de documentación de la aplicación) y trunk (se corresponde con se utiliza para alojar la línea base del desarrollo. Este directorio es específico para la base de datos y contendrá los scripts a ejecutar (dev/scripts) y el modelo de datos correspondiente (doc).
- **0.lineasbases**: directorio reservado para los equipos transversales: oficinas de calidad y pruebas, departamento de arquitectura y departamento de sistemas. En él se localizará la información compartida entre los diferentes equipos transversales y el equipo de desarrollo, así como los tags de pre (aquellos que no se han subido a PRO) y de pro (aquellos que se han subido a PRO). En el caso de 0.lineasbases de acro-proyxx, contendrá la carpeta intercambio subdividida en hitos (h0x) si dicho proyecto está dividido en iteraciones.
- **Vxx,** donde se encontrará el histórico de las versiones anteriores de la aplicación.

En la figura 1 se muestra un ejemplo de estructura de SVN para una **aplicación con subsistemas**, donde además, se incluyen ejemplos de nombrado de branches y tags:

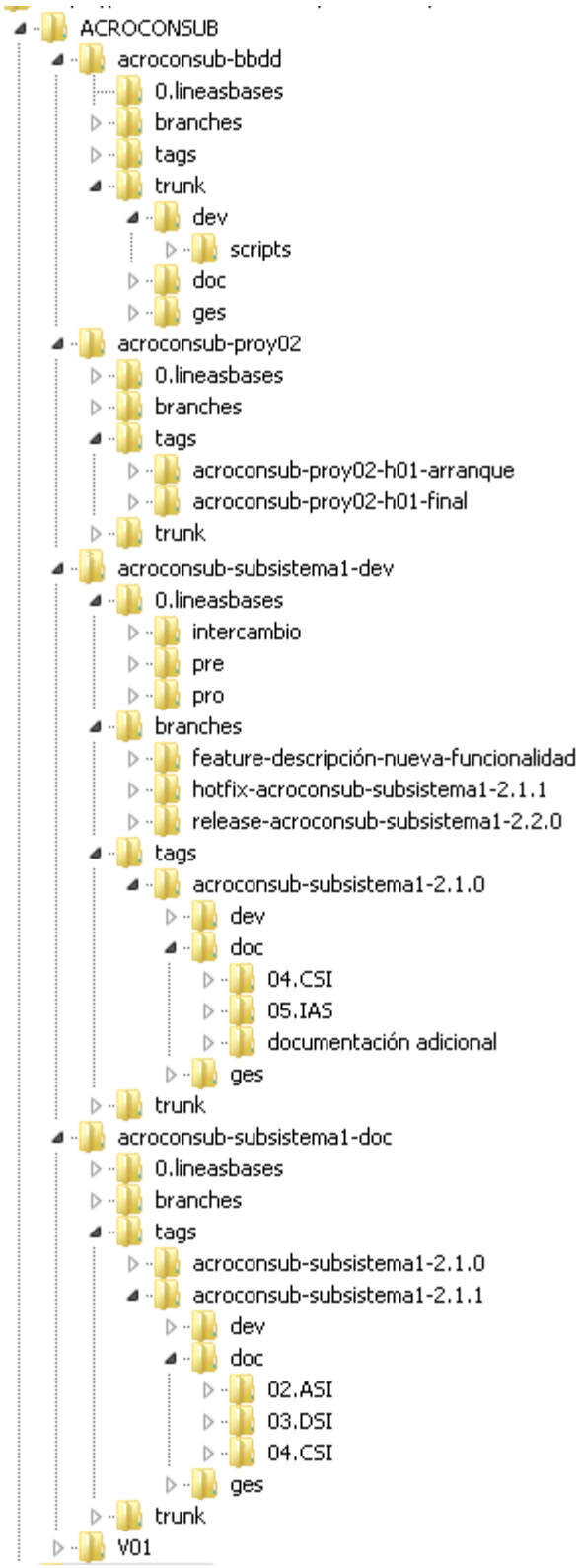

Ilustración 1: Ejemplo de estructura en SVN con subsistemas

En la figura 2 se muestra un ejemplo de estructura de SVN para una **aplicación sin subsistemas**, donde además, se incluyen ejemplos de nombrado de branches y tags:

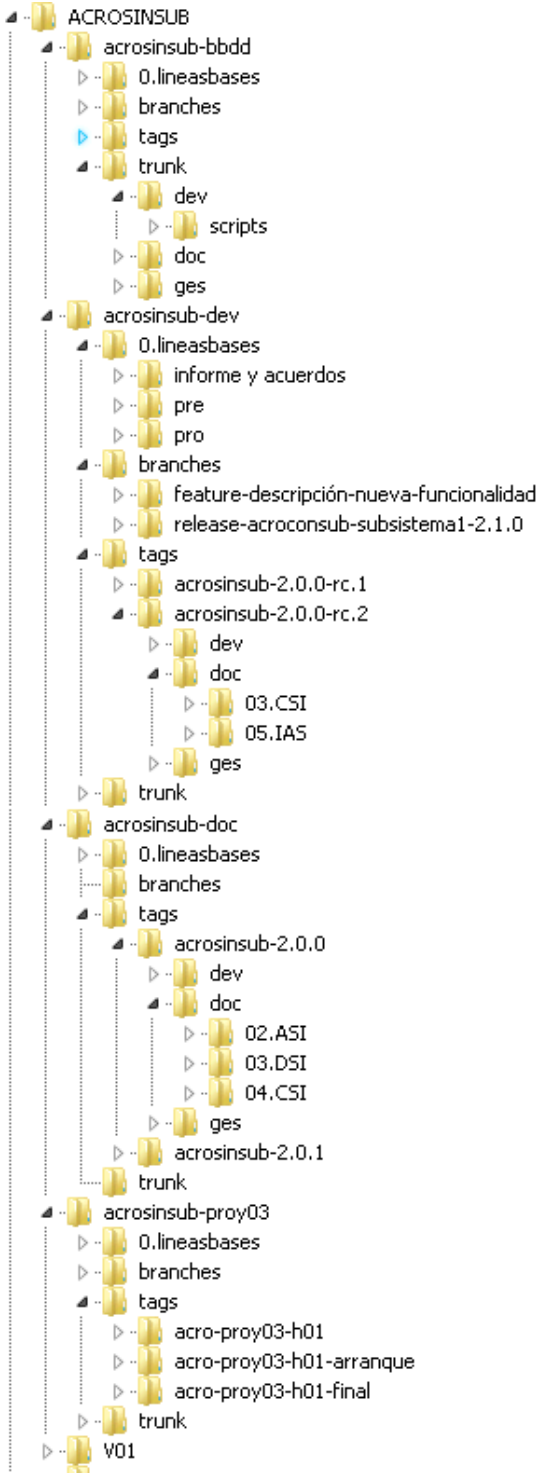

Ilustración 2: Estructura en SVN sin subsistemas

2.2 Estructura de SVN para aplicaciones nuevas

Para el caso de aplicaciones nuevas en DGT que no se encuentran dadas de alta en SVN, se ha definido una estructura basada en la creación de repositorios. El acrónimo de las aplicaciones será utilizado para nombrar los repositorios asociados al proyecto en Subversion, con las siguientes consideraciones:

- La documentación asociada a la gestión del proyecto se ubicará en el repositorio denominado "acrónimo", además de las actas y los informes de seguimiento contendrá la documentación de arranque del proyecto como puede ser el pliego, el Plan de proyecto, Acuerdos y ANS establecidos, etc.
- El código fuente de la aplicación se almacenará en el repositorio con sufijo dev (acrónimo-[subsistema]-dev) de la estructura del proyecto en SVN.
- La documentación técnica de la aplicación cuyo ciclo de vida es independiente del código se almacenará en el repositorio con sufijo doc (acrónimo-[subsistema]-doc).

Dentro de cada proyecto de desarrollo se establece la siguiente estructura normalizada para el repositorio de control de versiones:

https://subversion.trafico.es/subversion/**acronimo-aplicacion-dev**
- 0.lineasbases
- branches
- tags
- trunk

https://subversion.trafico.es/subversion/**acronimo-aplicacion-doc**
- 0.lineasbases
- branches
- tags
- trunk

https://subversion.trafico.es/subversion/**acronimo-bbdd**
- 0.lineasbases
- branches
- tags
- trunk

https://subversion.trafico.es/subversion/**acronimo**
- 0.lineasbases
- branches
- tags

trunk

Como consideración adicional, el acrónimo de los proyectos podrá tener **hasta 10 caracteres**. No se podrá reutilizar un nombre de acrónimo ya existente. Además, se recomienda el uso de vocales para facilitar su identificación.

2.3 Estructura de SVN para la documentación

En la siguiente tabla se describe la estructura que debe mantener el repositorio de un proyecto organizada según la fase en la que se elabora cada entregable:

Documentación del proyecto		
ELEMENTO	**LOCALIZACIÓN**	**ARCHIVOS**
Actas de reuniones	ges/actas/	Nomenclatura de las actas: “Acta Reunion.”
Informes de Seguimiento	ges/informes seguimiento/	Copia de los correos enviados por el proyecto al Jefe de proyecto DGT notificando riesgos del proyecto (Si procede). “Informe Seguimiento aaaa_mes_Proveedor.”
Planificación	ges/plan/	“Plan de Proyecto” “Cronograma_XXXX” “Hoja de Riesgos_XXXX” “Hoja de Acuerdos_XXXX” “acro-proy-[Hitox]” “Plan de implantación en producción” “Plan de contingencia” Etc.

Documentación del proyecto		
ELEMENTO	**LOCALIZACIÓN**	**ARCHIVOS**
Memorias / contrato	ges/contrato/	"MT" "PPT"
Matrices de trazabilidad del producto y del proyecto	ges/trazabilidad/	"Matrices de trazabilidad"
Correos relevantes del proyecto	ges/correos/	Copia de los correos relevantes del proyecto intercambiados entre los participantes del proyecto que necesiten ser almacenados para referencias posteriores.
Correos de validación de entregables	ges/validaciones/	Copia de los correos de confirmación enviados por el usuario, CAU, almacén de datos, etc. tras revisar los entregables que necesitan su firma.
Solicitudes	ges/solicitudes/	Solicitudes enviadas por parte del proyecto a los distintos actores y funciones transversales de DGT
Evidencias de auditorías	ges/informes y acuerdos	En esta carpeta se localizarán los informes de autoría resultantes de la realización de las auditorías internas de los proyectos.

Tabla 1. Estructura del repositorio del proyecto para la documentación del proyecto

A continuación se indican las rutas estándares donde ubicar la documentación del producto dependiendo del tipo de documento.

Documentación de la aplicación		
ELEMENTO	**LOCALIZACIÓN**	**ARCHIVOS**
Documentación del producto	doc/	

Documentación de la aplicación		
ELEMENTO	**LOCALIZACIÓN**	**ARCHIVOS**
Documentos de análisis	doc/02.ASI	/maqueta/"Maqueta Internet /.zip/.rar" , "Maqueta Intranet /.zip/.rar" , (si hubiera interfaz de usuario) /maqueta/"Servicios Interfaz" (para servicios Web) "Especificación de Requisitos" "Repositorio Cambios"
Documentación de diseño	doc/03.DSI	"Documento de arquitectura" "Requisitos Servicio de acceso a datos" "Requisitos Datapower"
	doc/04.CSI	"\Pruebas\Carga" con el documento generado desde el portal ALM "Plan pruebas de Carga" \Pruebas\Funcionales\" con el documento generado desde el portal ALM "Plan Pruebas Funcionales" NOTA: Los planes de pruebas se podrán auditar en diseño para una primera revisión pero su validación se realiza en la fase de construcción, por dicho motivo se guardan en doc/04.CSI
Documentación de cada aplicación	doc/04.CSI	"Manual de Usuario" "Manual CAU" "Manual de integración" "Descripción de Registros de Auditoría" "Documento Compilación"

Documentación de la aplicación		
ELEMENTO	**LOCALIZACIÓN**	**ARCHIVOS**
Documentación de implantación	doc/05.IAS	"Documento Implantación" "Descripción de Scripts de shell" "Perfiles de Usuario" "Manual de Explotación" "Acciones de Sistemas VXX_vYY_RZZZ_ENNN" "Plan de Paso a Producción" "Pruebas de Despliegue de Servicios" Carpeta \pruebas\funcionales\, con los documentos de resultados de las pruebas funcionales. Carpeta \pruebas\carga\, con los documentos de resultados de las pruebas de rendimiento, sobrecarga y escalabilidad Carpeta \pruebas\seguridad\, con los documentos de resultados de las pruebas de seguridad. Carpeta \pruebas\movilidad\, con los documentos de resultados de las pruebas de movilidad. Carpeta \pruebas\usabilidad\, con los documentos de resultados de las pruebas de usabilidad Carpeta \pruebas\compatibilidad y portabilidad\, con los documentos de resultados de las pruebas de compatibilidad y portabilidad.
Documentación de mantenimiento	doc/06.MSI	

Tabla 2. Estructura del repositorio del proyecto para la documentación de la aplicación

Se dispone además, de lo siguiente:

- En el directorio de distribución se proporciona un fichero comprimido en formato RAR "EstructuraCarpetasVacias.rar" donde está generada la estructura de carpetas del SVN para facilitar la tarea de generación del repositorio del proyecto.
- Para poder manejar toda la documentación adicional no sujeta a la metodología DGT útil para el proyecto y con el fin de no perderla una vez terminado el mismo, se podrán crear nuevos directorios con nombre "documentacion adicional" dentro de las carpetas "doc" y "ges" siempre que no se modifique la estructura básica del repositorio.

3 Gestión de la configuración

Todas las entregas de código fuente y documentación que se realicen, deberán llevarse a cabo utilizando etiquetas (tags). Todas las etiquetas se deberán generar en Subversion en la carpeta “tags”, con la nomenclatura que se define en el siguiente punto. Las ramas (branches) también utilizarán la nomenclatura designada.

3.1 Convención de versionado

El patrón de nombrado de versiones es el siguiente: x.y.z[-calificador] , asociado a la convención de versionado semántico (ver Anexo 38 Normativa de Versionado). En la tabla siguiente se indica el significado y posible valor de cada una de las partes que la componen:

Variable	Valor	Descripción
xx	Entero positivo. [1..99]	Número de versión mayor. Aumenta cuando se añaden cambios significativos a la aplicación, asociados con modificaciones de diseño o implementación no retro-compatibles, es decir, la aplicación contendrá un cambio funcional que impedirá utilizarse como en versiones anteriores, por ejemplo, cambios significativos en servicios expuestos, o cambios sustanciales de su interfaz de usuario.
yy	Entero positivo. [0..99]	Número de versión menor. Aumenta cuando se añaden pequeños cambios y/o funcionalidades que no supongan un problema de compatibilidad con el uso realizado con anterioridad.
zzz	Entero positivo. [0..999]	Número de revisión. Aumenta con los cambios realizados para solucionar errores detectados en las versiones liberadas, se asocia con soluciones hotfix de errores producidos en producción.
calificador	Calificador	Esta etiqueta marca un estado dentro del ciclo de vida del desarrollo, indicando la fase en la que se encuentra a través de los calificadores del tipo alpha, beta, RC, etc. Para obtener más información sobre los calificadores permitidos, se puede consultar el Anexo 38. Dentro del estándar de versionado explicado en la normativa, podemos encontrar el calificador release candidate, RC, que podrá ser utilizado para marcar aquellas versiones en proceso de pruebas, cuyo destino sea el despliegue en el entorno PRE (ejemplo 1.0.0-

Variable	Valor	Descripción
		RC). Aquellas versiones release, se marcarán a partir de la inexistencia de cualquier calificador, de esta forma la versión se convertirá en final, (ejemplo 1.0.0), estableciendo que su destino es el despliegue en entorno productivo, PRO.

Tabla 3. Convención de versionado

3.2 Nomenclatura de nombrado de etiquetas (tags)

Todas las entregas de fuentes y documentación se realizan generando una etiqueta con Subversion en la carpeta "tags", dentro de la estructura de SVN, con la nomenclatura designada para las etiquetas.

- La nomenclatura de etiqueta a usar para las **carpetas de gestión del proyecto** creadas en proyxx/tags será: **acronimo-proyxx_[h|i|eyy]-[resumen]**, indicando:
  - hyy: si el proyecto tiene hitos parciales, se podrán identificar como h01, h02, etc.
  - iyy: si el proyecto tiene iteraciones, se podrán identificar como i01, i02, etc.
  - eyy: si el proyecto tiene asociadas épicas (Jira), se podrán identificar como e01, e02, etc.
  - faseyy: si el proyecto tiene fases, se podrán identificar como i01, i02, etc.
  - resumen: descripción breve (una palabra) del contenido del tag. Como básicas para un proyecto se identifican el tag "arranque" y el tag "cierre".
- La nomenclatura de etiqueta a usar para las carpetas de **documentación técnica** creadas en acronimo-[subsistema]-doc/tags será: **acrónimo-[subsistema]-x.y.z**. Teniendo en cuenta que las dos primeras coordenadas de la versión de la documentación deberán coincidir con las de la versión del código correspondiente, dejando la última coordenada para posibles correcciones de errores propios de la documentación.

- La nomenclatura de etiqueta a usar para las carpetas de **código** creadas en acronimo-[subistema]-dev/tags será: **acrónimo-[subsistema]-x.y.z[-calificador].** En la imagen siguiente se muestra un resumen:

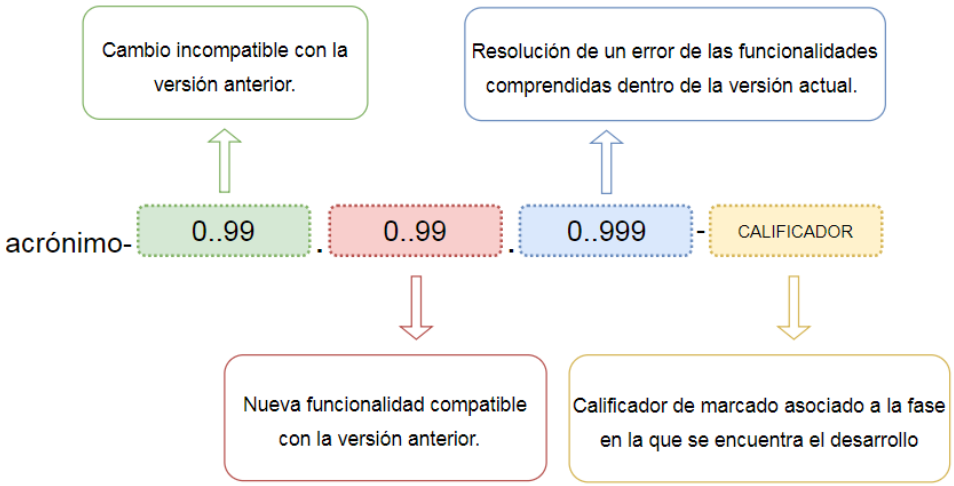

Ilustración 3. Nomenclatura de etiquetas

En acronimo-[subsistema]-x.y.z se incluirán los directorios dev, doc y ges:

- En "dev" se localizará el código fuente necesario para generar los artefactos a desplegar.
- En "doc" se localizará toda aquella documentación asociada al ciclo de vida del software, cuyo objetivo sea la compilación, despliegue, redespliegue, marcha atrás, etc. de la aplicación. Entre ellos se encuentran:
    - 03.CSI/Documento de compilación
    - 05.IAS/Documento de implantación
    - 05.IAS/Acciones de Sistemas
    - 05.IAS/Manual de Explotación
    - 05.IAS/Descripción de scripts de shell
    - 05.IAS/Pruebas de despliegue de servicios

- En “ges” se podrá incluir información propia del proyecto que quiera incorporarse por estar acoplada al ciclo de vida del producto que se está etiquetando.

A continuación se muestran ejemplos de nombrado en función del tipo de modificación realizado sobre el proyecto:

- Si se comienza el desarrollo de una versión nueva y no retrocompatible, se incrementa la versión mayor, restableciendo las demás versiones:

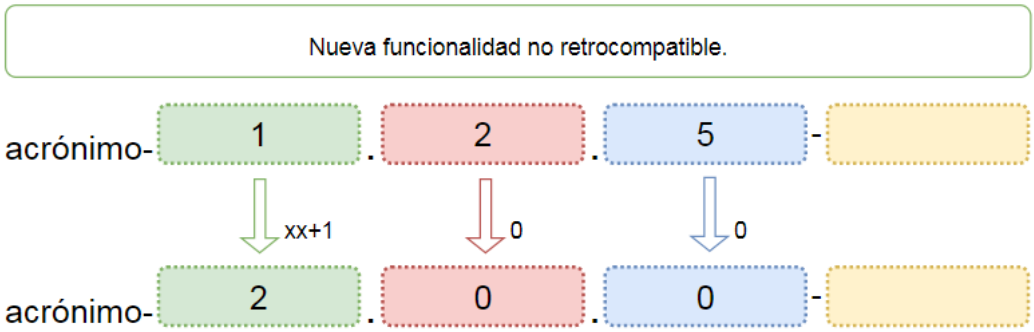

Ilustración 4. Nombrado de versión nueva no retrocompatible

- Si se cambia la funcionalidad en el contexto de la versión actual o se crea una nueva funcionalidad retrocompatible, se incrementa la versión menor:

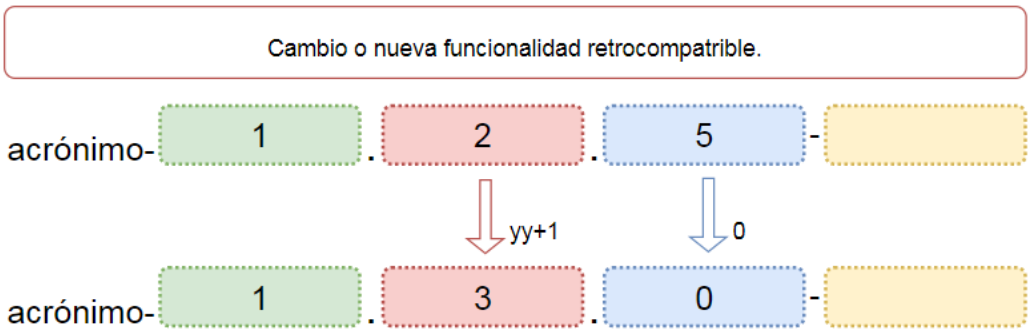

Ilustración 5. Cambio de funcionalidad o nueva funcionalidad retrocompatible

- Si se resuelve un error de código, se incrementa la **versión de revisión**:

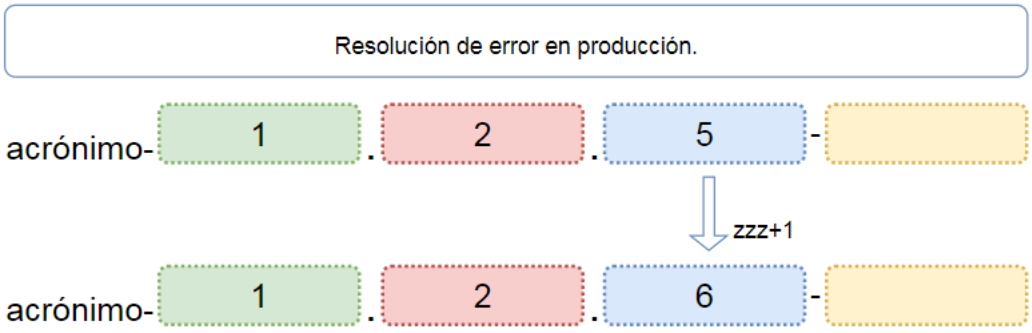

Ilustración 6. Error de código

- Si se desea indicar que su despliegue está asociado únicamente al entorno de PRE:

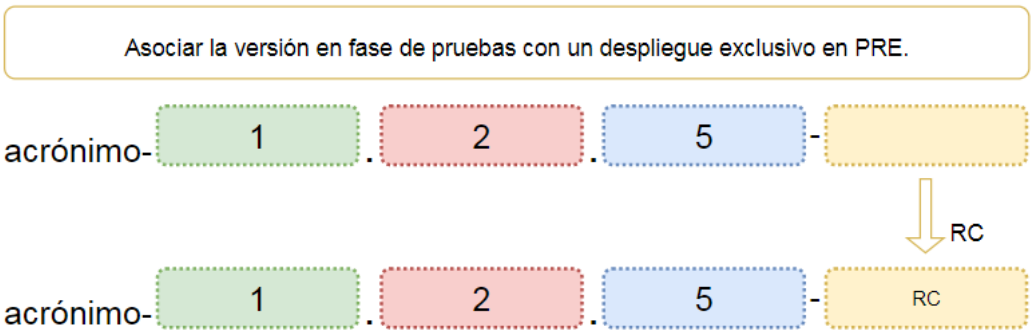

Ilustración 7. Despliegue asociado al entorno de pre

Dentro del proceso de versionado y liberación del proyecto, es necesario tener en cuenta que, tanto la documentación como el código fuente, deben seguir un versionado paralelo, es decir, **las versiones asignadas a las etiquetas de código y de su documentación, deben tener la misma versión mayor y menor**, dejando el número de revisión y el calificador para poder disponer de un ciclo de vida independiente entre código y documentación.

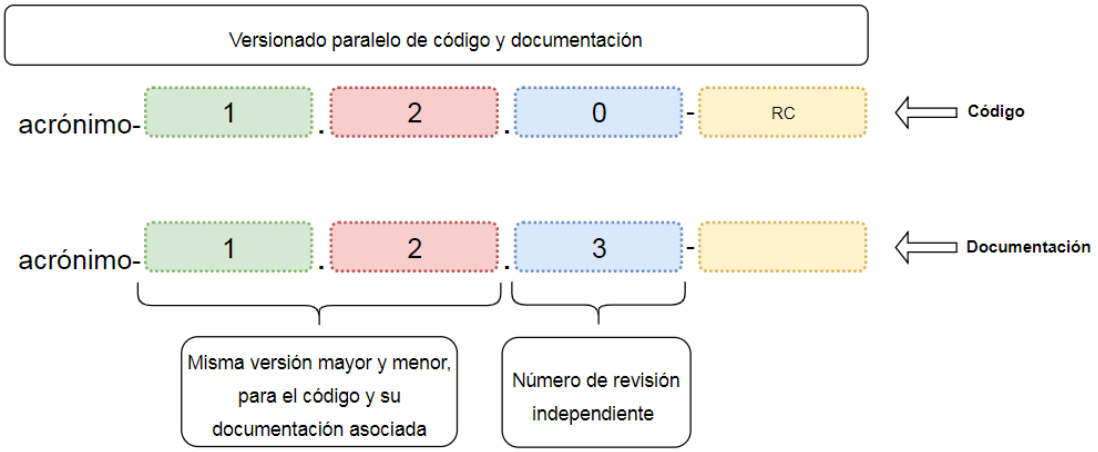

Ilustración 8. Versionado paralelo de código y documentación

Es necesario tener en cuenta que la versión indicada en las etiquetas de código, debe corresponderse con la versión indicada en el proyecto Maven o Gradle.

3.3 Nomenclatura de nombrado de ramas (branches)

Se generará una rama en la carpeta "branches" de la estructura de SVN, siguiendo la nomenclatura designada para las ramas. Todas las copias/ramas son versiones que en un futuro se pueden borrar una vez cumplido su cometido.

Las ramas deben utilizar la semántica de los siguientes prefijos:

- **hotfix**: para marcar una rama destinada a la resolución de incidencias en producción, encontradas en código liberado. Ejemplo: hotfix-acrónimo-1.2.5.
- **feature**: para marcar una rama destinada a la implementación a largo plazo, y de forma paralela, de una nueva funcionalidad. Ejemplo: feature-descripción-nueva-funcionalidad
- **release**: para marcar una rama de mantenimiento o desarrollo paralelo a la rama principal del repositorio. Ejemplo: release-acrónimo-1.2.x.

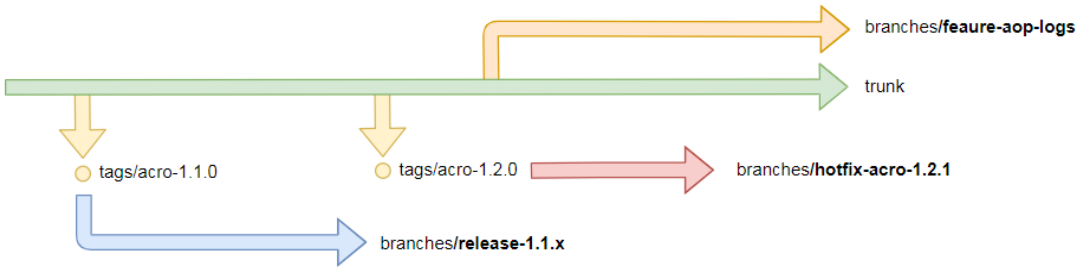

Ilustración 9. Nombrado de ramas

4 Ciclo de vida de desarrollo con Subversion

Atendiendo a la estructura del control de versiones para el código fuente:

- **Trunk** (línea base): se utiliza para alojar la línea base del desarrollo. Es la carpeta en la que se actualizarán los cambios continuos con los que implementar las nuevas versiones.
- **Branches** (ramas): puede contener las copias/ramas. En el caso en que se deban mantener varias versiones en paralelo, se utilizarán ramas para gestionar las distintas líneas de desarrollo concurrentes.
- **Tags** (etiquetas): contiene las etiquetas asociadas con la liberación del producto mediante versiones release o release candidate (despliegue de pruebas en PRE.)

La línea base, o trunk, es la línea principal de desarrollo con evolución constante, siendo el equipo de desarrollo el encargado de mantener el código fuente del proyecto sin errores durante la compilación y ejecución de pruebas unitarias.

Las etiquetas son instantáneas de los archivos de la línea base o rama en un momento determinado del ciclo de vida del software, utilizadas para liberar versiones finales o destinadas a procesos de pruebas y aprobación.

Las etiquetas acrónimo-[subsistema]-x.y.z-RC representan aquellas versiones de las aplicaciones que están siendo sometidas a pruebas de integración en el entorno de pre-producción. En el transcurso de las pruebas pueden surgir defectos a reparar, volviendo a empezar el ciclo.

Cuando una versión de la aplicación, pasa las pruebas de integración (entorno de pre-producción), puede ser promovida y marcada como liberada, eliminado cualquier tipo de calificador o fijando el calificador adecuado para el contexto de su liberación.

4.1 Desarrollo correctivo

En el mantenimiento correctivo se corrigen los defectos detectados después de la entrega del producto; es la forma básica de mantenimiento y consiste en localizar averías o defectos y corregirlos.

Durante el ciclo de desarrollo correctivo, se localiza el defecto y corrige generando una nueva versión en SVN que es desplegada en el entorno de PRE para su testeo. Si todo es correcto, se genera

y despliega en Producción cumpliendo con los procedimientos establecidos de despliegue entre entornos.

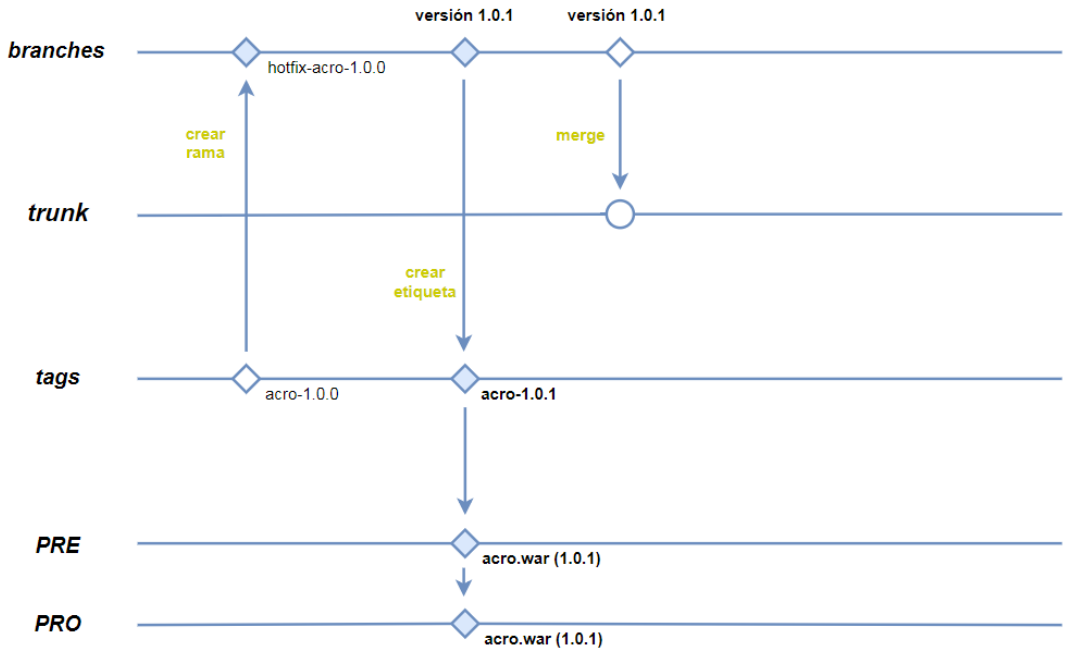

Ilustración 10. Desarrollo correctivo

4.2 Desarrollo evolutivo

Un mantenimiento evolutivo es aquel que pretende modificar algo que funciona y es correcto, con el objeto de aumentar, disminuir o cambiar las funcionalidades del sistema, ya sea por las necesidades del usuario o por otras causas, por ejemplo, cambios normativos.

Todos los desarrolladores trabajan sobre la rama trunk, en la que realizarán commit diarios de sus implementaciones, siempre que no contengan errores detectados por SonarLint (ver Anexo 39 Guia tecnica de uso de SonarLint), ni fallen las pruebas unitarias lanzadas en local. De esta forma se dispondrá de una rama continuamente actualizada y probada, de la que saldrán las versiones con las que generar los tags.

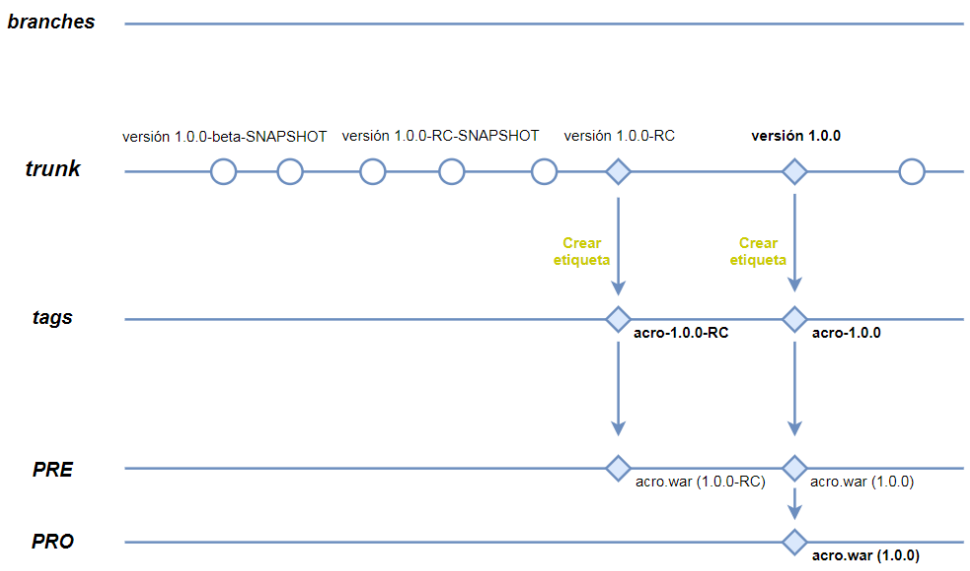

Ilustración 11. Desarrollo evolutivo

4.3 Desarrollo evolutivo y correctivo

En desarrollos evolutivos más largos pueden surgir cambios en la línea principal y tener que realizar desarrollos correctivos, éstos una vez terminados se integrarán desde la línea base a la rama de la nueva funcionalidad.

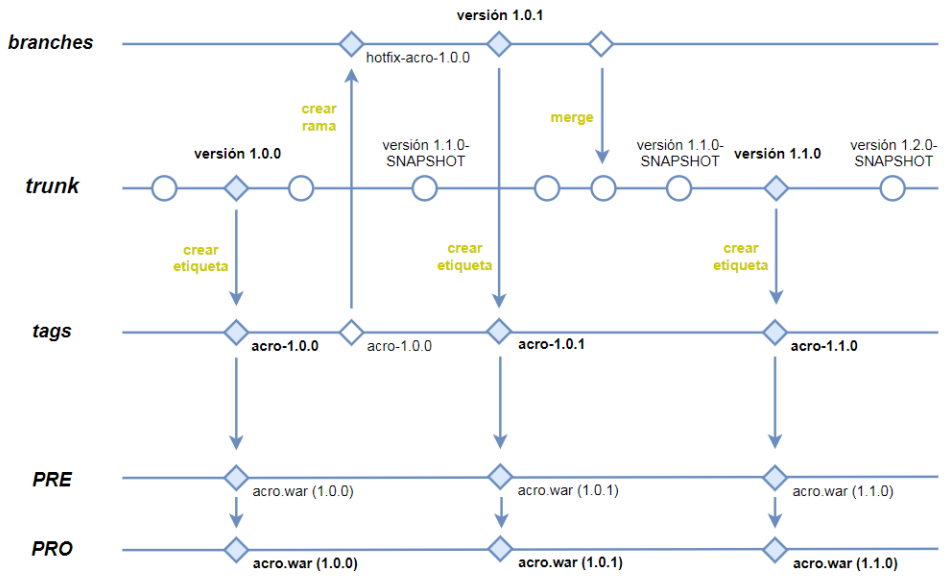

Ilustración 12. Desarrollo evolutivo y correctivo

5 Utilización Clientes SVN

Desde la máquina de cada persona integrante de un proyecto, se puede acceder al repositorio SVN a través de la aplicación cliente TortoiseSVN o cualquiera que se quiera utilizar para el acceso a repositorios SVN. Incluso puede utilizarse un navegador de internet y poner las URL de acceso al repositorio.

5.1 TortoiseSVN

A continuación se detalla la ruta de instalación y las operaciones más comunes a realizar con TortoiseSVN.

5.1.1 Instalación del cliente TortoiseSVN

Debe solicitarse al CAU enviando un mail al 3333@dgt.es o a través de la Herramienta de Gestión de Servicios TI de DGT.

5.1.2 Obtener repositorio con TortoiseSVN

Con TortoiseSVN instalado, se crea una carpeta en el sistema de ficheros local y se realiza la operación “SVN Checkout”, tras la cual aparecerá la ventana que se muestra en la ilustración Ilustración 13: Checkout con TortoiseSVN.

Se ha de poner la URL de la máquina servidora (https://subversion.trafico.es/subversion/) y la ruta al repositorio/carpeta que se quiere obtener. Nos solicitará usuario y password, se ha de poner el **usuario y password** que se tenga de **LDAP** y a partir de ese momento, se tendrá una copia en el sistema de ficheros local del repositorio seleccionado.

Si se quiere una versión diferente, se puede indicar en el cuadro “Revision” cuando se hace un SVN checkout.

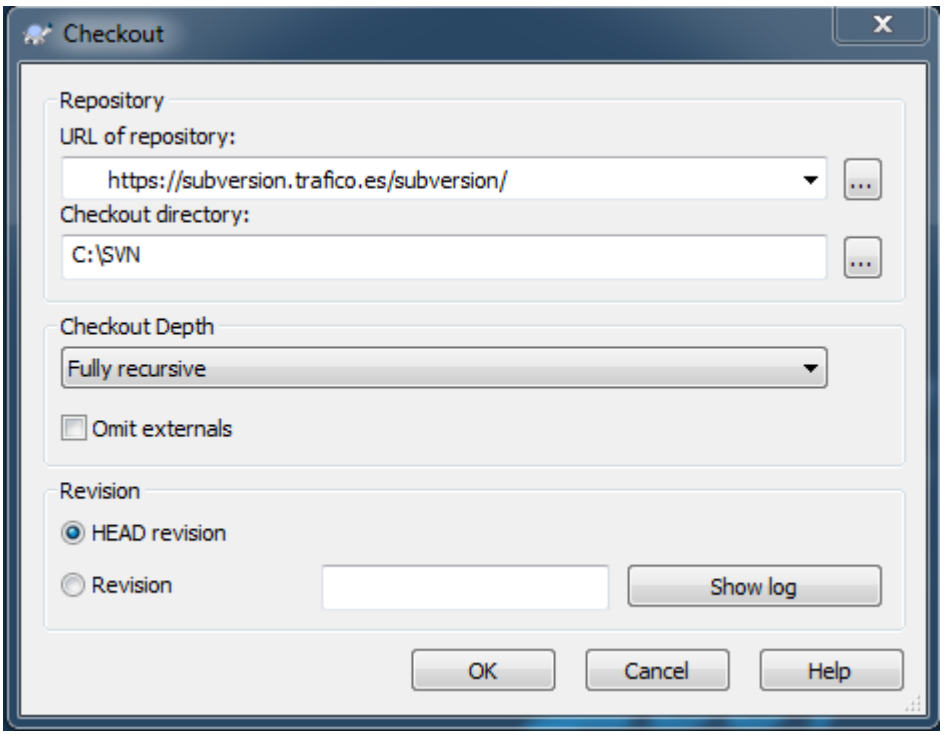

Ilustración 13: Checkout con TortoiseSVN

5.1.3 Añadir ficheros/carpetas al repositorio con TortoiseSVN

Para subir un fichero/carpeta al repositorio, en el explorador de proyectos nos posicionaremos en la ruta donde queremos incluir el archivo o carpeta y seleccionaremos la opción TortoiseSVN➤ Add, tal y como se muestra en la siguiente figura:

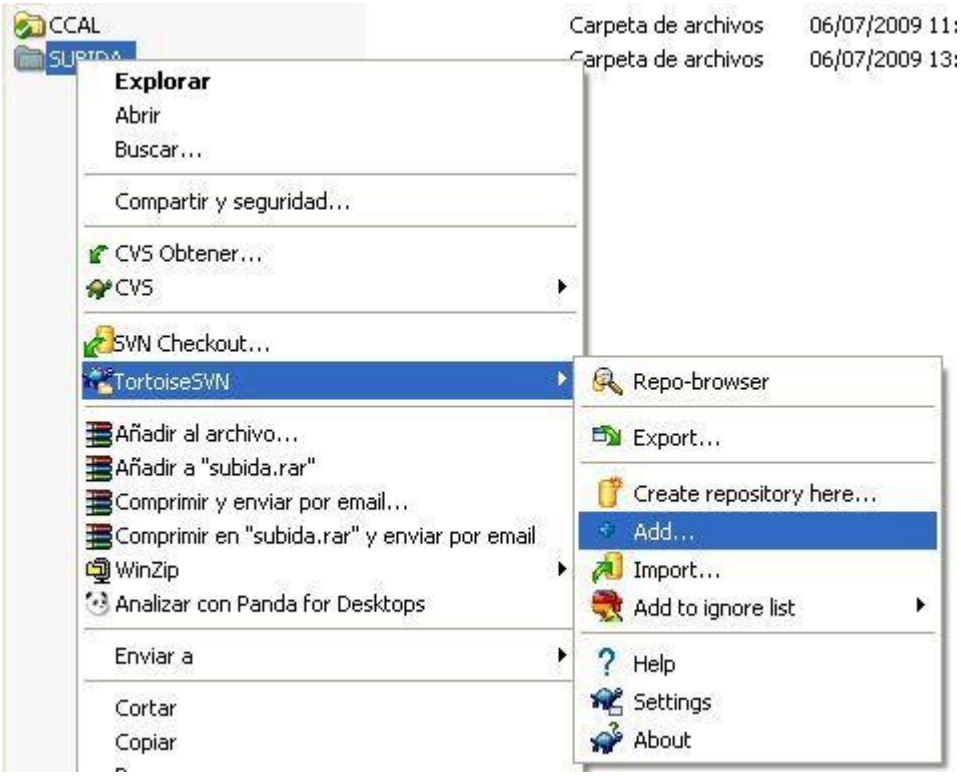

Ilustración 14: Añadir carpeta con TortoiseSVN

- Una vez añadida la carpeta se nos mostrará el siguiente icono:

Ilustración 15: Cambio de icono.

- Para confirmar la subida de la carpeta al repositorio, tendremos que confirmar la operación con la opción de menú “SVN Commit”.

Ilustración 16: Confirmar carpeta

5.1.4 Actualización de archivos en el repositorio con SVN

Con TortoiseSVN, una vez modificado un archivo se nos mostrará el siguiente icono:

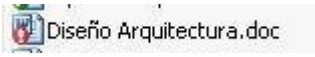

Ilustración 17: Estado de fichero modificado en Tortoise SVN

Para realizar la actualización del fichero en el repositorio se debe realizar la operación SVN **Commit**:

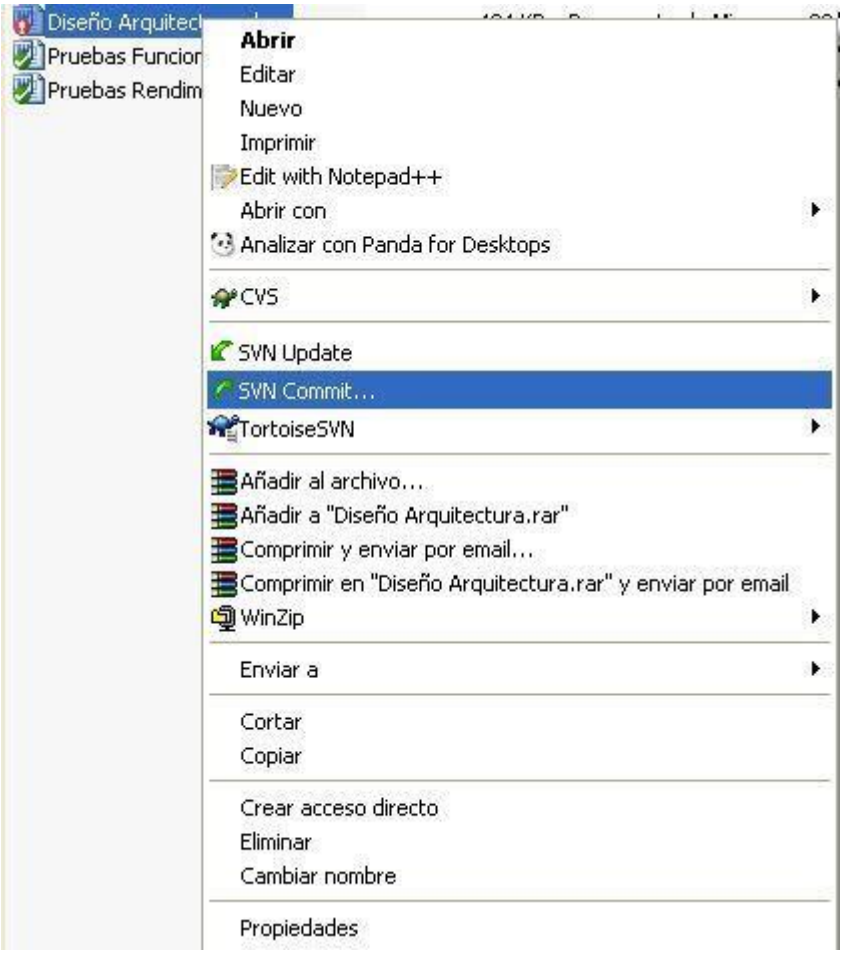

Ilustración 18: Actualización de ficheros en Tortoise SVN

Una vez realizado el **commit** el fichero quedará actualizado en el repositorio.

Ilustración 19: Fichero Actualizado en Tortoise SVN

5.1.5 Bloqueo de Ficheros en SVN

En SVN, es posible trabajar con apropiación de ficheros, de esta forma, se evitarán los conflictos de actualización del mismo fichero por distintas personas en un mismo momento.

Para realizar el bloqueo de un fichero seleccionaremos el fichero y en el menú contextual seleccionaremos la opción Get Lock

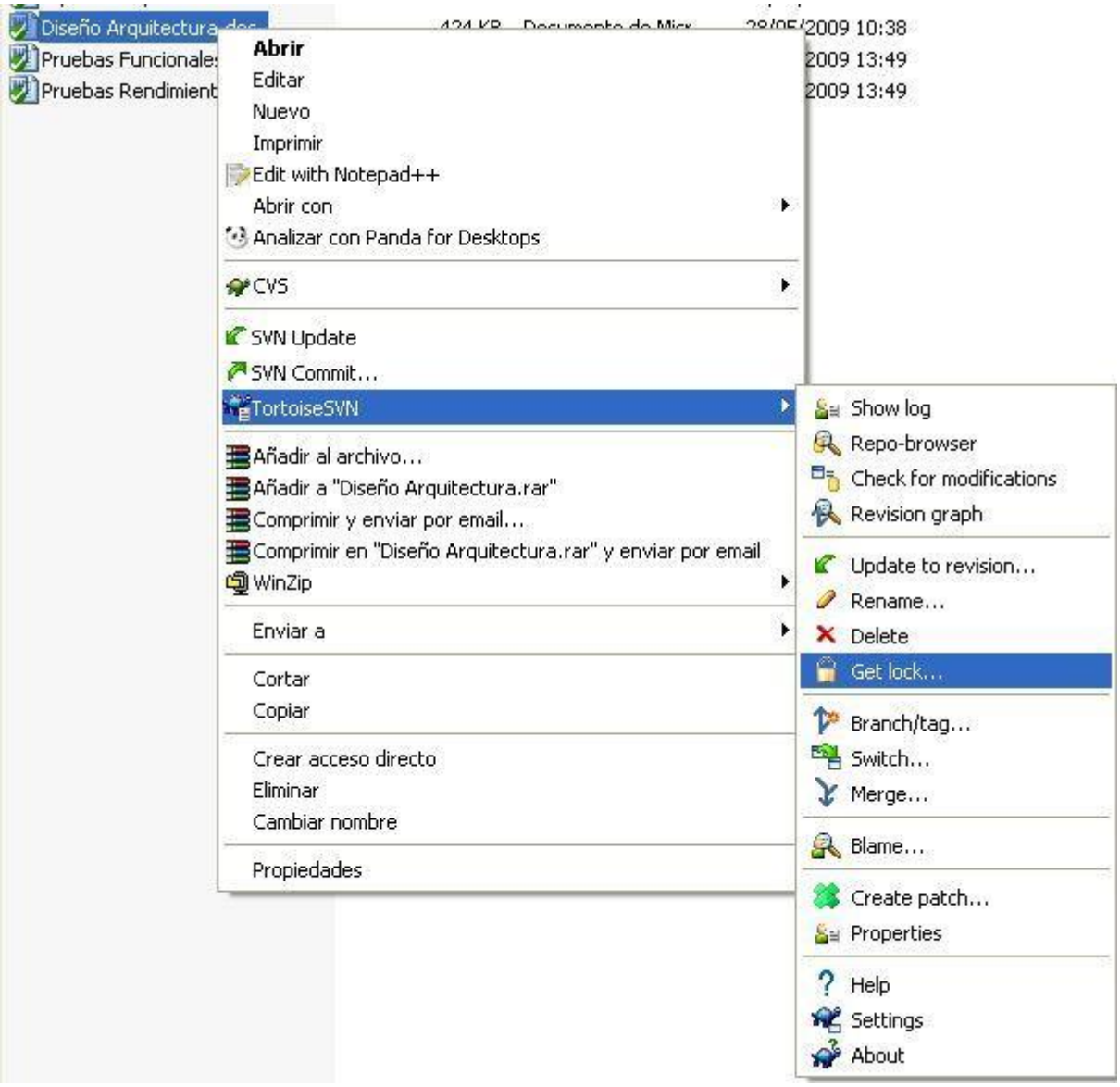

Ilustración 20: Bloqueo de ficheros en Tortoise SVN

Únicamente cuando un fichero este bloqueado nos aparecerá la opción de desbloquearlo:

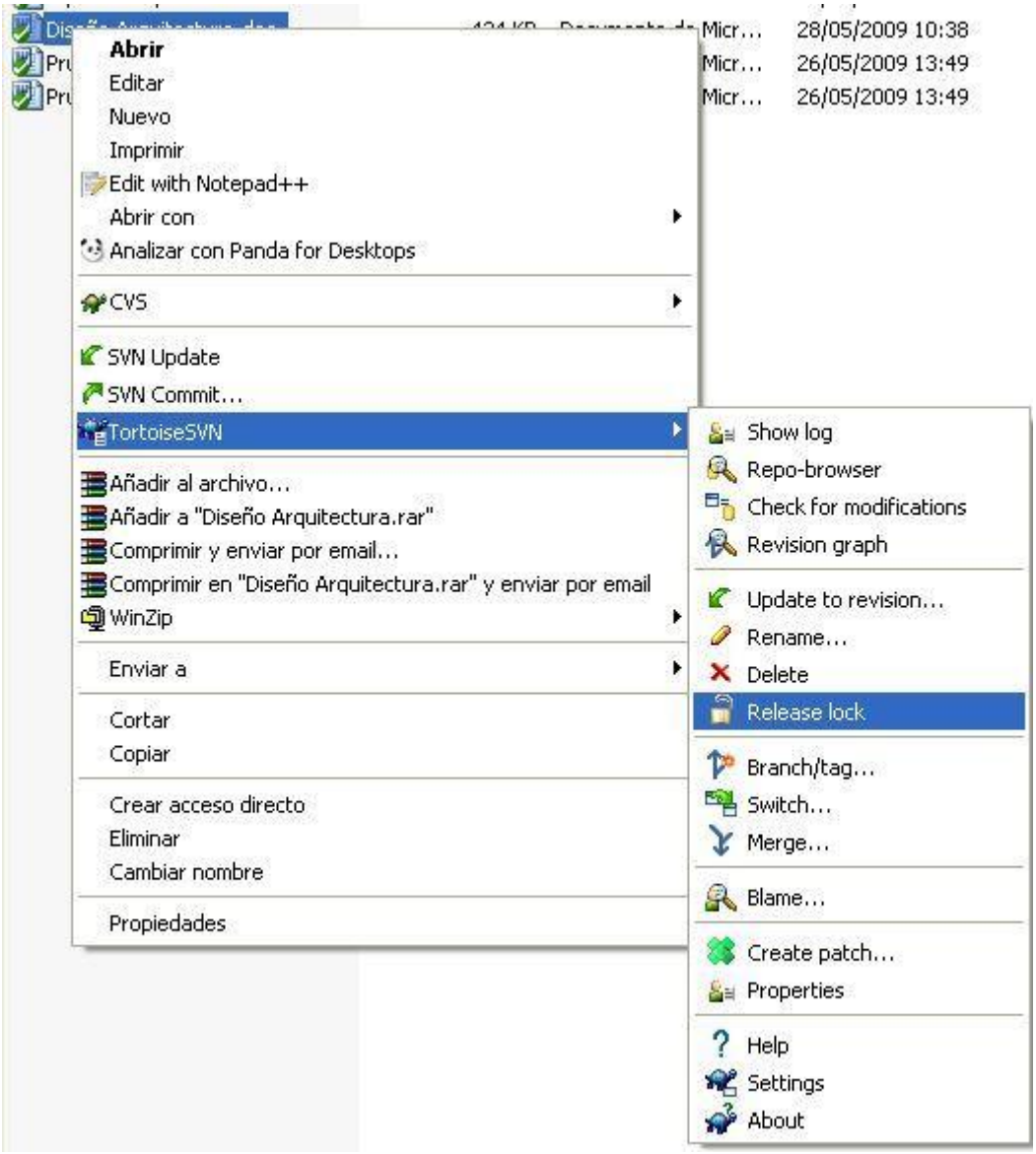

Ilustración 21: Desbloqueo de ficheros en Tortoise SVN

Cuando se hace un Commit y se tenía bloqueado el fichero queda liberado el lock.

5.1.6 Creación de una rama/etiqueta con TortoiseSVN

La creación de una rama/etiqueta conlleva la creación de una copia de toda la carpeta seleccionada en una ubicación diferente. La copia se hace mediante enlaces simbólicos, lo que quiere decir que no es una copia real de toda la información.

A continuación se describe con más detalle los pasos a seguir para la creación de una rama y/o de una etiqueta.

5.1.6.1 Creación de una etiqueta con TortoiseSVN

Existen varias maneras para realizar una etiqueta. A continuación mostramos una de ellas.

Para crear una etiqueta, nos posicionamos en la carpeta trunk del repositorio, y en la carpeta o archivos que queramos etiquetar hacemos clic con el botón derecho. Después se selecciona la entrada de TortoiseSVN denominada “Branch/tag”. Con esta opción se nos muestra la siguiente ventana:

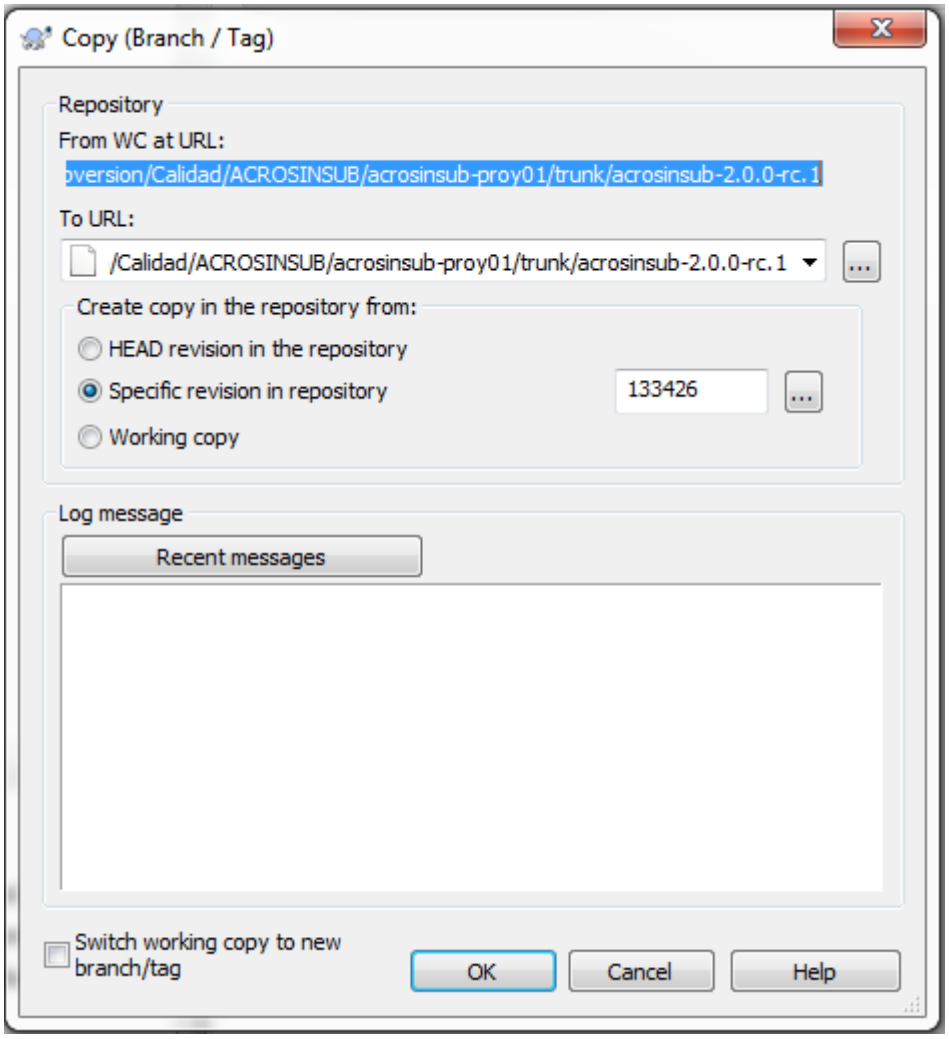

Ilustración 22: Creación de una etiqueta

En esta ventana aparecen los siguientes campos:

- From WC at URL: Indicamos la carpeta que queremos etiquetar.
- To URL: Indicamos la carpeta destino de la etiqueta. Para ello vamos a seguir los siguientes pasos:
    - Se debe tener en cuenta que todas las etiquetas del repositorio se deben ubicar en la carpeta *tags*, además se debe cumplir el estándar de etiquetado especificado en la guía de desarrollo, por lo que el primer paso será la creación de la carpeta con el nombre de la etiqueta correspondiente dentro de la carpeta *tags*, para la creación de dicha carpeta pulsaremos el botón de explorar directorios de la pantalla anterior y se mostrará la siguiente ventana :

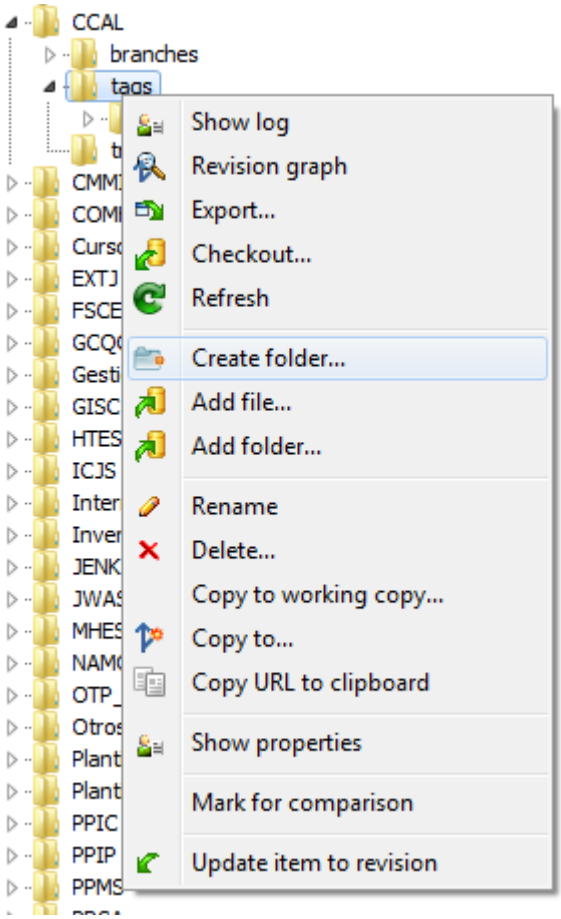

Ilustración 23: Creación de carpeta de etiqueta

    - Seleccionaremos la carpeta *tags*, pulsaremos botón derecho para desplegar el menú contextual y elegiremos la opción Create folder.

- Crearemos una carpeta con el nombre de la etiqueta (acrosinsub-2.0.0-rc.2, por ejemplo), y seleccionaremos esa carpeta como destino de nuestra etiqueta.

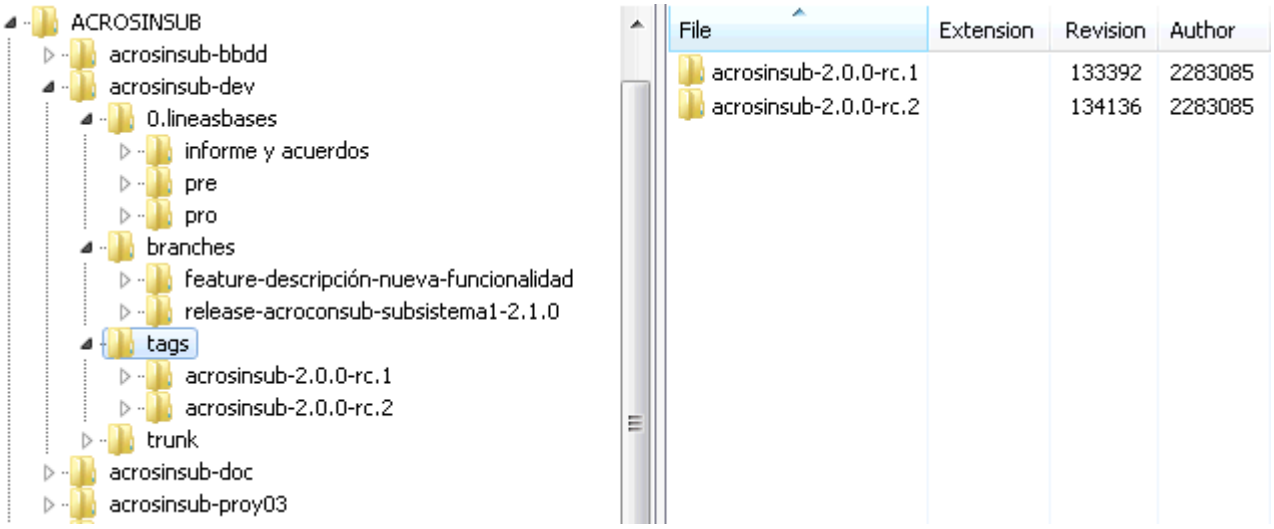

Ilustración 24: Carpeta destino etiqueta

- En esta ruta, añadiremos la carpeta que queremos etiquetar (dev)

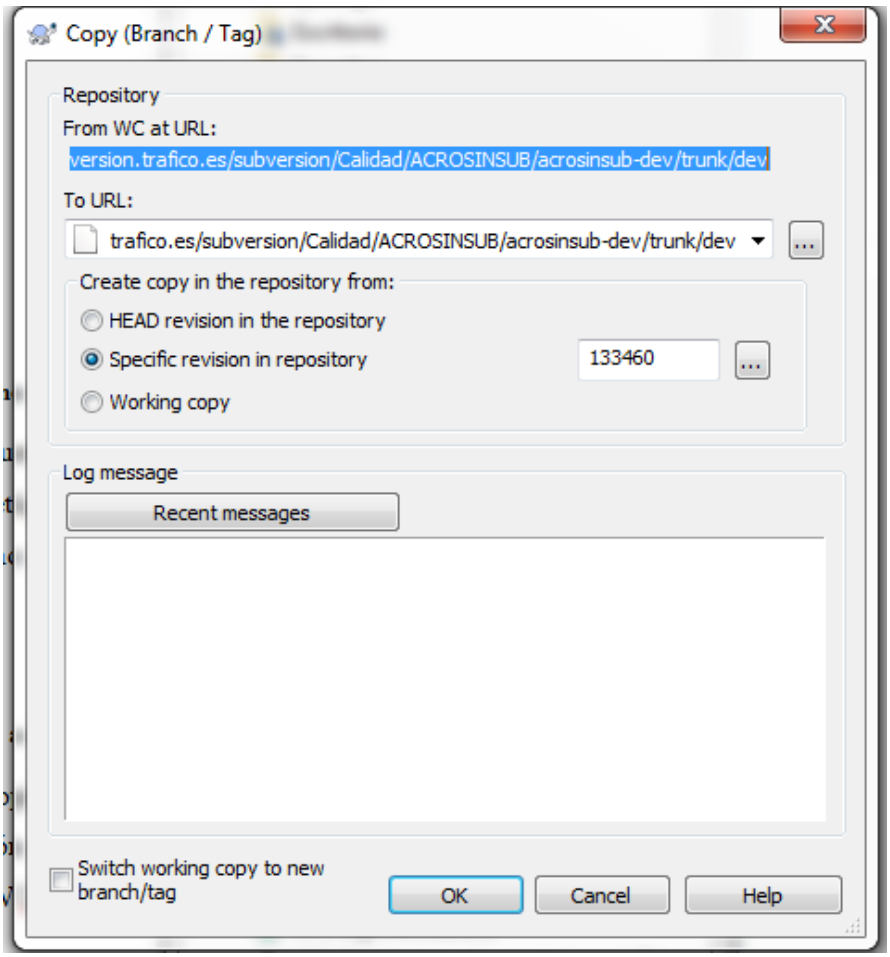

Ilustración 25: incluir carpeta a etiquetar

Se debe elegir la opción Working copy si se quiere etiquetar la copia de trabajo, se debe realizar además un commit previo, ya que si no la etiqueta no se aplicará al contenido que no esté en dicho estado.

- Pulsaremos el botón OK, y se creará la copia etiquetada en el repositorio en el directorio *tags*. Para comprobar la creación de dicha etiqueta , solo tendremos que actualizar la carpeta *tags* (realizar un *SVN Update*)

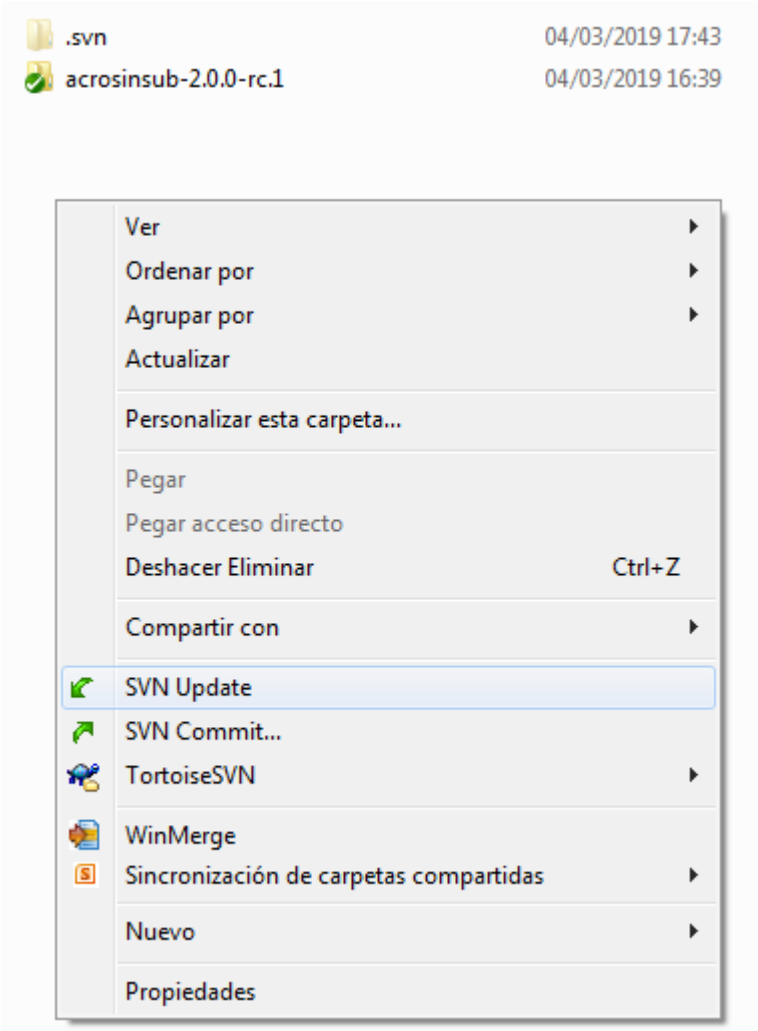

Ilustración 26: Actualización de la carpeta

5.1.6.2 Creación de una rama con TortoiseSVN

Existen varias maneras para realizar una rama. A continuación mostramos una de ellas.

Para crear una rama, nos posicionamos en la carpeta trunk del repositorio, y en la carpeta o archivos que queramos enramar. Después se selecciona la entrada de TortoiseSVN denominada “Branch/tag”. Con ésta opción se nos muestra la siguiente ventana:

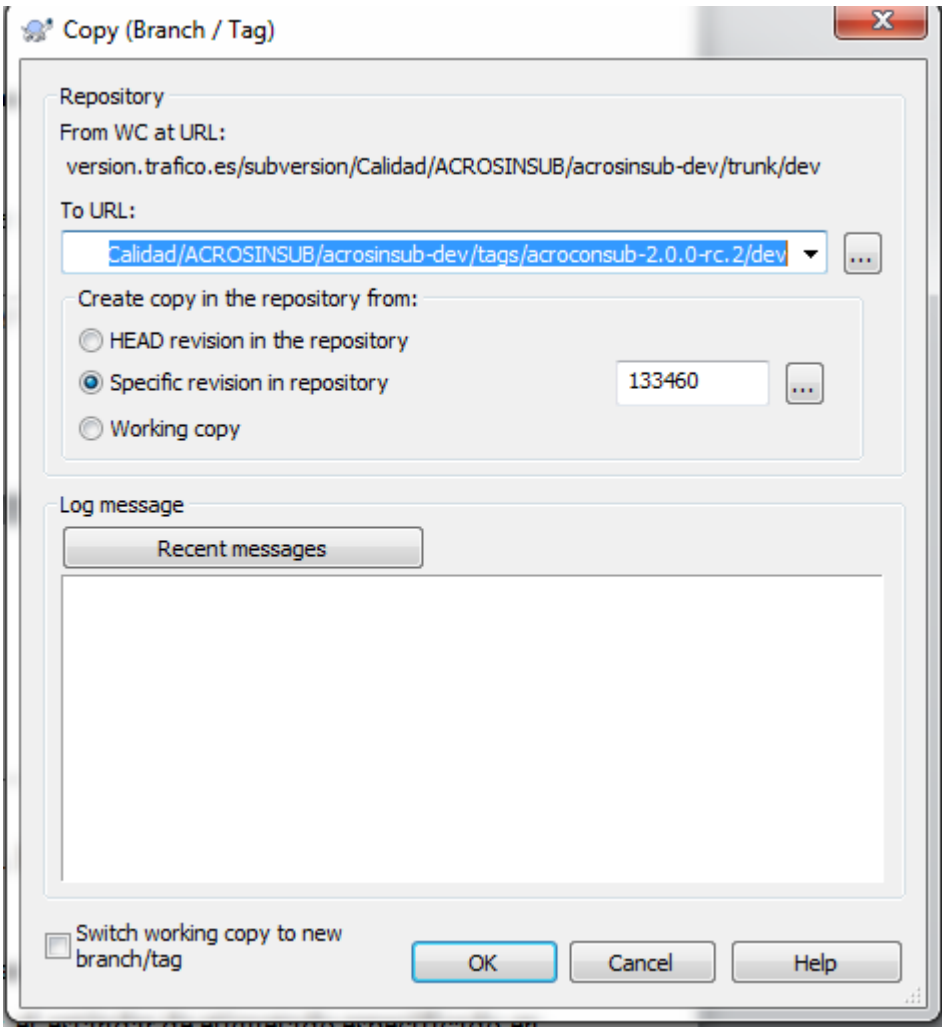

Ilustración 27: Creación de una etiqueta

En esta ventana aparecen los siguientes campos:

- <u>From WC at URL</u>: Indicamos la carpeta de la que queremos crear la rama.
- <u>To URL:</u> Indicamos la carpeta destino de la rama. Para ello vamos a seguir los siguientes pasos:
    - Se debe tener en cuenta que todas las ramas del repositorio se deben ubicar en la carpeta *branches*, además se debe cumplir el estándar de etiquetado especificado en la guía de desarrollo, por lo que el primer paso será la creación de la carpeta con el nombre de la rama correspondiente dentro de la carpeta *branches*, para la creación

de dicha carpeta pulsaremos el botón de explorar directorios de la pantalla anterior y se mostrará la siguiente ventana :

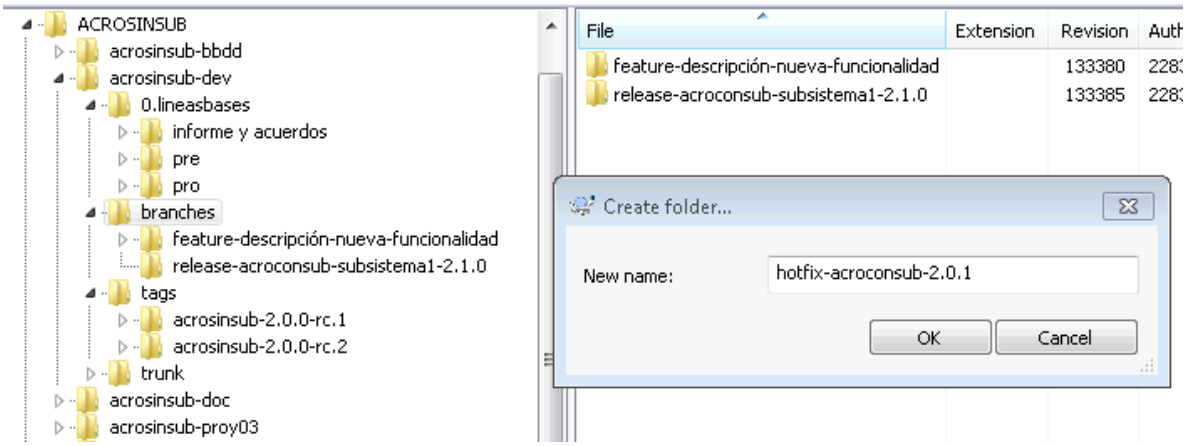

Ilustración 28: Creación de carpeta de rama

- Seleccionaremos la carpeta *branches*, pulsaremos botón derecho para desplegar el menú contextual y elegiremos la opción Create folder.
- Crearemos una carpeta con el nombre de la rama (hotfix-acrconsub-2.0.1, por ejemplo), y seleccionaremos esa carpeta como destino de nuestra rama.
- En esta ruta, añadiremos la carpeta a enramar (dev)

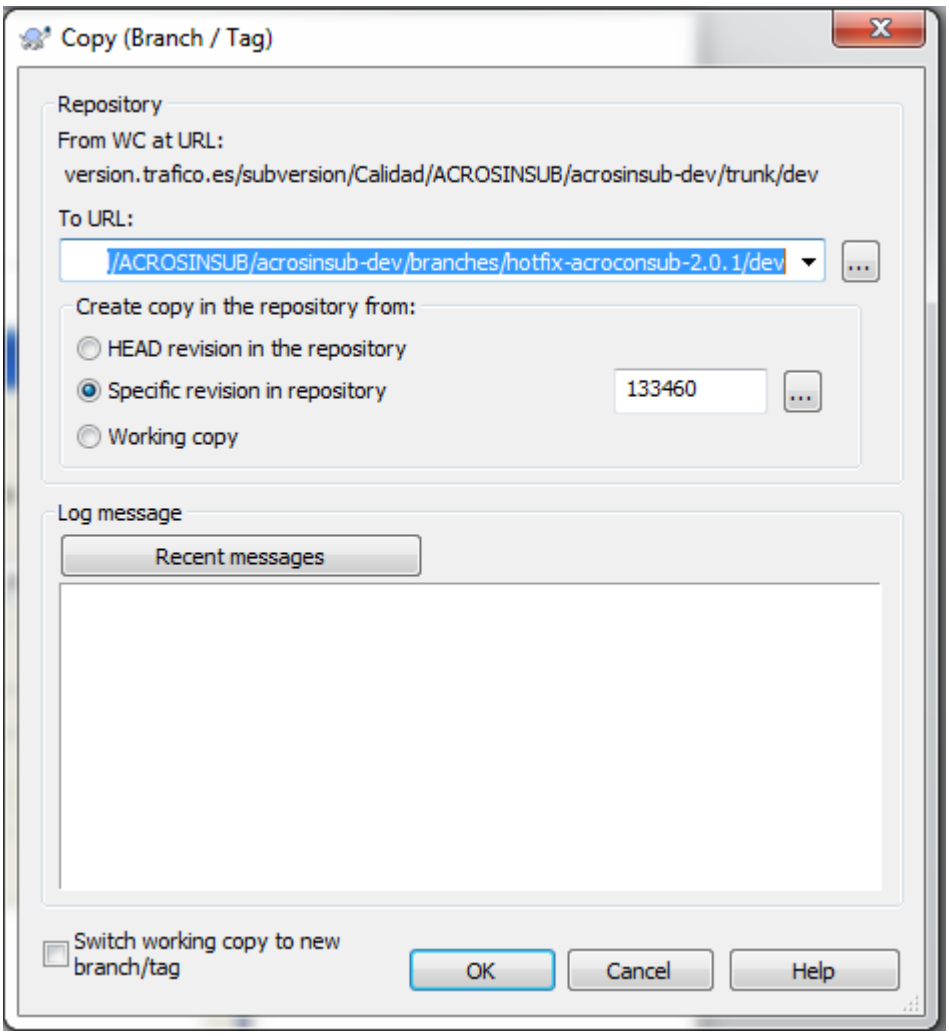

Ilustración 29: Carpeta destino de la rama

- Pulsaremos el botón OK, y se creará la copia etiquetada en el repositorio en el directorio *branches*. Para comprobar la creación de dicha etiqueta , solo tendremos que actualizar la carpeta *branches* (realizar un *SVN Update*)

5.1.7 Fusión de ramas con SVN

Con SVN, como se explica en el apartado anterior, existe la posibilidad de trabajar con ramas, para la fusión de dichas ramas se deben seguir los siguientes pasos:

- Nos posicionamos en la copia de trabajo, carpeta *trunk*, en la que queremos actualizar los cambios de la rama y en el menú contextual seleccionamos la opción Merge. Nos aparecerá la siguiente pantalla:

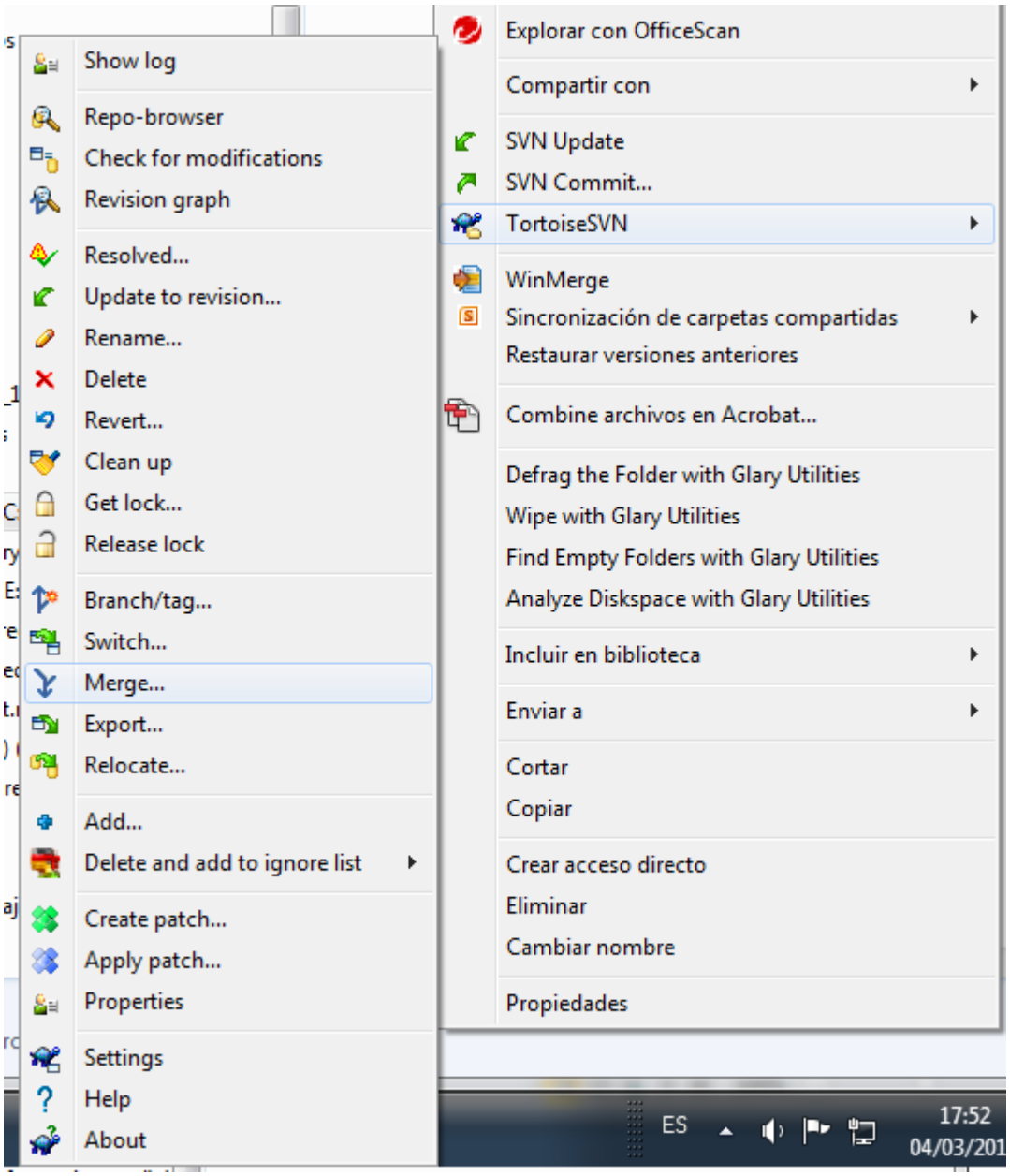

Ilustración 30: Fusión de ramas

- Seleccionaremos la versión de la carpeta de trabajo que queremos fusionar con la rama. Para ello, pulsaremos el botón *Show log* y elegiremos la versión a fusionar
- Pulsaremos el botón Ok, y volveremos a la pantalla anterior, donde ya tendremos la *Revision* a fusionar.

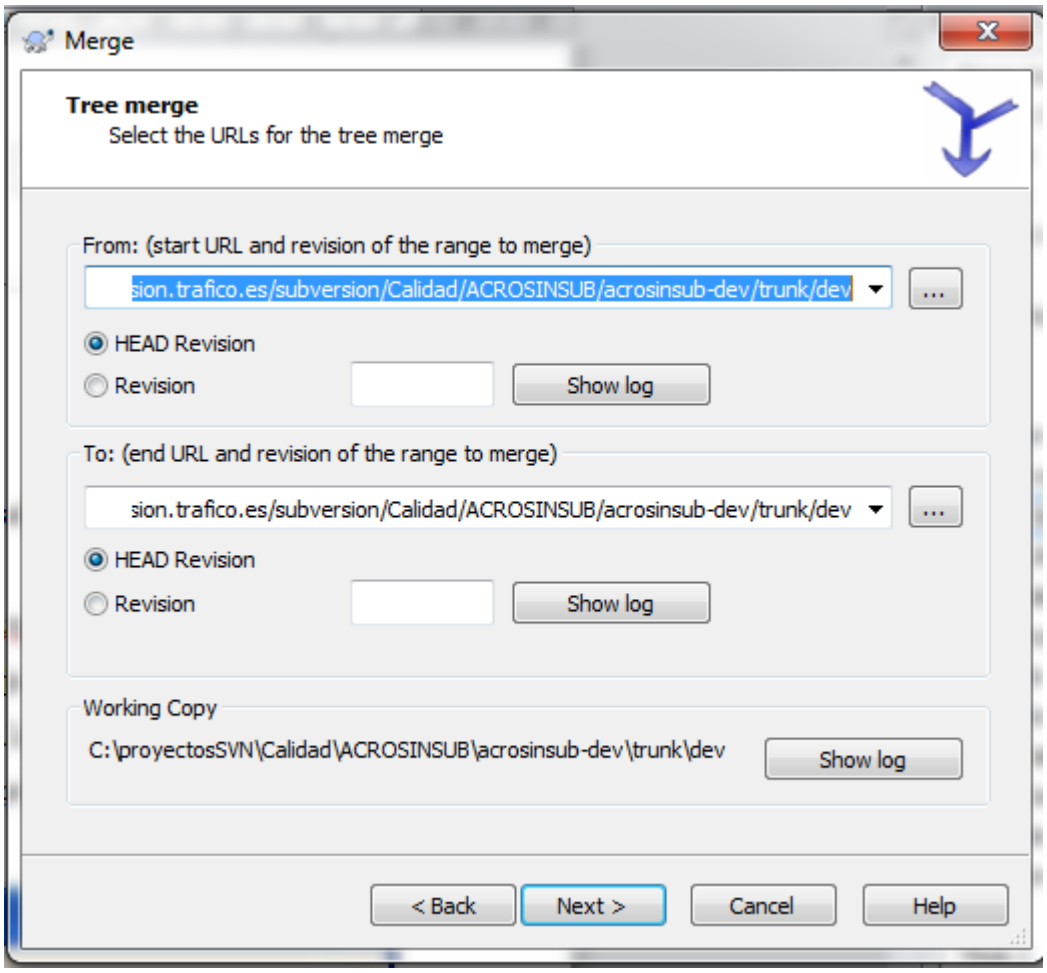

Ilustración 31: Inclusión de la versión

- TortoiseSVN, por defecto, indica una versión menos de la seleccionada, con lo que **habrá que indicar una más** de la que indica por defecto.
- En el apartado "*To(en URL and revisión of the range to merge*) ", indicaremos la carpeta de la rama que queremos fusionar con nuestra carpeta de trabajo (seleccionaremos la rama de la carpeta *branches*). Para ello, desmarcaremos la casilla "*Use From:URL (revisión range is selectable in the above log dialog)*" , y marcaremos la opción "*HEAD revision*".

- Una vez tengamos toda la información rellena, pulsaremos el botón Merge, y la fusión se realizará en local. Una vez que la fusión se ha realizado en local, se realizará la operación SVN Commit para terminar el proceso.

5.1.8 Renombrado de Ficheros/Carpetas en TortoiseSVN

Con TortoiseSVN existe la posibilidad de renombrar archivos y/o carpetas sin perder el historial de los mismos. Para ello se deben seguir los siguientes pasos:

- Nos posicionamos en el fichero o carpeta a renombrar, pulsamos botón derecho, opción TortoiseSVN, opción Rename como se indica en la siguiente imagen:

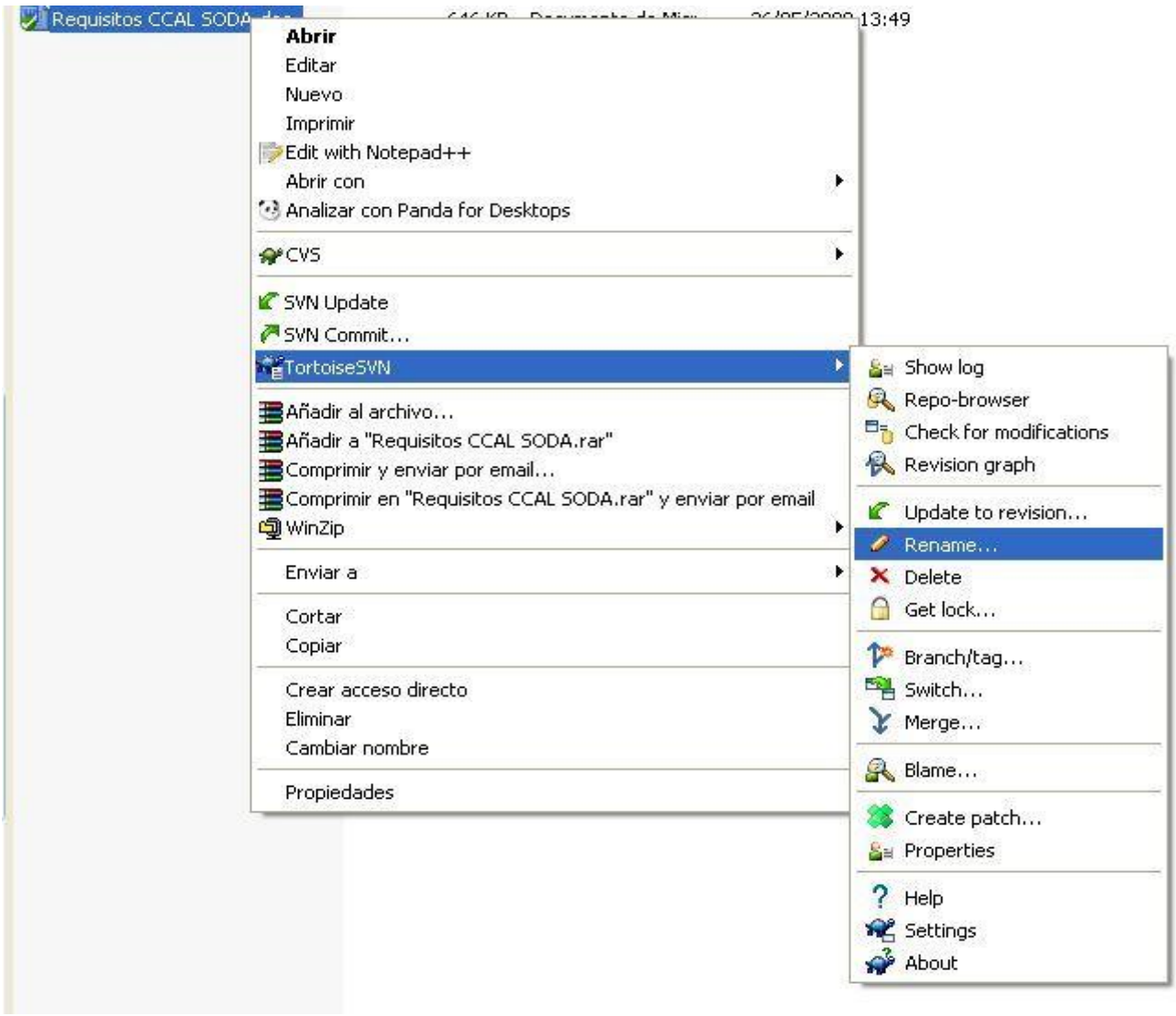

Ilustración 32: Selección del fichero a Renombrar

- Una vez seleccionada la opción introduciremos el nuevo nombre del fichero carpeta.
- Una vez renombrado, se realiza un SVN Commit para guardar los cambios en el servidor.

5.1.9 Reubicación de Ficheros/Carpetas en TortoiseSVN

Con TortoiseSVN existe la posibilidad de mover archivos y/o carpetas sin perder el historial de los mismos. Para ello se deben seguir los siguientes pasos:

- Seleccionamos con el botón derecho del ratón carpeta/fichero a reubicar y sin soltarlo lo arrastramos a la nueva ubicación.

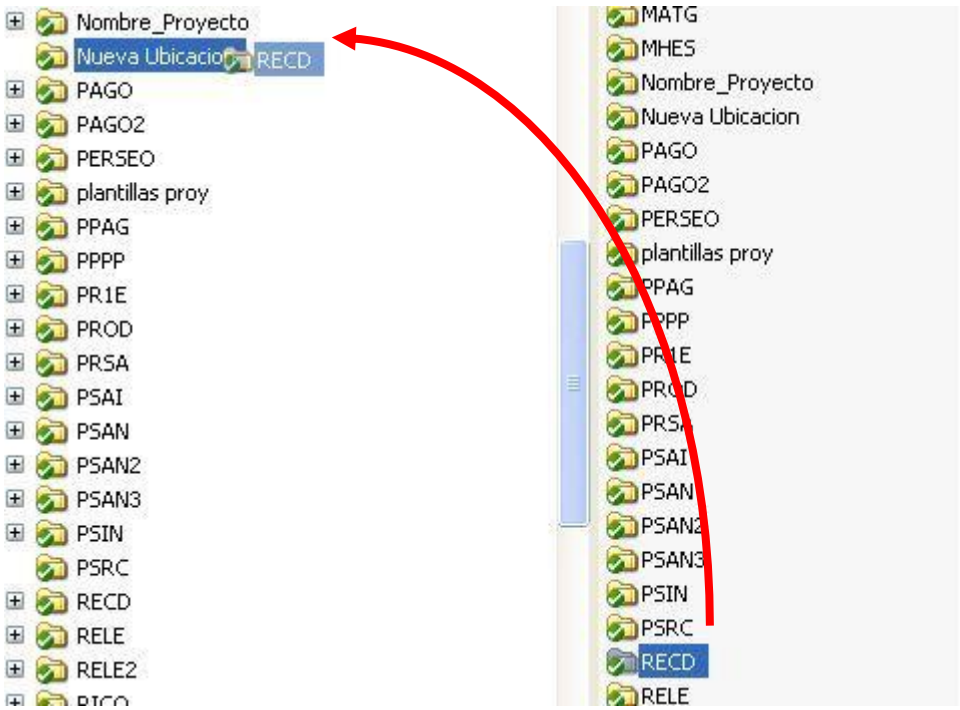

Ilustración 33: Selección de la carpeta a reubicar

- Una vez que soltemos la carpeta en la nueva ubicación nos aparecerá el siguiente menú contextual:

Ilustración 34: Menú Contextual Mover Carpeta

- Seleccionamos la opción marcada en la imagen anterior, "SVN Move versioned files here", y la carpeta quedará reubicada.
- Se debe realizar un SVN Commit tanto en la nueva ubicación como en la anterior para guardar los cambios:

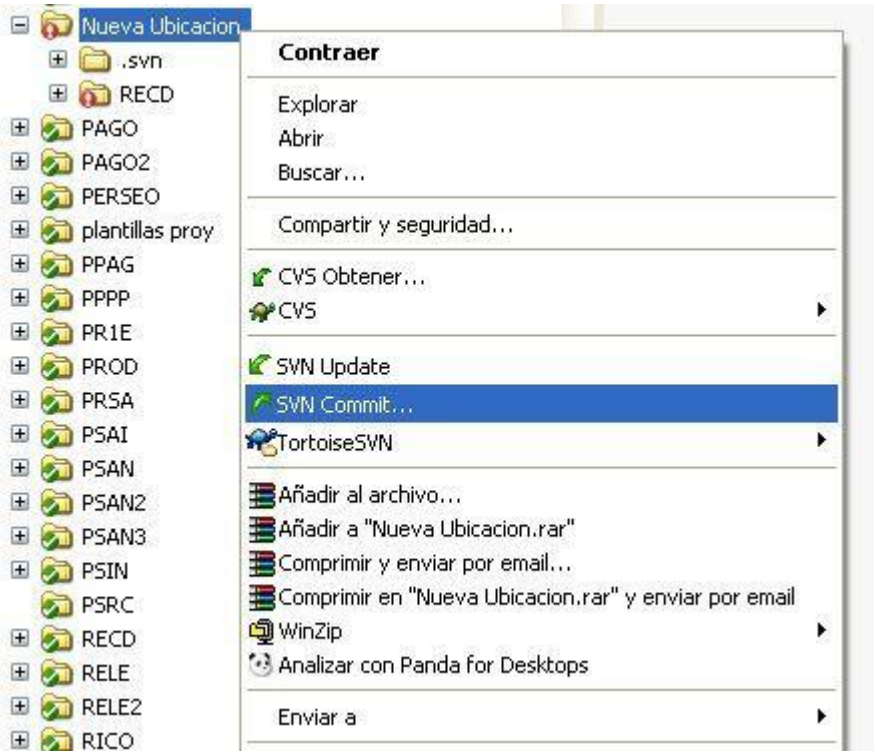

Ilustración 35: Confirmación en la nueva ubicación

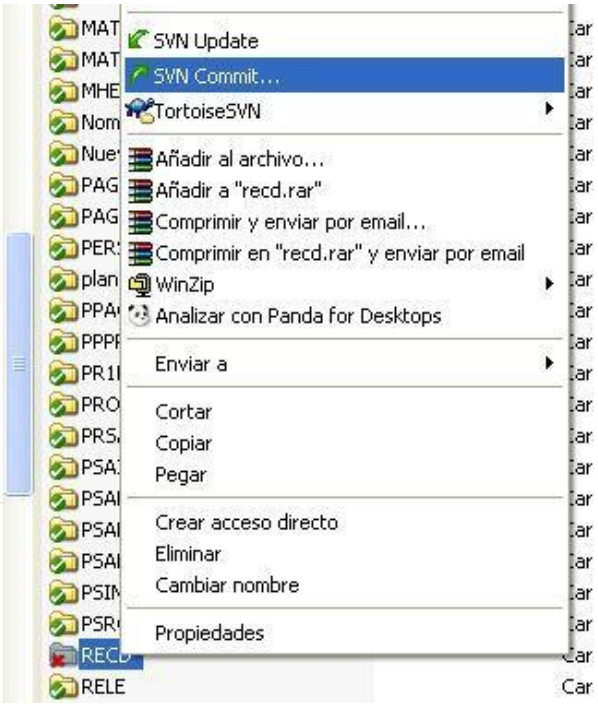

Ilustración 36: Confirmación en la antigua ubicación

5.2 Plugin de Subversion para Eclipse

Tigris.org, la misma comunidad que desarrolla SVN, también mantiene **Subclipse**, un plugin que integra SVN en Eclipse.

5.2.1 Instalación del Plugin

Antes de la instalación del plugin se debe configurar el proxy en RSA, ya que para la instalación necesitamos la conexión a internet desde RSA.

Para configurar el proxy se deben seguir los siguientes pasos:

Acceder al menú Ventana→ Preferencias→ Internet→Valores de proxy.

Nos aparecerá la ventana de configuración del proxy donde deberemos indicar los parámetros que aparecen en la siguiente imagen:

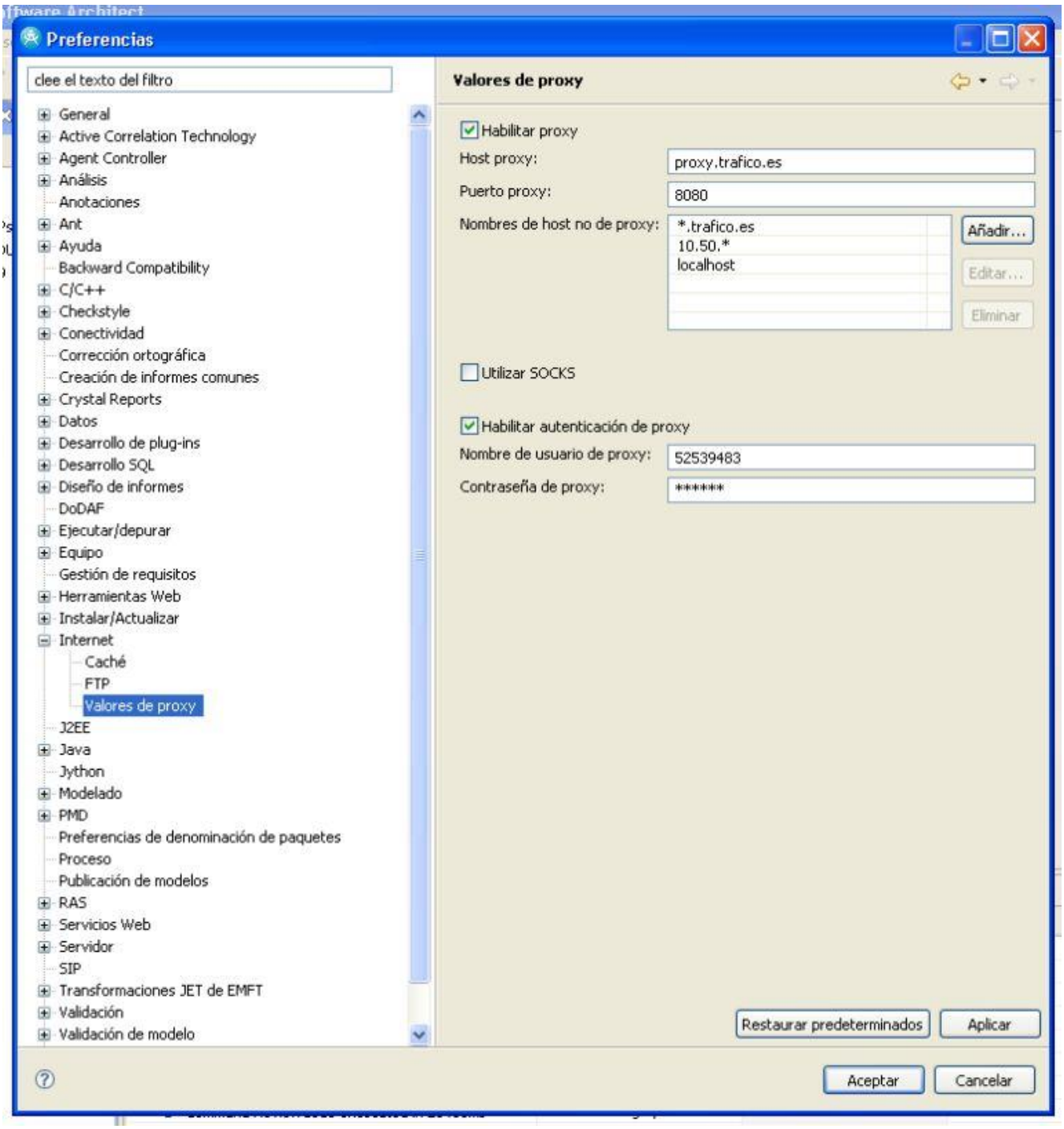

Ilustración 37: Configuración del Proxy en RSA

A continuación se describen los pasos a seguir para realizar la instalación de Subclipse:

Acceder al menú Ayuda→ Actualizaciones de Software→ Buscar e instalar.

Y seleccionamos buscar características nuevas para instalar, pulsando siguiente.

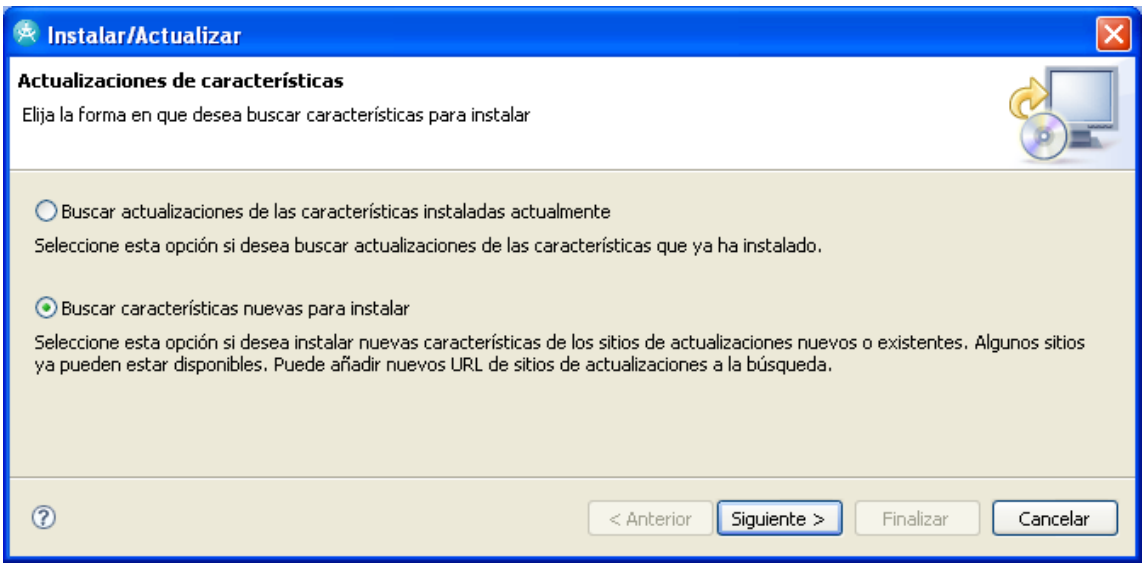

Ilustración 38: Instalación Subclipse

Seleccionamos sitio remoto nuevo e introducimos:

Nombre: Subclipse

URL: http://subclipse.tigris.org/update_1.6.x/

Ahora seleccionamos el nuevo sitio en la lista y le damos a finalizar.

A continuación, dentro de subclipse, seleccionamos solo Subclipse y desechamos Integrations. Aceptamos los términos de licencia y pulsamos finalizar.

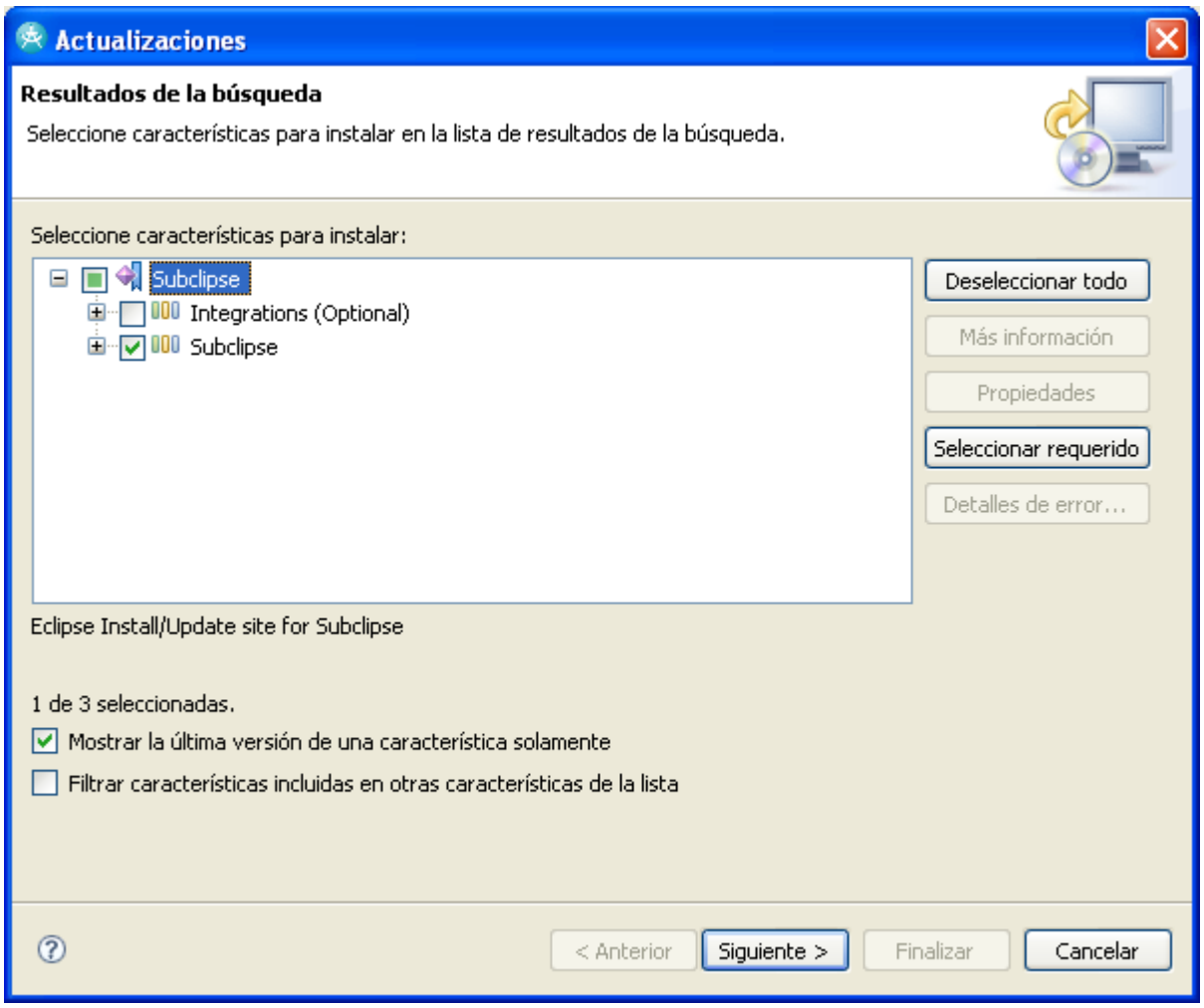

Ilustración 39: Selección de módulos

De esta manera, ya se tendría instalado el plugin Subclipse de SVN para Eclipse y estaría listo para utilizar.

5.2.2 Utilización del Plugin

Una vez instalado el plugin, se nos ha añadido una nueva perspectiva *"SVN Repository Exploring"*, y ya podemos utilizar subclipse para realizar checkouts y commits de nuestros proyectos desde el entorno de Eclipse. A continuación se describe como hacer estas tareas con el nuevo plugin:

5.2.3 Importar un proyecto SVN existente

Desde la perspectiva *"SVN Repository Exploring"* y dentro de la pestaña SVN Repository, pulsando botón derecho del ratón elegimos *"new repository location"* e introducimos la dirección del proyecto dentro del repositorio.

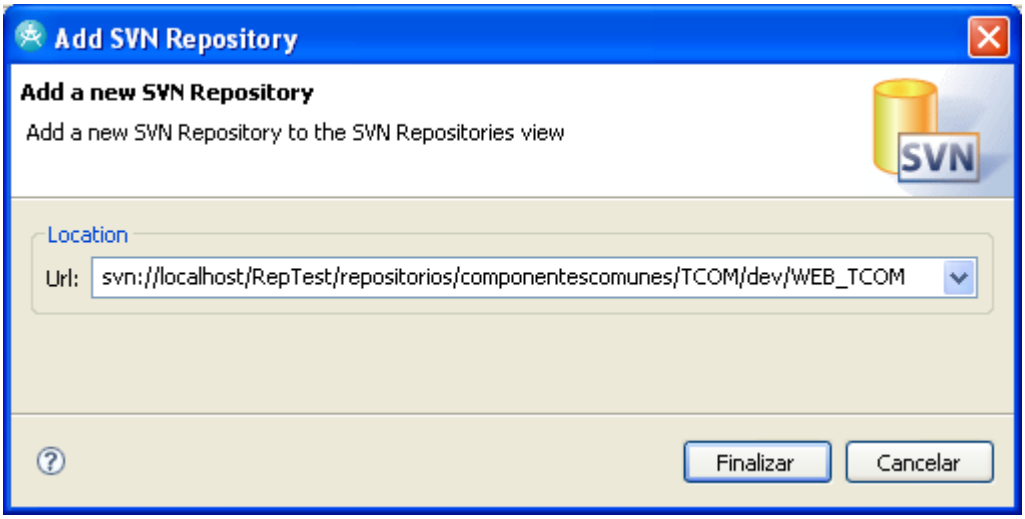

Ilustración 40: Selección de la dirección del repositorio

Ahora en la pestaña SVN Repository habrá aparecido una nueva entrada con la nueva ruta introducida para el repositorio. Pulsando botón derecho sobre la nueva entrada, seleccionamos "*checkout*", para crear un nuevo proyecto en el espacio de trabajo, pudiendo seleccionar el número de revisión que queremos obtener del repositorio.

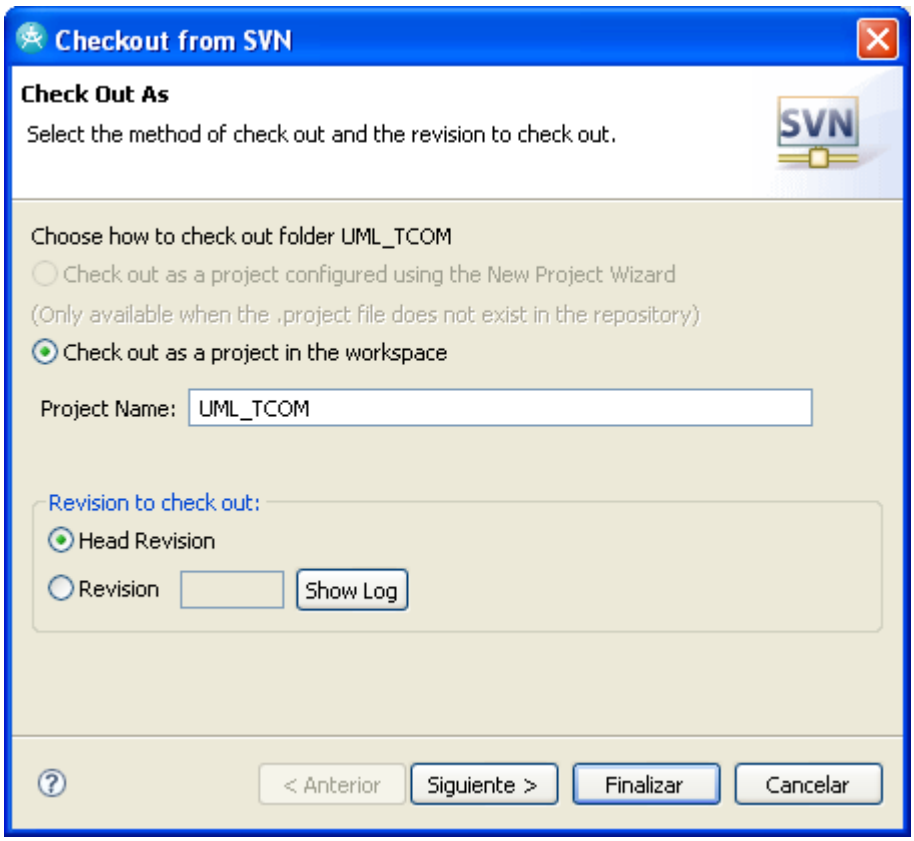

Ilustración 41: Realizar el checkout del proyecto

A continuación si accedemos a la perspectiva Java, ya dispondríamos del proyecto en el "Explorador de Paquetes" para poder trabajar con él.

5.2.4 Importar un proyecto SVN existente con selección de carpetas.

Desde la perspectiva *"SVN Repository Exploring"* y dentro de la pestaña SVN Repository, pulsando botón derecho del ratón elegimos *"new repository location"* e introducimos la dirección del proyecto dentro del repositorio.

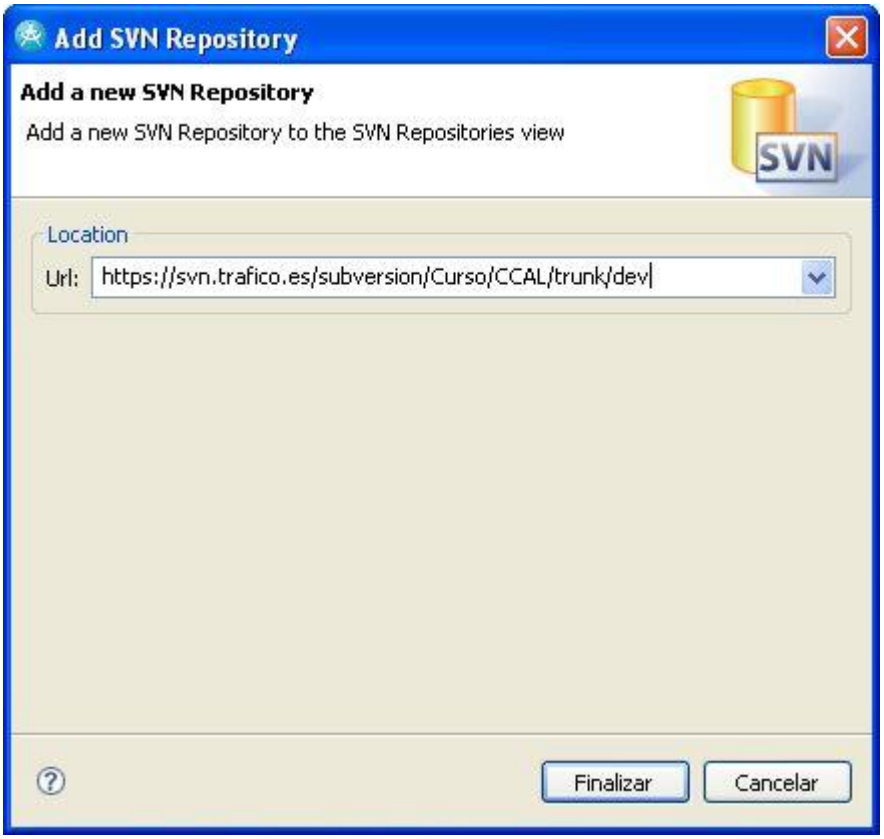

Ilustración 42: Selección de la dirección del repositorio

Ahora en la pestaña SVN Repository aparecerá una nueva entrada con la nueva ruta introducida para el repositorio. Pulsando sobre la nueva entrada, se expandirán las carpetas existentes que podrán ser seleccionadas por el usuario a su conveniencia.

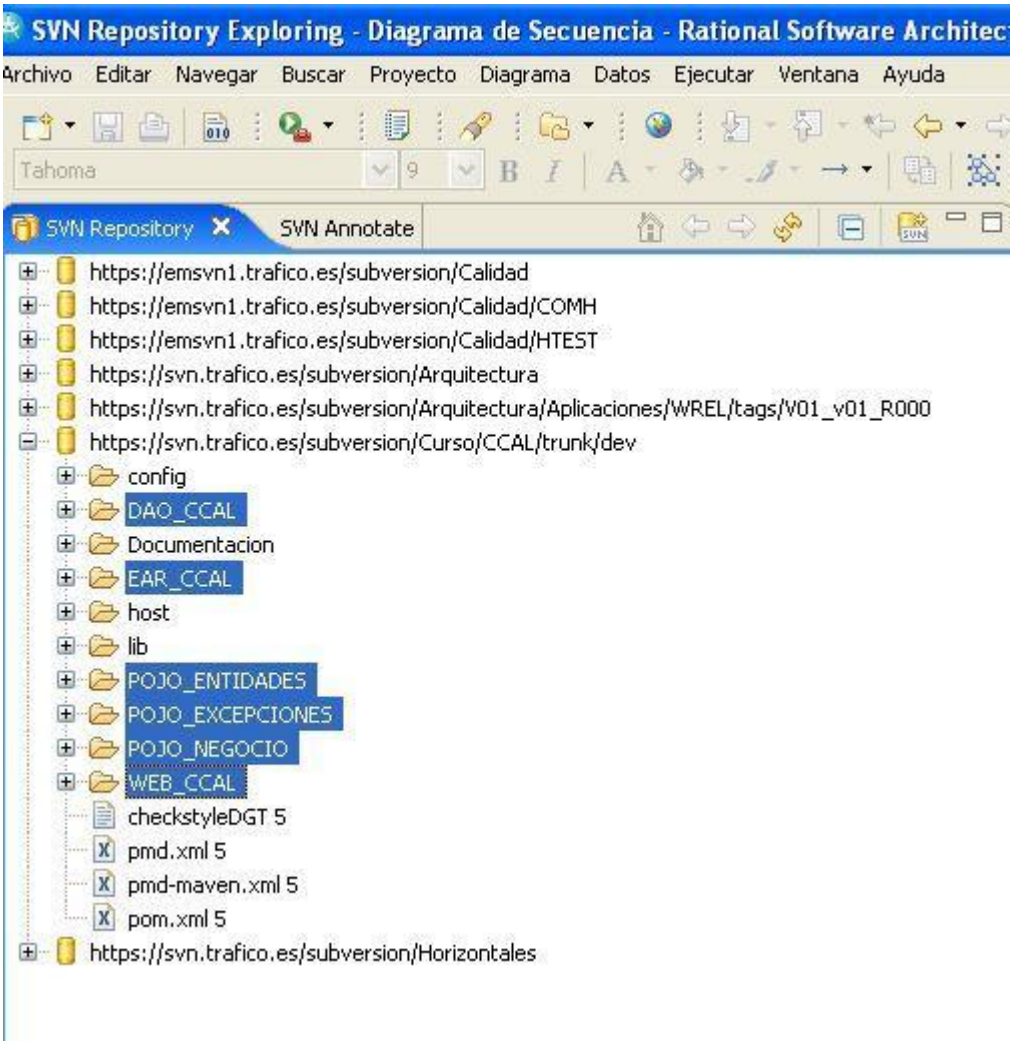

Ilustración 43: Selección de las carpetas/módulos

Con la selección realizada pulsamos la opción "*checkout*" sobre el menú de contextual, para crear el conjunto de proyectos en el espacio de trabajo, pudiendo seleccionar el número de revisión que queremos obtener del repositorio.

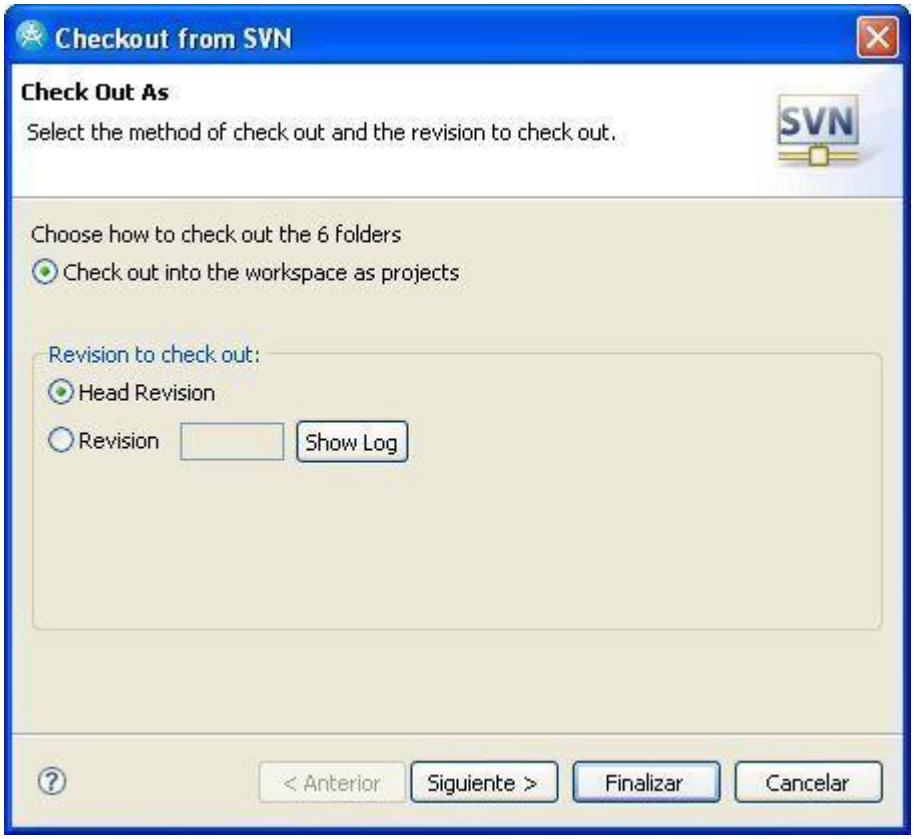

Ilustración 44: Realizar el checkout de los proyectos

A continuación si accedemos a la perspectiva Java, ya dispondríamos de los proyectos en el "Explorador de Paquetes" para poder trabajar con ellos.

5.2.5 Añadir un proyecto de Eclipse a SVN

Igual que se puede importar a Eclipse un proyecto de SVN, también se puede hacer lo contrario: crear un proyecto en un repositorio de SVN desde un proyecto local de Eclipse. Para ello se seguirán los siguientes pasos:

En el "Explorador de Paquetes", pulsando el botón derecho sobre el proyecto que queremos exportar, seleccionamos Equipo→Compartir proyecto.

Seleccionamos como tipo de repositorio SVN e introducimos una nueva URL para el repositorio en el que queremos subir el proyecto. Esto nos pedirá usuario y password.

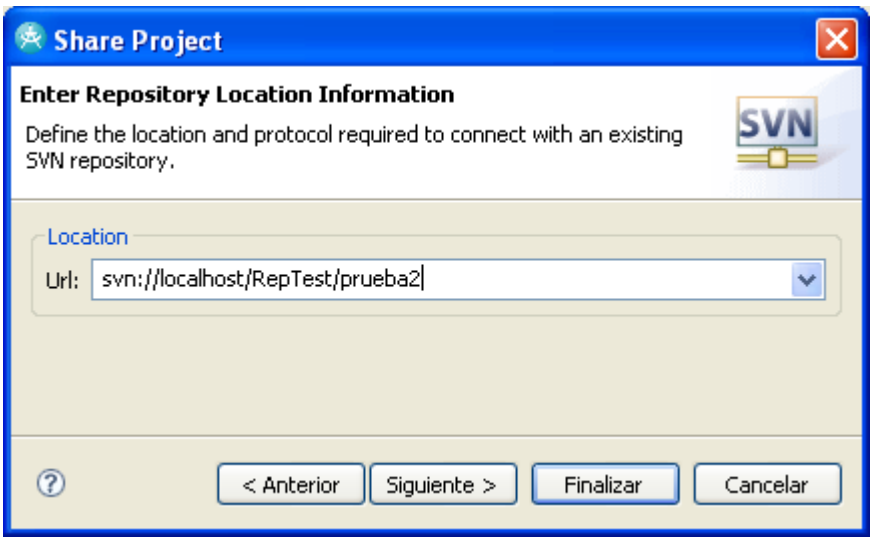

Ilustración 45: Selección del repositorio donde subir el proyecto

Aparecerá una pantalla que nos permitirá seleccionar los archivos del proyecto que queremos subir así como de incluir un comentario que haga referencia a la subida.

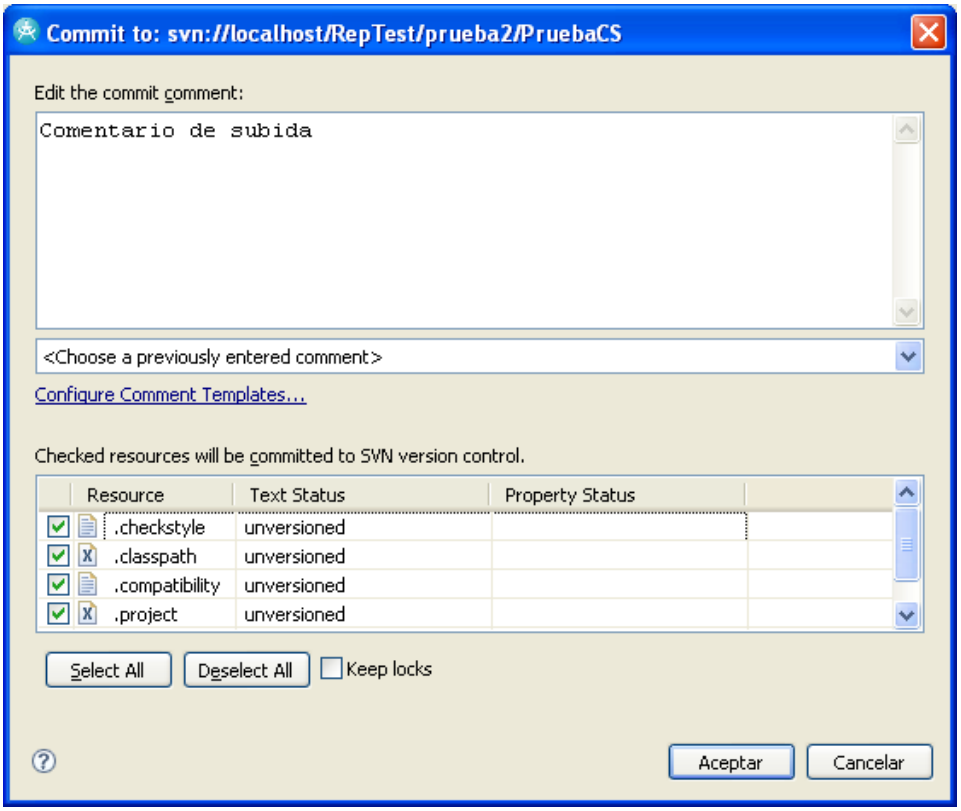

Ilustración 46: Selección de archivos e introducción de comentario

5.2.6 Interfaz hacia SVN desde Eclipse

Una vez que se dispone de una copia local del proyecto vinculada a un repositorio, el manejo de svn es directo. Se trabaja con los ficheros del proyecto en local hasta que se quiera hacer un *update* para actualizar la copia local o un *commit* para subir los cambios. Para tener acceso a las acciones que permite SVN, desde el "Explorador de Paquetes" pulsamos el botón derecho sobre el archivo, conjunto de archivos o el proyecto completo y seleccionamos "Equipo". Se nos mostrara un menú con las opciones de la siguiente figura:

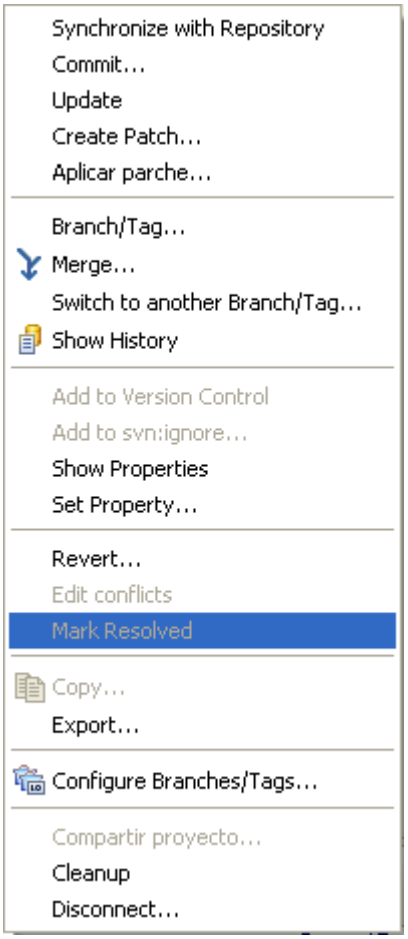

Ilustración 47: Menú de opciones de SVN desde Eclipse

Básicamente esto es todo lo necesario para poder utilizar SVN desde Eclipse mediante el plugin de Subclipse.