

TortoiseSVN

Un cliente de Subversion para Windows

Version 1.4.0

**Stefan Küng
Lübbe Onken
Simon Large**

TortoiseSVN: Un cliente de Subversion para Windows: Version 1.4.0

por Stefan Küng, Lübbe Onken, y Simon Large

Traducción: Fernando P. Najera Cano (yo@FernandoNajera.com)

publicado 2006/09/26 12:24:40 (r7622)

Tabla de contenidos

Prefacio	xi
1. Audiencia	xi
2. Guía de Lectura	xi
3. ¡TortoiseSVN es gratis!	xii
4. Comunidad	xii
5. Reconocimientos	xii
6. Terminología usada en este documento	xii
1. Introducción	1
1.1. ¿Qué es TortoiseSVN?	1
1.2. Historia de TortoiseSVN	1
1.3. Características de TortoiseSVN	1
1.4. Instalando TortoiseSVN	3
1.4.1. Requerimientos del sistema	3
1.4.2. Instalación	3
1.4.3. Packs de idiomas	3
1.4.4. Corrector ortográfico	3
2. Conceptos básicos	5
2.1. El repositorio	5
2.2. Modelos de versionado	5
2.2.1. El problema de compartir ficheros	6
2.2.2. La solución bloquear-modificar-desbloquear	6
2.2.3. La solución copiar-modificar-fusionar	8
2.2.4. ¿Qué hace Subversion?	10
2.3. Subversion en acción	10
2.3.1. Copias de trabajo	10
2.3.2. Revisiones	13
2.3.3. Cómo se unen las copias de trabajo al repositorio	14
2.4. Sumario	15
3. Preparando un servidor	16
3.1. Servidor basado en Apache	16
3.1.1. Introducción	16
3.1.2. Instalando Apache	16
3.1.3. Instalando Subversion	17
3.1.4. Configuración	18
3.1.5. Múltiples repositorios	19
3.1.6. Autorización basada en rutas	21
3.1.7. Autentificación con un dominio de Windows	22
3.1.8. Múltiples orígenes de autentificación	23
3.1.9. Asegurando el servidor con SSL	24
3.2. Servidor Basado en Svnserve	26
3.2.1. Introducción	26
3.2.2. Instalando svnserve	27
3.2.3. Ejecutando svnserve	27
3.2.4. Autentificación con svnserve	28
3.2.5. Autentificación con svn+ssh	29
3.2.6. Autorización basada en rutas con svnserve	30
4. El repositorio	31
4.1. Creación de repositorios	31
4.1.1. Creando un repositorio con el cliente de línea de comandos	31
4.1.2. Creando el repositorio con TortoiseSVN	31
4.1.3. Acceso local al repositorio	32
4.2. Copia de seguridad del Repositorio	32
4.3. Scripts gancho	33
4.4. Enlaces de obtener	33
5. Guía de uso diario	34
5.1. Empezando	34
5.1.1. Iconos sobreimpresionados	34

5.1.2. Menús contextuales	34
5.1.3. Arrastrar y soltar	36
5.1.4. Atajos comunes	37
5.1.5. Autentificación	37
5.2. Importando datos en un repositorio	38
5.2.1. Organización del repositorio	38
5.2.2. Importar	40
5.2.3. Ficheros especiales	41
5.2.4. Proyectos referenciados	42
5.3. Obteniendo una copia de trabajo	43
5.4. Enviando sus cambios al repositorio	44
5.5. Actualice su copia de trabajo con los cambios de otros	49
5.6. Resolviendo conflictos	50
5.7. Obteniendo información del estado	51
5.7.1. Iconos sobreimpresionados	51
5.7.2. Columnas de TortoiseSVN en el Explorador de Windows	53
5.7.3. Estado local y remoto	53
5.7.4. Viendo diferencias	55
5.8. Diálogo de Registro de revisiones	55
5.8.1. Invocando el diálogo de Registro de revisiones	56
5.8.2. Obteniendo información adicional	57
5.8.3. Obteniendo más mensajes de registro	59
5.8.4. Cambiando el mensaje de registro y el autor	60
5.8.5. Filtrando los mensajes de registro	60
5.8.6. Información estadística	61
5.9. Viendo diferencias	63
5.9.1. Diferencias de ficheros	63
5.9.2. Comparando carpetas	65
5.9.3. Diferenciando imágenes utilizando TortoiseIDiff	65
5.9.4. Herramientas externas de diferencias/fusión	66
5.10. Añadiendo nuevos ficheros y directorios	67
5.11. Ignorando ficheros y directorios	68
5.11.1. Expansión de comodines en los nombres de los ficheros de la lista de ignorados	69
5.12. Borrando, renombrando y moviendo	70
5.12.1. Renombrando un fichero sólo en mayúsculas/minúsculas	71
5.13. Deshacer cambios	72
5.14. Limpieza	73
5.15. Configuración del proyecto	74
5.15.1. Propiedades de Subversion	76
5.15.2. Propiedades de TortoiseSVN	78
5.16. Haciendo ramas / etiquetas	79
5.16.1. Crando una rama o etiqueta	79
5.16.2. Obtener o cambiar... ..	81
5.17. Fusionando	82
5.17.1. Fusionando un rango de revisiones	83
5.17.2. Fusionando dos árboles diferentes	85
5.17.3. Previsualizando los resultados de la fusión	85
5.17.4. Ignorando a los ancestros	86
5.18. Bloqueando	86
5.18.1. Cómo trabaja el bloqueo en Subversion	86
5.18.2. Obteniendo un bloqueo	87
5.18.3. Quitando un Bloqueo	88
5.18.4. Comprobando el estado de los bloqueos	88
5.18.5. Haciendo ficheros no-bloqueados como sólo-lectura	89
5.18.6. Los scripts ganchos de bloqueo	89
5.19. Creando y aplicando parches	89
5.19.1. Creando un fichero parche	89
5.19.2. Aplicando un fichero parche	90
5.20. ¿Quién cambió qué línea?	91
5.20.1. Autoría de ficheros	91
5.20.2. Autoría de las diferencias	92
5.21. El navegador de repositorios	93

5.22. Gráficos de revisión	94
5.23. Exportando una copia de trabajo de Subversion	96
5.24. Relocalizando una copia de trabajo	97
5.25. Integración con sistemas de control de errores / seguimiento de incidencias	98
5.26. Configuración de TortoiseSVN	100
5.26.1. Configuración general	100
5.26.2. Configuración del aspecto visual	102
5.26.3. Configuración de red	110
5.26.4. Configuración de programas externos	110
5.26.5. Datos de configuración almacenados	114
5.26.6. Configuraciones del registro	115
5.26.7. Carpetas de trabajo de Subversion	115
5.27. Último paso	115
6. El programa SubWCRev	116
6.1. La línea de comandos de SubWCRev	116
6.2. Sustitución de palabras clave	116
6.3. Ejemplo de palabras clave	117
A. Preguntas más frecuentes (FAQ)	119
B. ¿Cómo...?	120
B.1. Mover/copiar muchos ficheros de golpe	120
B.2. Obligar a los usuarios a introducir un mensaje de registro	120
B.2.1. Script gancho en el servidor	120
B.2.2. Propiedades del proyecto	120
B.3. Actualizar los ficheros seleccionados desde el repositorio	120
B.4. Deshacer revisiones en el repositorio	121
B.4.1. Utilice el diálogo Registro de revisiones	121
B.4.2. Utilice el diálogo Fusionar	121
B.4.3. Utilice svndumpfilter	121
B.5. Comparar dos revisiones de un fichero	122
B.6. Incluir un sub-proyecto común	122
B.6.1. Utilice svn:externals	122
B.6.2. Utilice una copia de trabajo anidada	122
B.6.3. Utilice una ruta relativa	123
B.7. Crear un acceso directo a un repositorio	123
B.8. Ignorar ficheros que ya están versionados	123
C. Administradores	125
C.1. Instalar TortoiseSVN utilizando políticas de grupo	125
C.2. Redirigir la comprobación de actualización	125
C.3. Estableciendo la variable de entorno SVN_ASP_DOT_NET_HACK	126
D. Automatizando TortoiseSVN	127
D.1. Comandos de TortoiseSVN	127
E. Referencia cruzada del interface de línea de comandos	130
E.1. Convenciones y reglas básicas	130
E.2. Comandos de TortoiseSVN	130
E.2.1. Obtener	130
E.2.2. Actualizar	130
E.2.3. Actualizar a la revisión	131
E.2.4. Confirmar	131
E.2.5. Diff	131
E.2.6. Mostrar registro	132
E.2.7. Comprobar modificaciones	132
E.2.8. Gráfico de revisión	132
E.2.9. Navegador de repositorios	132
E.2.10. Editar conflictos	133
E.2.11. Resuelto	133
E.2.12. Renombrar	133
E.2.13. Eliminar	133
E.2.14. Revertir	133
E.2.15. Limpieza	133
E.2.16. Obtener bloqueo	133
E.2.17. Quitar bloqueo	134
E.2.18. Ramas / Etiqueta	134
E.2.19. Cambiar	134

E.2.20. Fusionar	134
E.2.21. Exportar	134
E.2.22. Relocalizar	135
E.2.23. Crear repositorio aquí	135
E.2.24. Añadir	135
E.2.25. Importar	135
E.2.26. Autoría	135
E.2.27. Añadir a la lista de ignorados	135
E.2.28. Crear parche	135
E.2.29. Aplicar parche	136
Glosario	137
Índice	141

Lista de figuras

2.1. Un sistema típico cliente/servidor	5
2.2. El problema a evitar	6
2.3. La solución bloquear-modificar-desbloquear	7
2.4. La solución copiar-modificar-fusionar	8
2.5. ...Copiar-modificar-fusionar continuado	9
2.6. El sistema de ficheros del repositorio	11
2.7. El repositorio	13
4.1. El menú de TortoiseSVN para carpetas no versionadas	31
5.1. Explorador mostrando iconos sobreimpresionados	34
5.2. Menú contextual para un directorio bajo el control de versiones	35
5.3. Menú archivo del explorador para un acceso directo en una carpeta versionada	36
5.4. Menú de arrastre con el botón derecho para un directorio bajo el control de versiones ..	37
5.5. Diálogo de autenticación	38
5.6. El diálogo Importar	41
5.7. El diálogo Obtener	43
5.8. El diálogo de Confirmación	45
5.9. El corrector ortográfico del diálogo de Confirmación	47
5.10. El diálogo Progreso mostrando el progreso de una confirmación	48
5.11. Diálogo de progreso mostrando una actualización terminada	49
5.12. Explorador mostrando iconos sobreimpresionados	52
5.13. Comprobar modificaciones	54
5.14. El diálogo de Registro de revisiones	56
5.15. El panel superior del diálogo de Registro de revisiones con el menú contextual	57
5.16. Menú contextual del panel superior para 2 revisiones seleccionadas	58
5.17. El panel inferior del diálogo de Registro con el menú contextual	59
5.18. Histograma de confirmaciones por autor	61
5.19. Gráfico de tarta de confirmaciones por autor	62
5.20. Gráfico de confirmaciones por semana	63
5.21. El diálogo Comparar Revisiones	65
5.22. El visor de diferencias de imágenes	66
5.23. Menú contextual del explorador para ficheros no versionados	67
5.24. Menú contextual del explorador para ficheros no versionados	68
5.25. Menú contextual del explorador para ficheros versionados	70
5.26. Diálogo de Revertir	73
5.27. Página de propiedades del Explorador, pestaña Subversion	75
5.28. Página de propiedades de Subversion	76
5.29. Añadiendo propiedades	77
5.30. El diálogo Rama/Etiqueta	80
5.31. El diálogo Cambiar	82
5.32. El diálogo Fusionar	83
5.33. El diálogo Bloquear	87
5.34. El diálogo Comprobar modificaciones	88
5.35. El diálogo Crear parche	90
5.36. El diálogo Anotar / Autoría	91
5.37. TortoiseBlame	92
5.38. El navegador de repositorios	93
5.39. Un gráfico de revisiones	95
5.40. El diálogo Relocalizar	97
5.41. El diálogo de Configuración, página General	100
5.42. El diálogo Configuración, página de Aspecto visual	102
5.43. El diálogo Configuración, página de Aspecto visual	103
5.44. El diálogo Configuración, página de Conjunto de iconos	105
5.45. El diálogo Configuración, página de Diálogos 1	106
5.46. El diálogo Configuración, página de Diálogos 2	107
5.47. El Diálogo Configuración, Página de Colores	109
5.48. El diálogo Configuración, página de Red	110
5.49. El diálogo Configuración, página de Visor de diferencias	111
5.50. El diálogo Configuración, diálogo de Diferencias/Fusión avanzadas	113

5.51. El diálogo Configuración, página de Datos almacenados	114
C.1. El diálogo de Actualización	125

Lista de tablas

2.1. URLs de acceso al repositorio	11
3.1. Configuración del fichero httpd.conf de Apache	19
6.1. Lista de opciones de línea de comandos disponible	116
6.2. Lista de opciones de línea de comandos disponible	117
D.1. List of available commands and options	129

Prefacio



TortoiseSVN

- ¿Trabaja en equipo?
- ¿Alguna vez le ha ocurrido que estaba trabajando en un fichero, y alguien más también estaba trabajando en ese mismo fichero al mismo tiempo? ¿Perdió sus cambios en ese fichero por ese motivo?
- ¿Alguna vez ha grabado un fichero, y luego deseó deshacer los cambios que había hecho? ¿Alguna vez ha querido ver cómo estaba un fichero hace tiempo?
- ¿Alguna vez ha encontrado un error en su proyecto y ha querido saber cuándo se introdujo ese error en sus ficheros?

Si ha respondido sí a alguna de las preguntas anteriores, ¡entonces TortoiseSVN está hecho para usted! Siga leyendo para saber cómo puede TortoiseSVN ayudarle en su trabajo. No es tan difícil.

1. Audiencia

Este libro está escrito para usuarios informáticos que quieren usar Subversion para manejar sus datos, pero no están cómodos usando el cliente de línea de comandos para hacerlo. Dado que TortoiseSVN es una extensión del shell de Windows, se asume que el usuario está familiarizado con el Explorador de windows y sabe cómo usarlo.

2. Guía de Lectura

Este *Prefacio* explica un poco sobre el proyecto TortoiseSVN, la comunidad de gente que trabaja en él, y las condiciones de licencia para utilizarlo y distribuirlo.

La *Introducción* explica qué es TortoiseSVN, lo que hace, de dónde viene y las bases para instalarlo en su PC.

En los *Conceptos Básicos* damos una corta introducción al sistema de control de revisiones *Subversion* en el que TortoiseSVN se basa. Está prestado de la documentación del proyecto Subversion y explica las diferentes formas de control de versiones y cómo funciona Subversion.

Incluso muchos de los usuarios de Subversion nunca tendrán que preparar un servidor por ellos mismos. El siguiente capítulo trata sobre cómo preparar dicho servidor, y es útil para los administradores.

El capítulo sobre *El Repositorio* explica cómo preparar un repositorio local, lo que es útil para probar Subversion y TortoiseSVN utilizando un único PC. También explica un poco sobre administración de repositorios, lo que también es relevante para los repositorios que se encuentran en un servidor.

La *Guía de Uso Diario* es la sección más importante ya que explica todas las características principales de TortoiseSVN y cómo utilizarlas. Tiene la forma de un tutorial, empezando con obtener una copia de trabajo, modificarla, confirmar los cambios, etc. Luego avanza a temas más avanzados.

SubWCRev es un programa separado que se incluye con TortoiseSVN que puede extraer información de su copia de trabajo y escribirla en un fichero. Esto es útil para incluir información de compilación en sus proyectos.

La sección *¿Cómo...?* responde algunas preguntas frecuentes sobre la realización de tareas que no están cubiertas explícitamente en ninguna otra parte.

La sección *Automatizando TortoiseSVN* le muestra cómo puede invocar los diálogos GUI de TortoiseSVN desde la línea de comandos. Esto es útil para scripts donde aún necesite interacción del usuario.

La *Referencia Cruzada del Interface de Línea de Comandos* le muestra una correlación entre los comandos de TortoiseSVN y sus equivalentes en el cliente de línea de comandos de Subversion `svn.exe`.

3. ¡TortoiseSVN es gratis!

TortoiseSVN es gratis. No tiene que pagar por él, y puede usarlo para lo que quiera. Está desarrollado bajo la Licencia Pública General GNU (GPL).

TortoiseSVN es un proyecto de Código Abierto. Eso significa que tiene acceso completo al código fuente de este programa. Puede verlo en este enlace <http://tortoisesvn.tigris.org/svn/tortoisesvn/>. (Usuario:guest, deje en blanco la contraseña). La versión más reciente (donde estamos trabajando) se encuentra bajo `/trunk/`, las versiones oficiales están bajo `/tags/`.

4. Comunidad

Tanto TortoiseSVN como Subversion están siendo desarrollados por una comunidad de gente que trabaja en estos proyectos. Proviene de diferentes países por todo el mundo y se juntan para crear programas estupendos.

5. Reconocimientos

Tim Kemp
por fundar el proyecto TortoiseSVN

Stefan Küng
por el duro trabajo de llevar a TortoiseSVN a lo que es hoy

Lübbe Onken
por los bonitos iconos, logo, detección de errores y cuidar de la documentación

El libro de Subversion
por la gran introducción a Subversion y su capítulo 2 que hemos copiado aquí

El proyecto Tigris Style
por algunos de los estilos que están siendo reutilizados en esta documentación

Nuestros colaboradores
por los parches, informes de errores y nuevas ideas, y por ayudar a otros respondiendo preguntas de nuestra lista de correo.

Nuestros donantes
por la cantidad de horas de entretenimiento con la música que nos enviaron

6. Terminología usada en este documento

Para hacer más fácil la lectura de la documentación, los nombres de todas las pantallas y menús de TortoiseSVN están remarcados en un tipo de letra diferente. Por ejemplo, el **Diálogo de Registro**.

Las opciones de menú se indican con una flecha. **TortoiseSVN → Mostrar Registro** significa: seleccione *Mostrar Registro* desde el menú contextual *TortoiseSVN*.

Donde aparezca un menú contextual local dentro de uno de los diálogos de TortoiseSVN, se muestra-

rá así: Menú Contextual → Grabar como ...

Los botones del interfaz de usuario se indican como este: Pulse **OK** para continuar.

Las acciones del usuario se indican en negrita. **ALT+A**: pulse la tecla **ALT** en su teclado, y mientras la mantiene pulsada, pulse también la tecla **A**. Arrastre-con-botón-derecho: pulse el botón derecho del ratón y mientras lo mantiene pulsado, *arrastre* los ítems a su nuevo destino.

La salida del sistema y la entrada por teclado se indica con una fuente también diferente.



Importante

Las notas importantes están marcadas con un icono.



Sugerencia

Trucos que le facilitan la vida.



Atención

Lugares donde debe tener cuidado con lo que hace.



Aviso

Donde hay que tener un cuidado extremo, porque puede ocurrir corrupción de datos u otras cosas horribles si se ignoran estas advertencias.



Capítulo 1. Introducción

El control de versiones es el arte de manejar cambios en la información. Ha sido desde siempre una herramienta crítica para los programadores, quienes típicamente emplean su tiempo haciendo pequeños cambios al software y luego deshaciendo o comprobando esos cambios al día siguiente. Imagine un equipo de estos programadores trabajando concurrentemente - ¡y quizás también simultáneamente en los mismos ficheros! - y podrá ver por qué se necesita un buen sistema para *manejar el caos potencial*.

1.1. ¿Qué es TortoiseSVN?

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones *Subversion*. Esto es, TortoiseSVN maneja ficheros y directorios a lo largo del tiempo. Los ficheros se almacenan en un *repositorio* central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio. Esta es la razón por la que mucha gente piensa que Subversion, y los sistemas de control de versiones en general, son una especie de “máquinas del tiempo”.

Algunos sistemas de control de versiones también son sistemas de manejo de configuración del software (SCM). Estos sistemas están diseñados específicamente para manejar árboles de código fuente, y tienen muchas características que son específicas para el desarrollo de software - tales como el entendimiento nativo de los lenguajes de programación, o proporcionan herramientas para compilar software. Subversion, sin embargo, no es uno de estos sistemas; es un sistema general que puede ser utilizado para manejar *cualquier* colección de ficheros, incluyendo código fuente.

1.2. Historia de TortoiseSVN

En 2002, Tim Kemp se dio cuenta que Subversion era un sistema de control de versiones muy bueno, pero le faltaba un buen cliente GUI. La idea de tener un cliente de Subversion integrado en el shell de Windows se inspiró por el cliente similar que ya existía para CVS llamado TortoiseCVS.

Tim estudió el código fuente de TortoiseCVS y lo utilizó como base de TortoiseSVN. Entonces inició el proyecto, registró el dominio tortoisesvn.org y puso el código fuente online. Durante ese tiempo, Stefan Küng estaba buscando un sistema de control de versiones bueno y gratuito, y encontró Subversion y el código fuente de TortoiseSVN. Como TortoiseSVN todavía no estaba listo para usarse, se unió al proyecto y empezó a programarlo. Pronto reescribió la mayor parte del código existente y empezó a añadir comandos y características, hasta el punto de que no quedó nada del código original.

Según se fue estabilizando Subversion, atrajo más y más usuarios que también empezaron a utilizar TortoiseSVN como su cliente de Subversion. Los usuarios de TortoiseSVN se incrementaron rápidamente (y aún crecen día a día). Entonces Lübke Onken se ofreció a ayudar con algunos iconos más vistosos y un logo para TortoiseSVN. Y también se encarga de la página web y de manejar las traducciones.

1.3. Características de TortoiseSVN

¿Qué hace de TortoiseSVN tan buen cliente de Subversion? Aquí hay una pequeña lista de sus características.

Integración con el shell de Windows

TortoiseSVN se integra perfectamente en el shell de Windows (por ejemplo, el explorador). Esto significa que puede seguir trabajando con las herramientas que ya conoce. ¡Y que no tiene que cambiar a una aplicación diferente cada vez que necesite las funciones del control de versiones!

Y ni siquiera está obligado a usar el Explorador de Windows. Los menús contextuales de TortoiseSVN también funcionan en otros administradores de archivos, y en el diálogo Fichero/Abrir que es común a la mayoría de aplicaciones estándar de Windows. Sin embargo, debe tener en cuenta que TortoiseSVN está desarrollado con la mirada puesta en hacerle extensión del Explorador de Windows. Por este motivo, puede que en otras aplicaciones la integración no sea tan completa y que, por ejemplo, los iconos sobreimpresionados en las carpetas no se muestren.

Iconos sobreimpresionados

El estado de cada carpeta y fichero versionado se indica por pequeños iconos sobreimpresionados. De esta forma, puede ver fácilmente el estado en el que se encuentra su copia de trabajo.

Fácil acceso a los comandos de Subversion

Todos los comandos de Subversion están disponibles desde el menú contextual del explorador. TortoiseSVN añade su propio submenú allí.

Dado que TortoiseSVN es un cliente de Subversion, también queremos enseñarle algunas de las características del propio Subversion:

Versionado de carpetas

CVS sólo controla la historia de ficheros individuales, pero Subversion implementa un sistema “virtual” de ficheros versionados que sigue la pista de los cambios en todos los árboles de directorios en el tiempo. Los ficheros y los directorios están versionados. Como resultado, hay comandos reales en el lado del cliente como **mover** y **copiar** que operan en ficheros y directorios.

Confirmaciones atómicas

Una confirmación o bien entra en el repositorio completamente, o no entra en absoluto. Esto permite a los desarrolladores construir y confirmar cambios como unidades lógicas.

Metadatos versionados

Cada fichero y directorio tiene un conjunto invisible de “propiedades” adjuntos. Puede inventarse y almacenar cualquier par de clave/valor que desee. Las propiedades se versionan en el tiempo, igual que el contenido de los ficheros.

Elección de capas de red

Subversion tiene una noción abstracta del acceso al repositorio, haciendo que la gente pueda implementar nuevos mecanismos de red fácilmente. El “avanzado” servidor de red de Subversion es un módulo para el servidor web Apache, que habla una variante de HTTP llamada WebDAV/DeltaV. Esto dota a Subversion una gran ventaja en estabilidad e interoperatividad, y proporciona varias características importantes gratis: autenticación, autorización, compresión de la transmisión y navegación del repositorio, por ejemplo. También está disponible un proceso servidor de Subversion independiente. Este servidor habla un protocolo propio que puede encapsularse fácilmente sobre ssh.

Manejo de datos consistente

Subversion expresa las diferencias entre ficheros usando un algoritmo de diferenciación binario, que funciona exactamente igual tanto en ficheros de texto (legibles por los humanos) como en ficheros binarios (que no son legibles por nosotros). Ambos tipos de ficheros se almacenan igualmente comprimidos en el repositorio, y las diferencias se transmiten en ambas direcciones por la red.

Etiquetado y creación de ramas eficiente

El coste de crear una rama o una etiqueta no necesita ser proporcional al tamaño del proyecto. Subversion crea ramas y etiquetas simplemente copiando el proyecto, utilizando un mecanismo similar a los vínculos duros. Por tanto estas operaciones llevan un tiempo pequeño y constante, y muy poco espacio en el repositorio.

Extensibilidad

Subversion no tiene lastre histórico; está implementado como una colección de librerías C compartidas con APIs bien definidas. Esto hace que Subversion sea extremadamente mantenible y se pueda utilizar por otras aplicaciones y lenguajes.

1.4. Instalando TortoiseSVN

1.4.1. Requerimientos del sistema

TortoiseSVN se ejecuta en Windows 2000 SP2, Windows XP o superiores. Desde la versión de TortoiseSVN 1.2.0, no se contempla la instalación sobre los sistemas operativos Windows 98, Windows ME y Windows NT4, pero aún puede descargar versiones antiguas si realmente las necesita.

Si encuentra algún problema durante o después de la instalación de TortoiseSVN, por favor visite primero Apéndice A, *Preguntas más frecuentes (FAQ)*.

1.4.2. Instalación

TortoiseSVN incluye un instalador fácil de utilizar. Haga doble click en el fichero del instalador y siga las instrucciones. El instalador se encargará del resto. Si desea instalar TortoiseSVN para todos los usuarios, deberá tener privilegios de administrador en su sistema. Si no tiene estos derechos, TortoiseSVN automáticamente se instalará sólo para el usuario actual.



Importante

Si no tiene instaladas las últimas librerías de tiempo de ejecución de C y MFC, todavía deberá tener privilegios de Administrador para instalar TortoiseSVN. Pero una vez que esas librerías se instalan, puede actualizar o instalar TortoiseSVN sin dichos privilegios.

1.4.3. Packs de idiomas

The TortoiseSVN user interface has been translated into many different languages, so you may be able to download a language pack to suit your needs. You can find the language packs on our *translation status page* [http://tortoisesvn.net/translation_status]. And if there is no language pack available yet, why not join the team and submit your own translation ;-)

Cada pack de idioma está empaquetado como un instalador .exe. Sólo tiene que ejecutar el programa de instalación y seguir las instrucciones. La siguiente vez que reinicie, tendrá disponible la traducción.

1.4.4. Corrector ortográfico

TortoiseSVN incluye un corrector ortográfico que le permite comprobar sus mensajes de registro en las confirmaciones. Esto es especialmente útil si el idioma del proyecto no es su lengua materna. El corrector ortográfico utiliza los mismos diccionarios que *OpenOffice* [<http://openoffice.org>] y *Mozilla* [<http://mozilla.org>].

El instalador automáticamente añade los diccionarios de inglés de EE.UU. y de Reino Unido. Si desea tenerlos en otros idiomas, la opción más sencilla es simplemente instalar uno de los packs de idiomas de TortoiseSVN. Eso instalará los ficheros de diccionario adecuados junto con el interface de usuario de TortoiseSVN en ese idioma. La siguiente vez que reinicie, el diccionario también estará disponible.

O puede instalar los diccionarios usted mismo. Si tiene OpenOffice o Mozilla instalados, puede copiar esos diccionarios, que se encuentran en las carpetas de instalación de esas aplicaciones. Si no, deberá descargar los ficheros de diccionario necesarios desde http://linguocomponent.openoffice.org/spell_dic.html

Una vez que tenga los ficheros de diccionario, seguramente deberá cambiarlos de nombre para que los nombres de los ficheros sólo contengan los caracteres del idioma. Por ejemplo:

- es_ES.aff
- es_ES.dic

Luego sólo tiene que copiarlos en la subcarpeta bin de la carpeta de instalación de TortoiseSVN. Normalmente ésta será `C:\Archivos de programa\TortoiseSVN\bin`. Si no desea jugar con la subcarpeta bin, también puede poner los ficheros del corrector ortográfico en `C:\Archivos de programa\TortoiseSVN\Languages`. Si esa carpeta no está ahí, tendrá que crearla primero. La siguiente vez que inicie TortoiseSVN, podrá utilizar el corrector ortográfico.

Si instala múltiples diccionarios, TortoiseSVN utilizará estas reglas para seleccionar cuál utilizar.

1. Compruebe la configuración `tsvn:projectlanguage`. Lea Sección 5.15, “Configuración del proyecto” para encontrar información sobre cómo se establecen propiedades de proyecto.
2. Si no se ha establecido un idioma del proyecto, o ese idioma no está instalado, inténtelo con el idioma que corresponde al de Windows.
3. Si no funciona el idioma exacto de Windows, pruebe el idioma “Base”, por ejemplo, `es_MX` (Español-México) se transformaría en `es_ES` (Español).
4. Si nada de lo anterior funciona, entonces el idioma por defecto es el Inglés, que se incluye con la instalación estándar.

Capítulo 2. Conceptos básicos

Este capítulo es una versión ligeramente modificada del mismo capítulo en el libro de Subversion. Puede leer el libro de Subversion aquí: <http://svnbook.red-bean.com/>.

Este capítulo es una introducción corta e informal a Subversion. Si el control de versiones es nuevo para usted, este capítulo es definitivamente para usted. Empezamos con una discusión de los conceptos generales de control de versiones, nos hacemos camino dentro de las ideas específicas que hay tras Subversion, y mostramos algunos ejemplos sencillos de Subversion en acción.

Incluso aunque los ejemplos en este capítulo muestran a gente compartiendo colecciones de código fuente de programas, tenga en cuenta que Subversion puede manejar cualquier colección de ficheros - no está limitado a ayudar a los programadores de ordenadores.

2.1. El repositorio

Subversion es un sistema centralizado para compartir información. En su núcleo está un *repositorio*, que es un almacén central de datos. El repositorio almacena información en forma de un *árbol de ficheros* - una jerarquía típica de ficheros y directorios. Cualquier número de *clientes* se conectan al repositorio, y luego leen o escriben esos ficheros. Al escribir datos, el cliente hace que la información esté disponible para los otros; al leer los datos, el cliente recibe la información de los demás.

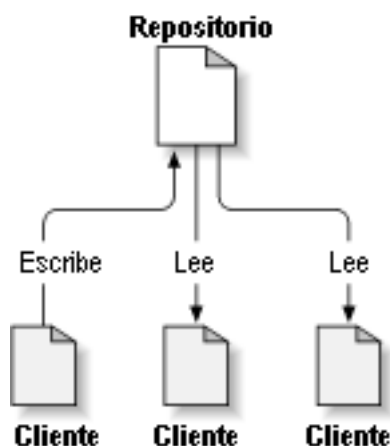


Figura 2.1. Un sistema típico cliente/servidor

¿Y esto por qué es interesante? Por ahora, eso suena a la definición típica de un servidor de ficheros típico. Y de hecho, el repositorio *es* una clase de servidores de ficheros, pero no el habitual. Lo que hace al repositorio de Subversion especial es que *recuerda todos los cambios* que alguna vez se hayan escrito en él: cada cambio en cada fichero, e incluso los cambios en el propio árbol de directorios, como el añadir, borrar o reorganizar ficheros y directorios.

Cuando un cliente lee datos de un repositorio, normalmente ve únicamente la última versión del árbol de ficheros. Pero el cliente también tiene la capacidad de ver estados *previos* del sistema de ficheros. Por ejemplo, un cliente puede hacer preguntas históricas, como "¿qué contenía este directorio el último miércoles?", o "¿quién fue la última persona que cambió este fichero, y qué cambios hizo?" Esta es la clase de preguntas que forman el corazón de cualquier *sistema de control de versiones*: son sistemas que están diseñados para guardar y registrar los cambios a los datos a lo largo del tiempo.

2.2. Modelos de versionado

Todos los sistemas de control de versiones tienen que resolver los mismos problemas fundamentales: ¿cómo permitirá el sistema compartir información entre usuarios, pero evitando que ellos acci-

dentalmente se pisen unos a otros? Es demasiado sencillo que los usuarios accidentalmente sobrescriban los cambios del otro en el repositorio.

2.2.1. El problema de compartir ficheros

Considere este escenario: suponga que tiene dos compañeros de trabajo, Harry y Sally. Cada uno decide editar el mismo fichero del repositorio a la vez. Si Harry graba sus cambios en el repositorio primero, el posible que (unos momentos después) Sally pueda accidentalmente sobrescribirlos con su propia versión nueva del fichero. Mientras que la versión del fichero de Harry no se ha perdido para siempre (porque el sistema recuerda cada cambio), cualquier cambio que Harry hizo *no estará* en la versión nueva del fichero de Sally, porque para empezar ella nunca vió los cambios de Harry. El trabajo de Harry está aún efectivamente perdido - o al menos falta en la última versión del fichero - y probablemente por accidente. ¡Esta es una situación que definitivamente tenemos que evitar!

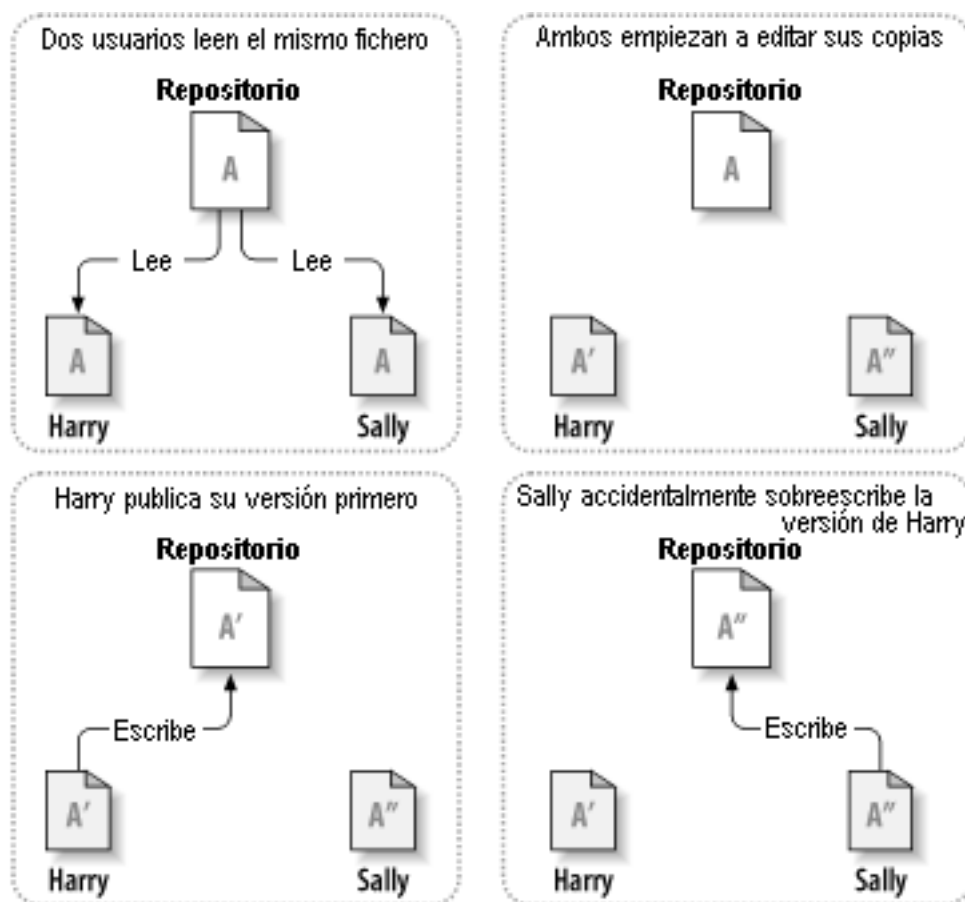


Figura 2.2. El problema a evitar

2.2.2. La solución bloquear-modificar-desbloquear

Muchos sistemas de control de versiones utilizan un modelo *bloquear-modificar-desbloquear* para enfrentarse a este problema, que es una solución muy simple. En estos sistemas, el repositorio sólo permite que una persona cambie un fichero al mismo tiempo. Harry primero debe "bloquear" el fichero antes de que pueda empezar a hacer cambios en él. Bloquear un fichero se parece mucho a tomar prestado un fichero de la biblioteca; si Harry ha bloqueado un fichero, entonces Sally no puede hacer ningún cambio en él. Si ella intenta bloquear el fichero, el repositorio entonces denegará la petición. Todo lo que ella puede hacer es leer el fichero, y esperar a que Harry termine sus cambios y libere su bloqueo. Después de que Harry desbloquee el fichero, se acabó su turno, y ahora le toca a Sally que puede bloquear y editar.

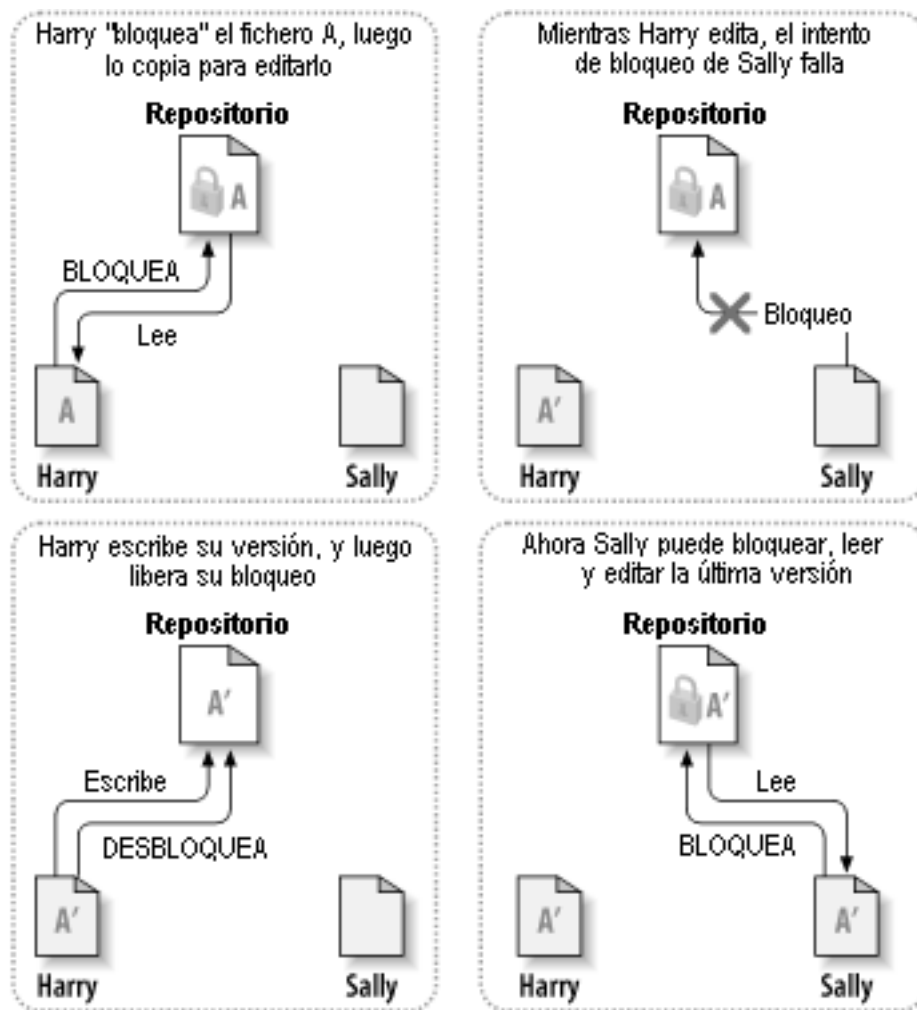


Figura 2.3. La solución bloquear-modificar-desbloquear

El problema con el modelo bloquear-modificar-desbloquear es que es un poco restrictivo, y a menudo se convierte en una calle cortada para los usuarios:

- *El bloqueo causa muchos problemas administrativos.* A veces Harry bloqueará un fichero y luego se olvidará de ello. Mientras tanto, dado que Sally está aún esperando para editar el fichero, sus manos están atadas. Y Harry se va de vacaciones. Ahora Sally tiene que buscar a un administrador para que libere el bloqueo de Harry. La situación acaba causando un montón de retraso y pérdida de tiempo innecesarios.
- *El bloqueo puede causar procesos en serie innecesarios.* ¿Qué ocurre si Harry está editando el inicio de un fichero de texto, y Sally simplemente quiere cambiar la parte final del mismo fichero? Esos cambios no se superponen en absoluto. Ellos podrían fácilmente editar el fichero de forma simultánea, y no habría ningún daño, asumiendo que los cambios se fusionaran correctamente. No hay necesidad de que se turnen en esta situación.
- *El bloqueo puede causar una falsa sensación de seguridad.* Imagine que Harry bloquea y edita el fichero A, mientras Sally simultáneamente bloquea y edita el fichero B. Pero suponga que A y B dependen uno del otro, y que los cambios hechos a cada uno son semánticamente incompatibles. De repente A y B ya no funcionan juntos. El sistema de bloqueo no tiene forma de prevenir este problema - sin embargo, de alguna forma dió una sensación de falsa seguridad. Es fácil para Harry y Sally imaginar que al bloquear los ficheros, cada uno está empezando una tarea segura y aislada, y por tanto les inhibe de discutir sus cambios incompatibles en un momento temprano.

2.2.3. La solución copiar-modificar-fusionar

Subversion, CVS y otros sistemas de control de versiones utilizan un modelo *copiar-modificar-fusionar* como alternativa al bloqueo. En este modelo, el cliente de cada usuario lee el repositorio y crea una *copia de trabajo* personal del fichero o del proyecto. Luego, los usuarios trabajan en paralelo, modificando sus copias privadas. Finalmente, las copias privadas se fusionan juntas en una nueva versión final. El sistema de control de versiones a menudo ofrece ayuda en la fusión, pero al final la persona es la responsable de hacer que ocurra correctamente.

Aquí hay un ejemplo. Digamos que Harry y Sally cada uno crean copias de trabajo del mismo proyecto, copiado del repositorio. Ellos trabajan concurrentemente, y hacen los cambios al mismo fichero "A" dentro de sus copias. Sally graba sus cambios al repositorio primero. Cuando Harry intenta grabar sus cambios más tarde, el repositorio le informa que su fichero A está *desactualizado*. En otras palabras, que el fichero A en el repositorio ha cambiado de alguna forma desde la última vez que lo copió. Por lo que Harry le pide a su cliente que *fusion*e cualquier cambio nuevo del repositorio dentro de su copia de trabajo del fichero A. Lo más seguro es que los cambios de Sally no se superpongan a los suyos; por lo que una vez que ambos conjuntos de cambios se han integrado, él graba su copia de trabajo de nuevo en el repositorio.

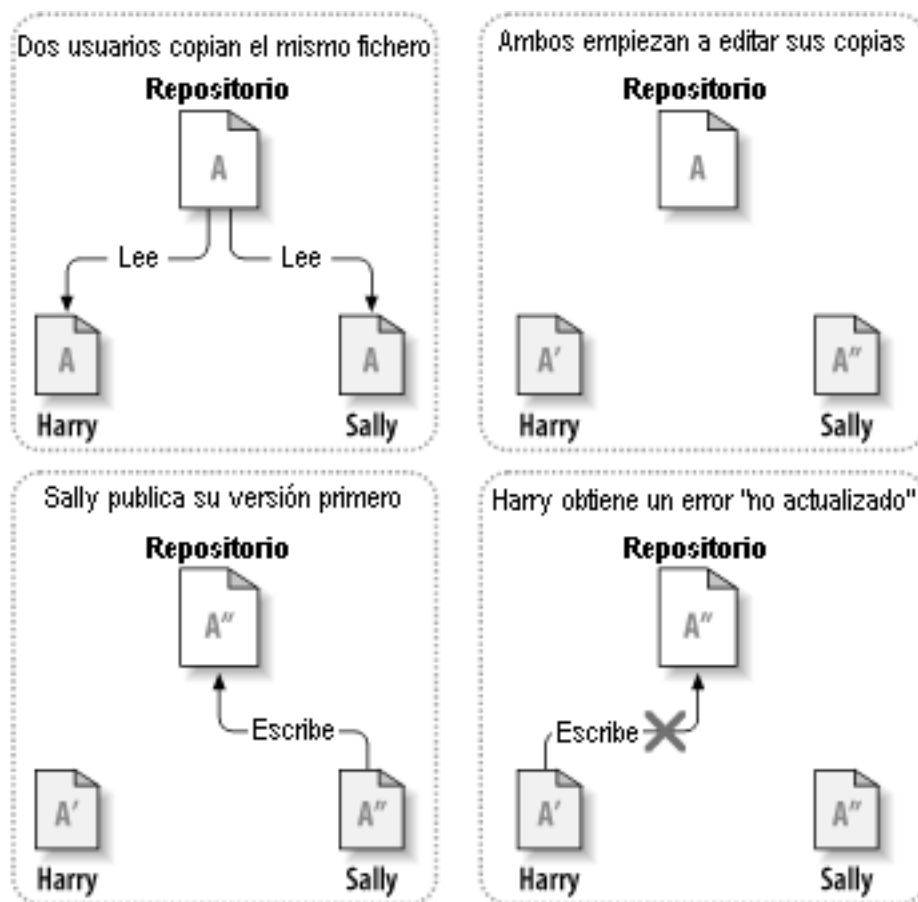


Figura 2.4. La solución copiar-modificar-fusionar

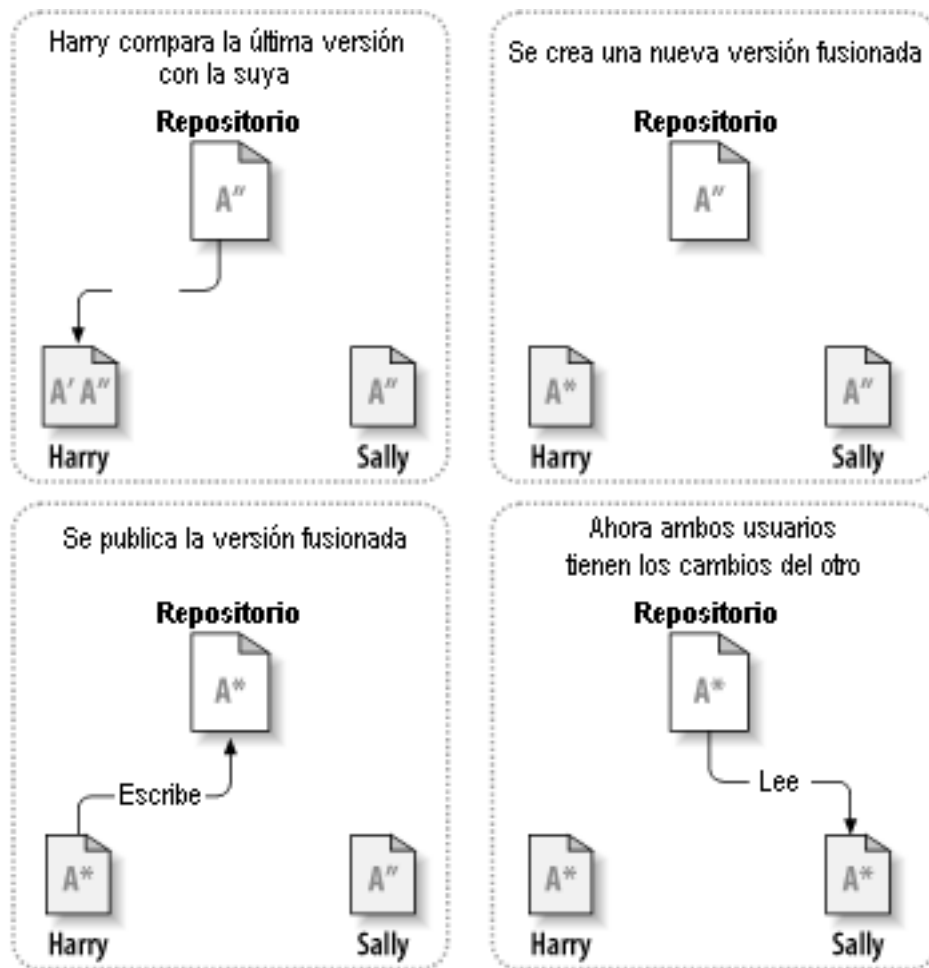


Figura 2.5. ...Copiar-modificar-fusionar continuado

¿Pero qué ocurre si los cambios de Sally *sí* se superponen a los cambios de Harry? ¿Qué hacemos entonces? La situación se denomina un *conflicto*, y normalmente no es mucho problema. Cuando Harry le pide a su cliente que fusione los últimos cambios del repositorio en su copia de trabajo, su copia del fichero A se marca de alguna forma como que está en un estado de conflicto: él será capaz de ver ambos conjuntos de cambios conflictivos, y manualmente podrá elegir entre ellos. Tenga en cuenta que el software no puede resolver conflictos automáticamente; sólo los humanos son capaces de entender y hacer las elecciones necesarias de forma inteligente. Una vez que Harry haya resuelto manualmente los cambios que se superponían (¡quizás discutiendo el conflicto con Sally!), puede volcar de forma segura el fichero fusionado al repositorio.

El modelo copiar-modificar-fusionar puede sonar un poco caótico, pero en la práctica, funciona extremadamente bien. Los usuarios pueden trabajar en paralelo, sin que tengan que esperar nunca uno por otro. Cuando trabajan en los mismos ficheros, resulta que la mayoría de los cambios concurrentes no se superponen en absoluto; los conflictos no son frecuentes. Y el tiempo que lleva resolver conflictos es mucho menor que el tiempo perdido por un sistema bloqueante.

Al final, todo se reduce a un factor crítico: la comunicación entre usuarios. Cuando los usuarios se comunican de forma pobre, aumentan los conflictos sintácticos y semánticos. No hay sistema capaz de forzar a los usuarios a comunicarse perfectamente, y no hay sistema que pueda detectar conflictos semánticos. Por lo que no hay motivo para que se le prometa falsamente que un sistema con bloqueos prevendrá de alguna forma los conflictos; en la práctica, el bloqueo parece inhibir la productividad más que otra cosa.

Hay una situación común donde el modelo bloquear-modificar-desbloquear resulta mejor, y es cuan-

do tiene ficheros no-fusionables. Por ejemplo si su repositorio contiene algunas imágenes gráficas, y dos personas cambian la imagen a la vez, no hay forma de fusionar esos cambios. O Harry o Sally perderán sus cambios.

2.2.4. ¿Qué hace Subversion?

Subversion utiliza la solución copiar-modificar-mezclar por defecto, y en muchos casos esto es todo lo que necesitará. Sin embargo, desde la Versión 1.2, Subversion también admite bloqueo de ficheros, por lo que si tiene ficheros no-fusionables, o si simplemente está forzado a una política de bloqueo por la dirección, Subversion seguirá teniendo las características que necesita.

2.3. Subversion en acción

2.3.1. Copias de trabajo

Ya ha oído hablar sobre las copias de trabajo; ahora le demostraremos cómo las crea y las utiliza el cliente de Subversion.

Una copia de trabajo de Subversion es un árbol de directorios ordinario en su sistema local, conteniendo una colección de ficheros. Puede editar estos ficheros como desee, y si son ficheros de código fuente, puede compilar su programa de la forma habitual. Su copia de trabajo es su propia área de trabajo privada: Subversion nunca incorporará los cambios de otra gente, ni hará que sus cambios estén disponibles para los demás, a menos que se lo pida explícitamente.

Después de que haya hecho algunos cambios en los ficheros dentro de su copia de trabajo y verifique que funcionan correctamente, Subversion le provee de comandos para "publicar" sus cambios para los demás que trabajan con usted en su proyecto (escribiendo en el repositorio). Si los demás publican sus propios cambios, Subversion le provee de comandos para fusionar esos cambios dentro de su copia de trabajo (leyendo desde el repositorio).

Una copia de trabajo también contiene algunos ficheros extra, creados y mantenidos por Subversion, para ayudarse a llevar a cabo esos comandos. En particular, cada directorio dentro de su copia de trabajo contiene un subdirectorio llamado `.svn`, también conocido como el *directorio administrativo* de la copia de trabajo. Los ficheros dentro de los directorios administrativos ayudan a Subversion a reconocer qué ficheros contienen cambios no publicados, y qué ficheros están desactualizados respecto al trabajo de los demás.

Un repositorio típico de Subversion a menudo contiene los ficheros (o el código fuente) de varios proyectos; usualmente, cada proyecto es un subdirectorio en el árbol de ficheros del repositorio. Con esta disposición, una copia de trabajo de un usuario normalmente corresponderán a un subárbol particular del repositorio.

Por ejemplo, suponga que tiene un repositorio que contiene dos proyectos de software.

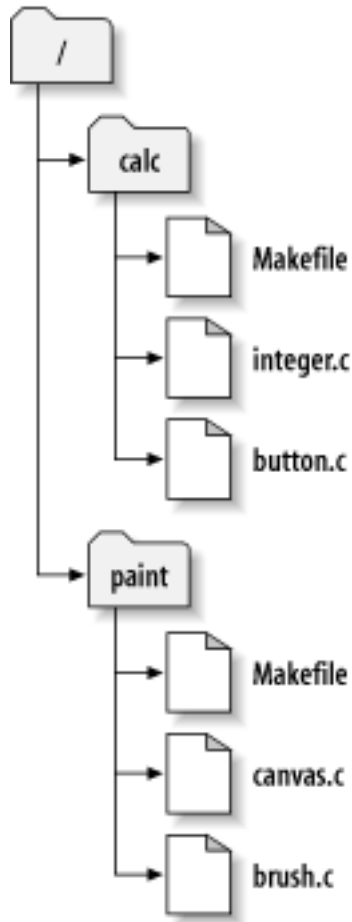


Figura 2.6. El sistema de ficheros del repositorio

En otras palabras, el directorio raíz del repositorio tiene dos subdirectorios, `paint` y `calc`.

Para obtener una copia de trabajo, primero debe *obtener* algún subárbol del repositorio. (El término "obtener" puede sonar como que tenga algo que ver con el bloqueo o la reserva de recursos, pero no es cierto; simplemente crea una copia privada del proyecto para usted).

URLs de repositorio

Los repositorios de Subversion pueden ser accedidos por muchos métodos diversos - en discos locales, o a través de varios protocolos de red. La ruta de un repositorio es siempre, sin embargo, una URL. El esquema URL indica el método de acceso:

Esquema	Método de acceso
<code>file://</code>	Acceso directo al repositorio en el disco local o de red.
<code>http://</code>	Acceso utilizando el protocolo WebDAV a un servidor Apache configurado para Subversion.
<code>https://</code>	Lo mismo que <code>http://</code> , pero con encriptación SSL.
<code>svn://</code>	Acceso TCP/IP sin autenticación utilizando un protocolo personalizado a un servidor <code>svnserve</code> .
<code>svn+ssh://</code>	Acceso TCP/IP autenticado y encriptado utilizando un protocolo propio a un servidor <code>svnserve</code> .

Tabla 2.1. URLs de acceso al repositorio

En su mayoría, las URLs de Subversion utilizan la sintaxis estándar, permitiendo que se especifiquen nombres de servidor y números de puertos como parte de la URL. El método de acceso `file:` se utiliza normalmente para el acceso local, aunque puede utilizarse con rutas UNC a equipos en red. La URL entonces toma la forma `file://nombredeequipo/ruta/al/repositorio`. Para la máquina local, la parte del nombre de equipo en la URL debe estar o ausente o ser `localhost`. Por esa razón, las rutas locales normalmente aparecen con tres barras, `file:///ruta/al/repositorio`.

Además, los usuarios del esquema `file:` en las plataformas Windows necesitarán utilizar una sintaxis “estándar” no oficial para acceder a los repositorios que están en la misma máquina, pero en una letra de unidad diferente de la unidad actual del cliente. Cualquiera de las siguientes sintaxis de URL funcionarán, donde X es la unidad en la que reside el repositorio:

```
file:///X:/ruta/al/repositorio
...
file:///X|/ruta/al/repositorio
...
```

Tenga en cuenta que las URLs utilizan las barras hacia delante (las de dividir) incluso aunque la forma nativa (no-URL) de una ruta en Windows utiliza las barras contrarias.

Puede acceder con seguridad a un repositorio FSFS utilizando una carpeta compartida de red, pero *no* puede acceder a un repositorio BDB de esta forma.



Aviso

No cree o acceda a un repositorio Berkeley DB en una unidad de red compartida. *No* puede existir en un sistema de archivos remoto. Ni siquiera si tiene la unidad de red mapeada a una letra de unidad. Si intenta usar Berkeley DB en una unidad de red compartida, los resultados son imprevisibles - puede ver desde el principio errores misteriosos, o pueden pasar meses antes de que descubra que su base de datos del repositorio está corrupta de una forma inimaginable.

Suponga que ha hecho cambios a `button.c`. Dado que el directorio `.svn` recuerda la fecha de modificación y los contenidos originales del fichero, Subversion puede decirle que ha cambiado el fichero. Sin embargo, Subversion no hace públicos sus cambios hasta que explícitamente se lo pida. El acto de publicar sus cambios se conoce más comúnmente como *confirmar* (o *enviar*) los cambios al repositorio.

Para publicar sus cambios para los demás, puede utilizar el comando de Subversion **commit**.

Ahora que sus cambios a `button.c` se han confirmado en el repositorio, si cualquier otro usuario obtiene una copia de trabajo de `/calc`, verán sus cambios en la última versión del fichero.

Suponga que tiene un colaborador, Sally, que obtuvo una copia de trabajo de `/calc` al mismo tiempo que usted. Cuando ha confirmado sus cambios en `button.c`, la copia de trabajo de Sally se queda sin cambios; Subversion sólo modifica las copias de trabajo cuando lo pide el usuario.

Para poner al día su proyecto, Sally puede pedirle a Subversion *actualizar* su copia de trabajo, utilizando el comando de Subversion **actualizar**. Esto incorporará sus cambios en la copia de trabajo de Sally, junto con cualquier otro que se haya confirmado desde que ella lo obtuvo.

Tenga en cuenta que Sally no necesita especificar qué ficheros actualizar; Subversion utiliza la in-

formación en el directorio `.svn`, y más información desde el repositorio, para decidir qué ficheros deben ponerse al día.

2.3.2. Revisiones

Una operación **svn commit** puede publicar los cambios de cualquier número de ficheros y carpetas como una única transacción atómica. En su copia de trabajo, puede cambiar el contenido de los ficheros, crear, borrar, renombrar y copiar ficheros y directorios, y luego confirmar el conjunto completo de cambios como una unidad.

En el repositorio, cada confirmación se trata como una transacción atómica: o bien todos los cambios de la confirmación se llevan a cabo, o bien ninguno de ellos se realiza. Subversion intenta retener esta atomicidad en caso de errores en el programa, errores del sistema, problemas de red, y otras acciones del usuario.

Cada vez que el repositorio acepta una confirmación, crea un nuevo estado del árbol de ficheros, llamado *revisión*. A cada revisión se le asigna un número natural único, un número mayor que la revisión anterior. La revisión inicial de un repositorio recién creado se numera como cero, y consiste únicamente en un directorio raíz vacío.

Una buena forma de visualizar el repositorio es como una serie de árboles. Imagine una fila de números de revisiones, empezando en 0, de izquierda a derecha. Cada número de revisión tiene un árbol colgando debajo, y cada árbol es una “foto” de cómo estaba el repositorio tras cada confirmación.

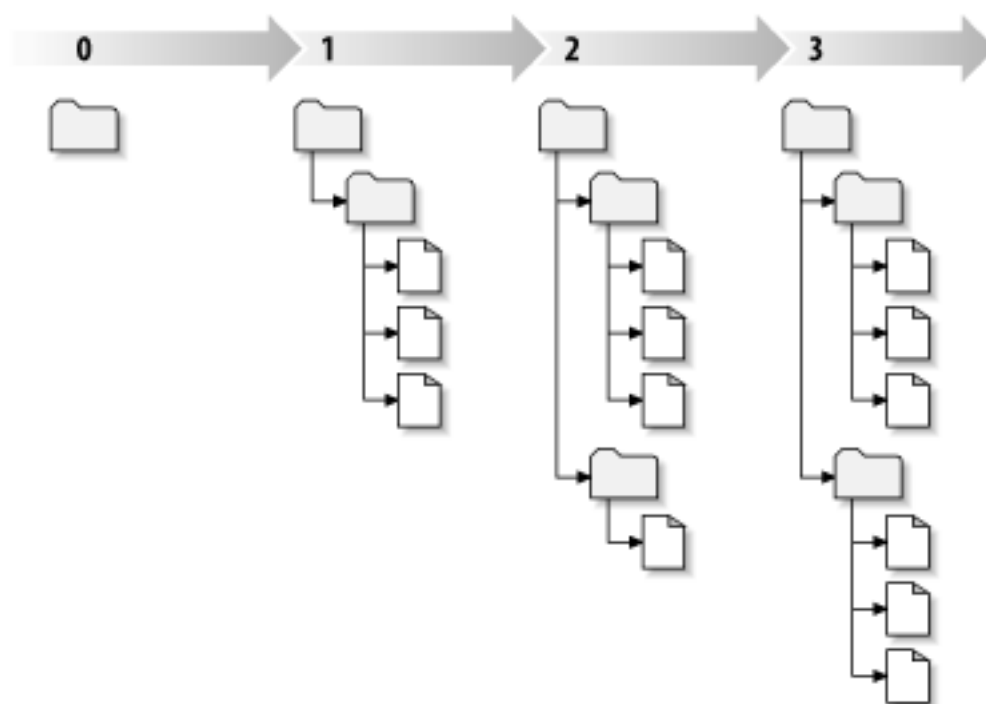


Figura 2.7. El repositorio

Números globales de revisión

Al contrario que la mayoría del resto de sistemas de control de versiones, los números de revisión de Subversion se aplican a *árboles completos*, no a los ficheros individuales. Cada número de revisión selecciona un árbol entero, un estado particular del repositorio tras algún cambio confirmado. Otra forma de verlo es pensar que la revisión N representa el estado del repositorio tras la confirmación N-ésima. Cuando un usuario de Subversion habla de la “revi-

sion 5 de `foo.c`", realmente quieren decir "`foo.c` tal y como estaba en la revisión 5". ¡Tenga en cuenta que, en general, las revisiones N y M de un fichero *no* tienen por qué ser diferentes!

Es importante que tenga en cuenta que las copias de trabajo no siempre se corresponden a una única revisión en el repositorio; pueden contener ficheros de varias revisiones. Por ejemplo, suponga que obtiene una copia de trabajo de un repositorio cuya revisión más reciente es la 4:

```
calc/Makefile:4
    integer.c:4
    button.c:4
```

En este momento, esta copia de trabajo corresponde exactamente a la revisión 4 en el repositorio. Sin embargo, suponga que ha hecho cambios al fichero `button.c`, y confirme ese cambio. Asumiendo que no se haya llevado a cabo ninguna otra confirmación, su confirmación creará la revisión 5 en el repositorio, y su copia de trabajo quedará así:

```
calc/Makefile:4
    integer.c:4
    button.c:5
```

Suponga que, en este punto, Sally hace un cambio a `integer.c`, creando la revisión 6. Si utiliza **svn update** para actualizar su copia de trabajo, obtendrá esto:

```
calc/Makefile:6
    integer.c:6
    button.c:6
```

Los cambios de Sally a `integer.c` aparecerán en su copia de trabajo, y su cambio estará aún presente en `button.c`. En este ejemplo, el texto de `Makefile` es idéntico en las revisiones 4, 5, y 6, pero Subversion marcará su copia de trabajo de `Makefile` con la revisión 6 para indicar que aún está actualizado. Por lo que, después de que haga una actualización limpia en la parte superior de su copia de trabajo, generalmente obtendrá exactamente una revisión del repositorio.

2.3.3. Cómo se unen las copias de trabajo al repositorio

Por cada fichero en un directorio de trabajo, Subversion grabará dos piezas esenciales de información en el área administrativa `.svn/`:

- en qué revisión se basa su fichero de trabajo (lo que se denomina la *revisión de trabajo*), y
- una fecha que indica cuándo se actualizó por última vez la copia local por el repositorio.

Con esta información, hablando con el repositorio, Subversion puede decirle en cuál de los siguientes cuatro estados está un fichero de trabajo:

Sin cambios, y actualizado

El fichero no se ha cambiado en el directorio de trabajo, y no se han confirmado cambios a ese fichero en el repositorio desde su revisión de trabajo. Una **confirmación** de ese fichero no hará nada, y una **actualización** de ese fichero no hará nada.

Cambiado localmente, y actualizado

El fichero ha sido cambiado en el directorio de trabajo, y no se ha confirmado ningún cambio a ese fichero en el repositorio desde su revisión base. Hay cambios locales que no se han confirmado al repositorio, por lo que al **confirmar** el fichero se conseguirá publicar sus cambios, y al

actualizar el fichero no se realizará nada.

Sin cambios, y desactualizado

El fichero no ha sido cambiado en el directorio de trabajo, pero ha sido cambiado en el repositorio. El fichero deberá ser actualizado en algún momento, para actualizarlo con la revisión pública. Un comando **confirmar** sobre el fichero no hará nada, y al **actualizar** el fichero se traerán los últimos cambios a su copia de trabajo.

Cambiado localmente, y desactualizado

El fichero se ha cambiado tanto en el directorio de trabajo como en el repositorio. Un comando **confirmar** sobre el fichero fallará con un error "desactualizado". El fichero debería actualizarse primero; al **actualizar** se intentará fusionar los cambios públicos con los cambios locales. Si Subversion no puede completar la fusión de una forma plausible automáticamente, le dejará al usuario la tarea de resolver el conflicto.

2.4. Sumario

Hemos cubierto un número de conceptos fundamentales de Subversion en este capítulo:

- Hemos introducido las nociones de un repositorio central, la copia de trabajo del cliente, y la lista de árboles de revisiones del repositorio.
- Hemos visto algunos ejemplos simples sobre cómo dos colaboradores pueden utilizar Subversion para publicar y recibir los cambios de uno a otro, utilizando el modelo 'copiar-modificar-fusionar'.
- Hemos hablado un poco sobre la forma en la que Subversion controla y maneja la información en una copia de trabajo.

Capítulo 3. Preparando un servidor

Para utilizar TortoiseSVN (o cualquier otro cliente de Subversion), necesita un lugar donde establecer sus repositorios. Puede o bien almacenar sus repositorios de forma local y acceder a ellos utilizando el protocolo *file://*, o puede ponerlos en un servidor y acceder a ellos con el protocolo *http://* o *svn://*. Los dos protocolos de servidor también pueden ser encriptados. Utilice *https://* o *svn+ssh://*. Este capítulo le muestra paso a paso cómo puede preparar uno de estos servidores en una máquina Windows.

Si no tiene un servidor y/o si trabaja solo, los repositorios locales son probablemente su mejor elección. Puede saltarse este capítulo e ir directamente al Capítulo 4, *El repositorio*.

3.1. Servidor basado en Apache

3.1.1. Introducción

La configuración más flexible de todas las instalaciones de servidor posibles para Subversion es la que se basa en Apache. Aunque es un poco más complicada de preparar, ofrece beneficios que otros servidores no pueden dar:

WebDAV

El servidor de Subversion basado en Apache utiliza el protocolo WebDAV que se utiliza por muchos otros programas. Por ejemplo, podría montar un repositorio como una "Carpeta Web" en el explorador de Windows y luego acceder a ella como cualquier otra carpeta en su sistema de ficheros

Navegando por el repositorio

Puede apuntar su navegador a la URL del repositorio y navegar por sus contenidos sin tener un cliente de Subversion. Esto da acceso a sus datos a un mayor círculo de usuarios.

Autenticación

Puede utilizar cualquier mecanismo de autenticación que Apache soporte, incluyendo SSPI y LDAP.

Seguridad

Dado que Apache es muy estable y seguro, automáticamente obtendrá la misma seguridad para su repositorio. Esto incluye la encriptación SSL.

3.1.2. Instalando Apache

La primera cosa que necesita antes de instalar Apache es un ordenador con Windows 2000, Windows XP con SP1, o Windows 2003.



Aviso

Por favor tenga en cuenta que utilizar Windows XP sin el Service Pack 1 corrompe datos de la red y por tanto ¡podría corromper su repositorio!

1. Descargue la última versión del servidor web Apache desde <http://httpd.apache.org/download.cgi>. Asegúrese de que descarga la versión > 2.0.54 - ¡las versiones 1.3.xx no sirven! Además, las versiones anteriores a la 2.0.54 no funcionará con Subversion 1.2 por culpa de un error en cómo se compiló Apache < 2.0.54 para Windows.
2. Una vez que tenga el instalador de Apache2 puede hacer doble click en él y le guiará a través del proceso de instalación. Asegúrese de que ha introducido la URL del servidor correctamente (si no tiene un nombre dns para su servidor introduzca la dirección IP). Le recomiendo que ins-

tale apache para Todos los usuarios, en Puerto 80, como un Servicio. Nota: si ya tiene IIS u otro programa ejecutándose que escuche en el puerto 80 la instalación puede fallar. Si esto ocurre, vaya al directorio de Archivos de Programa, \Apache Group\Apache2\conf y localice el fichero httpd.conf. Edite dicho fichero para cambiar Listen 80 por un puerto libre, por ejemplo, Listen 81. Luego reinicie la instalación - esta vez debería terminar sin problemas.

3. Ahora compruebe si el servidor web Apache funciona correctamente apuntando desde su navegador web a la dirección `http://localhost/` - debería aparecer un sitio web preconfigurado.



Atención

Si decide instalar Apache como un servicio, queda avisado de que por defecto se ejecutará con la cuenta de sistema local. Sería una práctica más segura que creara una cuenta separada para que Apache se ejecutara bajo ella.

Asegúrese de que la cuenta en el servidor bajo la que se ejecuta Apache tenga una entrada explícita en la lista de control de acceso del directorio del repositorio (click con el botón derecho en el directorio | propiedades | seguridad), con control total. Si no lo hace así, los usuarios no podrán confirmar sus cambios.

Incluso si Apache se ejecuta como sistema local, aún así necesitará dicha entrada (que en este caso debería ser la cuenta SYSTEM).

Si Apache no tiene este permiso configurado, sus usuarios tendrán mensajes de error "Acceso denegado", que se mostrarán en el registro de errores de Apache como error 500.

3.1.3. Instalando Subversion

1. Descargue la última versión de Subversion desde <http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=91>.
2. Ejecute el instalador de Subversion y siga las instrucciones. Si el instalador de Subversion reconoce que ha instalado Apache, habrá casi terminado. Si no puede encontrar un servidor de Apache entonces tendrá que realizar algunos pasos adicionales.
3. Utilizando el explorador de Windows, vaya al directorio de instalación de Subversion (normalmente C:\Archivos de programa\Subversion) y busque los ficheros / httpd/mod_dav_svn.so y mod_authz_svn.so. Copie estos ficheros al directorio de módulos de Apache (normalmente C:\Archivos de programa\Apache Group\Apache2\modules).
4. Copie el fichero /bin/libdb43.dll desde el directorio de instalación de Subversion al directorio de módulos de Apache.
5. Edite el fichero de configuración de Apache (normalmente C:\Archivos de Programa\Apache Group\Apache2\conf\httpd.conf) con un editor de texto como el Bloc de Notas y haga los siguientes cambios:

Descomente (quitando la marca '#') las siguientes líneas:

```
#LoadModule dav_fs_module modules/mod_dav_fs.so
```

```
#LoadModule dav_module modules/mod_dav.so
```

Añada las dos líneas siguientes al final de la sección `LoadModule`.

```
LoadModule dav_svn_module modules/mod_dav_svn.so
LoadModule authz_svn_module modules/mod_authz_svn.so
```

3.1.4. Configuración

Ahora ya ha preparado Apache y Subversion, pero Apache aún no sabe cómo manejar los clientes de Subversion como TortoiseSVN. Para que Apache sepa qué URL debe utilizarse para los repositorios de Subversion debe editar el fichero de configuración de Apache (normalmente está en `C:\Archivos de programa\Apache Group\Apache2\conf\httpd.conf`) con cualquier editor de texto que desee (por ejemplo, el Bloc de Notas):

1. Al final del fichero `Config` añada las siguientes líneas:

```
<Location /svn>
DAV svn
SVNListParentPath on
SVNParentPath D:\SVN
AuthType Basic
AuthName "Repositorios de Subversion"
AuthUserFile passwd
#AuthzSVNAccessFile svnaccessfile
Require valid-user
</Location>
```

Esto configura el Apache de forma que todos sus repositorios de Subversion están físicamente localizados bajo `D:\SVN`. Los repositorios se sirven al mundo exterior desde la URL: `http://MiServidor/svn/`. El acceso es restringido a los usuarios/contraseñas listados en el fichero `passwd`.

2. Para crear el fichero `passwd`, abra el Símbolo del sistema o línea de comandos (ventana DOS) de nuevo, cambie a la carpeta `Apache2` (normalmente `C:\Archivos de programa\Apache Group\Apache2`) y cree el fichero introduciendo

```
bin\htpasswd -c passwd <nombreusuario>
```

Esto creará un nuevo fichero con el nombre `passwd` que se utilizará para la autenticación. Se pueden crear usuarios adicionales con

```
bin\htpasswd passwd <nombreusuario>
```

3. Reinicie el servicio de Apache de nuevo.
4. Apunte su navegador a `http://MiServidor/svn/MiNuevoRepositorio` (donde `MiNuevoRepositorio` es el nombre del repositorio de Subversion que creó antes). Si todo ha ido bien debería ver una ventana preguntando por un usuario y una contraseña, y luego podrá ver los contenidos de su repositorio.

Una explicación corta de lo que acaba de introducir:

Configuración	Explicación
<code><Location /svn></code>	significa que los repositorios de Subversion están disponibles en la URL <code>http://MiServidor/svn/</code>
<code>DAV svn</code>	le dice a Apache qué módulo será responsable de servir esa URL - en este caso, el módulo de Subversion.

Configuración	Explicación
SVNListParentPath on	para Subversion 1.3 y superiores, esta directiva habilita el listado de todos los repositorios disponibles bajo SVNParentPath.
SVNParentPath D:\SVN	le dice a Subversion que busque repositorios bajo D:\SVN
AuthType Basic	se utiliza para activar la autenticación básica, es decir, Usuario/contraseña
AuthName "Repositorios de Subversion"	se utiliza cuando le aparezca un diálogo de autenticación al usuario como información para decirle para qué se necesita su autenticación
AuthUserFile passwd	especifica qué fichero de contraseñas se utiliza para la autenticación
AuthzSVNAccessFile	lugar del fichero de Acceso para las rutas dentro del repositorio de Subversion
Require valid-user	especifica que sólo los usuarios que hayan introducido un par usuario/contraseña válido podrán acceder a la URL

Tabla 3.1. Configuración del fichero httpd.conf de Apache

Pero eso es sólo un ejemplo. Hay muchas, muchas más posibilidades de lo que puede hacer con el servidor web Apache.

- Si desea que su repositorio tenga acceso de lectura para todo el mundo pero el acceso de escritura sólo para usuarios específicos, puede cambiar la línea

```
Require valid-user
```

por

```
<LimitExcept GET PROPFIND OPTIONS REPORT>
Require valid-user
</LimitExcept>
```

- Utilizando un fichero passwd se limita y se otorga acceso a todos sus repositorios como una unidad. Si desea más control sobre qué usuarios tienen acceso a cada carpeta dentro de un repositorio, puede descomentar la línea

```
#AuthzSVNAccessFile svnaccessfile
```

y crear un fichero de acceso de Subversion. Apache se asegurará que sólo los usuarios válidos pueden acceder su ruta /svn, y luego pasará el nombre de usuario al módulo de Subversion AuthzSVNAccessFile para que pueda forzar un acceso más granularizado basado en las reglas que se especifican en el fichero de acceso de Subversion. Tenga en cuenta que las rutas se especifican o bien como repos:ruta o simplemente ruta. Si no especifica un repositorio particular, la regla de acceso se aplicará a todos los repositorios bajo SVNParentPath. El formato del fichero de política de autorización utilizado por mod_authz_svn se describe en Sección 3.1.6, “Autorización basada en rutas”

3.1.5. Múltiples repositorios

Si ha utilizado la directiva SVNParentPath no necesita cambiar el fichero de configuración de Apache cada vez que añada un nuevo repositorios de Subversion. ¡Simplemente cree el nuevo repositorio bajo la misma ruta que el primer repositorio y ya está! En mi compañía yo tengo acceso directo a esa carpeta específica via SMB (el acceso normal de ficheros de Windows). Por lo que simplemente

creo una nueva carpeta allí, ejecuto el comando de TortoiseSVN TortoiseSVN → Crear repositorio aquí... y un nuevo proyecto que ya tiene hogar...

Si está utilizando Subversion 1.3 o posterior, puede utilizar la directiva `SVNListParentPath` on para permitir que Apache produzca un listado de todos los proyectos disponibles si apunta su navegador a la ruta raíz en vez de a un repositorio en concreto.

Si la versión de su servidor de Subversion es anterior a la 1.3, obtendrá una fastidiosa página de error. Para obtener una bonita lista de todos los proyectos disponibles en vez de ese error, puede utilizar el siguiente script PHP que genera el índice automáticamente. (Necesitará instalar PHP en su servidor para utilizar el script que aparece a continuación).

```
<html>
<head>
<title>Subversion Repositories</title>
</head>
<body>

<h2>Subversion Repositories</h2>
<p>
<?php
    $svnparentpath = "C:/svn";
    $svnparenturl = "/svn";

    $dh = opendir( $svnparentpath );
    if( $dh ) {
        while( $dir = readdir( $dh ) ) {
            $svndir = $svnparentpath . "/" . $dir;
            $svnbdir = $svndir . "/db";
            $svnfstypefile = $svnbdir . "/fs-type";
            if( is_dir( $svndir ) && is_dir( $svnbdir ) ) {
                echo "<a href=\"\" . $svnparenturl . "/" .
                    $dir . "\">" . $dir . "</a>\n";
                if( file_exists( $svnfstypefile ) ) {
                    $handle = fopen ( "$svnfstypefile", "r" );
                    $buffer = fgets( $handle, 4096 );
                    fclose( $handle );
                    $buffer = chop( $buffer );
                    if( strcmp( $buffer, "fsfs" ) == 0 ) {
                        echo " (FSFS) <br />\n";
                    } else {
                        echo " (BDB) <br />\n";
                    }
                } else {
                    echo " (BDB) <br />\n";
                }
            }
        }
        closedir( $dh );
    }
?>
</p>

</body>
</html>
```

Grabe las líneas anteriores en un fichero `svn_index.php` y grabe dicho fichero en su directorio web raíz. Ahora debe decirle a Apache que muestre dicha página en vez del error:

- Descomente (quite el caracter '#') de la siguiente línea en su fichero de configuración de Apache:

```
#LoadModule rewrite_module modules/mod_rewrite.so
```

- Añada las siguientes líneas justo después del bloque <Location> donde define sus parámetros de Subversion:

```
RewriteEngine on
RewriteRule ^/svn$ /svn_index.php [PT]
RewriteRule ^/svn/$ /svn_index.php [PT]
RewriteRule ^/svn/index.html$ /svn_index.php [PT]
```

3.1.6. Autorización basada en rutas

El módulo `mod_authz_svn` le permite un control detallado de los permisos de acceso basado en nombres de usuarios y rutas de repositorios. Esto está disponible con el servidor de Apache, y desde Subversion 1.3 también está disponible con `svnserve`.

Esto sería un fichero de ejemplo:

```
[groups]
admin = john, kate
devteam1 = john, rachel, sally
devteam2 = kate, peter, mark
docs = bob, jane, miguel
training = zak
# Regla de acceso por defecto para TODOS los repositorios
# Todo el mundo puede leer, los administradores pueden escribir,
# Donpe Ligro está excluido.
[/]
* = r
@admin = rw
donpeligro =
# Permitir a los desarrolladores acceso completo
# al repositorio de su proyecto
[proj1:/]
@devteam1 = rw
[proj2:/]
@devteam2 = rw
[bigproj:/]
@devteam1 = rw
@devteam2 = rw
trevor = rw
# Dar a los documentadores acceso de escritura
# a todas las carpetas de documentación
[/trunk/doc]
@docs = rw
# Dar a los becarios acceso de escritura
# sólo al repositorio de pruebas
[TrainingRepos:/]
@training = rw
```

Tenga en cuenta que comprobar cada ruta puede ser una operación costosa, particularmente en el caso del registro de revisiones. El servidor toma cada ruta cambiada en cada revisión y comprueba si se puede leer, lo que puede ser lento en revisiones que afecten a un gran número de ficheros.

La autenticación y la autorización son procesos separados. Si un usuario desea tener acceso a una ruta de un repositorio, tiene que cumplir *ambos* requisitos, los requerimientos usuales de autenticación y los requerimientos de autorización del fichero de acceso.

3.1.7. Autenticación con un dominio de Windows

Como habrá notado necesita introducir una entrada usuario/contraseña en el fichero `passwd` para cada usuario de forma separada. Y si (por razones de seguridad) quiere que sus usuarios cambien periódicamente sus contraseñas tendrá que hacer el cambio de forma manual.

Pero hay una solución a este problema - al menos si accede al repositorio desde dentro de una LAN con un controlador de dominio de Windows: `mod_auth_sspi`!

El módulo original de SSPI lo ofreció Syneapps incluyendo el código fuente. Pero su desarrollo se paró. No se desespere, la comunidad lo ha retomado y mejorado. Tiene un nuevo hogar en *SourceForge* [<http://sourceforge.net/projects/mod-auth-sspi/>].

- Descargue el módulo, copie el fichero `mod_auth_sspi.so` en la carpeta de módulos de Apache.
- Edite el fichero de configuración de Apache: añada la línea

```
LoadModule sspi_auth_module modules/mod_auth_sspi.so
```

a la sección de `LoadModule`. Asegúrese de insertar esta línea *antes* de la línea

```
LoadModule auth_module modules/mod_auth.so
```

- Para asegurarse de que la ruta de Subversion utiliza este tipo de autenticación tiene que cambiar la línea

```
AuthType Basic
```

a

```
AuthType SSPI
```

y también deberá añadir

```
SSPIAuth On
SSPIAuthoritative On
SSPIDomain <domaincontroller>
SSPIOfferBasic On
```

dentro del bloque `<Location /svn>`. Si no tiene un controlador de dominio, deje el nombre del control de dominio como `<domaincontroller>`.

Tenga en cuenta que si se autentifica utilizando SSPI, ya no necesitará la línea `AuthUserFile` para definir un fichero de contraseñas. En su lugar, Apache autentifica su usuario y contraseña contra su dominio de Windows. También necesitará actualizar la lista de usuarios en su `svnaccessfile` para referirse a `DOMINIO\usuario`.



Sugerencia

Los ficheros `AuthzSVNAccessFile` de Subversion distinguen mayúsculas y minúsculas respecto a los nombres de usuario ("JuanPerez" es diferente de "juanperez").

En el mundo de Microsoft, los dominios y nombres de usuarios de Windows no distinguen mayúsculas y minúsculas. Incluso así, a algunos administradores de redes les gusta crear las cuentas de usuario en "CamelCase" (por ejemplo, "JuanPerez").

La diferencia puede morderle cuando utilice la autenticación SSPI ya que el dominio de Windows y los nombres de usuario se pasan a Subversion exactamente como los haya tecleado el usuario en la ventana. Internet Explorer a menudo pasa el nombre de usuario a Apache automáticamente utilizando el formato con el que se creó la cuenta.

El resultado final es que puede necesitar al menos dos entradas en su fichero AuthzSVNAccessFile para cada usuario -- una entrada en minúsculas y una entrada con el formato que Internet Explorer pasa a Apache. También tendrá que aleccionar a sus usuarios para que escriban sus credenciales en minúsculas cuando accedan a los repositorios utilizando TortoiseSVN.

Los registros de Error y Acceso de Apache son su mejor aliado para descifrar problemas como estos, ya que le ayudarán a determinar la cadena de nombre de usuario pasada al módulo AuthzSVNAccessFile de Subversion. Puede que necesite experimentar con el formato exacto de la cadena del usuario en el svnaccessfile (por ejemplo, DOMINIO\usuario contra DOMINIO//usuario) para conseguir que todo funcione.



SSL e Internet Explorer

Si está asegurando su servidor con SSL y utiliza la autenticación contra un dominio de Windows se encontrará que la navegación de repositorios con el Internet Explorer ya no funcionará. No se preocupe - es sólo que Internet Explorer no se puede autenticar. Los demás navegadores no tienen ese problema y tanto TortoiseSVN como cualquier otro cliente de Subversion todavía podrán autenticarse.

Si todavía quiere utilizar IE para navegar en el repositorio, puede:

- Definir una directiva separada <Location /ruta> en el fichero de configuración de apache, y añada SSPIBasicPreferred On. Esto permitirá que IE se autentique de nuevo, pero los demás navegadores y Subversion no se podrán autenticar contra esa ruta.
- Ofrezca también la navegación sin autenticación encriptada (sin SSL). Extrañamente, IE no tiene ningún problema para autenticarse si la conexión no está asegurada con SSL.
- En la configuración "estándar" de ssl a menudo aparecen las siguientes líneas en el host virtual ssl de Apache:

```
SetEnvIf User-Agent ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
```

Hay (¿había?) buenas razones para esta configuración, lea http://www.modssl.org/docs/2.8/ssl_faq.html#ToC49 Pero si desea autenticación NTLM deberá utilizar keepalive: http://www.microsoft.com/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/qos_enablekeepalives.asp Si descomenta todo el "SetEnvIf" debería ser capaz de autenticar IE con la autenticación de Windows sobre SSL contra un Apache sobre Win32 con el módulo incluido mod_auth_sspi.

3.1.8. Múltiples orígenes de autenticación

También es posible tener más de un origen de autenticación para su repositorio de Subversion. Para conseguirlo, debe hacer que cada tipo de autenticación sea no-autoritario, para que Apache compruebe múltiples orígenes buscando un par usuario/contraseña que concuerden.

Un escenario común es utilizar tanto la autenticación de dominio de Windows como un fichero passwd, para que pueda dar acceso a SVN a usuarios que no tienen usuario en el dominio de Windows.

- Para habilitar tanto la autenticación de dominio de Windows como el fichero passwd, añada las siguientes entradas en el bloque <Location> de su fichero de configuración de Apache:

```
AuthAuthoritative Off
SSPIAuthoritative Off
```

Aquí hay un ejemplo de la configuración completa de Apache para combinar la autenticación de dominio de Windows y el fichero passwd:

```
<Location /svn>
DAV svn
SVNListParentPath on
SVNParentPath D:\SVN

AuthName "Repositorios de Subversion"
AuthzSVNAccessFile svnaccessfile.txt

# Usuarios de Dominio NT.
AuthType SSPI
SSPIAuth On
SSPIAuthoritative On
SSPIDomain <domaincontroller>
SSPIOfferBasic On

# Usuarios htpasswd.
AuthType Basic
AuthAuthoritative Off
AuthUserFile passwd

Require valid-user
</Location>
```

3.1.9. Asegurando el servidor con SSL

El servidor Apache no tiene soporte SSL instalado por defecto por culpa de las restricciones de exportación de EE.UU. Pero puede descargar el módulo necesario de algún otro lugar e instalarlo usted mismo.

1. Primero necesitará los ficheros necesarios para habilitar SSL. Puede encontrarlos en el paquete disponible en <http://hunter.campus.com/>. Simplemente descomprima el paquete y luego copie el fichero mod_ssl.so a la carpeta modules de Apache, y el fichero openssl.exe a la carpeta bin. También debe copiar el fichero conf/ssl.conf a la carpeta conf de Apache.
2. Abra el fichero ssl.conf en la carpeta de configuración del Apache con un editor de texto.
3. Ponga un carácter de comentario (#) al inicio de estas líneas:

```
DocumentRoot "c:/apache/htdocs"
ServerName www.example.com:443
ServerAdmin you@example.com
ErrorLog logs/error_log
TransferLog logs/access_log
```

4. cambie la línea

```
SSLCertificateFile conf/ssl.crt/server.crt
```

por

```
SSLCertificateFile conf/ssl/my-server.cert
```

la línea

```
SSLCertificateKeyFile conf/ssl.key/server.key
```

por

```
SSLCertificateKeyFile conf/ssl/my-server.key
```

y la línea

```
SSLMutex file:logs/ssl_mutex
```

por

```
SSLMutex default
```

5. Borre las líneas

```
<IfDefine SSL>
```

y

```
</IfDefine>
```

6. Abra el fichero de configuración de Apache (`httpd.conf`) y descomente la línea

```
#LoadModule ssl_module modules/mod_ssl.so
```

7. Openssl necesita un fichero de configuración. Puede descargar uno que funciona desde <http://tud.at/programm/openssl.cnf>. Grabe el fichero en `bin/openssl.cnf`. Por favor, tenga en cuenta que el fichero debe ser de tipo `*.cnf`. Windows trata dichos ficheros de una forma especial, ¡pero realmente es simplemente un fichero de texto!

8. Lo siguiente que debe hacer es crear un certificado SSL. Para hacer esto, abra un símbolo del sistema (Ventana DOS) y cambie a la carpeta de apache (por ejemplo, `C:\Archivos de programa\Apache Group\Apache2`) y escriba el siguiente comando:

```
bin\openssl req -config bin\openssl.cnf -new -out my-server.csr
```

Se le preguntará una contraseña. Por favor no utilice palabras sencillas, sino frases completas, por ejemplo, unos versos de un poema. Cuanto más larga sea la frase, mejor. También tendrá que introducir la URL de su servidor. Todas las demás respuestas son opcionales pero le recomendamos que las rellene también.

Normalmente el fichero `privkey.pem` se crea automáticamente, pero si no, necesitará teclear este comando para generarlo:

```
bin\openssl genrsa -out privkey.pem 2048
```

Ahora escriba los comandos

```
bin\openssl rsa -in privkey.pem -out my-server.key
```

y (en una única línea)

```
bin\openssl x509 -in my-server.csr -out my-server.cert  
-req -signkey my-server.key -days 4000
```

Esto creará un certificado que expirará en 4000 días. Y finalmente, introduzca (también en una única línea):

```
bin\openssl x509 -in my-server.cert -out my-server.der.crt -outform DER
```

Estos comandos crearán algunos ficheros en la carpeta Apache (my-server.der.crt, my-server.csr, my-server.key, .rnd, privkey.pem, my-server.cert). Copie los ficheros a la carpeta conf/ssl (por ejemplo, C:\Archivos de programa\Apache Group\Apache2\conf\ssl) - si esta carpeta no existe deberá crearla primero.

9. Reinicie el servicio de Apache.
10. Apunte su navegador a `https://nombredelservidor/svn/project ...`



Forzando el acceso SSL

Cuando haya preparado SSL para hacer su repositorio más seguro, puede que desee deshabilitar el acceso normal via no-ssl (http) y sólo permitir el acceso https. Para hacer esto, debería añadir otra directiva al bloque <Location> de Subversion: SSLRequireSSL.

Un bloque <Location> de ejemplo se parecería a éste:

```
<Location /svn>  
DAV svn  
SVNParentPath D:\SVN  
SSLRequireSSL  
AuthType Basic  
AuthName "Repositorios de Subversion"  
AuthUserFile passwd  
#AuthzSVNAccessFile svnaccessfile  
Require valid-user  
</Location>
```

3.2. Servidor Basado en Svnserve

3.2.1. Introducción

Puede haber situaciones donde no es posible utilizar Apache como servidor. Afortunadamente, Subversion incluye Svnserve - un ligero servidor aislado que utiliza un protocolo propio sobre una conexión TCP/IP ordinaria.

En la mayoría de los casos svnserve es más fácil de instalar y se ejecuta más rápido que el servidor

basado en Apache.

3.2.2. Instalando svnserve

1. Obtenga la última versión de Subversion desde <http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=91>.
2. If you already have a version of Subversion installed, and svnserve is running, you will need to stop it before continuing.
3. Run the Subversion installer. If you run the installer on your server you can skip step 4.
4. Open the windows-explorer, go to the installation directory of Subversion (usually C:\Program Files\Subversion) and in the bin directory, find the files `svnserve.exe`, `libdb44.dll`, `libeay32.dll` and `ssleay32.dll` - copy these files into a directory on your server e.g. `c:\svnserve`

3.2.3. Ejecutando svnserve

Now that svnserve is installed, you need it running on your server. The simplest approach is to run the following from a DOS shell or create a windows shortcut:

```
svnserve.exe --daemon
```

svnserve will now start waiting for incoming requests on port 3690. The `--daemon` switch tells svnserve to run as a daemon process, so it will always exist until it is manually terminated.

Si aún no ha creado un repositorio, siga las instrucciones dadas en la instalación del servidor basado en Apache Sección 3.1.4, “Configuración”.

Para comprobar que svnserve está funcionando, utilice TortoiseSVN → Navegador para ver un repositorio.

Asumiendo que su repositorio está en `c:\repos\TestRepo`, y que su servidor se llama `localhost`, introduzca:

```
svn://localhost/repos/TestRepo
```

cuando le pregunte el navegador de repositorios.

You can also increase security and save time entering Url's with svnserve by using the `--root` switch to set the root location and restrict access to a specified directory on the server:

```
svnserve.exe --daemon --root drive:\path\to\repository
```

Using the previous test as a guide, svnserve would now run as:

```
svnserve.exe --daemon --root c:\repos
```

And in TortoiseSVN our repo-browser Url is now shortened to:

```
svn://localhost/TestRepo
```

Note that the `--root` switch is also needed if your repository is located on a different partition or drive than the location of svnserve on your server.



Aviso

No cree o acceda a un repositorio Berkeley DB en una unidad de red compartida. *No* puede existir en un sistema de archivos remoto. Ni siquiera si tiene la unidad de red mapeada a una letra de unidad. Si intenta usar Berkeley DB en una unidad de red compartida, los resultados son imprevisibles - puede ver desde el principio errores misteriosos, o pueden pasar meses antes de que descubra que su base de datos del repositorio está corrupta de una forma inimaginable.

3.2.3.1. Ejecutar svnserve como un servicio

If you are concerned about always having a user logged in on your server, or worried about someone shutting down svnserve or forgetting to restart it after a reboot, it is possible to run svnserve as a windows service. Starting with Subversion 1.4, svnserve can be installed as a native windows service, in previous versions it can be installed using a wrapper.

To install svnserve as a native windows service, execute the following command all on one line to create a service which is automatically started when windows starts.

```
sc create svnserve binpath= "c:\svnserve\svnserve.exe --service  
--root c:\repos" displayname= "Subversion" depend= tcpip start= auto
```



Sugerencia

Microsoft now recommend services to be run as under either the Local Service or Network Service account. Refer to *The Services and Service Accounts Security Planning Guide*

[<http://www.microsoft.com/technet/security/topics/serversecurity/serviceaccount/default.msp>]. To create the service under the Local Service account, append the following to the example above.

```
obj= "NT AUTHORITY\LocalService"
```

Note that you would have to give the Local Service account appropriate rights to both Subversion and your repositories, as well as any applications which are used by hook scripts.

To install svnserve using a wrapper, one written specifically for svnserve is *SvnService*. Magnus Norddahl adapted some skeleton code from Microsoft, and further improvements have been made by Daniel Thompson. Daniel's version is available for download from *tigris.org* [<http://tortoisesvn.tigris.org/files/documents/406/29202/SVNServiceDT.zip>].

More generic tools like *firedaemon* [<http://www.firedaemon.com/>] will also work. Note that you will still need to run svnserve with the `--daemon` switch.

Finalmente, si tiene acceso al kit de recursos de Windows 2000/XP/2003 puede utilizar *SrvAny* [<http://support.microsoft.com/kb/q137890/>] de Microsoft. Esta es la forma oficial de Microsoft para ejecutar programas como servicios, pero es un poco liosa (necesita editar el registro) y si para el servicio matará svnserve inmediatamente sin dejarle terminar limpiamente. Si no desea instalar el kit de recursos entero, puede descargar *sólo* los componentes de SrvAny desde *Daniel Petri* [<http://www.petri.co.il/software/srvany.zip>].

3.2.4. Autenticación con svnserve

La configuración por defecto de svnserve proporciona acceso anónimo de sólo-lectura. Esto significa que puede utilizar una URL de tipo `svn : //` para obtener y actualizar, o utilizar el navegador de

repositorios en TortoiseSVN para ver el repositorio, pero no podrá confirmar ningún cambio.

Para permitir acceso de escritura en un repositorio, necesitará editar el fichero `conf/svnserve.conf` en el directorio de su repositorio. Este fichero controla la configuración del servicio `svnserve`, y también contiene información útil.

Puede habilitar el acceso anónimo para escritura simplemente poniendo:

```
[general]
anon-access = write
```

Sin embargo, no sabrá quién ha hecho cambios en el repositorio, dado que la propiedad `svn:author` estará vacía. Tampoco podrá controlar quién puede hacer cambios en el repositorio. ¡Esta es una configuración algo arriesgada!

Una forma de conseguir esto es crear una base de datos de contraseñas:

```
[general]
anon-access = none
auth-access = write
password-db = fichero de usuarios
```

Donde `fichero de usuarios` es un fichero que existe en el mismo directorio que `svnserve.conf`. Este fichero también puede estar en cualquier otro sitio de su sistema de ficheros (útil cuando tiene múltiples repositorios que necesitan los mismos derechos de acceso) y puede ser referenciado utilizando una ruta absoluta, o una ruta relativa del directorio `conf`. Si incluye una ruta, debe estar escrita en `/a/la/forma/de/unix`. No funcionará si utiliza `\` o letras de unidades. El `fichero de usuarios` debería tener una estructura como ésta:

```
[users]
usuario = contraseña
...
```

Este ejemplo denegaría cualquier acceso a los usuarios no autenticados (anónimo), y daría acceso de lectura-escritura a los usuarios listados en `fichero de usuarios`.



Sugerencia

Si mantiene múltiples repositorios utilizando la misma base de datos de contraseñas, la utilización de un dominio de autenticación le hará la vida más fácil a los usuarios, dado que TortoiseSVN puede almacenar en caché sus credenciales para que sólo tenga que introducirlas una vez. Se puede encontrar más información en el libro de Subversion, específicamente en las secciones *Create a 'users' file and realm* [<http://svnbook.red-bean.com/en/1.1/ch06s03.html#svn-ch-6-sect-3.2.1>] y *Client Credentials Caching* [<http://svnbook.red-bean.com/en/1.1/ch06s02.html#svn-ch-6-sect-2.2>]

3.2.5. Autenticación con svn+ssh

Otra forma de autenticar a los usuarios con un servidor basado en `svnserve` es utilizar un shell seguro (SSH) para encapsular las peticiones a través suyo.

Con esta aproximación, `svnserve` no se ejecuta como un servicio, en cambio, el shell seguro inicia `svnserve` por usted, ejecutándolo como el usuario autenticado SSH. Para habilitar esto, necesita un servicio de shell seguro en su servidor.

It is beyond the scope of this documentation to detail the installation and setup of a secure shell, however you can find further information in the *TortoiseSVN FAQ* [<http://tortoisesvn.net/faq>]. Search for “SSH”.

Puede encontrar más información sobre svnserve en el *libro de SVN* [<http://svnbook.red-bean.com>].

3.2.6. Autorización basada en rutas con svnserve

Empezando con Subversion 1.3, svnserve soporta el mismo esquema de autorización basada en rutas que está disponible en el módulo `mod_authz_svn` de Apache. Necesita editar el fichero `conf/svnserve.conf` dentro del directorio de su repositorio y añadir una línea refiriéndose a su fichero de autorización.

```
[general]
authz-db = authz
```

Aquí, `authz` es un fichero que debe existir y que define los permisos de acceso. Puede utilizar un fichero separado por cada repositorio, o utilizar el mismo fichero para varios repositorios. Si desea una descripción del formato del fichero, lea Sección 3.1.6, “Autorización basada en rutas”.

Capítulo 4. El repositorio

Sin importar el protocolo que use para acceder a sus repositorios, siempre necesita crear al menos un repositorio. Esto puede hacerse o bien con el cliente de línea de comandos de Subversion o con TortoiseSVN.

Si todavía no ha creado un repositorio de Subversion, ahora es el momento de hacerlo.

4.1. Creación de repositorios

Puede crear un repositorio bajo el sistema FSFS o bien con el viejo pero estable formato Berkeley Database (BDB). El formato FSFS es más rápido y ahora funciona en carpetas compartidas y en Windows 98 sin problemas. El formato BDB es más estable porque se ha probado durante más tiempo. Lea el *Capítulo 5, Repository Data-Stores* [<http://svnbook.red-bean.com/en/1.1/ch05.html#svn-ch-5-sect-1.2.A>] en el libro SVN si desea más información.

4.1.1. Creando un repositorio con el cliente de línea de comandos

1. Cree una carpeta vacía con el nombre SVN (por ejemplo, D:\SVN\), que se usará como la raíz de todos sus repositorios.
2. Cree otra carpeta MiNuevoRepositorio dentro de D:\SVN\
3. Abra un símbolo del sistema (o ventana MS-DOS), vaya a D:\SVN\ y escriba

```
svnadmin create --fs-type bdb MiNuevoRepositorio
```

o

```
svnadmin create --fs-type fsfs MiNuevoRepositorio
```

Ahora ya tiene un nuevo repositorio que se encuentra en D:\SVN\MiNuevoRepositorio.

4.1.2. Creando el repositorio con TortoiseSVN

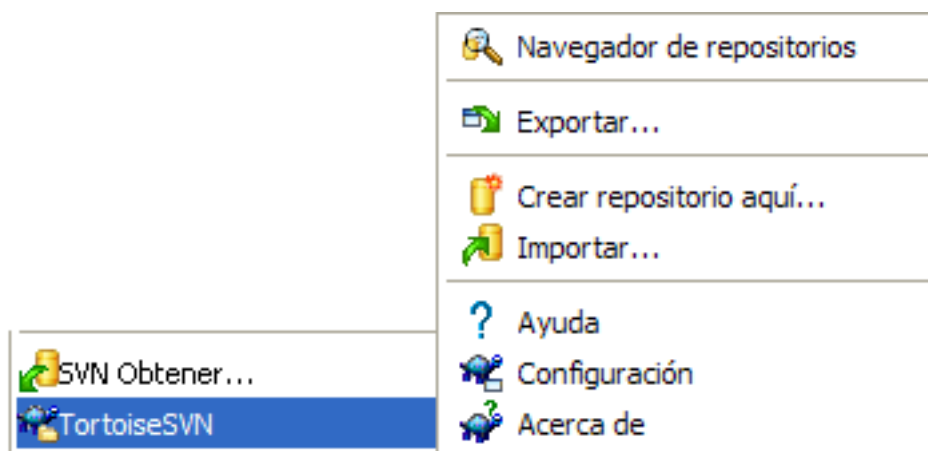


Figura 4.1. El menú de TortoiseSVN para carpetas no versionadas

1. Abra el explorador de Windows
2. Cree una nueva carpeta y llámela por ejemplo SVNRepositorio
3. Haga click con el botón derecho sobre la carpeta recién creada y seleccione TortoiseSVN → Crear Repositorio aquí....

Entonces se creará un repositorio dentro de la nueva carpeta. *¡¡No edite los ficheros!!!*. Si obtiene algún error asegúrese de que la carpeta esté vacía y que no esté protegida contra escritura.

4.1.3. Acceso local al repositorio

Para acceder a su repositorio local, necesita la ruta a esa carpeta. Recuerde que Subversion espera todas las rutas de repositorios con el formato `file:///C:/RepositorioSVN/`. Tenga en cuenta el uso de las barras de dividir.

Para acceder a un repositorio que se encuentre en una unidad de red compartida puede o bien utilizar mapeado de unidades, o bien usar la ruta UNC. El formato de una ruta UNC es `file://NombreDelServidor/ruta/al/repositorio/`. Observe que sólo hay 2 barras invertidas aquí.

Antes de SVN 1.2, las rutas UNC tenían que estar dadas en la forma más oscura `file:///\\NombreDelServidor/ruta/al/repositorio`. Esta forma aún se puede utilizar, pero no está recomendada.



Aviso

No cree o acceda a un repositorio Berkeley DB en una unidad de red compartida. *No* puede existir en un sistema de archivos remoto. Ni siquiera si tiene la unidad de red mapeada a una letra de unidad. Si intenta usar Berkeley DB en una unidad de red compartida, los resultados son imprevisibles - puede ver desde el principio errores misteriosos, o pueden pasar meses antes de que descubra que su base de datos del repositorio está corrupta de una forma inimaginable.



Sugerencia

Si realmente necesita acceder a un repositorio a través de una unidad de red compartida, cree el repositorio con el formato `fsfs`. Si también tiene que dar acceso via servidor, necesitará un Servidor de Subversion 1.1 o posterior.

4.2. Copia de seguridad del Repositorio

Cualquiera que sea el tipo de repositorio que use, es de vital importancia que mantenga copias de seguridad regulares, y que verifique la copia. Si el servidor falla, puede ser capaz de acceder a la versión más reciente de sus ficheros, pero sin el repositorio toda su historia se perderá para siempre.

La manera más sencilla (pero no recomendada) es tan simple como copiar la carpeta del repositorio a un medio de backup. Sin embargo, tiene que estar absolutamente seguro de que no hay ningún proceso accediendo a los datos. En este contexto, acceder significa *cualquier* tipo de acceso. En un repositorio BDB se escribe incluso cuando la operación sólo parece que necesite leer, como obtener el estado. Si se accede a su repositorio durante la copia (se deja un navegador web abierto, WebSVN, etc.) la copia puede que no valga para nada.

El método recomendado es ejecutar

```
svnadmin hotcopy ruta/al/repositorio ruta/al/backup --clean-logs
```

para crear una copia del repositorio de forma segura. Entonces hacer una copia de seguridad de la copia. La opción `--clean-logs` no es necesaria, pero quita cualquier fichero de log redundante cuando hace una copia de seguridad de un repositorio BDB; lo que puede ahorrar algo de espacio.

La herramienta `svnadmin` se instala automáticamente cuando instala el cliente de línea de comandos de Subversion. Si está instalando las herramientas de línea de comandos en un PC con Windows, la mejor forma de hacerlo es descargar la versión con Windows installer. Está comprimido de forma más eficiente que la versión `.zip`, por lo que la descarga es menor, y se encarga de establecer las rutas por usted. Puede descargar la última versión del cliente de línea de comandos desde <http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=91>.

4.3. Scripts gancho

Un script gancho es un programa que se activa por algún evento del repositorio, tal como la creación de una nueva revisión o la modificación de una propiedad no versionada. A cada gancho se le proporciona suficiente información para decirle qué evento es, sobre qué destino(s) se está operando, y el nombre de usuario de la persona que lanzó el evento. Dependiendo de la salida del gancho o del estado de vuelta, el programa gancho puede continuar la acción, pararla, o suspenderla de alguna forma. Por favor vea el capítulo sobre *Scripts Gancho* [<http://svnbook.red-bean.com/en/1.1/ch05s02.html#svn-ch-5-sect-2.1>] en el Libro de Subversion donde encontrará todos los detalles sobre la forma en la que están implementados los ganchos.

Puede encontrar scripts de ganchos en el directorio `hooks` del repositorio. Estos scripts de ejemplo son válidos para servidores Unix/Linux pero necesitan modificarse si su servidor está basado en Windows. El gancho puede ser un fichero batch o un ejecutable. El siguiente ejemplo muestra un fichero batch que puede ser usado para implementar un ganch pre-revprop-change.

```
rem Solo se permite cambiar mensajes de registro.
if "%4" == "svn:log" exit 0
echo No se puede cambiar la property '%4' >&2
exit 1
```

Tenga en cuenta que cualquier cosa que mande a la salida estándar se descartará. Si desea que aparezca un mensaje en el diálogo Confirmación Rechazada debe enviarlo a la salida de error. En un fichero batch esto se consigue usando `>&2`

4.4. Enlaces de obtener

Si desea hacer disponible su repositorio de Subversion a los demás, puede que desea incluir un vínculo a él desde su sitio web. Una forma de hacer esto más accesible es incluir un *enlace de obtener* para otros usuarios de TSVN.

Cuando se instala TortoiseSVN, se registra un nuevo protocolo `tsvn:`. Cuando un usuario de TSVN pulsa en un enlace con ese protocolo, aparece el diálogo obtener automáticamente con la URL del repositorio ya rellena.

Para incluir dicho enlace en su propia página html, necesita añadir un código que será similar a este:

```
<a href="tsvn:https://tortoisesvn.tigris.org/svn/tortoisesvn/trunk"></a>
```

Por supuesto será más bonito si añade una imagen adecuada. Puede utilizar el *logotipo de TortoiseSVN* [<http://tortoisesvn.tigris.org/images/TortoiseCheckout.png>] o puede proporcionar su propia imagen.

```
<a href="tsvn:https://tortoisesvn.tigris.org/svn/tortoisesvn/trunk">
<img src=TortoiseCheckout.png></a>
```

Capítulo 5. Guía de uso diario

Este documento describe el uso diario del cliente TortoiseSVN. *No* es una introducción a los sistemas de control de versiones, y *no* es una introducción a Subversion (SVN). Es más como un lugar donde puede venir cuando sepa qué quiere hacer, pero no recuerde exactamente cómo hacerlo.

Si necesita una introducción al control de versiones con Subversion, le recomendamos que se lea el fantástico libro: *Subversion: La Guía Definitiva* [<http://svnbook.red-bean.com/>].

Este documento es también un trabajo en progreso, igual que lo son TortoiseSVN y Subversion. Si encuentra algún error, por favor háganoslo saber en la lista de correo para que podamos actualizar la documentación. Algunas de las capturas de pantalla en la Guía de Uso Diario (GUD) puede que no reflejen el estado actual del software. Le rogamos nos disculpe. Trabajamos en TortoiseSVN en nuestro tiempo libre.

- Debe tener ya instalado TortoiseSVN.
- Debe estar familiarizado con los sistemas de control de versiones.
- Debe conocer las bases de Subversion.
- Debe haber preparado un servidor y/o tener acceso a un repositorio de Subversion.

5.1. Empezando

5.1.1. Iconos sobreimpresionados

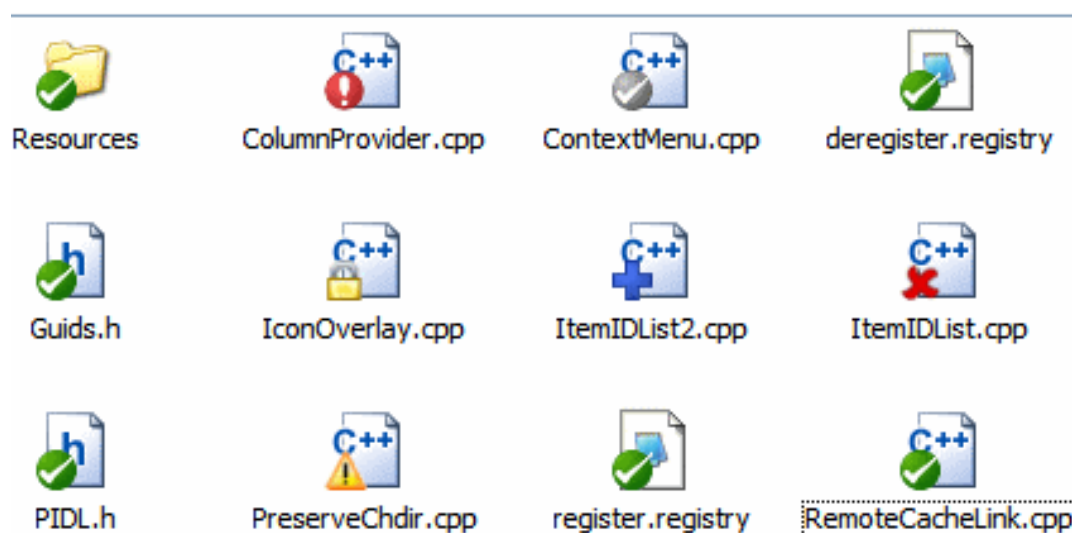


Figura 5.1. Explorador mostrando iconos sobreimpresionados

Una de las funciones más visibles de TortoiseSVN son los iconos sobreimpresionados que aparecen en los ficheros de su copia de trabajo. Estos le muestran de un vistazo qué ficheros han sido modificados. Lea Sección 5.7.1, “Iconos sobreimpresionados” para averiguar qué representa cada icono sobreimpresionado.

5.1.2. Menús contextuales

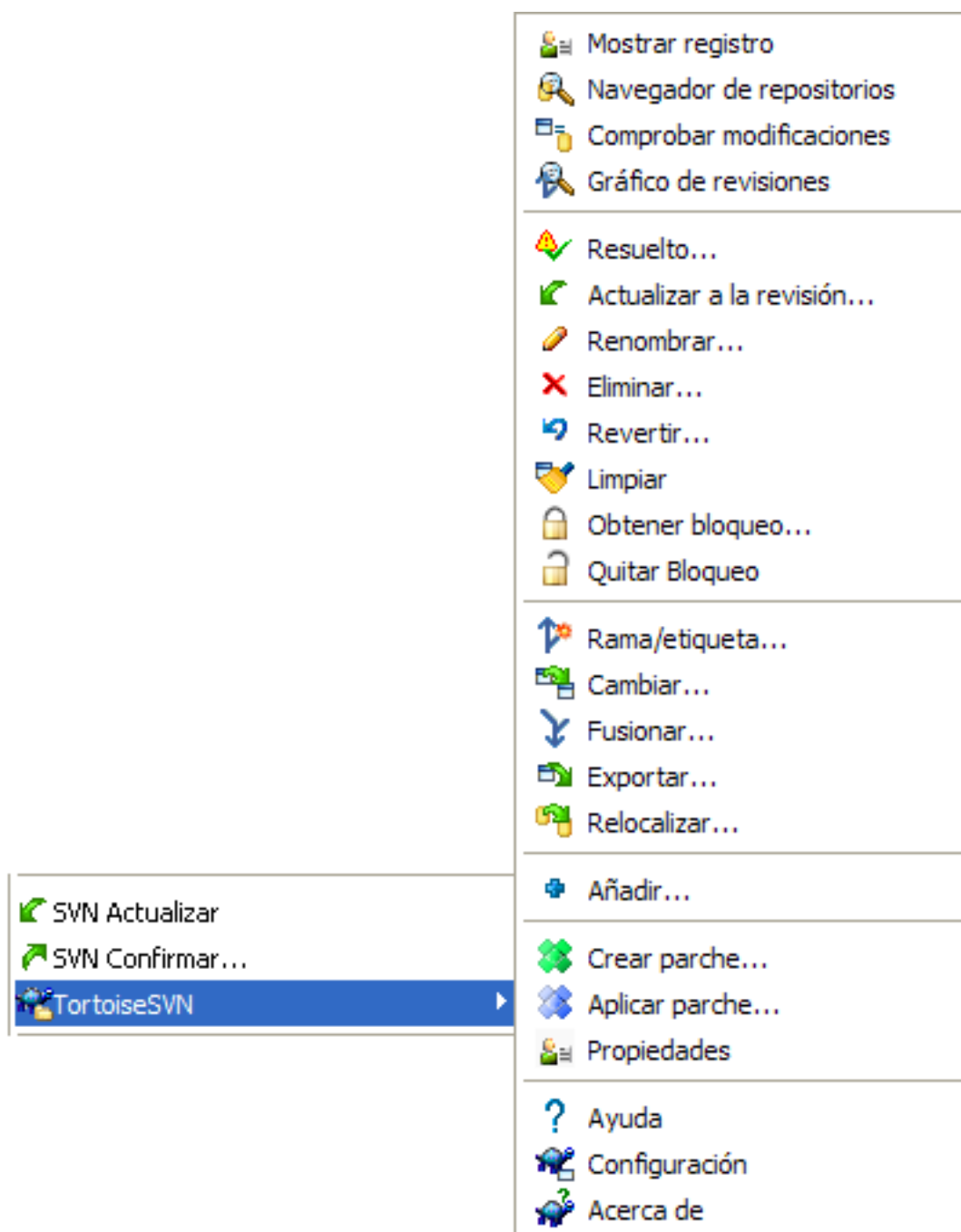


Figura 5.2. Menú contextual para un directorio bajo el control de versiones

Todos los comandos de TortoiseSVN se invocan desde el menú contextual del explorador de Windows. La mayoría se ven directamente, cuando hace click con el botón derecho en un fichero o una carpeta. Los comandos disponibles dependen de si el fichero o la carpeta o su carpeta padre está bajo el control de versiones o no. También puede ver el menú de TortoiseSVN como parte del menú archivo del explorador.

En algunos casos puede ver varias entradas de TortoiseSVN. ¡Esto no es un error!

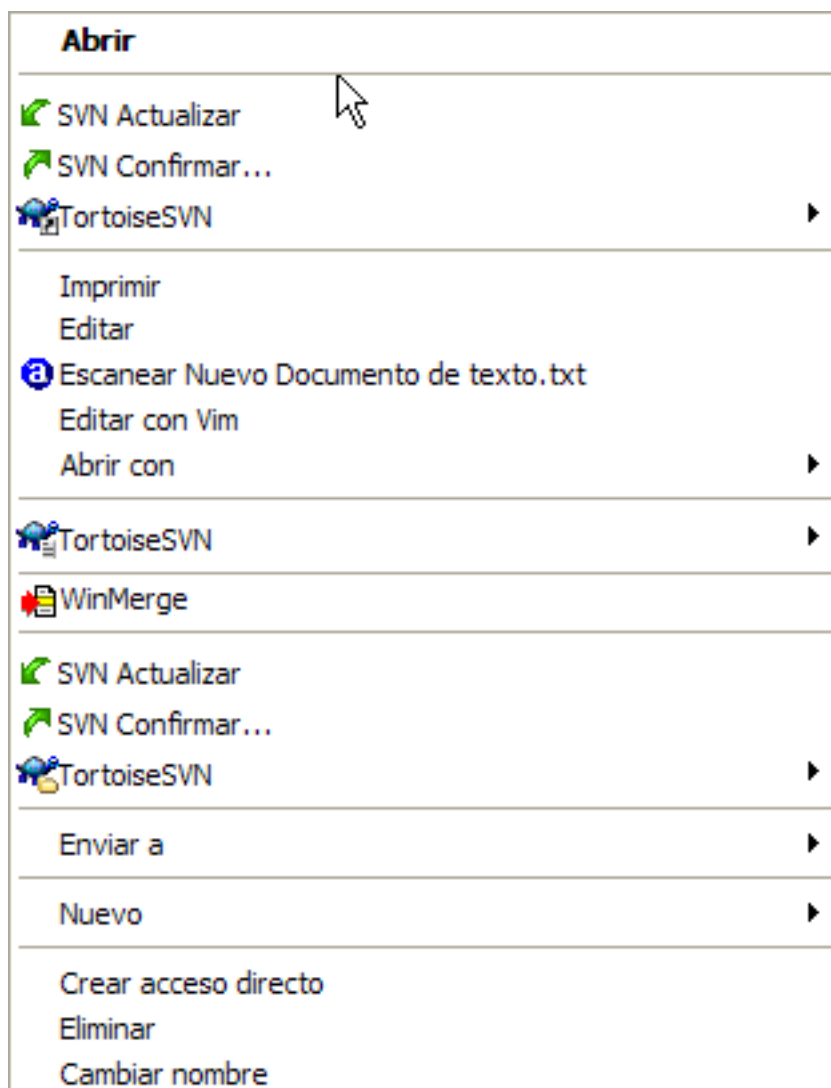


Figura 5.3. Menú archivo del explorador para un acceso directo en una carpeta versionada

Este ejemplo es para un acceso directo sin versionar dentro de una carpeta versionada, y en el menú de archivo del Explorador hay *tres* entradas para TortoiseSVN. Una es para la carpeta, otra para el acceso directo en sí mismo, y otra para el objeto al que apunta el acceso directo. Para ayudarle a distinguir entre ellos, los iconos tienen un indicador en la esquina inferior derecha para mostrarle que la entrada del menú es para un fichero, una carpeta, un acceso directo o para múltiples ítems seleccionados.

5.1.3. Arrastrar y soltar

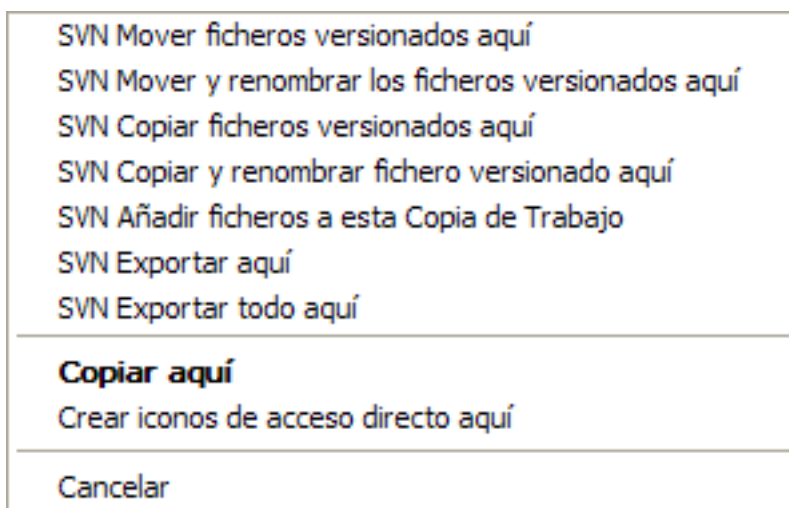


Figura 5.4. Menú de arrastre con el botón derecho para un directorio bajo el control de versiones

Otros comandos están disponibles como manejadores de arrastre, cuando arrastra con el botón derecho ficheros o carpetas a un nuevo destino dentro de copias de trabajo, o cuando arrastra con el botón derecho un fichero o una carpeta no versionados a un directorio que está bajo el control de versiones.

5.1.4. Atajos comunes

Algunas operaciones comunes tienen atajos de Windows bien conocidos, pero no aparecen en botones o en los menús. Si no puede averiguar cómo hacer algo obvio, como refrescar una vista, pulse aquí.

F1

La ayuda, por supuesto.

F5

Refresca la vista actual. Este es quizás el comando de una tecla más útil. Por ejemplo... en el Explorador esto refresca los iconos sobreimpresionados en su copia de trabajo. En el diálogo de confirmación volverá a reescanear la copia de trabajo para ver qué puede necesitar ser confirmado. En el diálogo de Mostrar Registro contactará con el repositorio de nuevo buscando los cambios más recientes.

Ctrl-A

Selecciona todo. Esto puede ser útil si obtiene un mensaje de error y quiere copiar y pegarlo en un email. Utilice Ctrl-A para seleccionar el mensaje de error y luego...

Ctrl-C

... copia el texto seleccionado.

5.1.5. Autenticación

Si el repositorio al que intenta acceder está protegido por contraseña, aparecerá un diálogo de autenticación.

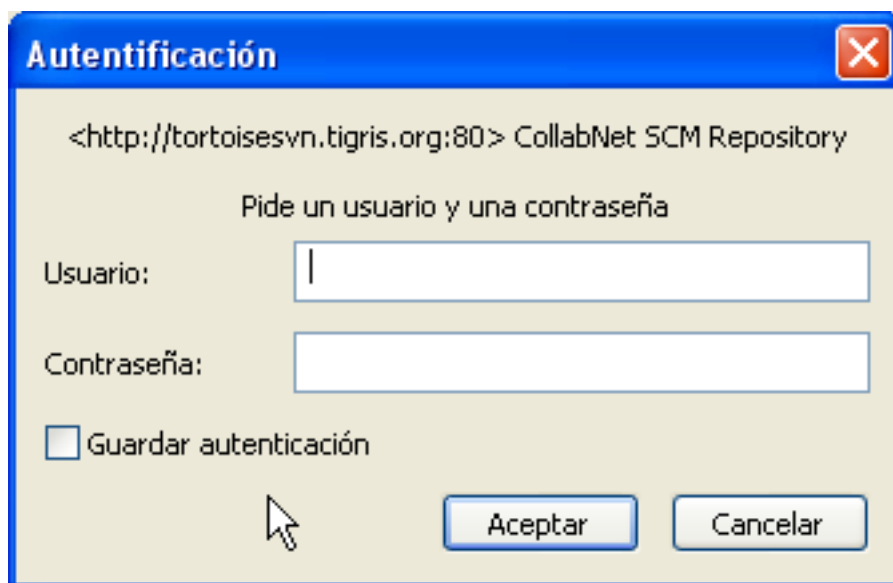


Figura 5.5. Diálogo de autenticación

Introduzca su usuario y contraseña. La casilla le permite que TortoiseSVN almacene las credenciales en el directorio por defecto de Subversion: `$APPDATA\Subversion\auth`, en tres subdirectorios:

- `svn.simple` contiene las credenciales para la autenticación básica (usuario/contraseña).
- `svn.ssl.server` contiene los certificados SSL de servidor.
- `svn.username` contiene las credenciales para autenticación sólo por usuario (sin necesidad de contraseña).

Se crea un fichero por cada servidor al que acceda, formateado como texto plano, así que puede utilizar un editor de texto para comprobar a qué servidor se aplica. Si desea que Subversion y TortoiseSVN (y cualquier otro cliente de Subversion) olviden los credenciales para un servidor en particular, tendrá que borrar el fichero correspondiente.

Si desea eliminar la caché de autenticación para *todos* los servidores, puede hacerlo desde la página **General** del diálogo de configuración de TortoiseSVN. Ese botón borrará todos los datos de autenticación cacheados de los directorios de autenticación de Subversion, así como cualquier dato de autenticación almacenado en el registro por versiones anteriores de TortoiseSVN. Lea Sección 5.26.1, “Configuración general”.

Para más información sobre cómo preparar su servidor para la autenticación y el control de acceso, vea Capítulo 3, *Preparando un servidor*



Sugerencia

Si tiene que autenticarse contra un dominio Windows NT, introduzca su usuario incluyendo el nombre de dominio, así: `MIDOMINIO/juanfdez`.

5.2. Importando datos en un repositorio

5.2.1. Organización del repositorio

Antes de que importe sus datos al repositorio, primero debería pensar cómo quiere organizar sus da-

tos. Si utiliza uno de los patrones recomendados lo tendrá luego mucho más fácil.

Hay algunas formas estándar y recomendadas de organizar un repositorio. La mayoría de la gente crea un directorio `trunk` (tronco) para alojar la "línea principal" del desarrollo, un directorio `branches` (ramas) para que contenga las copias/ramas, y un directorio `tags` (etiquetas) para contener las copias/etiquetas. Si un repositorio sólo aloja un proyecto, entonces la gente a menudo crea estos directorios en el primer nivel:

```
/trunk
/branches
/tags
```

Si un repositorio contiene múltiples proyectos, la gente a menudo indexa por ramas:

```
/trunk/paint
/trunk/calc
/branches/paint
/branches/calc
/tags/paint
/tags/calc
```

...o por proyecto:

```
/paint/trunk
/paint/branches
/paint/tags
/calc/trunk
/calc/branches
/calc/tags
```

Indexar por proyecto tiene sentido si los proyectos no están muy relacionados y cada uno se obtiene de forma individual. Para proyectos relacionados donde puede querer obtener todos los proyectos de golpe, o donde los proyectos están unidos y forman un paquete de distribución único, a menudo es mejor indexar por rama. De esta forma sólo tendrá un tronco para obtener, y las relaciones entre sub-proyectos se ven más fácilmente.

Si adopta una aproximación de primer nivel `/trunk /tags /branches`, no es necesario decir que tendrá que copiar el tronco completo para cada rama y etiqueta, y de alguna forma esta estructura ofrece la mayor flexibilidad.

Para proyectos que no tienen que ver, puede preferir utilizar repositorios separados. Cuando confirma los cambios, lo que cambia es el número de revisión del repositorio completo, no el número de revisión del proyecto. Si tiene 2 proyectos que no tengan que ver compartiendo un repositorio, eso puede llevar a que ocurran grandes lagunas en los números de revisión. Los proyectos Subversion y TortoiseSVN aparecen en el mismo servidor, pero son dos repositorios totalmente separados que permiten un desarrollo independiente, y no hay confusión sobre los números de compilación.

Por supuesto, tiene libertad para ignorar estos patrones comunes. Puede crear cualquier variación, la que mejor le venga a usted o a su equipo. Recuerde que cualquiera que sea la que elija, no es una elección inamovible. Puede reorganizar su repositorio en cualquier momento. Dado que las ramas y las etiquetas son directorios normales, TortoiseSVN puede mover o renombrarlas como desee.

Cambiar de una disposición a otra es sólo una cuestión de ejecutar una serie de movimientos en el lado del servidor; si no le gusta la forma en la que están organizadas las cosas en el repositorio, sólo tiene que ir moviendo los directorios.

Así que si no ha creado ya una estructura básica de carpetas en su repositorio debería hacerlo ahora:

1. cree una nueva carpeta en su disco duro
2. cree la estructura de carpetas de primer nivel dentro de esa carpeta - ¡no ponga ningún fichero allí todavía!
3. importe esta estructura en el repositorio via click con el botón derecho en la carpeta y seleccionando **TortoiseSVN → Importar....** Esto importará su carpeta temporal a la raíz del repositorio para crear la estructura básica del repositorio.

Tenga en cuenta que el nombre de la carpeta que está importando no aparece en el repositorio, sólo sus contenidos. Por ejemplo, cree la siguiente estructura de carpetas:

```
C:\Temp\New\trunk
C:\Temp\New\branches
C:\Temp\New\tags
```

Importe `C:\Temp\New` en la raíz del repositorio, que entonces contendrá:

```
/trunk
/branches
/tags
```

También puede usar el navegador de repositorios para crear nuevas carpetas directamente en el repositorio.

5.2.2. Importar

Antes de importar su proyecto en un repositorio debería:

1. Quitar todos los ficheros que no se necesitan para construir el proyecto (ficheros temporales, ficheros que se generan por un compilador como los *.obj, binarios compilados, ...)
2. Organizar los ficheros en carpetas y subcarpetas. Aunque es posible renombrar/mover los ficheros más tarde, ¡es muy recomendable que tenga la estructura del proyecto antes de importarlo!

Ahora seleccione la carpeta superior de la estructura de directorios del proyecto en el explorador de Windows, y haga click con el botón derecho para abrir el menú contextual. Seleccione el comando **TortoiseSVN → Importar...** y aparecerá un cuadro de diálogo:

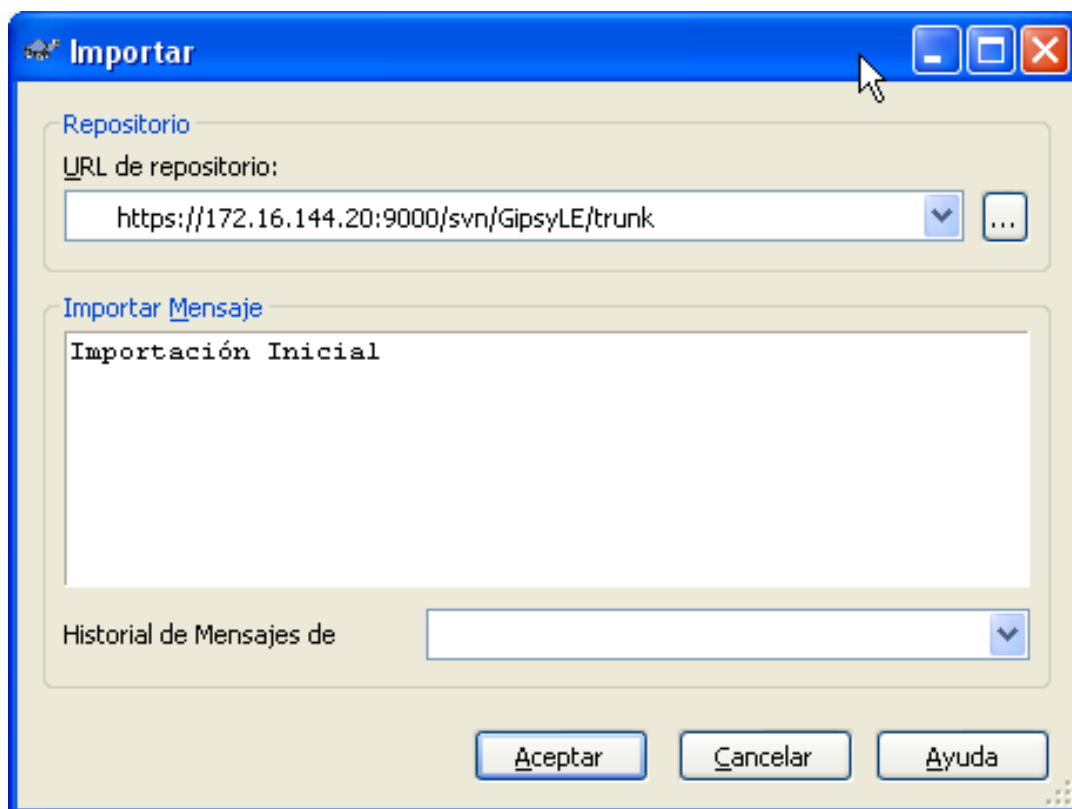


Figura 5.6. El diálogo Importar

En este diálogo tiene que introducir la URL del repositorio en el que quiere importar su proyecto.

El mensaje de importación se utiliza como un mensaje de registro.

Por defecto, los ficheros y carpetas que concuerden con los patrones globales de ignorar *no* se importan. Para cambiar este comportamiento, puede utilizar la casilla **Incluir ficheros ignorados**. Lea Sección 5.26.1, “Configuración general” para más información sobre cómo establecer un patrón global de ignorar.

Tan pronto como presione **Aceptar**, TortoiseSVN importa el árbol completo de directorios incluyendo todos los ficheros en el repositorio. Como antes, el nombre de la carpeta que está importando no aparece en el repositorio, sólo los contenidos de los ficheros. El proyecto ahora está almacenado en el repositorio bajo el control de versiones. Por favor tenga en cuenta que la carpeta que ha importado *¡NO* está bajo el control de versiones! Para obtener una *copia de trabajo* bajo el control de versiones necesita Obtener la versión que acaba de importar.

5.2.3. Ficheros especiales

A veces necesitará tener un fichero bajo control de versiones que contenga datos específicos del usuario. Esto significa que tiene un fichero que cada desarrollador/usuario necesita modificar para que se ajuste a su configuración local. Pero versionar ese fichero es difícil, porque cada usuario haría confirmaciones de sus cambios cada vez en el repositorio.

En estos casos le sugerimos que utilice ficheros *plantilla*. Cree un fichero que contenga todos los datos que sus desarrolladores puedan necesitar, añádalo al control de versiones y haga que sus desarrolladores lo obtengan. Entonces, cada desarrollador tiene que *hacer una copia* de ese fichero y renombrar esa copia. Después de eso, modificar la copia no vuelve a ser un problema.

Por poner un ejemplo, puede mirar el script de compilación de TortoiseSVN. Se invoca a un fichero llamado `TortoiseVars.bat` que no existe en el repositorio. Sólo existe el fichero `Tortoise-`

`Vars.tmpl`. `TortoiseVars.tmpl` es el fichero plantilla del que cada desarrollador tiene que hacer una copia y cambiarla de nombre a `TortoiseVars.bat`. Dentro de ese fichero, hemos añadido comentarios para que los usuarios vean qué líneas tienen que editar y cambiar de acuerdo a sus configuraciones locales para que funcione.

Para no molestar a los usuarios, también hemos añadido el fichero `TortoiseVars.bat` a la lista de ignorados de su carpeta padre, es decir, hemos cambiado la propiedad de Subversion `svn:ignore` para incluir ese nombre de fichero. De esta forma no se mostrará como no versionado en cada confirmación.

5.2.4. Proyectos referenciados

A veces es útil construir una copia de trabajo que está hecho de un número de obtenciones diferentes. Por ejemplo, puede querer que diferentes carpetas vengan de orígenes distintos de un repositorio, o quizás de repositorios diferentes. Si desea que todos los usuarios tengan la misma estructura, puede definir propiedades `svn:externals`.

Digamos que obtiene una copia de trabajo de `/project1` en `D:\dev\project1`. Seleccione la carpeta `D:\dev\project1`, haga click con el botón derecho y seleccione **Menú de Windows → Propiedades** del menú contextual. Aparece el **Diálogo Propiedades**. Vaya a la pestaña de Subversion. Allí puede establecer las propiedades. Seleccione la propiedad `svn:externals` del desplegable y escriba en el cuadro de edición la URL del repositorio con el formato `nombre url` o, si desea especificar una revisión en concreto, `name -rREV url`. Puede añadir múltiples proyectos externos, uno por línea. Tenga en cuenta que las URLs deben estar correctamente codificadas o no funcionarán bien. Por ejemplo, debe reemplazar cada espacio por `%20`. Tenga en cuenta que no es posible utilizar nombres de carpetas con espacios en ellos. Suponga que ha establecido estas propiedades en `D:\dev\project1`:

```
sounds      http://sounds.red-bean.com/repos
quick_graphs http://graphics.red-bean.com/repos/fast%20graphics
skins/toolkit -r21 http://svn.red-bean.com/repos/skin-maker
```

Ahora pulse **Establecer** y confirme sus cambios. Cuando usted (o cualquier otro usuario) actualice su copia de trabajo, Subversion creará una subcarpeta `D:\dev\project1\sounds` y obtendrá el proyecto `sounds`, otra subcarpeta `D:\dev\project1\quick_graphs` que contendrá el proyecto `graphics`, y finalmente una subcarpeta anidada `D:\dev\project1\skins\toolkit` que contiene la revisión 21 del proyecto `skin-maker`.

Si los proyectos externos están en el mismo repositorio, cualquier cambio que haga allí se incluirá en la lista de confirmación cuando confirme su proyecto principal.

Si los proyectos externos están en repositorios diferentes, cualquier cambio que haga en el proyecto externo se notificará cuando confirme el proyecto principal, pero tendrá que confirmar esos cambios externos de forma separada.

Tenga en cuenta que si cambia la URL en una propiedad `svn:externals`, la siguiente vez que actualice su copia de trabajo Subversion borrará la carpeta externa antigua y obtendrá una copia de trabajo nueva, por lo que verá que los ficheros serán **Añadidos**, en vez de ser **Actualizados** como podría esperarse. Esta situación puede ocurrir si se refiere a una etiqueta de otro proyecto. Cuando se lanza una nueva versión de ese proyecto, cambia su referencia externa para apuntar a la nueva rama.



Sugerencia

Debería considerar seriamente utilizar números de revisión explícitos en todas sus definiciones de externos. Hacerlo significa que tiene que decidir cuándo tirar de una instantánea diferente de información externa, y exactamente de qué instantánea tirar. Además del aspecto de sentido común de no ser sorprendido por cambios en repositorios de terceras partes sobre los que puede que no tenga control, utilizar números de revisión explícitos también significa que cuando retroceda su copia de trabajo a una revisión anterior, sus definiciones externas también se revertirán a la forma a la que tenían en esa revisión, lo que a su vez significa que las copias de trabajo externas se actuali-

zarán para concordar en la forma en la que *ellas* tenían cuando su repositorio estaba en esa revisión anterior. Para proyectos de software, esto puede ser la diferencia entre una compilación válida o fallida de una vieja intentánea de su complejo código.

Las definiciones de URL en `svn:externals` son absolutas. No puede utilizar URLs relativas. Si relozaliza su copia de trabajo, o si se relocaliza el repositorio externo, estas URLs no se actualizarán automáticamente. Además, si crea una rama de un proyecto que tiene referencias externas dentro del mismo repositorio, las URLs en la copia de la rama no se cambiarán; quizás desee entonces cambiar las referencias del tronco con referencias de la rama.

Si necesita más información sobre cómo TortoiseSVN maneja las Propiedades, lea Sección 5.15, “Configuración del proyecto”.

Para averiguar los diferentes métodos para acceder a subproyectos comunes lea Sección B.6, “Incluir un sub-proyecto común”.

5.3. Obteniendo una copia de trabajo

Para tener una copia de trabajo necesita *obtener* una de un repositorio.

Seleccione un directorio en el explorador de Windows donde quiera poner su copia de trabajo. Haga click con el botón derecho para mostrar el menú contextual y seleccione el comando TortoiseSVN → Obtener..., que mostrará el siguiente cuadro de diálogo:

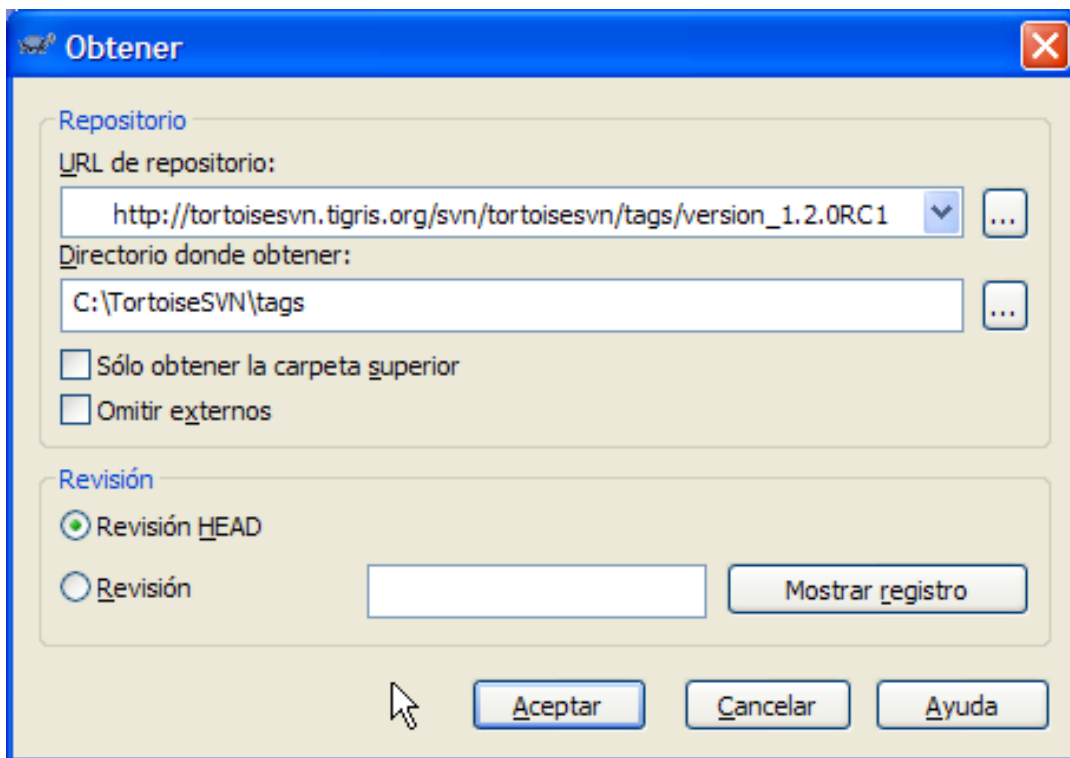


Figura 5.7. El diálogo Obtener

Si introduce un nombre de carpeta que no aún no exista, se creará un directorio con ese nombre.



Importante

Sólo debería obtener en un directorio vacío. Si quiere obtener su árbol de código en el

mismo directorio desde el que realizó la importación, Subversion le dará un mensaje de error porque no puede sobrescribir los ficheros no versionados que ya existen con los ficheros versionados. Deberá obtener en un directorio diferente o borrar antes la carpeta con el árbol de código existente.

Si desea obtener sólo la carpeta superior y omitir todas las subcarpetas, utilice la casilla **Sólo obtener la carpeta superior**.

Si el proyecto contiene referencias a proyectos externos que *no* desea que se obtengan al mismo tiempo, utilice la casilla **Omitir externos**.



Importante

Si marca cualquiera de estas opciones, tendrá que realizar actualizaciones en su copia de trabajo utilizando TortoiseSVN → Actualizar a la Revisión... en vez de TortoiseSVN → Actualizar. La actualización estándar incluirá todas las subcarpetas y todos los vínculos externos.

Le recomendamos que obtenga únicamente la parte `trunk` del árbol de directorios. Si especifica la ruta padre del árbol de directorios en la URL, al final acabará con un disco duro lleno ¡porque obtendrá una copia del árbol completo del repositorio, incluyendo cada rama y etiqueta de su proyecto!



Exportando

A veces puede querer crear una copia local sin ninguno de esos directorios `.svn`, por ejemplo para crear un fichero comprimido de sus fuentes. Lea Sección 5.23, “Exportando una copia de trabajo de Subversion” para averiguar cómo hacerlo.

5.4. Enviando sus cambios al repositorio

Enviar los cambios que ha hecho al repositorio se conoce como *confirmar* los cambios. Pero antes de confirmar tiene que estar seguro de que su copia de trabajo está actualizada. Puede o bien ejecutar TortoiseSVN → Actualizar directamente, o bien ejecutar TortoiseSVN → Comprobar Modificaciones primero, para ver qué se ha cambiado localmente o en el servidor.

Si su copia de trabajo está actualizada y no hay conflictos, ya está preparado para confirmar sus cambios. Seleccione los ficheros y/o carpetas que desee confirmar y seleccione TortoiseSVN → Confirmar....

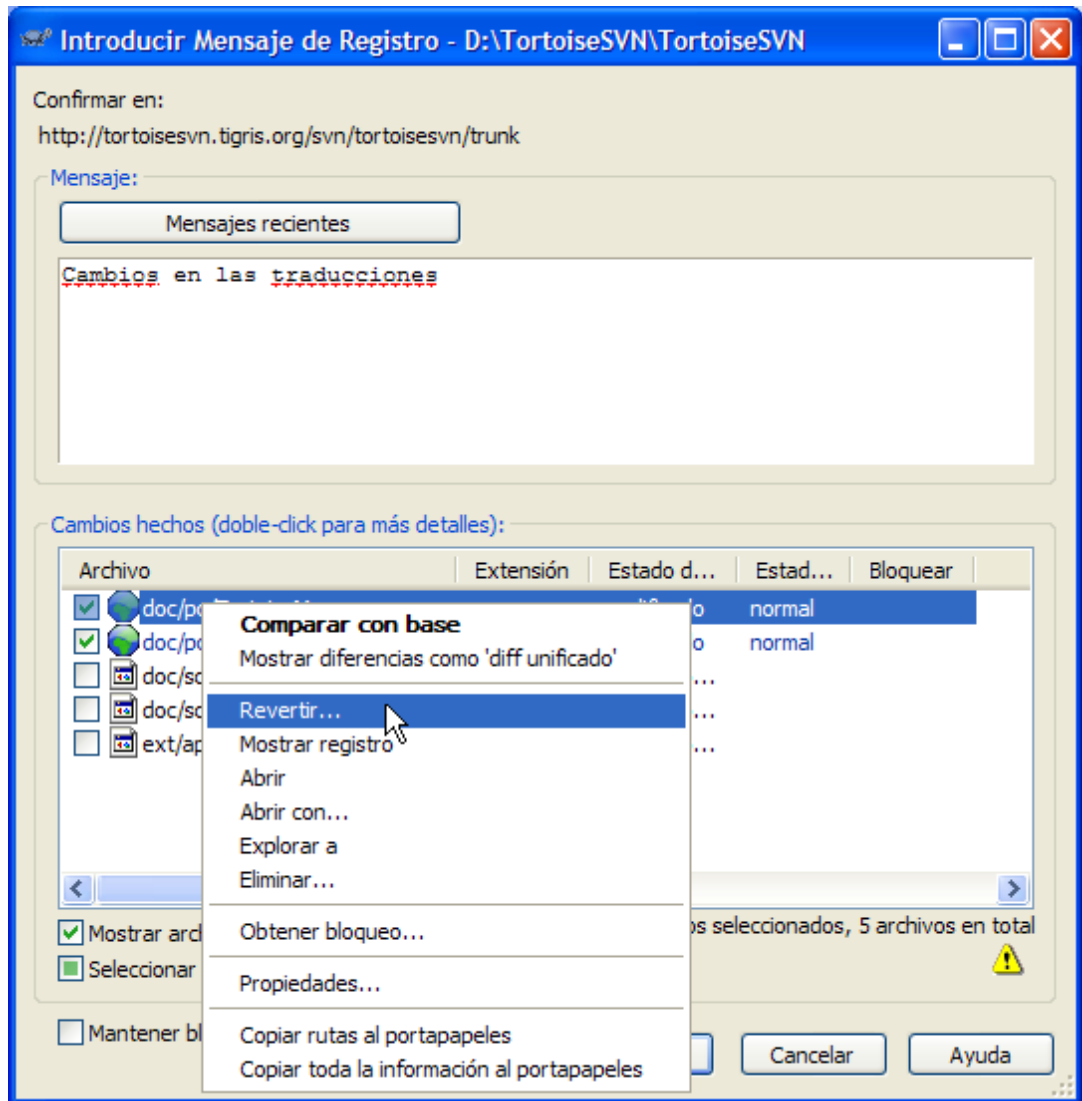


Figura 5.8. El diálogo de Confirmación

El diálogo de confirmación le mostrará todos los ficheros cambiados, incluso los ficheros añadidos, borrados o no versionados. Si no desea que un fichero cambiado se confirme, simplemente desmarque ese fichero. Si desea incluir un fichero no versionado, márquelo para añadirlo a la confirmación.

Los ítems que han sido cambiados a una ruta de repositorio diferente también se indican utilizando un marcador (s). Puede haber cambiado algo mientras trabaja en una rama y habersele olvidado volver a cambiarlo al tronco. ¡Este es su signo de advertencia!



¿Confirmar ficheros o carpetas?

Cuando confirma ficheros, el diálogo de confirmación sólo le enseña los ficheros que ha seleccionado. Cuando confirma una carpeta el diálogo de confirmación seleccionará los ficheros que han cambiado de forma automática. Si se olvidó un fichero nuevo que haya creado, al confirmar la carpeta lo encontrará. Confirmar una carpeta *no* significa que todos los ficheros se marquen como cambiados; sólo le hace la vida más fácil haciendo más trabajo por usted.

Si ha modificado ficheros que han sido incluidos desde un repositorio diferente utilizando

`svn:externals`, esos cambios no pueden ser incluidos en la misma confirmación atómica. Aparecerá un símbolo de advertencia bajo la lista de ficheros que le indicará si esto ha ocurrido, y el texto de ayuda le explicará que esos ficheros externos deben confirmarse de forma separada.



Muchos ficheros sin versionar en el diálogo de confirmar

Si cree que el diálogo de confirmación de TSVN le muestra demasiados ficheros no versionados (por ejemplo, ficheros generados por el compilador o copias de seguridad hechas por su editor), hay varias formas de manejar esta situación. Puede:

- añadir el fichero (o una extensión con máscara) a la lista de ficheros a ignorar en la página de configuración. Esto afectará a todas las copias de trabajo que tenga.
- añadir el fichero a la lista de `svn:ignore` utilizando TortoiseSVN → Añadir a la lista de ignorados. Esto únicamente afectará al directorio en el que establezca la propiedad `svn:ignore`. Puede cambiar la propiedad `svn:ignore` de un directorio utilizando el Diálogo de Propiedades SVN.

Para más información, lea la Sección 5.11, “Ignorando ficheros y directorios”.

Haciendo doble click en cualquier fichero modificado en el diálogo de confirmaciones, se lanzará la herramienta externa de diferenciar para mostrarle sus cambios. El menú contextual le proporciona más opciones, como se ve en la captura de pantalla. También puede arrastrar ficheros desde aquí a otra aplicación, como un editor de textos o un IDE.

Las columnas que se muestran en el panel inferior son personalizables. Si hace click con el botón derecho en cualquier cabecera de columna verá un menú contextual que le permite seleccionar qué columnas se muestran. También puede cambiar el ancho de la columna utilizando el manejador de arrastre que aparece cuando mueve el cursor sobre el límite de una columna. Estas personalizaciones se mantienen, por lo que verá los mismos encabezados la siguiente vez. Tenga en cuenta que por un detalle de implementación, puede ver dos diferentes iconos de arrastre, dependiendo de dónde exactamente ha puesto el ratón sobre el borde. Uno tiene una barra vertical sólida y el otro tiene dos líneas verticales finas. Sólo funcionará el puntero con la barra sólida.



Arrastrar y soltar

Puede arrastrar ficheros hasta el diálogo de confirmación desde cualquier otro lugar, siempre y cuando las copias de trabajo sean del mismo repositorio. Por ejemplo, puede tener una copia de trabajo enorme con diferentes ventanas del explorador abiertas en carpetas distantes de la jerarquía. Si quiere evitar confirmar desde la carpeta más alta (lo que implica una lenta operación de búsqueda de cambios) puede abrir el diálogo de confirmar para una carpeta y arrastrar desde las otras ventanas para incluir ficheros dentro de la misma confirmación atómica.

Asegúrese de introducir un mensaje de registro que describa los cambios que está confirmando. Esto le ayudará a saber qué ocurrió y cuando según navegue por los mensajes de registro del proyecto en el futuro. El mensaje puede ser tan extenso o escueto como desee; muchos proyectos tienen directrices sobre qué debe incluirse en ellos, el idioma que debe utilizarse, y a veces incluso un formato estricto.

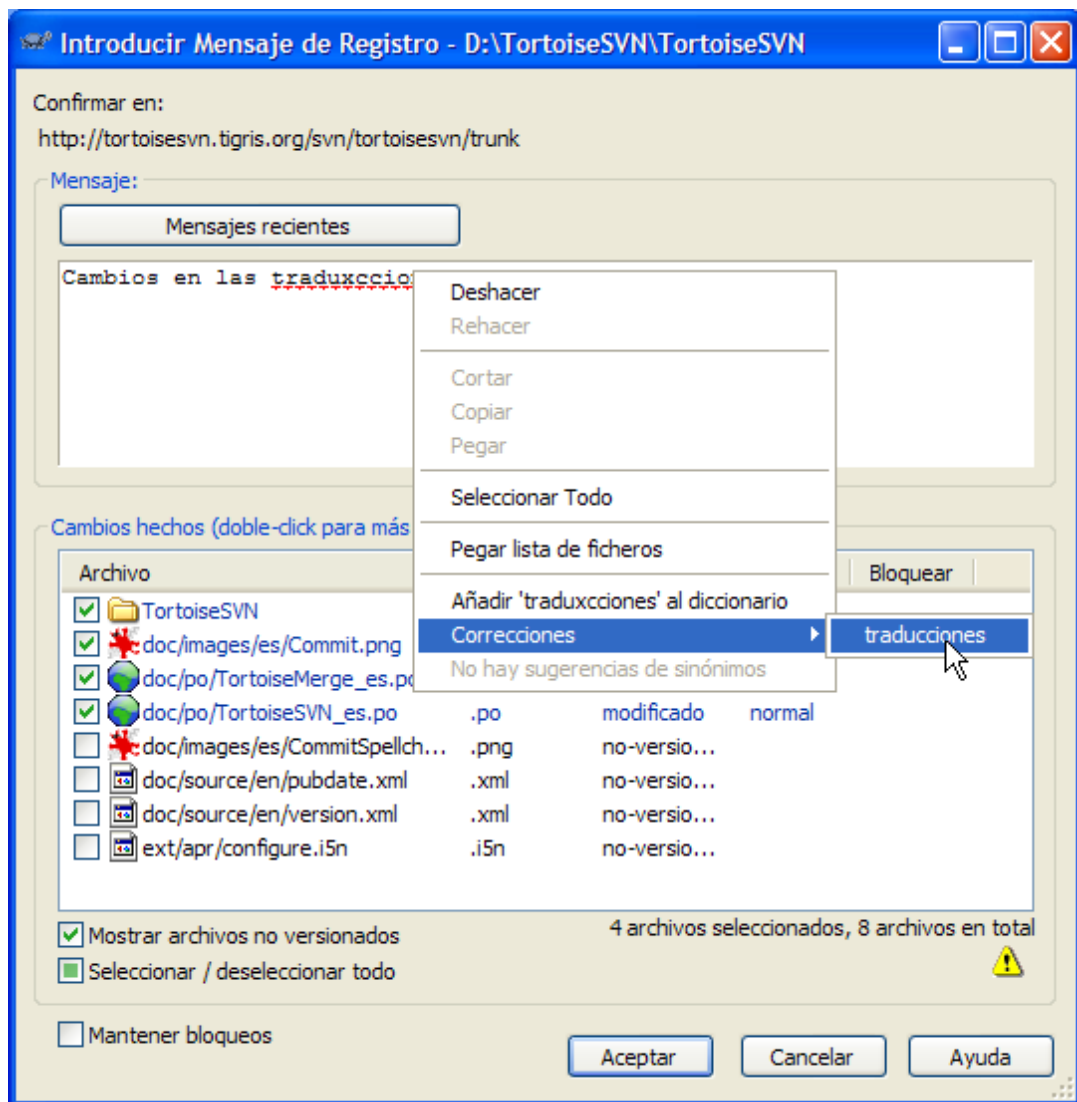


Figura 5.9. El corrector ortográfico del diálogo de Confirmación

TortoiseSVN incluye un corrector ortográfico para ayudarle a escribir sus mensajes de registro correctamente. Este corrector señalará cualquier palabra mal escrita. Utilice el menú contextual para acceder a las correcciones sugeridas. Por supuesto, el corrector no conoce *todos* los términos técnicos que utiliza, así que a veces palabras bien escritas aparecerán como errores. Pero no se preocupe. Puede simplemente añadirlas a su diccionario personal utilizando el menú contextual.

La ventana de mensajes de registro también incluye una facilidad de autocompletar nombres de ficheros y funciones. Esto utiliza expresiones regulares para extraer clases y nombres de funciones de los ficheros (de texto) que está confirmando, y también los propios nombres de ficheros. Si una palabra que está tecleando concuerda con algo en la lista (después de haber tecleado al menos 3 caracteres), aparecerá un desplegable que le permitirá seleccionar el nombre completo. Las expresiones regulares suministradas con TortoiseSVN se mantienen en la carpeta bin de la instalación de TortoiseSVN. También puede definir sus propias expresiones regulares y almacenarlas en %APPDATA%\TortoiseSVN\autolist.txt. Por supuesto su lista privada no se sobrescribirá cuando actualice su instalación de TortoiseSVN. Si no está familiarizado con las expresiones regulares, eche un vistazo a la documentación y al tutorial en línea en <http://www.regular-expressions.info/>.

Tras pulsar **Aceptar** aparece un diálogo mostrando el progreso de la confirmación.

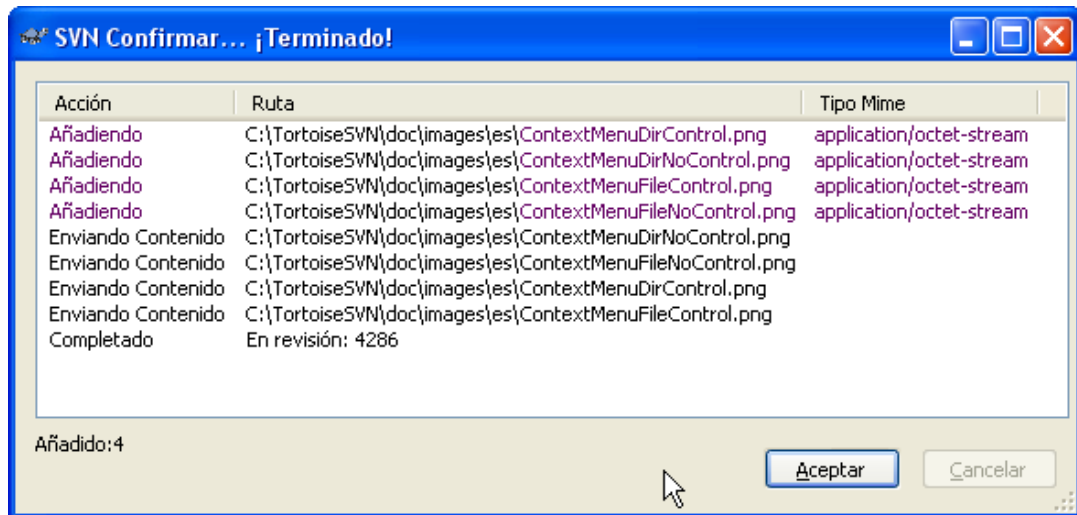


Figura 5.10. El diálogo Progreso mostrando el progreso de una confirmación

El diálogo de progreso utiliza una codificación de colores para resaltar las diferentes acciones de confirmación:

- Azul
Confirmando una modificación.
- Púrpura
Confirmando un ítem añadido.
- Rojo oscuro
Confirmando un borrado o un reemplazo.
- Negro
Todos los demás ítems.

Este es el esquema de colores por defecto, pero puede personalizar dichos colores utilizando el diálogo de configuración. Para más información, lea Sección 5.26.2.5, “Configuración de colores de TortoiseSVN”.



Propiedades especiales de carpetas

Hay diversas propiedades especiales de carpeta que pueden usarse para darle mayor control sobre el formato de los mensajes de registro de las confirmaciones y el idioma que utiliza el módulo del corrector ortográfico. Para más información, lea Sección 5.15, “Configuración del proyecto”.



Integración con herramientas de control de errores

Si ha activado el sistema de control de errores, puede poner una o más Incidencias en el cuadro de texto Bug-ID / Issue-Nr:. Debería separar las múltiples incidencias por comas. Alternativamente, si está utilizando el soporte de control de errores basado en expresiones regulares, simplemente añada las referencias de las incidencias como parte del mensaje de registro. Si desea saber más, vea Sección 5.25, “Integración con sistemas de control de errores / seguimiento de incidencias”.

5.5. Actualice su copia de trabajo con los cambios de otros

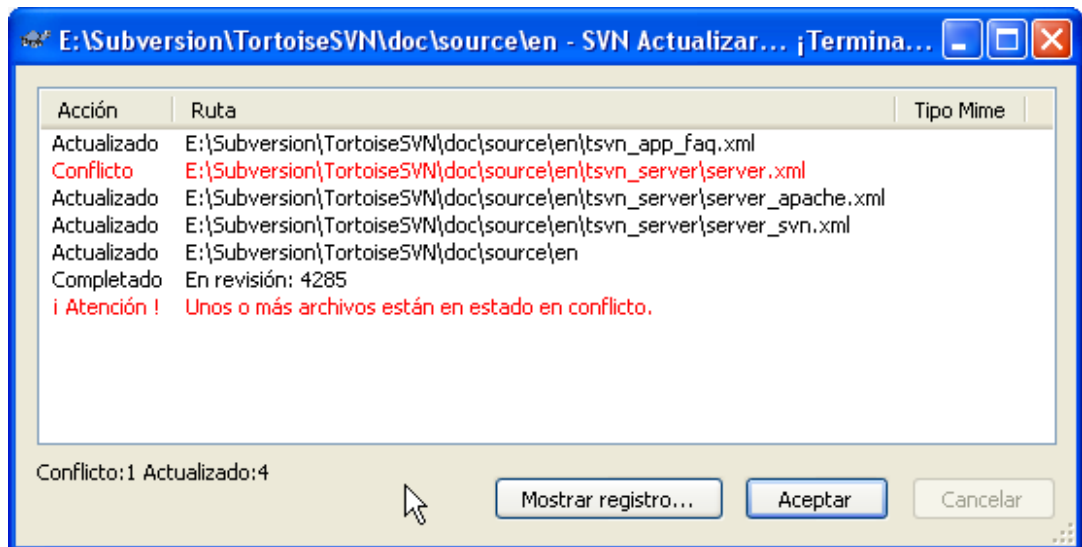


Figura 5.11. Diálogo de progreso mostrando una actualización terminada

Periódicamente, debería asegurarse que los cambios que hacen los demás se incorporen en su copia de trabajo local. El proceso de incorporar los cambios desde el servidor a su copia de trabajo local se conoce como *actualización*. La actualización puede hacerse en ficheros sueltos, en un conjunto de ficheros, o recursivamente en jerarquías completas de directorios. Para actualizar, seleccione los ficheros y/o directorios que desee, haga click con el botón derecho y seleccione TortoiseSVN → Actualizar en el menú contextual del explorador. Aparecerá una ventana con el progreso de la actualización según se ejecuta. Los cambios que los demás hayan hecho se fusionarán con sus ficheros, guardando cualquier cambio que usted haya hecho en los mismos ficheros. El repositorio *no* se ve afectado por una actualización.

El diálogo de progreso utiliza un código de colores para resaltar diferentes acciones de actualización:

Púrpura

Nuevo ítem añadido a su copia de trabajo

Rojo oscuro

Ítem redundante borrado de su copia de trabajo, o ítem faltante reemplazado en su copia de trabajo.

Verde

Cambios del repositorio que se han fusionado satisfactoriamente con sus cambios locales.

Rojo brillante

Cambios del repositorio fusionados con sus cambios locales, pero que han dado lugar a conflictos que debe resolver.

Negro

Items sin cambios en su copia de trabajo actualizados con una versión más nueva desde el repositorio.

Este es el esquema de colores por defecto, pero puede personalizar dichos colores utilizando el diálogo de configuración. Para más información, lea Sección 5.26.2.5, “Configuración de colores de TortoiseSVN”.

Si obtiene algún *conflicto* durante una actualización (esto puede suceder si los demás han cam-

biado las mismas líneas del mismo fichero a la vez que usted y esos cambios no concuerdan), el diálogo muestra esos conflictos en rojo. Puede hacer doble click en esas líneas para iniciar la herramienta de fusión externa para resolver los conflictos.

El comando Actualizar estándar no tiene opciones y simplemente actualiza su copia de trabajo a la revisión HEAD del repositorio, lo que es el caso de uso más común. Si desea más control sobre el proceso de actualización, debería utilizar TortoiseSVN → Actualizar a la revisión.... Esto le permite actualizar su copia de trabajo a una revisión específica, no sólo a la más reciente. Suponga que su copia de trabajo está en la revisión 100, pero quiere que refleje el estado que tenía en la revisión 50 - entonces simplemente actualice a la revisión 50. En ese mismo diálogo también puede decidir si desea actualizar sólo la carpeta actual (sin sus subcarpetas) y si desea ignorar todos los proyectos externos en la actualización (es decir, los proyectos referenciados utilizando `svn:externals`).



Atención

Si actualiza un fichero o una carpeta a una revisión en concreto, no debería hacer cambios en esos ficheros. ¡Obtendrá mensajes `out of date` (desactualizado) cuando intente confirmarlos! Si desea deshacer los cambios de un fichero y empezar de nuevo desde una revisión anterior, puede revertir a una revisión previa desde el diálogo de registro de revisiones. Lea Sección B.4, “Deshacer revisiones en el repositorio” si desea más instrucciones, y métodos alternativos.

Actualizar a la revisión puede ser útil a veces para ver cómo estaba su proyecto en un momento anterior en su historia. Pero en general, actualizar ficheros individuales a una revisión anterior no es una buena idea, ya que deja su copia de trabajo en un estado inconsistente. Si el fichero que está actualizando ha cambiado de nombre, incluso puede encontrar que ese fichero ha desaparecido de su copia de trabajo porque en esa revisión no había ningún fichero con ese nombre. Si simplemente desea una copia local de una versión antigua de un fichero, es mejor utilizar para ese fichero el comando Menú Contextual → Grabar la revisión como... desde el diálogo de registro.



Múltiples ficheros/carpetas

Si selecciona múltiples ficheros y carpetas en el explorador y luego selecciona Actualizar, todos esos ficheros/carpetas se actualizan uno a uno. ¡TortoiseSVN se asegura de que todos los ficheros/carpetas del mismo repositorio se actualicen exactamente a la misma revisión! Incluso si entre esas actualizaciones ocurrió alguna confirmación.



El fichero local ya existe

A veces cuando intente actualizar, la actualización falla con un mensaje para decir que ya existe un fichero local con el mismo nombre. Esto típicamente ocurre cuando Subversion intenta obtener un fichero recién versionado, y se encuentra un fichero no versionado del mismo nombre en su copia de trabajo. Subversion nunca sobrescribirá un fichero no versionado - puede contener algo en lo que está trabajando, y que casualmente tiene el mismo nombre de fichero que otro desarrollador ha utilizado para su recién confirmado fichero.

Si obtiene este mensaje de error, la solución es simplemente renombrar el fichero local sin versionar. Tras completar la actualización, puede comprobar si el fichero renombrado sigue siendo necesario.

Si sigue obteniendo mensajes de error, utilice mejor el comando TortoiseSVN → Comprobar modificaciones para mostrar todos los ficheros con problemas. De esa forma puede lidiar con ellos de un golpe.

5.6. Resolviendo conflictos

De vez en cuando, obtendrá un *conflicto* cuando actualiza sus ficheros desde el repositorio. Un conflicto ocurre cuando dos o más desarrolladores han hecho cambios en las mismas líneas de un fichero. Dado que Subversion no sabe nada de su proyecto, delega la resolución de los conflictos en los desarrolladores. Cuando se le informa de un conflicto, debería abrir el fichero en cuestión, y buscar líneas que empiecen con el texto <<<<<<. El área conflictiva se marca así:

```
<<<<<< nombre-del-fichero
      sus cambios
=====
      código fusionado del repositorio
>>>>>> revisión
```

Además, para cada fichero en conflicto Subversion deja tres ficheros adicionales en su directorio:

nombre-del-fichero.ext.mine

Este es su fichero tal y como estaba en su copia de trabajo antes de que actualizara su copia de trabajo - esto es, sin marcadores de conflicto. Este fichero tiene sus últimos cambios en él y nada más.

nombre-del-fichero.ext.rREV-ANTIGUA

Este es el fichero que era la revisión BASE antes de que actualizara su copia de trabajo. Esto es, el fichero que obtuvo antes de empezar a hacer sus últimos cambios.

nombre-del-fichero.ext.rREV-NUEVA

Este es el fichero que su cliente de Subversion acaba de recibir desde el servidor del que actualizó su copia de trabajo. Este fichero corresponde a la revisión HEAD del repositorio.

Puede o bien lanzar una herramienta externa de fusiones / editor de conflictos con el menú contextual TortoiseSVN → Editar Conflictos o bien utilizar otro editor manualmente para resolver el conflicto. Debe decidir cómo tiene que quedar el código, hacer los cambios necesarios, y grabar el fichero.

Después, ejecute el comando TortoiseSVN → Resolver y confirme sus modificaciones al repositorio. Tome nota de que el comando Resolver realmente no resuelve el conflicto. Simplemente elimina los ficheros `filename.ext.mine` y `filename.ext.r*`, dejándole confirmar sus cambios.

Si tiene conflictos con ficheros binarios, Subversion no intentará mezclar dichos ficheros por sí mismo. El fichero local se mantendrá sin cambios (exactamente tal y como lo había cambiado usted) y obtendrá unos ficheros `nombrefichero.ext.r*`. Si desea descartar sus cambios y quedarse con la versión del repositorio, utilice el comando Revertir. Si desea mantener su versión y sobrescribir la versión del repositorio, utilice el comando Resuelto y luego confirme su versión.

Puede utilizar el comando Resuelto para múltiples ficheros si pulsa con el botón derecho en la carpeta padre y selecciona TortoiseSVN → Resuelto... Esto mostrará un diálogo con todos los ficheros en conflicto dentro de esa carpeta, y le permitirá seleccionar cuáles marcar como resueltos.

5.7. Obteniendo información del estado

Mientras está trabajando en su copia de trabajo a menudo necesitará saber qué ficheros ha cambiado/añadido/borrado o renombrado, o incluso qué ficheros han sido cambiados y confirmados por los demás.

5.7.1. Iconos sobreimpresionados

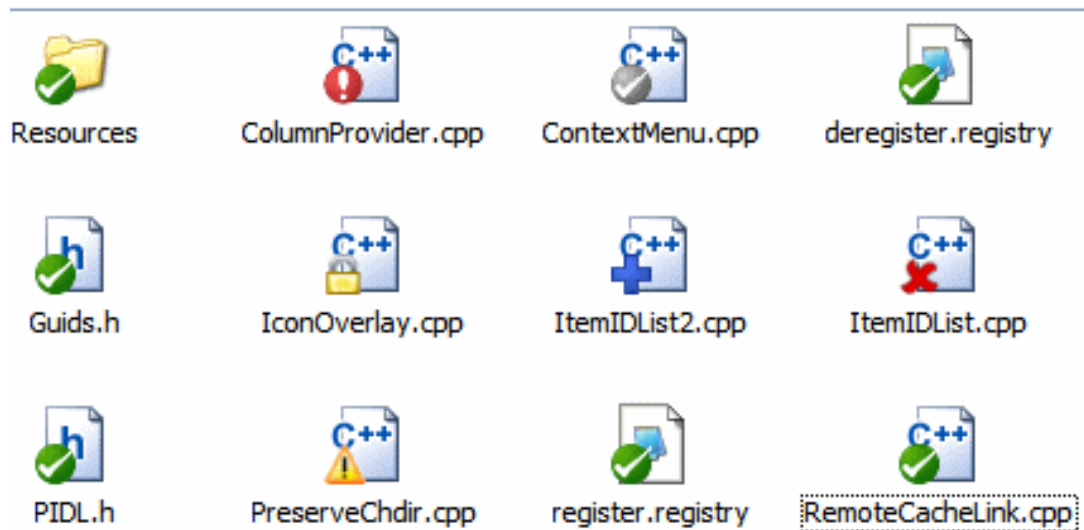


Figura 5.12. Explorador mostrando iconos sobreimpresionados

Ahora que ha obtenido una copia de trabajo desde un repositorio de Subversion, puede ver sus ficheros en el explorador de Windows con los iconos cambiados. Ésta es una de las razones por las que TortoiseSVN es tan popular. TortoiseSVN añade lo que se llama un icono sobreimpresionado al icono de cada fichero que se superpone al icono original del fichero. Dependiendo del estado en Subversion del fichero, el icono sobreimpresionado es diferente.



Una copia de trabajo recién obtenida tiene una marca verde como sobreimpresión. Esto significa que el estado de Subversion es `normal`.



En cuanto empiece a editar un fichero, el estado cambia a `modificado` y el icono sobreimpresionado cambia entonces a una marca de exclamación roja. De esta forma puede ver fácilmente qué ficheros se han cambiado desde la última vez que actualizó su copia de trabajo, y que necesitan ser confirmados.



Si durante una actualización ocurre un `conflicto`, el icono cambia a un signo de exclamación amarillo.



Si ha establecido la propiedad `svn:needs-lock` en un fichero, Subversion pone a ese fichero como de sólo-lectura hasta que obtenga un bloqueo en dicho fichero. Los ficheros de sólo-lectura tienen esta sobreimpresión para indicarle que debe obtener un bloqueo antes de que pueda editar ese fichero.



Si ha obtenido un bloqueo sobre un fichero, y el estado de Subversion es `normal`, este icono sobreimpresionado le recordará que debería liberar el bloqueo si no lo está utilizando para permitir a

los demás que puedan confirmar sus cambios al fichero.



Este icono le muestra que algunos ficheros o carpetas dentro de la carpeta actual se han marcado para ser borrados del control de versiones, o bien que falta un fichero bajo control de versiones dentro de una carpeta.



El signo más le indica que el fichero o carpeta está programado para ser añadido al control de versiones.

Al contrario que TortoiseCVS (la integración shell de CVS) no se muestran iconos sobreimpresionados para los ficheros no versionados. Hacemos ésto porque el número de iconos sobreimpresionados está limitado por el sistema y debe economizarse su utilización.

De hecho, puede ser que no se utilicen todos estos iconos en su sistema. Esto se debe a que el número de sobreimpresiones permitidas por Windows está limitado a 15. Windows utiliza 4 de ellas, y las 11 restantes pueden ser utilizados por otras aplicaciones. Si también está utilizando TortoiseCVS, entonces no hay suficientes huecos de sobreimpresión disponibles, por lo que TortoiseSVN intenta ser un "Buen Ciudadano (TM)" y limita su uso de sobreimpresiones para darles una oportunidad al resto de aplicaciones.

- Normal, Modificado y En conflicto siempre se cargan y están visibles.
- Borrado se carga si es posible, pero se cambia por Modificado si no hay suficientes huecos.
- Sólo-lectura se carga si es posible, pero se cambia por Normal si no hay suficientes huecos.
- Bloqueado se carga sólo si hay menos de 13 sobreimpresiones ya cargadas. Se cambia por Normal si no hay suficientes huecos.
- Añadido se carga sólo si hay menos de 14 sobreimpresiones ya cargadas. Se cambia por Modificado si no hay suficientes huecos.

5.7.2. Columnas de TortoiseSVN en el Explorador de Windows

Se puede ver la misma información que está disponible en los iconos sobreimpresionados (y mucha más) como columnas adicionales en la Vista Detalles del Explorador de Windows.

Simplemente haga click con el botón derecho en las cabeceras de una columna y seleccione Más... del menú contextual que aparece. Aparecerá un diálogo donde puede especificar las columnas y su orden, que se mostrarán en la "Vista Detalles". Baje hasta que vea las entradas que empiezan por SVN. Marque aquellas que desee mostrar y cierre el diálogo pulsando **Aceptar**. Las columnas aparecerán a la derecha de las que ya se mostraban. Puede reorganizarlas utilizando arrastrar y soltar, o cambiarlas de tamaño, para que se ajusten a sus necesidades.



Sugerencia

Si desea que la organización actual se muestre en todas sus copias de trabajo, puede que desee convertirla en su vista por defecto.

5.7.3. Estado local y remoto

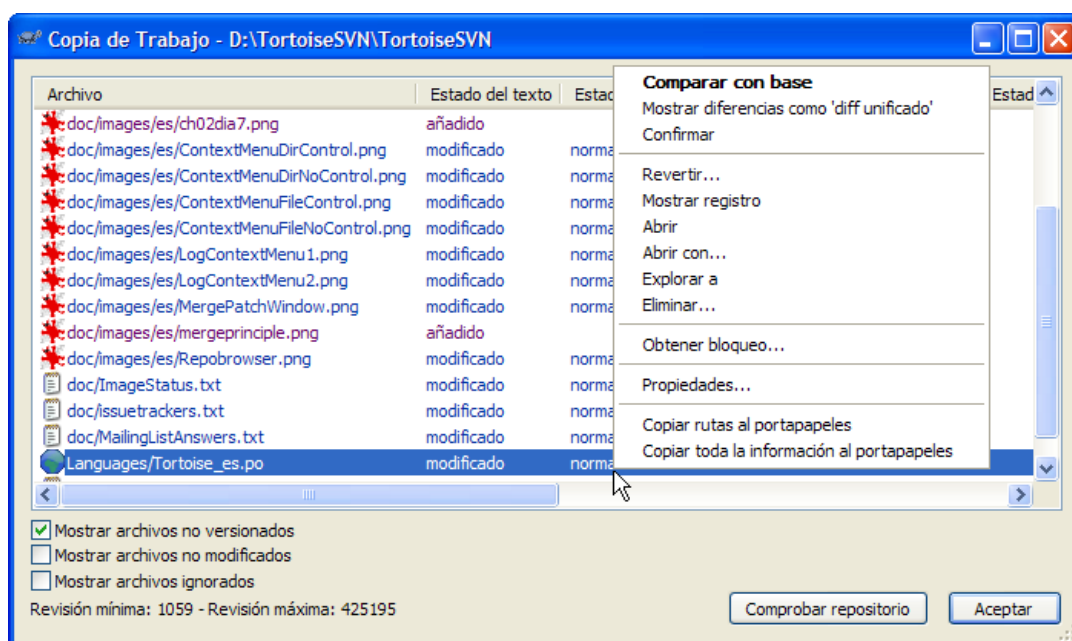


Figura 5.13. Comprobar modificaciones

A menudo es muy útil saber qué ficheros ha cambiado y también qué ficheros han cambiado y confirmado los demás. Ahí es donde viene bien el comando TortoiseSVN → Comprobar Modificaciones.... Este diálogo le muestra todos los ficheros que ha cambiado de alguna forma en su copia de trabajo, y además todos los ficheros no versionados que pueda tener.

Si pulsa en el botón Comprobar Repositorio también puede comprobar los cambios en el repositorio. De esa forma puede comprobar antes de hacer una actualización si es posible que haya un conflicto. También puede actualizar los ficheros seleccionados desde el repositorio sin actualizar la carpeta completa.

El diálogo utiliza un código de colores para resaltar el estado.

Azul

Ítems modificados localmente.

Púrpura

Ítems añadidos. Los ítems que han sido añadidos con historia tienen un signo + en la columna Estado del texto, y una etiqueta de ayuda que le muestra de dónde ha sido copiado.

Rojo oscuro

Ítems faltantes o borrados.

Verde

Ítems modificados localmente y en el repositorio. Los cambios se fusionarán al actualizar. Ésto puede producir conflictos al actualizar.

Rojo brillante

Ítems modificados localmente y borrados en el repositorio, o modificados en el repositorio y borrados localmente. Esto producirá conflictos al actualizar.

Negro

Ítems sin cambios y sin versionar.

Este es el esquema de colores por defecto, pero puede personalizar dichos colores utilizando el diá-

logo de configuración. Para más información, lea Sección 5.26.2.5, “Configuración de colores de TortoiseSVN”.

Los ítems que han sido cambiados a una ruta de repositorio diferente también se indican utilizando un marcador (⌘). Puede haber cambiado algo mientras trabaja en una rama y habérsele olvidado volver a cambiarlo al tronco. ¡Este es su signo de advertencia!

Desde el menú contextual del diálogo puede mostrar un resumen de los cambios. Compruebe los cambios locales que *usted* ha hecho utilizando **Menú Contextual → Comparar con Base**. Compruebe los cambios en el repositorio hechos por los demás utilizando **Menú Contextual → Mostrar Diferencias como Diff Unificado**.

También puede revertir cambios en ficheros individuales. Si ha borrado un fichero de forma accidental, se mostrará como **Falta** y puede utilizar **Revertir** para recuperarlo.

Si desea examinar un fichero más en detalle, puede arrastrarlo desde aquí a otra aplicación, tal como un editor de textos o un IDE.

Las columnas que se muestran en el panel inferior son personalizables. Si hace click con el botón derecho en cualquier cabecera de columna verá un menú contextual que le permite seleccionar qué columnas se muestran. También puede cambiar el ancho de la columna utilizando el manejador de arrastre que aparece cuando mueve el cursor sobre el límite de una columna. Estas personalizaciones se mantienen, por lo que verá los mismos encabezados la siguiente vez. Tenga en cuenta que por un detalle de implementación, puede ver dos diferentes iconos de arrastre, dependiendo de dónde exactamente ha puesto el ratón sobre el borde. Uno tiene una barra vertical sólida y el otro tiene dos líneas verticales finas. Sólo funcionará el puntero con la barra sólida.



Sugerencia

Si quiere una vista llana de su copia de trabajo, por ejemplo mostrando todos los ficheros y carpetas en todos los niveles de su jerarquía de carpetas, entonces el diálogo **Comprobar modificaciones** es la forma más sencilla de conseguirlo. Simplemente seleccione la casilla **Mostrar archivos no modificados** para ver todos los ficheros de su copia de trabajo.

5.7.4. Viendo diferencias

A menudo querrá mirar dentro de sus ficheros, para echar un vistazo a lo que ha cambiado. Puede llevar esto a cabo seleccionando un fichero que haya cambiado, y seleccionando **Diferenciar** desde el menú contextual de TortoiseSVN. Esto inicia el visor externo de diferencias, que comparará el fichero actual con la copia prístina (revisión BASE), que se guardó tras su obtención o tras la última actualización.



Sugerencia

Puede mostrar diferencias incluso cuando no está dentro de una copia de trabajo, o cuando tiene múltiples versiones del fichero alrededor:

Seleccione los dos ficheros que desea comparar en el explorador (por ejemplo, utilizando la tecla **Ctrl** y el ratón) y seleccione **Diferenciar** del menú contextual de TortoiseSVN. El fichero que haya pulsado en último lugar (el que tiene el foco, es decir, el rectángulo con puntos) será tomado como más nuevo.

5.8. Diálogo de Registro de revisiones

Para cada cambio que haga y confirme, debería proporcionar un mensaje de registro de ese cambio. Así podrá averiguar después qué cambios hizo y por qué, y tendrá un registro detallado para su proceso de desarrollo.

El diálogo de Registro de revisiones recopila todos esos mensajes de registro y se los enseña. La pantalla se divide en tres paneles.

- El panel superior le muestra una lista de revisiones donde se confirmaron cambios a los ficheros/ carpetas. Este resumen incluye la fecha y la hora, la persona que confirmó la revisión y el inicio del mensaje de registro.

Las líneas azules indican que algo se ha copiado a esta línea de desarrollo (quizás desde una rama).

- El panel medio le muestra el mensaje de registro completo para la revisión seleccionada.
- El panel inferior le muestra una lista de todos los ficheros y carpetas que se cambiaron como parte de la revisión seleccionada.

Pero hace mucho más que eso - le proporciona comandos del menú contextual que puede utilizar para obtener aún más información de la historia del proyecto.

5.8.1. Invocando el diálogo de Registro de revisiones

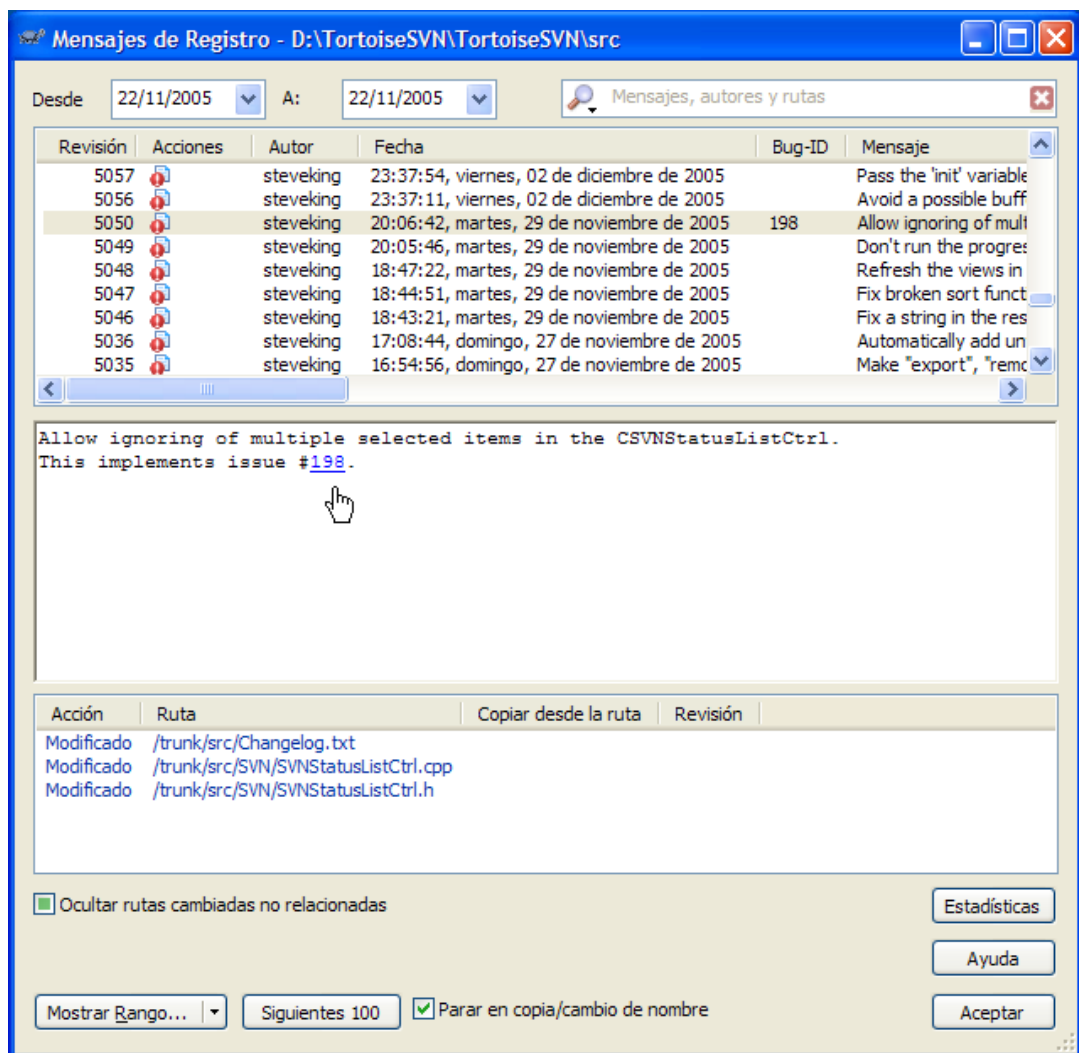


Figura 5.14. El diálogo de Registro de revisiones

Hay varios lugares desde los que puede mostrar el diálogo de Registro:

- Desde el submenú contextual de TortoiseSVN
- Desde la página de propiedades
- Desde el diálogo de Progreso después de que termine una actualización. En ese caso el diálogo de Registro sólo le mostrará aquellas revisiones que cambiaron desde su última actualización

5.8.2. Obteniendo información adicional

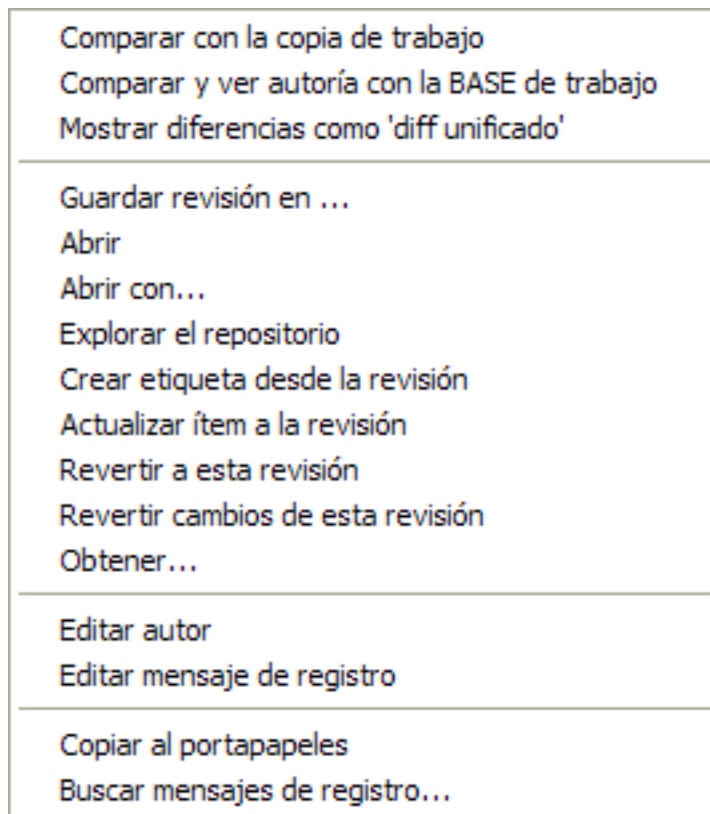


Figura 5.15. El panel superior del diálogo de Registro de revisiones con el menú contextual

El panel superior del diálogo de Registro tiene un menú contextual que le permite

- Comparar las revisiones seleccionadas con su copia de trabajo. La Herramienta de Diferencias por defecto es TortoiseMerge que se proporciona con TortoiseSVN. Si el diálogo de registro es de una carpeta, esto mostrará una lista de sus ficheros cambiados, y le permitirá revisar los cambios hechos a cada fichero individualmente.
- Obtener la información de autoría de la revisión seleccionada, y del fichero en su BASE de trabajo, y comparar los resultados utilizando una herramienta visual de diferencias. Lea Sección 5.20.2, “Autoría de las diferencias” para más detalles.
- Ver los cambios hechos en la revisión seleccionada como fichero de diff unificado (formato de parche GNU). Esto le muestra sólo las diferencias con unas pocas líneas de contexto. Es más difícil de leer que una comparación visual de ficheros, pero le mostrará todos los cambios juntos en un formato compacto.
- Grabar la revisión seleccionada a un fichero para que pueda tener una versión más antigua de ese fichero. Esta opción sólo está disponible cuando accede al registro de un fichero, y almacena

únicamente una versión de ese fichero en concreto.

- Abrir el navegador de repositorios para examinar la carpeta seleccionada. Esta opción sólo está disponible cuando accede al registro de un directorio.
- Crear una rama/etiqueta desde una revisión seleccionada. Esto es útil por ejemplo si se le olvidó crear una etiqueta y ya ha confirmado algunos cambios que no se supone que tengan que ir en esa versión.
- Actualizar su copia de trabajo a la revisión seleccionada. Útil si quiere hacer que su copia de trabajo refleje un momento en el pasado. Es mejor actualizar un directorio completo en su copia de trabajo, no sólo un fichero, ya que si no su copia de trabajo será inconsistente y no podrá confirmar ningún cambio.
- Revertir los cambios que se hicieron en la revisión seleccionada. Estos cambios se revierten en su copia de trabajo, ¡por lo que esta operación *no* afecta al repositorio en absoluto! Tenga en cuenta que esto deshacerá únicamente los cambios hechos en esa revisión. No reemplaza su copia de trabajo con el fichero entero en la revisión anterior. Es muy útil para deshacer un cambio anterior cuando se han hecho además otros cambios que no tienen que ver.
- Revertir a una revisión anterior. Si ha hecho varios cambios, y luego decide que realmente desea volver a dejar las cosas como estaban en la revisión N, este es el comando que necesita. De nuevo, los cambios se revierten en su copia de trabajo, por lo que esta operación *no* afecta al repositorio a menos que confirme los cambios. Tenga en cuenta que esto deshacerá *todos* los cambios que realizó tras la revisión seleccionada, reemplazando el fichero o carpeta con la versión anterior.
- Editar el mensaje de registro o el autor adjunto a una confirmación anterior. Lea Sección 5.8.4, “Cambiar el mensaje de registro y el autor” para averiguar cómo funciona esto.
- Buscar en los mensajes de registro el texto que desee. Esto busca en los mensajes de registro que ha introducido, y también en los sumarios de acción creados por Subversion (mostrados en el panel inferior). La búsqueda no distingue mayúsculas y minúsculas.

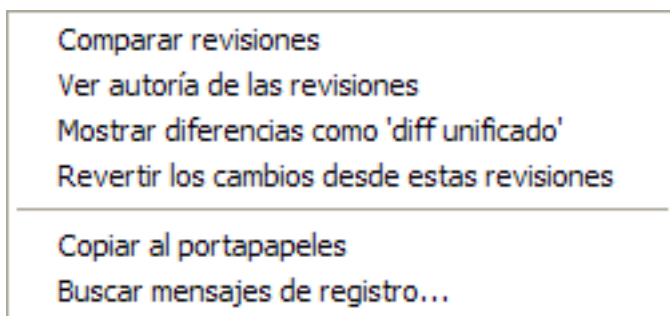


Figura 5.16. Menú contextual del panel superior para 2 revisiones seleccionadas

Si selecciona dos revisiones a la vez (utilizando el modificador habitual **Ctrl**), el menú contextual cambia y le ofrece menos opciones:

- Compara las dos revisiones seleccionadas utilizando una herramienta de diferencias visual. La herramienta de diferencias por defecto es TortoiseMerge que se proporciona con TortoiseSVN.

Si selecciona esta opción para una carpeta, aparecerá un diálogo posterior mostrando los ficheros cambiados y ofreciéndole más opciones de diferenciación. Lea más sobre el diálogo Comparar Revisiones en Sección 5.9.2, “Comparando carpetas”.

- Obtener la información de autoría de las dos revisiones y comparar los resultados utilizando una herramienta visual de diferencias. Lea Sección 5.20.2, “Autoría de las diferencias” para más detalles.
- Ver las diferencias entre las dos revisiones seleccionadas como un fichero diff unificado. Esto funciona para ficheros y carpetas.
- Editar el mensaje de registro o el autor como se describe más arriba.
- Buscar mensajes de registro como se describe más arriba.

Si selecciona múltiples revisiones consecutivas (utilizando los modificadores habituales **Ctrl** o **Mayúsculas**), el menú contextual incluirá una entrada para Revertir todos los cambios que se hicieron en ese rango de revisiones. Ésta es la forma más sencilla para deshacer un grupo de revisiones de golpe.

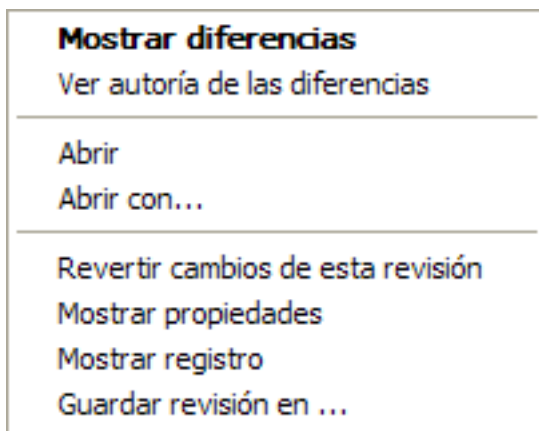


Figura 5.17. El panel inferior del diálogo de Registro con el menú contextual

El panel inferior del diálogo Registro también tiene un menú contextual que le permite

- Mostrar las diferencias hechas en la revisión seleccionada sobre el fichero seleccionado. Este menú contextual sólo está disponible para los ficheros que se muestran como **Modificados**.
- Obtener la información de autoría de la revisión seleccionada y de la revisión anterior del fichero seleccionado, y comparar los resultados utilizando una herramienta visual de diferencias. Lea Sección 5.20.2, “Autoría de las diferencias” para más detalles.
- Abrir el fichero seleccionado, bien con el visor por defecto para ese tipo de fichero, o bien con el programa que elija.
- Revertir los cambios hechos al fichero seleccionado en esa revisión.
- Ver las propiedades de Subversion del ítem seleccionado.
- Mostrar el registro de revisiones para ese fichero seleccionado.
- Grabar la revisión seleccionada a un fichero, para que pueda tener una versión antigua de ese fichero.

5.8.3. Obteniendo más mensajes de registro

El diálogo Registro no siempre le muestra todos los cambios que se hayan hecho alguna vez por unos cuantos motivos:

- En un repositorio grande, puede haber cientos o incluso miles de cambios y obtenerlos todos puede llevar mucho tiempo. Normalmente estará interesado sólo en los cambios más recientes. Por defecto, el número de mensajes de registro obtenidos se limita a 100, pero puede cambiar este valor en TortoiseSVN → Propiedades (Sección 5.26, “Configuración de TortoiseSVN”),
- Cuando se marca la casilla **Parar en copia/renombrado**, **Mostrar Registro** se parará en el punto en el que el fichero o carpeta seleccionado se copió de algún otro lugar en el repositorio. Esto puede ser útil para buscar ramas (o etiquetas) porque se para en la raíz de esa rama, y le da una indicación rápida de los cambios hechos únicamente en esa rama.

Normalmente querrá dejar esta opción sin marcar. TortoiseSVN recuerda el estado de la casilla, por lo que respetará su preferencia.

Cuando se invoca el diálogo **Mostrar Registro** desde el diálogo **Fusionar**, la casilla siempre se marca por defecto. Esto es porque al fusionar lo más probable es buscar cambios en las ramas, y retroceder a la raíz de la rama no tiene sentido en ese caso.

Tenga en cuenta que Subversion actualmente implementa el renombrado como un par de copia/borrado, por lo que renombrar un fichero o carpeta también provocará que el diálogo de registro se pare si se marca esta opción.

Si desea ver más mensajes de registro, pulse **Siguientes 100** para obtener los siguientes 100 mensajes de registro. Puede repetir esto tantas veces como sea necesario.

Al lado de este botón hay un botón multifunción que recuerda la última opción que utilizó. Pulse en la flecha para ver las otras opciones ofrecidas.

Utilice **Mostrar rango ...** si desea ver un rango específico de revisiones. Aparecerá un diálogo que le preguntará por la revisión de inicio y de fin.

Utilice **Mostrar Todos** si desea ver *todos* los mensajes de registro desde HEAD hasta la revisión 1.

5.8.4. Cambiando el mensaje de registro y el autor

A veces querrá cambiar un mensaje de registro que introdujo en su día, quizás porque hay un error ortográfico en él o porque quiere mejorar el mensaje o cambiarlo por otras razones. O quizás quiera cambiar el autor de una confirmación porque se le olvidó preparar la autenticación, o ...

Subversion le deja cambiar tanto el mensaje de registro como el autor de las revisiones en cualquier momento. Pero como estos cambios no se pueden deshacer (estos cambios no se versionan), esta característica está deshabilitada por defecto. Para hacer que esto funcione, debe preparar un gancho *pre-revprop-change*. Por favor consulte el capítulo sobre *Scripts Gancho* [<http://svnbook.red-bean.com/en/1.1/ch05s02.html#svn-ch-5-sect-2.1>] en el Libro de Subversion para tener más detalles sobre cómo hacerlo. Lea Sección 4.3, “Scripts gancho” para encontrar más notas sobre cómo implementar ganchos en una máquina Windows.

Una vez que haya preparado su servidor con los ganchos necesarios, puede cambiar tanto el autor como el mensaje de registro de cualquier revisión, utilizando el menú contextual del panel superior del diálogo Registro.



Aviso

Dado que las propiedades de revisión de Subversion no se versionan, al hacer modificaciones a estas propiedades (por ejemplo, la propiedad del mensaje `svn:log`) se sobrescribirá el valor anterior de esa propiedad *para siempre*.

5.8.5. Filtrando los mensajes de registro

Si desea restringir los mensajes de registro para mostrar sólo en los que está interesado en vez de tener que navegar en una lista de cientos, puede utilizar los controles de filtro en la parte superior del

Diálogo de Registro. Los controles de fecha de inicio y de fin le permite restringir la salida a un rango de fechas conocido. La caja de texto de búsqueda le permite mostrar sólo los mensajes que contengan una frase en particular.

Tenga en cuenta que estos filtros actúan sobre los mensajes ya obtenidos. Ellos no controlan la descarga de mensajes desde el repositorio.

También puede filtrar los nombres de las rutas en el panel inferior utilizando la casilla **Ocultar rutas cambiadas no relacionadas**. Las rutas relacionadas son aquellas que contienen la ruta utilizada para mostrar el registro. Si está obteniendo el registro de una carpeta, eso significa todo lo que esté en esa carpeta o debajo de ella. Para un fichero, significa sólo ese fichero. La casilla tiene tres estados: puede mostrar todas las rutas, poner en gris las rutas no relacionadas, u ocultar esas rutas completamente.

5.8.6. Información estadística

El botón **Estadísticas** lanza un cuadro de diálogo que muestra algunas informaciones interesantes sobre las revisiones que se muestran en el diálogo Registro. Le muestra cuántos autores han estado trabajando, cuántas confirmaciones han hecho, el progreso por semanas, y mucho más. Ahora puede ver de un vistazo quién ha trabajado duro y quién se ha tocado la barriga ;-)

5.8.6.1. Página de estadísticas

Esta página le proporciona todas las cifras que pueda necesitar, en particular el período y el número de revisiones cubiertas, y algunos valores mínimos/máximos/medios.

5.8.6.2. Página de confirmaciones por autor

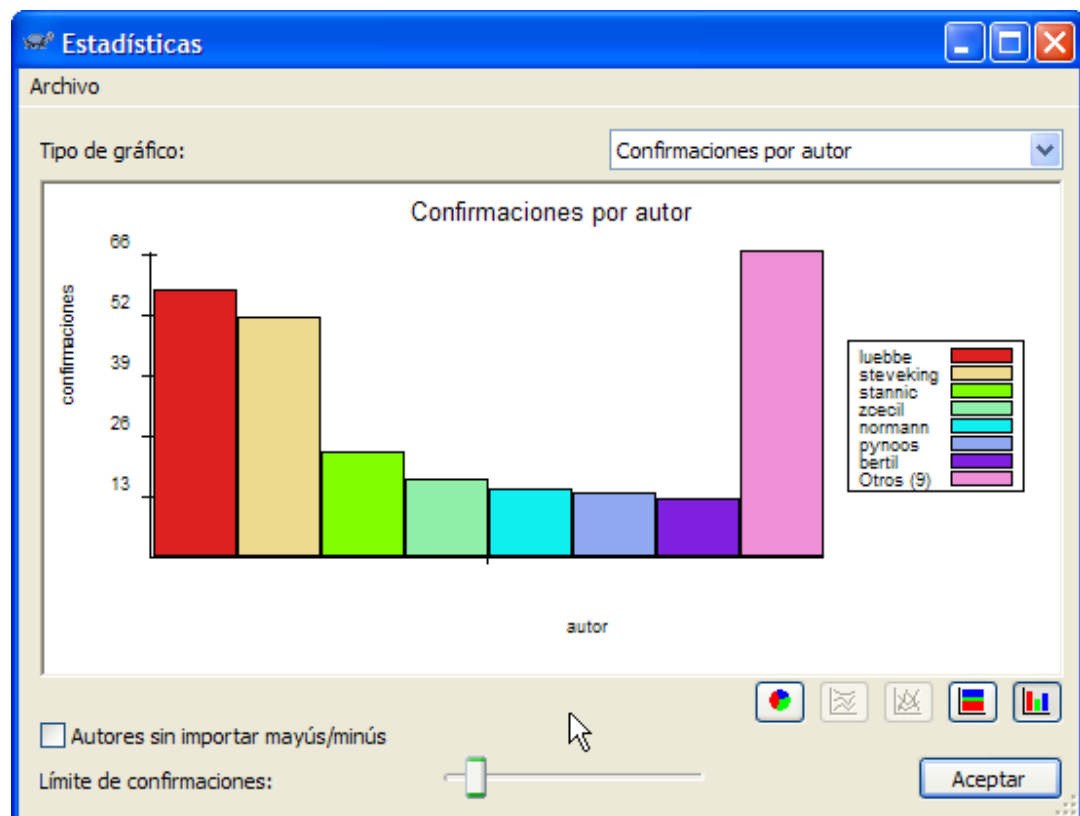


Figura 5.18. Histograma de confirmaciones por autor

Este gráfico le muestra qué autores han estado activos en el proyecto como un simple histograma, un histograma apilado o un gráfico de tarta.

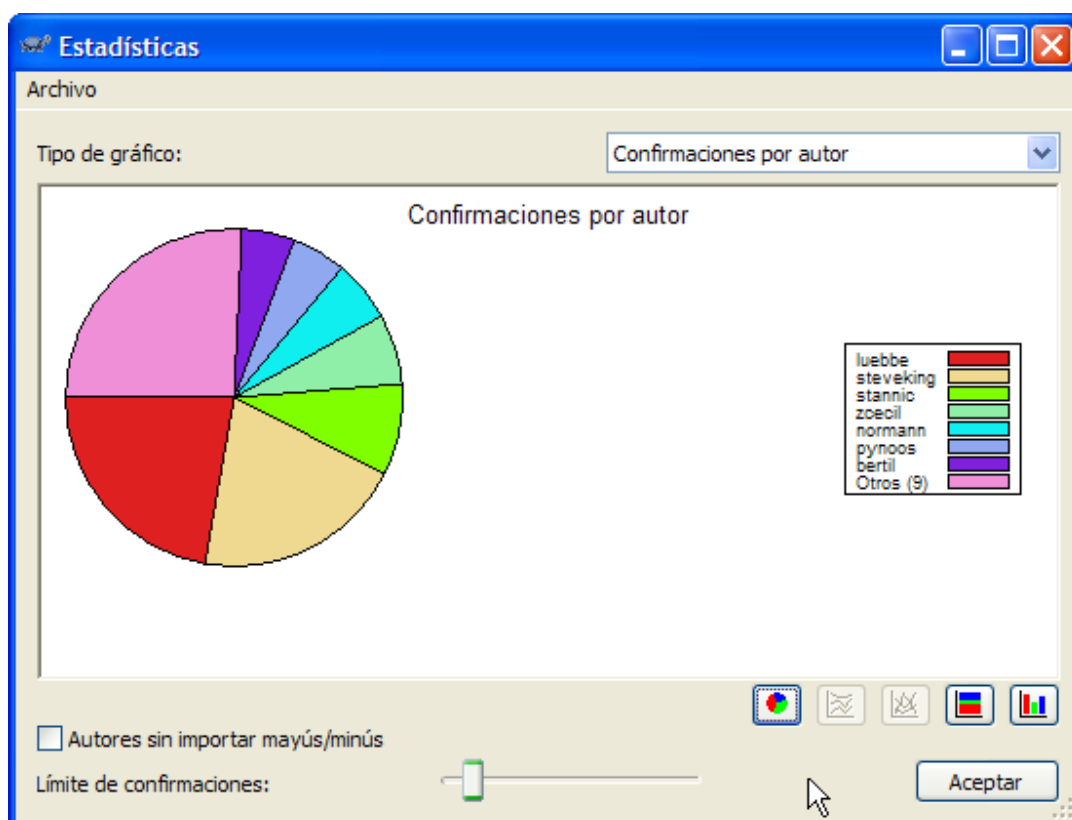


Figura 5.19. Gráfico de tarta de confirmaciones por autor

Cuando hay unos pocos autores muy activos y muchos pequeños contribuyentes, el número de segmentos pequeños puede hacer que el gráfico sea más difícil de leer. El selector deslizante en la parte inferior le permite establecer un límite (el porcentaje sobre el total de confirmaciones) bajo el cual cualquier actividad se agrupa en una categoría *Otros*.

5.8.6.3. Página de confirmaciones por semana

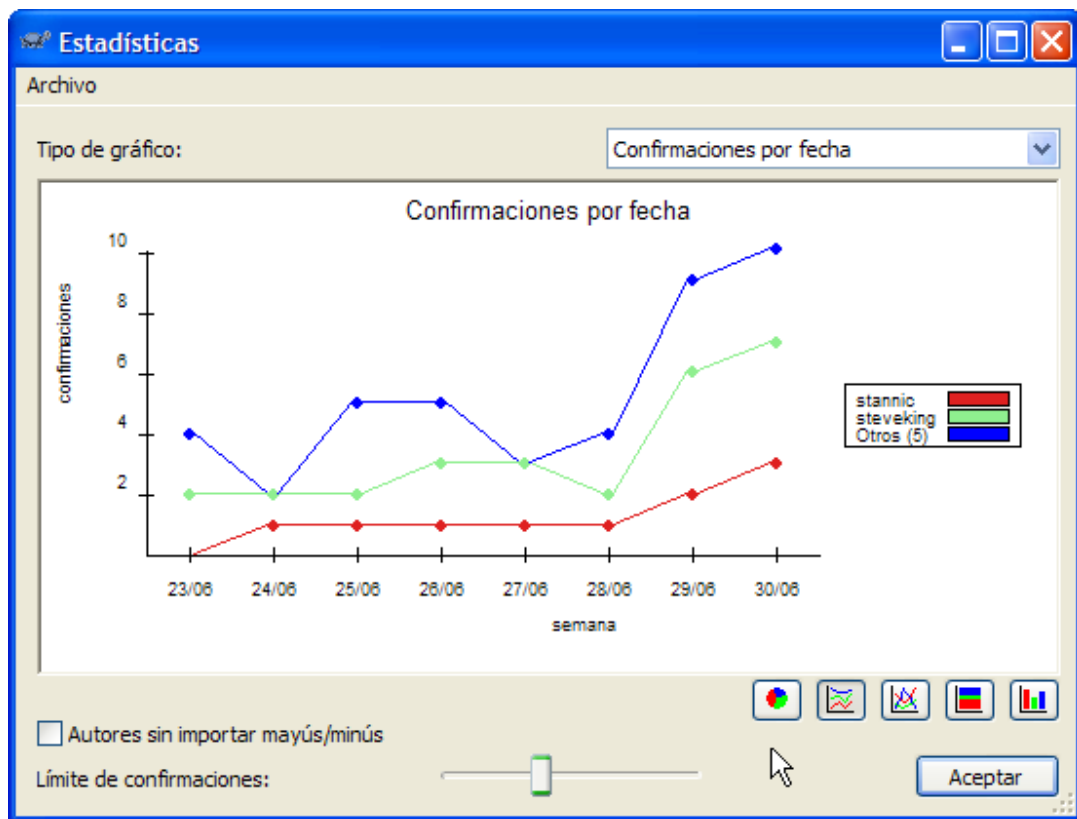


Figura 5.20. Gráfico de confirmaciones por semana

Esta página proporciona una representación gráfica de la actividad del proyecto en términos del número de confirmaciones y autor. Esto da una idea de cuándo se ha trabajado en un proyecto, y quién estaba trabajando en cada momento.

Cuando hay varios autores, puede obtener muchas líneas en el gráfico. Hay dos vistas disponibles aquí: *normal*, en la que la actividad de cada autor se refiere a la línea base, y *apilado*, donde la actividad de cada autor se refiere a la línea subyacente. La última opción evita que se crucen las líneas, lo que puede permitir un gráfico más sencillo de leer, pero es menos fácil ver la salida de cada autor.

Por defecto el análisis distingue mayúsculas y minúsculas, por lo que los usuarios PeterEgan y PeteRegan se tratan como autores diferentes. Sin embargo, en muchos casos los nombres de usuario no distinguen mayúsculas y minúsculas, y a veces se introducen de forma inconsistente, por lo que puede querer que DavidMorgan y davidmorgan se traten como la misma persona. Utilice la casilla Autores sin importar mayús/minús para controlar este comportamiento.

Tenga en cuenta que las estadísticas cubren el mismo período que el diálogo Registro. Si sólo se está mostrando una revisión, las estadísticas no le dirán mucho.

5.9. Viendo diferencias

Uno de los requisitos más comunes en el desarrollo de proyectos es ver qué ha cambiado. Puede querer ver las diferencias entre dos revisiones del mismo fichero, o las diferencias entre dos ficheros separados. TortoiseSVN provee una herramienta integrada llamada TortoiseMerge para ver las diferencias entre ficheros de texto. Para ver las diferencias entre ficheros de imagen, TortoiseSVN también tiene una herramienta llamada TortoiseIDiff. Por supuesto, puede utilizar su herramienta de diferencias favorita si lo desea.

5.9.1. Diferencias de ficheros

Cambios locales

Si desea ver qué cambios ha hecho *usted* en su copia de trabajo, simplemente utilice el menú contextual del explorador y seleccione **TortoiseSVN → Diferenciar**.

Diferenciar con otra rama/etiqueta

Si desea ver qué ha cambiado en el tronco (si está trabajando en una rama) o en una rama específica (si está trabajando en el tronco), puede utilizar el menú contextual del explorador. Simplemente sostenga la tecla **Mayúsculas** mientras hace click con el botón derecho en el fichero. Luego, seleccione **TortoiseSVN → Diferenciar con URL**. En el siguiente diálogo, especifique la URL del repositorio con la que quiere comparar su fichero local.

También puede utilizar el navegador de repositorios y seleccionar dos árboles para diferenciar, quizás dos ramas, o una rama/etiqueta y el tronco. Ahí, el menú contextual le permite comparar los utilizando **Comparar revisiones**. Lea más en Sección 5.9.2, “Comparando carpetas”.

Diferenciar desde una revisión anterior

Si desea ver las diferencias entre una revisión en concreto y su copia de trabajo, utilice el diálogo **Registro de Revisiones**, seleccione la revisión de interés, y luego seleccione **Comparar con la copia de trabajo** desde el menú contextual.

Diferenciar entre dos revisiones antiguas

Si desea ver las diferencias entre dos revisiones que ya se confirmaron, utilice el diálogo **Registro de Revisiones** y seleccione las dos revisiones que desea comparar (utilizando el modificador habitual **Ctrl**). Luego seleccione **Comparar revisiones** desde el menú contextual.

Si hizo esto desde el historial de revisiones de una carpeta, aparece un diálogo **Comparar Revisiones**, mostrando una lista de ficheros cambiados en esa carpeta. Lea más en Sección 5.9.2, “Comparando carpetas”.

Todos los cambios hechos en una confirmación

Si desea ver los cambios hechos a todos los ficheros en una revisión en particular de una vez, puede utilizar la salida **diff unificado** (formato de parche GNU). Esto le muestra sólo las diferencias con unas pocas líneas de contexto. Es más difícil de leer que una comparación visual de ficheros, pero le mostrará todos los cambios juntos. Desde el diálogo **Registro de Revisiones**, seleccione la revisión de interés, y luego seleccione **Mostrar Diferencias como Diff Unificado** desde el menú contextual.

Diferencias entre ficheros

Si desea ver las diferencias entre dos ficheros diferentes, puede hacerlo directamente en el explorador seleccionando ambos ficheros (utilizando el modificador habitual **Ctrl**). Luego desde el menú contextual del explorador seleccione **TortoiseSVN → Diferenciar**.

Diferencias entre un fichero/carpeta en la copia de trabajo y una URL

Si desea ver las diferencias entre un fichero en su copia de trabajo, y un fichero en cualquier repositorio de Subversion, puede hacerlo directamente en el explorador seleccionando el fichero y pulsando la tecla **Mayúsculas** mientras hace click con el botón derecho para obtener el menú contextual. Seleccione **TortoiseSVN → Diferenciar con URL**. Puede hacer lo mismo para una carpeta de copia de trabajo. **TortoiseMerge** muestra esas diferencias de la misma forma que muestra un fichero de parche - una lista de ficheros cambiados que puede ver de uno en uno.

Diferencias con información de autoría

Si desea ver no sólo las diferencias sino también el autor, revisión y la fecha en la que se hicieron los cambios, puede combinar los informes de diferencias y de autoría desde dentro del diálogo del historial de revisiones. Lea Sección 5.20.2, “Autoría de las diferencias” para más detalles.

Diferencias entre carpetas

Las herramientas proporcionadas con **TortoiseSVN** no le permiten ver las diferencias entre jerarquías de directorios. Pero si tiene una herramienta externa que soporte esta funcionalidad, puede utilizarla. En Sección 5.9.4, “Herramientas externas de diferencias/fusión” le contamos algunas de las herramientas que hemos utilizado.

5.9.2. Comparando carpetas

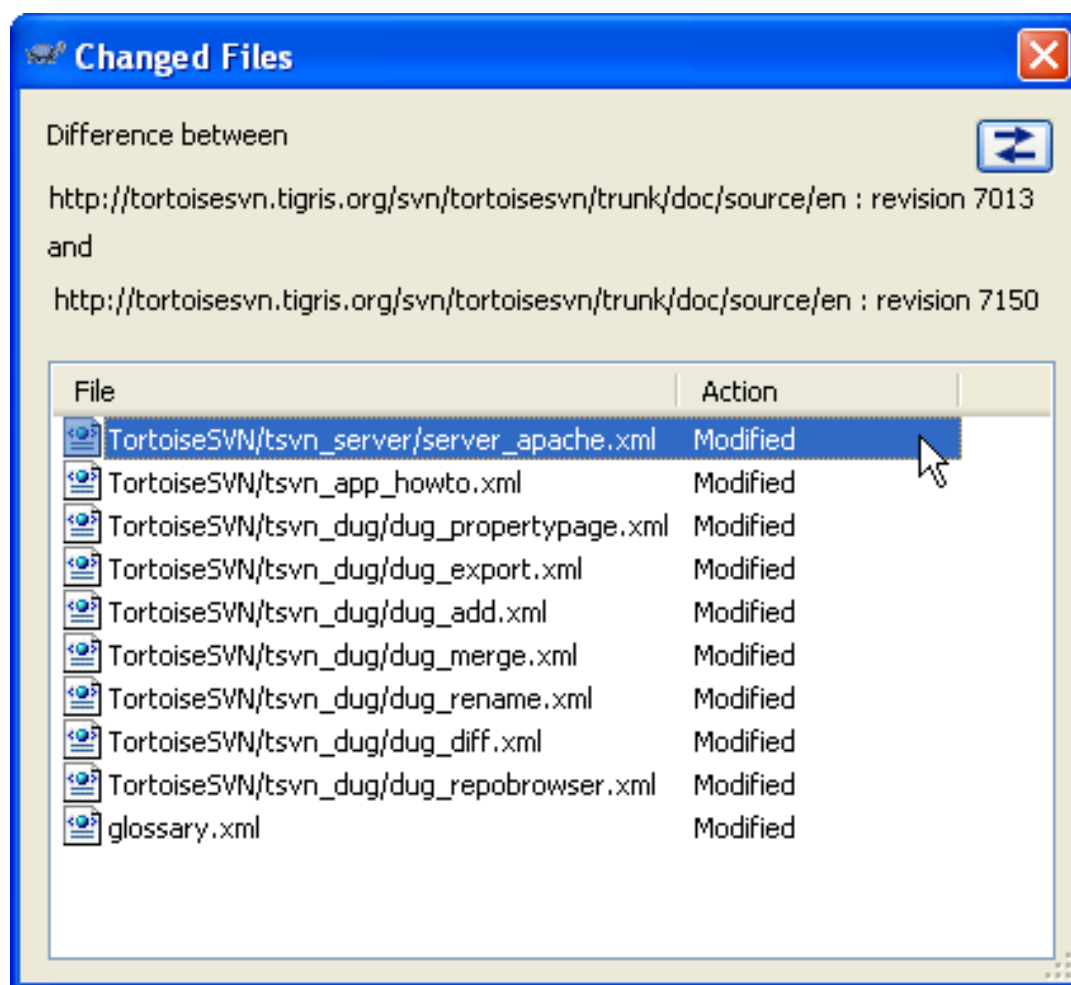


Figura 5.21. El diálogo Comparar Revisiones

Cuando seleccione dos árboles dentro del navegador de repositorios, o cuando seleccione dos revisiones de una carpeta en el diálogo de registro, puede Menú contextual → Comparar revisiones.

Este diálogo muestra una lista de todos los ficheros que han cambiado y le permite comparar o ver la autoría de cada uno individualmente utilizando el menú contextual.

También puede exportar la lista de ficheros cambiados a un fichero de texto, o puede exportar los propios ficheros cambiados a una carpeta. Esta operación funciona únicamente sobre los ficheros seleccionados, por lo que deberá seleccionar los ficheros de interés - normalmente eso quiere decir seleccionarlos a todos.

If you want to export the list of files *and* the actions (modified, added, deleted) as well, you can do that using the keyboard shortcuts Ctrl-A to select all entries and Ctrl-C to copy the detailed list to the clipboard.

El botón en la parte superior le permite cambiar la dirección de la comparación. Puede mostrar los cambios necesarios para ir de A a B, o si lo prefiere, de B a A.

5.9.3. Diferenciando imágenes utilizando TortoiseIDiff

Hay muchas herramientas disponibles para diferenciar ficheros de texto, incluyendo nuestro propio TortoiseMerge, pero a veces nos encontramos en la situación de querer ver también cómo ha cambiado un fichero de imagen. Por eso es por lo que hemos creado TortoiseIDiff.

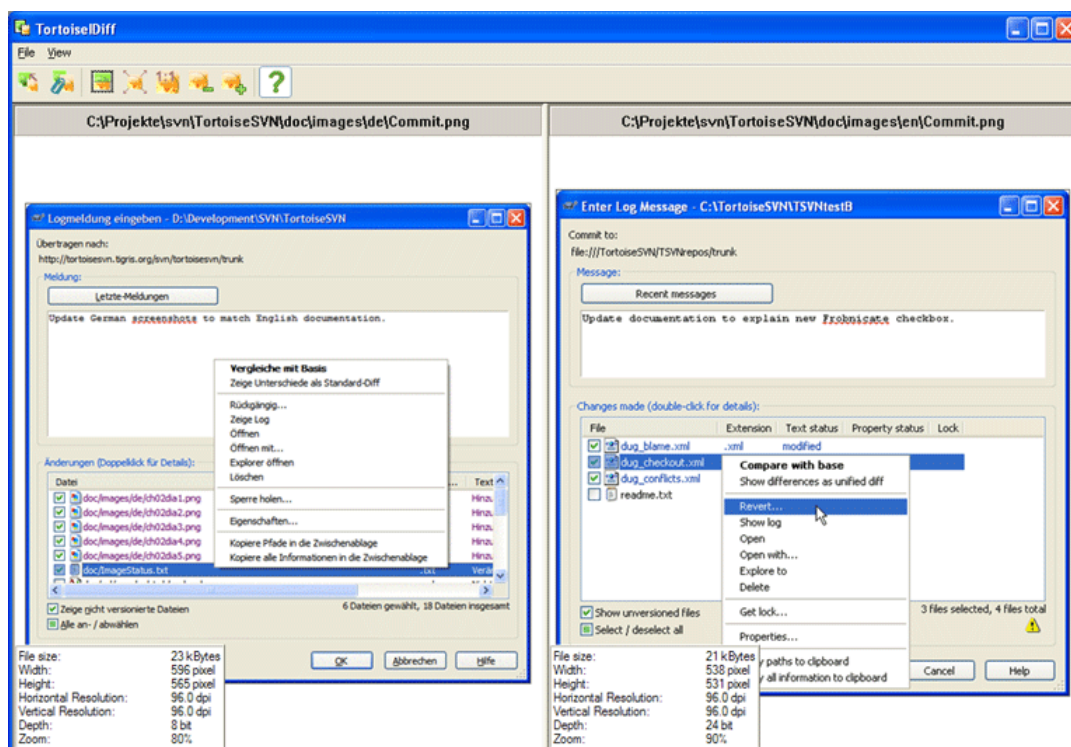


Figura 5.22. El visor de diferencias de imágenes

TortoiseSVN → Diff for any of the common image file formats will start TortoiseIDiff to show image differences. By default the images are displayed side-by-side but you can use the View menu or toolbar to switch to a top-bottom view instead, or if you prefer, you can overlay the images and pretend you are using a lightbox. A slider at the top controls the relative intensity of the images (alpha blend). You can also use Ctrl-Shift-Wheel to change the blend.

Naturally you can also zoom in and out and pan around the image. If you select the Link images together option, then the pan controls (scrollbars, wheelmouse) on both images are linked.

Un cuadro de información de imagen le muestra detalles sobre el fichero de imagen, tales como el tamaño en pixeles, la resolución y la profundidad de color. Si la caja le molesta, utilice Ver → Información de imagen para ocultarla.

5.9.4. Herramientas externas de diferencias/fusión

Si las herramientas que le proporcionamos no son lo que necesita, pruebe alguno de los muchos programas de código abierto o comerciales disponibles. Cada uno tiene sus propios favoritos, y esta lista por supuesto no está completa, pero aquí hay algunas que debería considerar:

WinMerge

WinMerge [<http://winmerge.sourceforge.net/>] es un muy buen visor de diferencias y herramienta de fusión de código abierto, que también puede manejar directorios.

Perforce Merge

Perforce es un RCS comercial, pero puede descargar la herramienta de diferencias/fusiones gratuitamente. Obtenga más información en *Perforce* [<http://www.perforce.com/perforce/products/merge.html>].

KDiff3

KDiff3 es una herramienta de diferencias gratuita que también maneja directorios. Puede descargarla desde *aquí* [<http://kdif3.sf.net/>].

ExamDiff

ExamDiff Standard es gratuito. Puede manejar ficheros pero no directorios. ExamDiff Pro es shareware y añade un número de ventajas que incluyen diferenciación de directorios y capacidades de edición. En ambos casos, la versión 3.2 y las siguientes pueden manejar también unicode. Puede descargarlas desde *PrestoSoft* [<http://www.prestosoft.com/>].

Beyond Compare

Similar a ExamDiff Pro, esta es una herramienta shareware de diferencias que puede manejar diferenciación de directorios y unicode. Descárgela desde *Scooter Software* [<http://www.scootersoftware.com/>].

Araxis Merge

Araxis Merge es una útil herramienta comercial para ver diferencias y fusionar tanto ficheros como carpetas. Hace comparaciones a tres bandas en las fusiones y tiene vínculos de sincronización para utilizar si ha cambiado el orden las funciones. Descárgelo desde *Araxis* [<http://www.araxis.com/merge/index.html>].

SciTE

Este editor de texto incluye sintaxis coloreada para ficheros diff unificados, haciéndolos mucho más fáciles de leer. Descárgelo desde *Scintilla* [<http://www.scintilla.org/SciTEDownload.html>].

Notepad2

Notepad2 está diseñado como un reemplazo del programa Bloc de Notas estándar de Windows, y se basa en el control de edición de código abierto Scintilla. Además de ser bueno para ver diffs unificados, es mucho mejor que el bloc de notas de Windows para la mayoría de trabajo. Descárgelo gratuitamente *aquí* [<http://www.flos-freeware.ch/notepad2.html>].

Lea Sección 5.26.4, “Configuración de programas externos” para más información sobre cómo preparar TortoiseSVN para utilizar estas herramientas.

5.10. Añadiendo nuevos ficheros y directorios

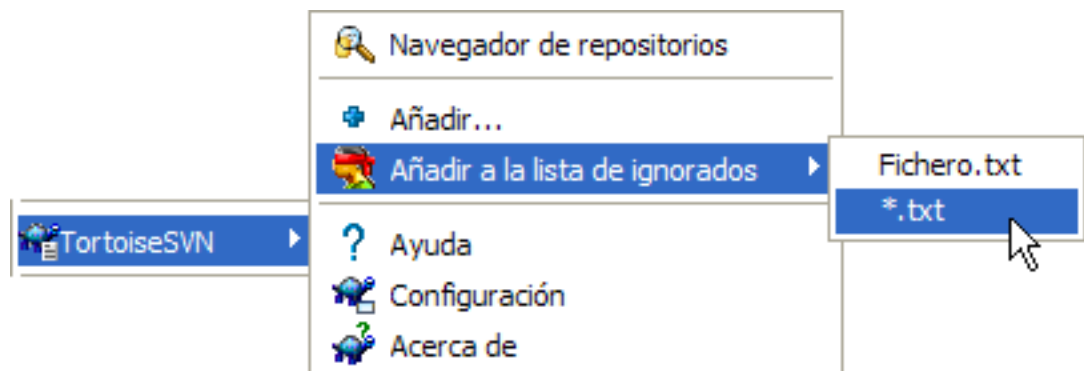


Figura 5.23. Menú contextual del explorador para ficheros no versionados

Si ha creado nuevos ficheros y/o directorios durante su proceso de desarrollo, necesitará añadirlos también al control de código. Seleccione los ficheros y/o directorios y utilice TortoiseSVN → Añadir.

Después de que añada los ficheros o directorios al control de código, el fichero aparece con una sobreimpresión de icono añadido que significa que primero debe confirmar su copia de trabajo para que esos ficheros y directorios estén disponibles para otros desarrolladores. ¡Añadir un fichero/directorio *no* afecta al repositorio!



Añadir muchos

También puede usar el comando Añadir en carpetas que ya estén versionadas. En ese caso, el diálogo de añadir le mostrará todos los ficheros sin versionar dentro de la carpeta versionada. Esto ayuda si tiene muchos ficheros nuevos y necesita añadirlos todos de golpe.

Para añadir ficheros desde fuera de su copia de trabajo puede usar el manejador de arrastrar-y-soltar:

1. seleccione los ficheros que desea añadir
2. arrástrelos con el botón derecho a su nuevo destino dentro de la copia de trabajo
3. suelte el botón derecho del ratón
4. seleccione Menú contextual → SVN Añadir ficheros a esta Copia de Trabajo. En ese momento los ficheros se copiarán a la copia de trabajo y se añadirán al control de versiones.

5.11. Ignorando ficheros y directorios

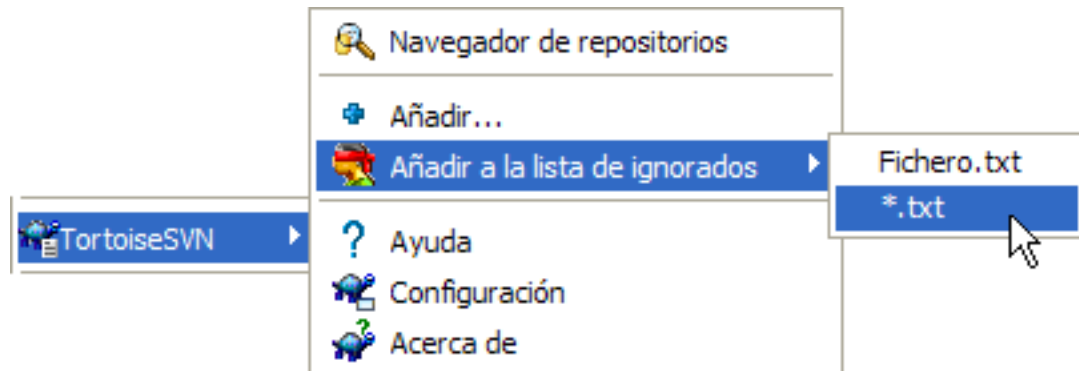


Figura 5.24. Menú contextual del explorador para ficheros no versionados

En la mayoría de los proyectos tendrá ficheros y carpetas que no deberán estar sujetos al control de versiones. Éstas pueden incluir ficheros creados por el compilador, *.obj, *.lst, quizás incluyan una carpeta de salida donde se almacenan los ejecutables. Cuando confirma los cambios, TSVN le muestra los ficheros no versionados, que rellenan a la lista de ficheros del diálogo de confirmar. Por supuesto que puede desactivarlos para que no se muestren, pero entonces quizás se olvide de añadir un nuevo fichero fuente.

La mejor forma de evitar estos problemas es añadir los ficheros derivados a la lista de ignorados del proyecto. De esta forma nunca se mostrarán en el diálogo de confirmar, pero se seguirán señalando los genuinos ficheros de código no versionados.

Si hace click con el botón derecho en un único fichero no versionado, y selecciona el comando TortoiseSVN → Ignorar del menú contextual, aparecerá un submenú que le permitirá seleccionar sólo ese fichero, o todos los ficheros con la misma extensión. Si selecciona múltiples ficheros, no hay submenú y sólo puede añadir esos ficheros o carpetas específicos.

Si desea eliminar uno o más ítems de la lista de ignorados, haga click con el botón derecho en dichos ítems y seleccione TortoiseSVN → Eliminar de la Lista de Ignorados También puede acceder a la propiedad svn:ignore de la carpeta directamente. Esto le permitirá especificar patrones más generales utilizando expansión de comodines en los nombres de los ficheros, descrito en la sección siguiente. Para más información sobre establecer propiedades directamente, lea Sección 5.15, “Configuración del proyecto”.



La lista global de ignorados

Otra forma de ignorar ficheros es añadirlos a la *lista global de ignorados*. La mayor diferencia aquí es que la lista global de ignorados es una propiedad del cliente. Se aplica a *todos* los proyectos de Subversion, pero sólo en el PC cliente. En general es mejor utilizar la propiedad `svn:ignore` donde sea posible, porque puede aplicarse a áreas del proyecto específicas, y funciona para todos los que obtengan el proyecto. Para más información, lea la Sección 5.26.1, “Configuración general”.



Ignorando ítems versionados

Los ficheros y las carpetas versionadas nunca pueden ser ignoradas - esta es una característica de Subversion. Si ha versionado un fichero por error, lea Sección B.8, “Ignorar ficheros que ya están versionados” para ver las instrucciones sobre cómo “desversionarlo”.

5.11.1. Expansión de comodines en los nombres de los ficheros de la lista de ignorados

El patrón de ignorados de Subversion hace uso de la expansión de comodines en los nombres de los ficheros, una técnica que originariamente se utilizaba en Unix para especificar ficheros utilizando meta-caracteres como son los comodines. Los siguientes caracteres tienen un significado especial:

*

Concuerda con cualquier cadena de caracteres, incluyendo la cadena vacía (sin caracteres).

?

Concuerda con un único carácter cualquiera.

[...]

Concuerda con cualquiera de los caracteres incluidos entre los corchetes. Dentro de los corchetes, un par de caracteres separados por “-” concuerda con cualquier carácter existente entre los dos en el orden lexicográfico. Por ejemplo, `[AGm-p]` concuerda con cualquiera de estos: A, G, m, n, o o p.

La expansión de comodines en los nombres de los ficheros la realiza Subversion, por lo que el separador de rutas es siempre `/` y no la barra invertida de Windows.

La concordancia de patrones distingue entre mayúsculas y minúsculas, lo que puede causar problemas en Windows. Puede forzar que no se distingan a lo bruto, emparejando los caracteres; es decir, por ejemplo, para ignorar `*.tmp` sin tener en cuenta las mayúsculas y minúsculas, puede utilizar un patrón como `*.[Tt][Mm][Pp]`.

Si hay nombres de directorios presentes en una ruta, se incluyen en la concordancia, por lo que el patrón `Fred.*` concuerda con `Fred.c` pero no con `subdir/Fred.c`. Esto es importante sobre todo si añade una carpeta que contenga algunos ficheros que desea ignorar, dado que dichos ficheros vendrán precedidos por el nombre de la carpeta.

Para ignorar todas las carpetas CVS debería especificar o bien un patrón `*CVS` o, mejor aún, el par `CVS */CVS`. La primera opción funciona, pero también excluye algo llamado `EstoNoEsCVS`. Utilizar `*/CVS` por sí mismo no funcionará para los hijos inmediatos de la carpeta CVS, y CVS solo no funcionará en las subcarpetas.

Si desea una definición oficial de la expansión de comodines en los nombres de ficheros, puede encontrarla en las especificaciones de la IEEE para el lenguaje de comandos del shell *Notación de Patrones* de *Concordancia*.

[http://www.opengroup.org/onlinepubs/009695399/utilities/xcu_chap02.html#tag_02_13].

5.12. Borrando, renombrando y moviendo

Al contrario que CVS, Subversion le permite renombrar y mover ficheros y carpetas. Por tanto hay entradas de menú para borrar y renombrar en el submenú TortoiseSVN.

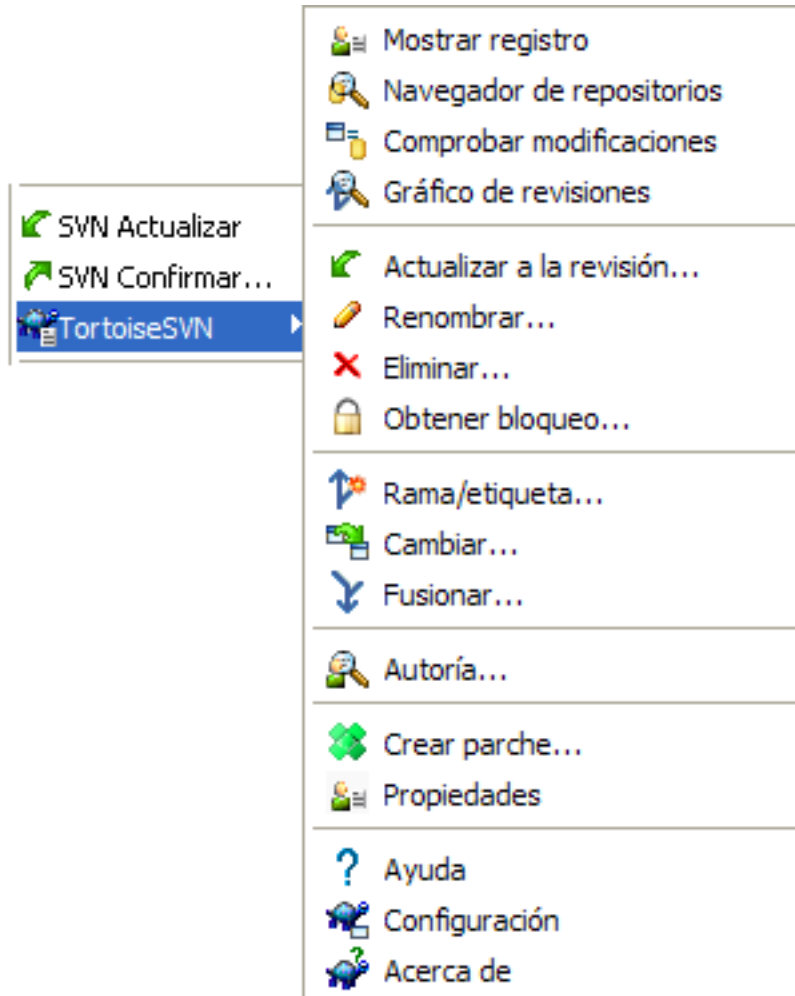


Figura 5.25. Menú contextual del explorador para ficheros versionados

Si borra un fichero/directorio utilizando TSVN, el fichero se elimina de su copia de trabajo y se marca para ser borrado. La carpeta padre del fichero muestra un icono sobreimpresionado "borrado". Siempre puede recuperar de nuevo el fichero borrado, si llama a TortoiseSVN → revertir en la carpeta padre.

Si quiere mover ficheros dentro de una copia de trabajo, puede usar de nuevo el manejador de arrastrar-y-soltar:

1. seleccione los ficheros o directorios que desea mover
2. arrástrelos con el botón derecho a su nuevo destino dentro de la copia de trabajo
3. suelte el botón derecho del ratón
4. en el menú contextual seleccione Menú contextual → SVN Mover ficheros versionados aquí



No SVN Mueva externos

NO debería utilizar los comandos TortoiseSVN Mover o Renombrar en una carpeta que ha sido creada utilizando `svn:externals`. Esta acción puede provocar que el ítem externo se elimine de su repositorio padre, probablemente molestando a muchas otras personas. Si necesita mover una carpeta externa, debería moverla como lo hace con el resto de ficheros sin versionar, y luego ajustar las propiedades `svn:externals` de las carpetas padres origen y destino.

Si se borra un *fichero* utilizando el Explorador en vez de usar el menú contextual de TortoiseSVN, el diálogo de confirmación le muestra esos ficheros y le deja eliminarlos del control de versiones antes de confirmar. Sin embargo, si actualiza su copia de trabajo, Subversion se dará cuenta de que falta un fichero y lo reemplazará con la última versión del repositorio. Si necesita borrar un fichero bajo control de versiones, utilice siempre TortoiseSVN → Eliminar para que Subversion no tenga que averiguar qué es lo que realmente quiere hacer usted.

Si se borra una *carpeta* utilizando el Explorador en vez de utilizar el menú contextual de TortoiseSVN, su copia de trabajo se romperá y no podrá confirmar. Si actualiza su copia de trabajo, Subversion reemplazará la carpeta que falta con la última versión del repositorio, y luego podrá eliminarla de la forma correcta, utilizando TortoiseSVN → Eliminar.



Confirmar la carpeta padre

Dado que las operaciones mover y renombrar se realizan como un borrado seguido de un añadir, debe confirmar la carpeta padre de los ficheros movidos/renombrados para que se muestre la parte de borrado de la operación en el diálogo de confirmación. Si no confirma la parte eliminada de los ficheros movidos/renombrados, se quedarán en el repositorio y cuando sus compañeros se actualicen no se eliminará el fichero antiguo, es decir, que ellos tendrán *ambas* copias, la antigua y la nueva.

Usted *debe* confirmar un cambio de nombre de carpeta antes de cambiar cualquier fichero dentro de la carpeta; si no, su copia de trabajo realmente puede quedar estropeada.



Recuperando un fichero o una carpeta borrados

Si ha borrado un fichero o una carpeta y ya ha confirmado esa operación de borrado en el repositorio, entonces un comando TortoiseSVN → Revertir normal no los podrá recuperar. Pero el fichero o la carpeta borrados no están perdidos para siempre. Si sabe la revisión en la que se borraron el fichero o la carpeta (si no lo sabe, utilice el diálogo de registro para averiguarlo), abra el navegador de repositorios y cambie a esa revisión. Luego seleccione el fichero o la carpeta que ha borrado, haga click con el botón derecho y seleccione Menú Contextual → Copiar a..., y como destino de esa operación de copia seleccione la ruta de su copia de trabajo.

5.12.1. Renombrando un fichero sólo en mayúsculas/minúsculas

En el caso de que tenga dos ficheros en el repositorio con el mismo nombre pero diferenciándose únicamente en las mayúsculas (por ejemplo, TEST.TXT y test.txt), ya no podrá actualizar u obtener el directorio donde están esos ficheros.

En ese caso, necesita decidir cuál desea mantener y borrar (o renombrar) el otro en el repositorio.

Hay (al menos) dos soluciones posibles para renombrar un fichero sin perder su historia de registro. Es importante renombrarlo dentro de Subversion. ¡¡¡Renombrarlo únicamente en el explorador corromperá su copia de trabajo!!!

Solución A) (recomendada)

1. Confirme los cambios en su copia de trabajo.
2. Renombrar el fichero de MAYusculas a mayUSCULAS directamente en el repositorio utilizando el Navegador de Repositorios.
3. Actualice su copia de trabajo.

Solución B)

1. Renombre de MAYusculas a MAYusculas_ con el comando Renombrar del submenú de TortoiseSVN.
2. Confirme los cambios.
3. Renombre de MAYusculas_ a mayUSCULAS.
4. Confirme los cambios.



Evitando dos ficheros con el mismo nombre

Hay un script gancho de servidor disponible en: <http://svn.collab.net/repos/svn/trunk/contrib/hook-scripts/> que evitará confirmaciones cuyo resultado sea dos ficheros con conflicto en las mayúsculas.

5.13. Deshacer cambios

Si desea deshacer todos los cambios que ha hecho en un fichero desde la última actualización, necesita seleccionar el fichero, hacer click con el botón derecho para sacar el menú contextual, y luego seleccionar el comando TortoiseSVN → Revertir. Aparecerá un diálogo que le muestra los ficheros que ha cambiado y que puede revertir. Seleccione los que desee revertir y pulse Aceptar.

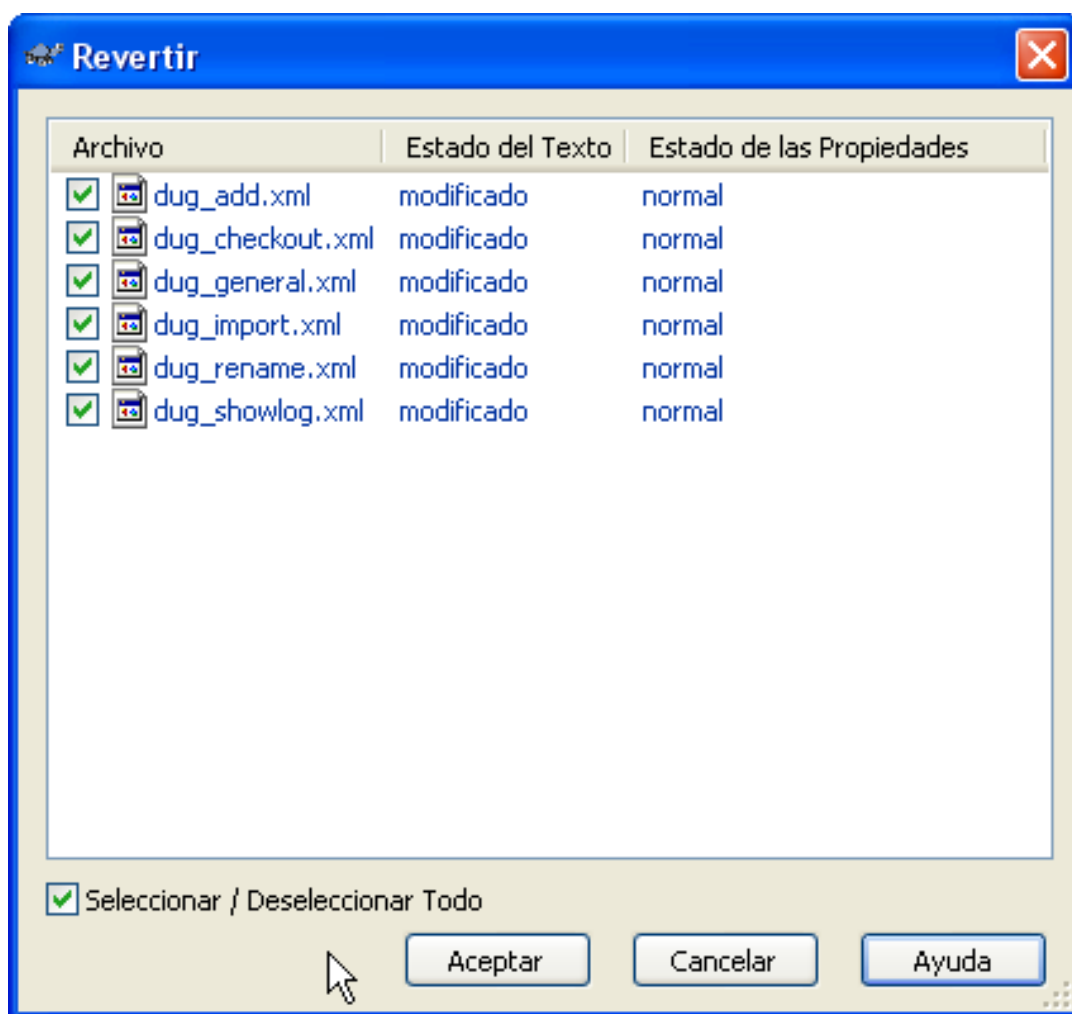


Figura 5.26. Diálogo de Revertir

Las columnas en este diálogo pueden ser personalizadas de la misma forma que las columnas en el diálogo Comprobar modificaciones. Para más detalles, lea Sección 5.7.3, “Estado local y remoto”.



Deshaciendo cambios que han sido confirmados

Revertir sólo deshacerá sus cambios locales. *No* deshace ningún cambio que ya haya sido confirmado. Si desea deshacer todos los cambios que se confirmaron en una revisión en particular, lea Sección 5.8, “Diálogo de Registro de revisiones” para obtener más información.

5.14. Limpieza

Si un comando de Subversion no puede completarse de forma correcta, quizás por problemas en el servidor, su copia de trabajo puede quedarse en un estado inconsistente. En ese caso deberá utilizar TortoiseSVN → Limpiar en la carpeta. Es una buena idea hacerlo en la rama superior de la copia de trabajo.

La limpieza tiene otro efecto secundario muy útil. Si ha cambiado la fecha de un fichero pero no su contenido, Subversion no puede determinar si el fichero ha cambiado realmente excepto haciendo una comparación byte-a-byte con la copia prístina. Si tiene muchos ficheros en este estado, compro-

bar su estado será muy lento, lo que hará que muchos diálogos ralenticen su respuesta. Ejecutando una Limpieza en su copia de trabajo reparará esas fechas “rotas” y restaurará las comprobaciones de estado a la máxima velocidad.



Utilizar fechas de confirmación

Algunas versiones anteriores de Subversion estaban afectadas con un error que causaba errores en la fecha cuando obtenía con la opción **Utilizar fechas de confirmación** activada. Utilice el comando Limpieza para acelerar esas copias de trabajo.

5.15. Configuración del proyecto

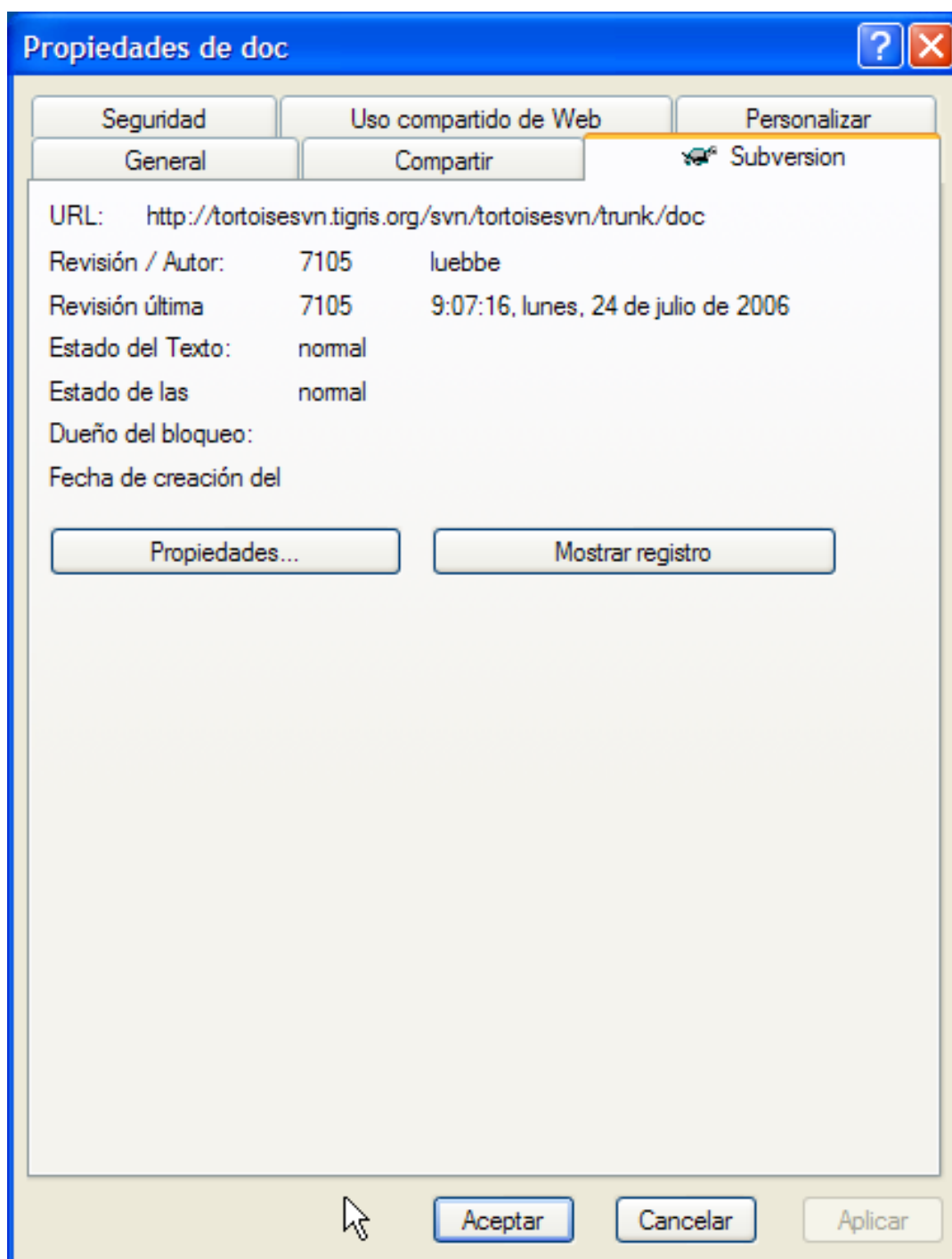


Figura 5.27. Página de propiedades del Explorador, pestaña Subversion

A veces desea obtener información más detallada sobre un fichero o directorio que la que proporcionan los iconos sobreimpresionados. Puede obtener toda la información que Subversion proporciona en el diálogo de propiedades del explorador. Simplemente seleccione el fichero o directorio y seleccione Menú de Windows → Propiedades en el menú contextual (atención: ésta es la entrada normal del menú de propiedades que el explorador proporciona, ¡no la que está en el submenú de TortoiseSVN!). En el cuadro de diálogo de propiedades, TortoiseSVN ha añadido una nueva página de propiedades para los ficheros y carpetas bajo control de Subversion, donde puede ver toda la información relevante sobre el fichero/directorio seleccionado.

5.15.1. Propiedades de Subversion

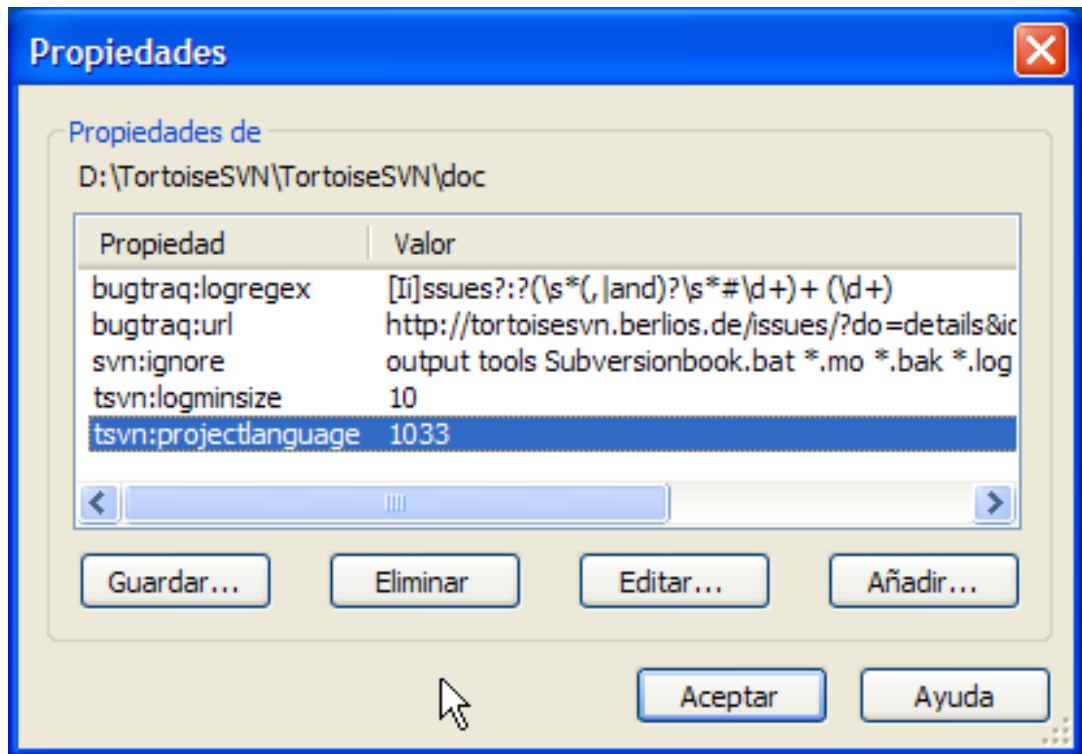


Figura 5.28. Página de propiedades de Subversion

Puede leer y establecer las propiedades de Subversion desde el diálogo de propiedades de Windows, pero también desde TortoiseSVN → Propiedades y dentro de las listas de estado de TSVN's, desde Menú contextual → Propiedades.

Puede añadir sus propias propiedades, o algunas propiedades con un significado especial en Subversion. Éstas empiezan con `svn:`. `svn:externals` es una de esas propiedades; vea cómo manejar externos en Sección 5.2.4, “Proyectos referenciados”. Para más información sobre las propiedades en Subversion vea el *Manual de Subversion* [<http://svnbook.red-bean.com/en/1.1/ch07s02.html#svn-ch-7-sect-2.3>].

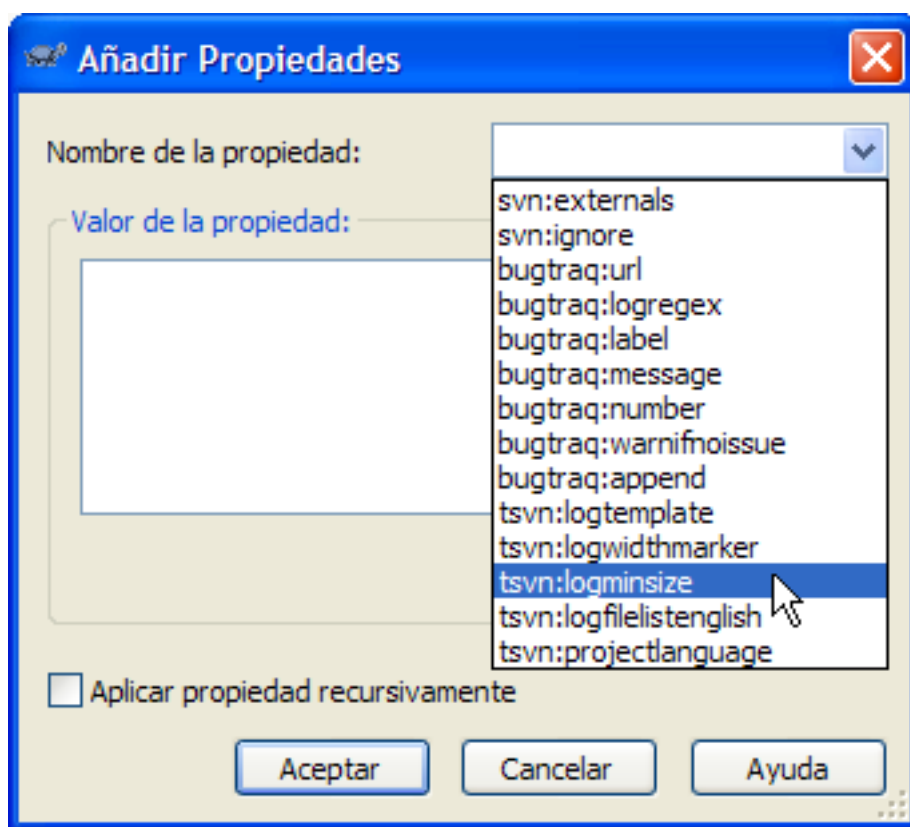


Figura 5.29. Añadiendo propiedades

Para añadir una nueva propiedad, primero pulse en **Añadir...** Seleccione el nombre de la propiedad deseado del cuadro desplegable, o escriba un nombre de su elección, y luego introduzca un valor en la caja de texto que hay debajo. Para las propiedades que admiten múltiples valores, como una lista de ignorados, éstos se pueden introducir en múltiples líneas. Pulse en **Aceptar** para a

Si desea aplicar una propiedad a muchos ítems a la vez, seleccione los ficheros/carpetas en el explorador, y luego seleccione **Menú contextual** → **propiedades**

Si desea aplicar la propiedad a *cada* fichero y carpeta en la jerarquía debajo de la carpeta actual, seleccione la casilla **Recursivo**.

Algunas propiedades, por ejemplo `svn:needs-lock`, pueden aplicarse únicamente a ficheros, por lo que ese nombre de propiedad no aparece en la lista desplegable para las carpetas. Aún así puede aplicar dicha propiedad recursivamente a todos los ficheros en una jerarquía, pero tendrá que escribir el nombre de la propiedad usted mismo.

Si desea editar una propiedad existente, primero seleccione dicha propiedad de la lista de propiedades existentes, y luego pulse en **Editar...**

Si desea eliminar una propiedad existente, seleccione dicha propiedad de la lista de propiedades existentes, y luego pulse en **Eliminar**.

La propiedad `svn:externals` se puede utilizar para traer otros proyectos del mismo repositorio o de otro completamente distinto. Para más información, lea Sección 5.2.4, “Proyectos referenciados”.

TortoiseSVN puede manejar valores de propiedades binarios utilizando ficheros. Para leer un valor de propiedad binario, utilice **Grabar...** para almacenarlo en un fichero. Para establecer un valor binario, utilice un editor hexadecimal u otra herramienta adecuada para crear un fichero con el contenido que necesite, y luego utilice **Cargar...** para leer ese fichero.



Confirmar propiedades

Las propiedades de Subversion están versionadas. Después de que cambie o añada una propiedad tiene que confirmar sus cambios.



Conflictos en las propiedades

Si hay un conflicto al confirmar los cambios, porque otro usuario ha cambiado la misma propiedad, Subversion genera un fichero `.prej`. Borre este fichero después de que haya resuelto del conflicto.



Establecer propiedades automáticamente

Puede configurar Subversion para que establezca propiedades automáticamente en ficheros y carpetas cuando se añaden al repositorio. Para más información, lea Sección 5.26, “Configuración de TortoiseSVN”.

5.15.2. Propiedades de TortoiseSVN

TortoiseSVN tiene unas pocas propiedades especiales para sí mismo, y estas empiezan con `tsvn:`.

- `tsvn:logminsize` establece la longitud mínima de un mensaje de registro para una confirmación. Si introduce un mensaje más corto de lo especificado aquí, la confirmación se deshabilita. Esta característica es muy útil para que se acuerde de proporcionar un mensaje descriptivo apropiado para cada confirmación. Si esta propiedad no se establece, o el valor es cero, se permiten mensajes de registro vacíos.

`tsvn:lockmsgminsize` establece la longitud mínima para un mensaje de bloqueo. Si introduce un mensaje más corto de lo especificado aquí, el bloqueo se deshabilita. Esta característica es muy útil para que se acuerde de proporcionar un mensaje descriptivo apropiado para cada bloqueo que obtenga. Si esta propiedad no se establece, o el valor es cero, se permiten mensajes de bloqueo vacíos.

- `tsvn:logwidthmarker` se utiliza con proyectos que necesitan que los mensajes de registro se formateen con algún ancho máximo (típicamente, 80 caracteres) antes de un salto de línea. Al establecer esta propiedad a un valor distinto de cero ocurren dos cosas en el diálogo de entrada de mensajes de registro: pone un marcador para indicar el ancho máximo, y deshabilita el ajuste de línea en la pantalla, para que pueda ver si el texto que ha introducido es muy largo. Tenga en cuenta que esta característica sólo funcionará correctamente si ha seleccionado una fuente de ancho fijo para los mensajes de registro.
- `tsvn:logtemplate` se utiliza en proyectos que tienen reglas sobre el formato de mensajes. Esta propiedad alberga una cadena de texto de múltiples líneas que se insertará en el cuadro del mensaje de confirmación cuando inicie una confirmación. Puede editarlo para incluir la información necesaria. Tenga en cuenta que si estaba utilizando también `tsvn:logminsize`, debe asegurarse de establecer una longitud mayor que la de la plantilla o perderá el mecanismo de protección.
- En el diálogo de Confirmar tiene la opción de pegar la lista de ficheros cambiados, incluyendo el estado de cada fichero (añadido, modificado, etc). `tsvn:logfilelistenglish` define si el estado del fichero se inserta en inglés o en el idioma traducido. Si la propiedad no se establece, el valor por defecto es `true`.
- TortoiseSVN puede utilizar los módulos de corrector ortográfico que se utilizan también por OpenOffice y Mozilla. Si tiene éstos instalados, esta propiedad determinará qué corrector orto-

gráfico utilizar, es decir, en qué lenguaje deberían estar escritos los mensajes de registro para su proyecto. `tsvn:projectlanguage` establece el módulo de idioma que el motor del corrector ortográfico debería utilizar cuando introduzca un mensaje de registro. Puede encontrar los valores para su idioma en esta página: *MSDN: Language Identifiers* [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/intl/nls_238z.asp].

Puede introducir este valor en decimal, o en hexadecimal si se precede por 0x. Por ejemplo, Español (España, ordenación tradicional) puede introducirse como 0x040a or 1034.

Algunas propiedades `tsvn:` necesitan un valor `true/false` (verdad/mentira). TSVN también entiende `yes` (sí) como un sinónimo de `true` y `no` como un sinónimo de `false`.



Establecer las propiedades tsvn: en carpetas

Estas propiedades `tsvn:` deben estar establecidas en *carpetas* para que el sistema funcione. Cuando confirma un fichero o una carpeta, se leen las propiedades de esa carpeta. Si no se encuentran allí las propiedades, TortoiseSVN las buscará hacia arriba en el árbol de carpetas para encontrarlas, hasta que llega a una carpeta sin versionar, o se encuentra la raíz del árbol (por ejemplo, `C:\`). Si puede estar seguro de que cada usuario obtiene sólo desde por ejemplo `trunk/` y no desde alguna subcarpeta, entonces es suficiente establecer las propiedades en `trunk/`. Si no puede estar seguro, debería establecer las propiedades recursivamente en cada subcarpeta. Una propiedad establecida hacia las profundidades de la jerarquía del proyecto tiene preferencia sobre las propiedades establecidas en niveles más altos (más cerca de `trunk/`).

Para las propiedades `tsvn:` sólo puede utilizar la casilla **Recursivo** para establecer la propiedad en todas las subcarpetas de la jerarquía, sin establecera en todos los ficheros.

TortoiseSVN puede integrarse con algunas herramientas de control de errores. Esto utiliza propiedades, que comienzan con `bugtraq:`. Para más información, lea Sección 5.25, “Integración con sistemas de control de errores / seguimiento de incidencias”.

5.16. Haciendo ramas / etiquetas

Una de las características de los sistemas de control de versiones es la posibilidad de aislar cambios en una línea separada de desarrollo. Esto se conoce como una *rama*. Las ramas se utilizan a menudo para probar nuevas características sin molestar la línea principal del desarrollo con errores de compilación y errores. Tan pronto como la nueva característica es lo suficiente estable, la rama de desarrollo se *fusiona* de nuevo en la rama principal (`trunk`, troncal).

Otra característica de los sistemas de control de versiones es la posibilidad de marcar revisiones particulares (por ejemplo, una versión lanzada a producción), para que pueda en cualquier momento recrear un cierto entorno o compilación. Este proceso se conoce como *etiquetar*.

Subversion no tiene comandos especiales para hacer ramas o etiquetas, pero en cambio utiliza lo que se denomina *copias baratas*. Las copias baratas son similares a los vínculos duros de Unix, que significa que en vez de hacer una copia completa en el repositorio, se crea un vínculo interno, apuntando a una revisión y árbol específicos. Como resultado, las ramas y las etiquetas son muy rápidas de crear, y casi no conllevan espacio extra en el repositorio.

5.16.1. Crando una rama o etiqueta

Si ha importado su proyecto con la estructura de directorios recomendados, crear una rama o una etiqueta es muy simple:

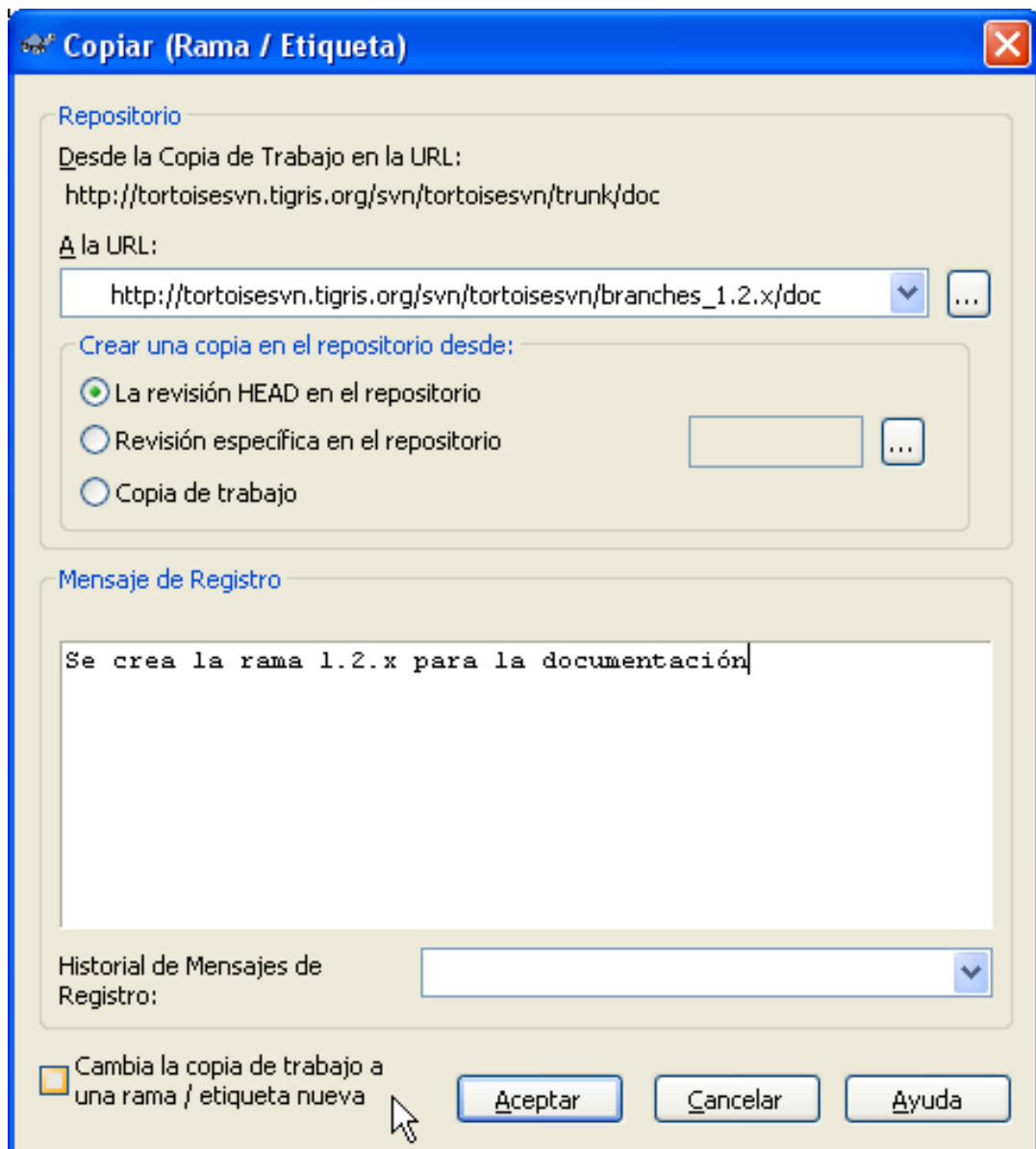


Figura 5.30. El diálogo Rama/Etiqueta

Seleccione la carpeta en su copia de trabajo de la que desea hacer una rama o una etiqueta, y luego seleccione el comando TortoiseSVN → Rama/Etiqueta....

La URL de destino por defecto para la nueva rama será la URL de origen en la que se basa su copia de trabajo. Necesitará editar esa URL con la nueva ruta para su rama/etiqueta. Así que en vez de

```
http://svn.collab.net/repos/NombreDelProyecto/trunk
```

querrá ahora usar algo como

```
http://svn.collab.net/repos/NombreDelProyecto/tags/Version_1.10
```

Si no puede recordar la convención de nombres que usó la última vez, pulse el botón a la derecha para abrir el navegador de repositorios, para que pueda ver la estructura actual del repositorio.

Ahora debe elegir el origen de la copia. Aquí tiene tres opciones:

Revisión HEAD en el repositorio

La nueva rama se copia directamente en el repositorio desde la revisión HEAD. No se necesita transferir datos desde su copia de trabajo, y la rama se crea muy rápidamente.

Revisión específica en el repositorio

La nueva rama se copia directamente en el repositorio, pero puede elegir una versión anterior. Esto es útil si se olvidó de crear una etiqueta cuando lanzó una versión de su proyecto la semana pasada. Si no puede acordarse del número de revisión, pulse el botón a la derecha para mostrar el registro de revisiones, y seleccione el número de revisión desde allí. De nuevo no se transfiere datos desde su copia de trabajo, y la rama se crea muy rápidamente.

Copia de trabajo

La nueva rama es una copia idéntica de su copia de trabajo local. Si ha cambiado algunos ficheros a una revisión anterior en su copia de trabajo, o si ha hecho cambios locales, esto es exactamente lo que irá a la copia. Naturalmente, esta clase de etiquetado complejo conlleva transferir datos desde su copia de trabajo al repositorio si no existe ya allí.

Si desea que su copia de trabajo se cambie automáticamente a la rama recién creada, utilice la casilla **Cambiar la copia de trabajo a la nueva rama/etiqueta**. Pero si lo hace, asegúrese primero de que su copia de trabajo no contenga modificaciones. Si las tiene, esos cambios se mezclarán en la copia de trabajo de la rama cuando se haga el cambio.

Pulse **Aceptar** para confirmar la nueva copia al repositorio. No se olvide de proporcionar un mensaje de registro. Tenga en cuenta que la copia se crea *dentro del repositorio*.

Tenga en cuenta que crear una Etiqueta o una Rama *no* afecta a su copia de trabajo. Incluso si copia su copia de trabajo, estos cambios se confirmarán en la rama nueva, no en el tronco, así que su copia de trabajo todavía se marcará como modificada.

5.16.2. Obtener o cambiar...

...esa (realmente no) es la cuestión. Mientras que obtener obtiene todo de la rama elegida a su directorio de trabajo, TortoiseSVN → **Cambiar...** sólo transfiere los datos cambiados a su copia de trabajo. Bueno para la carga de la red, bueno para su paciencia. :-)

Para poder trabajar con su copia recientemente generada tiene varias formas de manejarlo. Puede:

- TortoiseSVN → **Obtener** para obtener una copia nueva en una carpeta vacía. Puede obtener la copia de trabajo en cualquier parte de su disco duro, y puede crear tantas copias de trabajo de su repositorio como desee.
- Cambiar su copia de trabajo actual a la copia recién creada en el repositorio. De nuevo seleccione la carpeta superior de su proyecto y utilice TortoiseSVN → **Cambiar...** del menú contextual.

En el siguiente diálogo, introduzca la URL de la rama que acaba de crear. Deje la opción **Revisión HEAD** activada y pulse **Aceptar**. Su copia de trabajo se cambia a la nueva rama/etiqueta.

Cambiar trabaja igual que Actualizar, en el sentido de que nunca pierde sus cambios locales. Cualquier cambio que haya hecho a su copia de trabajo que todavía no se hayan confirmado se fusionarán cuando haga el Cambio. Si no desea que esto ocurra, entonces debe o bien confirmar los cambios antes de cambiar, o revertir su copia de trabajo a una revisión ya-confirmada (típicamente HEAD).

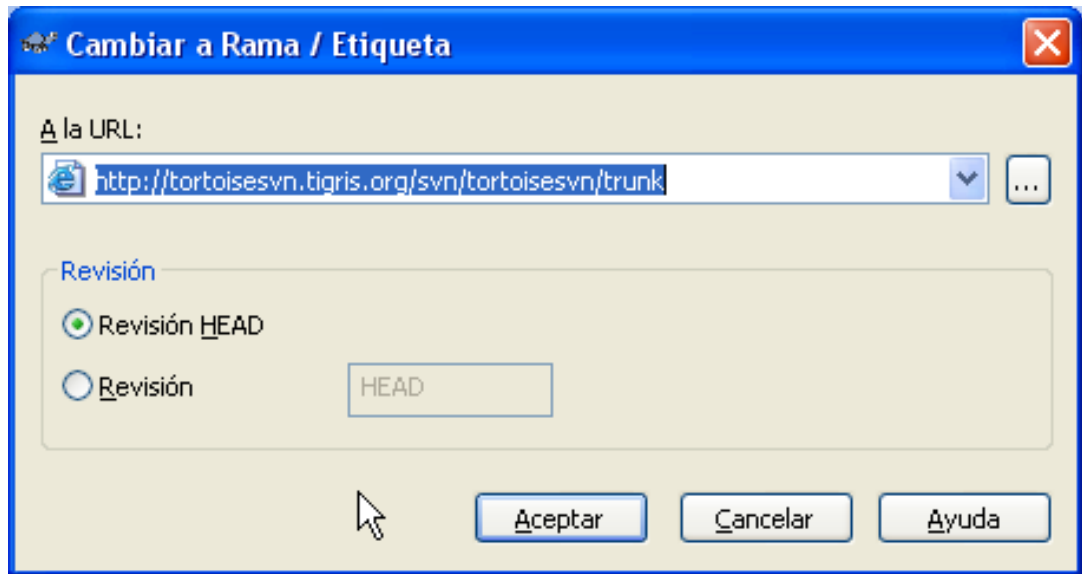


Figura 5.31. El diálogo Cambiar

Aunque Subversion por sí mismo no hace ninguna distinción entre etiquetas y ramas, la forma en la que normalmente se usan difiere un poco.

- Las etiquetas se usan típicamente para crear una copia estática de un proyecto en una etapa concreta. Como tales normalmente no se utilizan para el desarrollo - eso es para lo que se utilizan las ramas, y por esa razón recomendamos la estructura del repositorio `/trunk /branches /tags` en primer lugar. Trabajando en una revisión etiquetada *no es una buena idea*, pero dado que sus ficheros locales no están protegidos, no hay nada que le impida hacer esto por error. Sin embargo, si intenta confirmar a una ruta en el repositorio que contenga `/tags/` (en inglés), TortoiseSVN le avisará.
- Puede ser que necesite hacer más cambios a una versión que ya había etiquetado. La forma correcta de manejar esta situación es crear primero una nueva rama desde la etiqueta. Haga sus cambios en esta rama, y luego cree una nueva etiqueta para esta rama, por ejemplo `Version_1.0.1`.
- Si modifica una copia de trabajo creada desde una rama y confirma, entonces los cambios irán a la nueva rama y *no* en el tronco. Sólo se guardan las modificaciones. El resto continúa siendo una copia barata.

5.17. Fusionando

Mientras que las ramas se utilizan para mantener líneas de desarrollo separadas, en alguna etapa tendrá que fusionar los cambios hechos en una rama de vuelta en el tronco, o viceversa.

Es importante entender cómo funcionan las ramas y la mezcla en Subversion antes de empezar a utilizarlos, porque puede convertirse en algo bastante complejo. Le recomendamos encarecidamente que lea el capítulo *Branching and Merging* [<http://svnbook.red-bean.com/en/1.1/ch04.html>] en el *Libro de Subversion* [<http://svnbook.red-bean.com/>] que le proporciona una descripción completa, y muchos ejemplos de cómo se utiliza.

Un punto importante a recordar es que Fusionar está muy relacionado con Diferenciar. El proceso de fusión trabaja generando una lista de diferencias entre dos puntos del repositorio, y aplicando esas diferencias a su copia de trabajo. Por ejemplo, si desea fusionar los cambios hechos en la revisión N, entonces tendrá que comparar la revisión N con la revisión (N-1). Los novatos a menudo preguntan “Por qué tengo que restar 1 a la revisión de inicio.” Recuerde el proceso de Diferenciación que hay

por debajo y se hará más claro. Para hacerlo más sencillo, cuando utilice **Mostrar Registro** para seleccionar un rango de revisiones a fusionar, TortoiseSVN hará el ajuste automáticamente por usted.

En general es una buena idea realizar fusiones en una copia de trabajo sin modificar. Si ha hecho otros cambios en su copia de trabajo, confírmelos primero. Si la fusión no funciona como espera, puede querer revertir los cambios, y el comando **Revertir** descartará *todos* los cambios, incluidos cualquiera que haya hecho antes de la fusión.

Hay dos casos de uso comunes para fusionar que se manejan de formas ligeramente diferentes, como se describen a continuación.

5.17.1. Fusionando un rango de revisiones

Este método cubre el caso en el que ha hecho una o más revisiones a una rama (o al tronco) y desea portar estos cambios a una rama diferente.

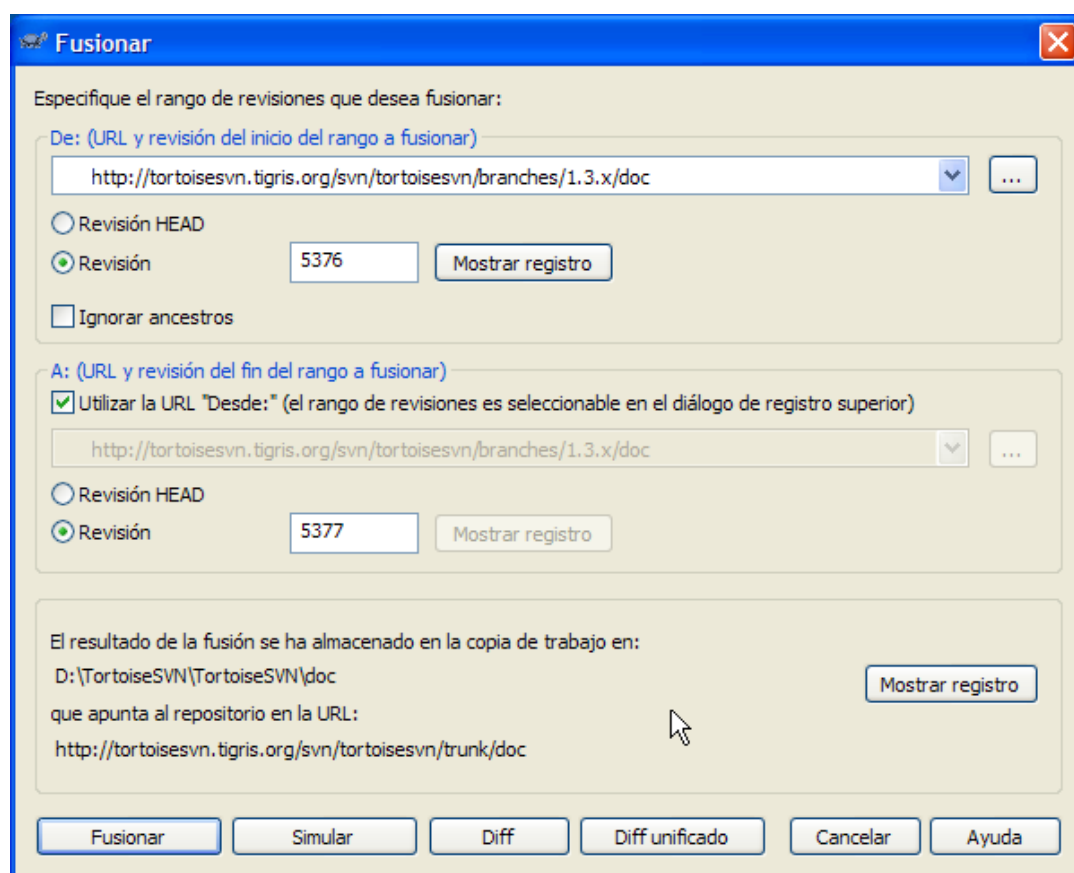


Figura 5.32. El diálogo Fusionar

Para fusionar revisiones necesita ir a una copia de trabajo de la rama en la que desee recibir los cambios, a menudo el tronco. Seleccione **TortoiseSVN → Fusionar...** desde el menú contextual.

1. En el campo **Desde:** introduzca la URL completa de la carpeta de la rama o la etiqueta que contiene los cambios que desea portar a su copia de trabajo. También puede pulsar ... para navegar el repositorio y encontrar la rama deseada. Si ha fusionado desde esta rama antes, entonces utilice la lista desplegable que le muestra una historia de las URLs utilizadas anteriormente.
2. Dado que está portando un rango de revisiones desde la misma rama en su copia de trabajo, asegúrese de que la casilla **Usar URL "Desde:"** está marcada.

3. En el campo **Desde la Revisión** introduzca el número de revisión inicial. Esta es la revisión *anterior* a los cambios que desea fusionar. Recuerde que Subversion creará un fichero de diferencias para realizar la fusión, por lo que el punto de inicio debe estar justo antes del primer cambio en el que está interesado. Por ejemplo, sus mensajes de registro pueden ser parecidos a estos:

```
Rev Comentarios
39. Trabajando en MiRama
38. Trabajando en trunk
37. Trabajando en MiRama
36. Crear rama MiRama
35. Trabajando en trunk
34. Trabajando en trunk
...
```

Si ahora desea fusionar todos los cambios de MiRama en el tronco, deberá elegir 36 como valor de **Desde la Revisión**, no 37 como podría pensar. Si selecciona la revisión 37 como punto de inicio, entonces el motor de diferencias compara el punto final con la revisión 37, y se perderá los cambios que hizo en la propia revisión 37. Si esto suena complicado, no se preocupe, hay una forma más fácil en TortoiseSVN...

La manera más sencilla de seleccionar el rango de revisiones que necesita es pulsar en **Mostrar Registro**, dado que esto mostrará los cambios recientes con sus comentarios de registro. Si desea fusionar los cambios de una única revisión, seleccione esa revisión. Si desea fusionar los cambios de varias revisiones, entonces seleccione ese rango (utilizando el modificador habitual **Mayúsculas**). Pulse **Aceptar** y *los dos* números de revisión **Desde la Revisión** y **Hasta la Revisión** se rellenarán por usted en el diálogo **Fusionar**.

Cuando se marca la casilla **Utilizar URL "Desde:"**, sólo se habilita un botón **Mostrar registro**. Esto es así porque el diálogo **Mostrar Registro** establece las dos revisiones **Desde:** y **A:**, por lo que necesita utilizar el método múltiple de selección descrito anteriormente.

Si ya ha fusionado algunos cambios desde esta rama, esperemos que haya apuntado la última revisión fusionada en el mensaje de registro cuando confirmó ese cambio. En ese caso, puede utilizar **Mostrar Registro** en la Copia de Trabajo para buscar ese mensaje de registro. Utilice el punto final de la última fusión como punto de inicio de esta fusión. Por ejemplo, si la última vez fusionó las revisiones de la 37 a la 39, entonces el punto de inicio de esta fusión sería la revisión 39.

4. Si no ha utilizado **Mostrar Registro** para seleccionar el rango de revisiones, necesitará establecer el campo **Hasta la Revisión** manualmente. Introduzca el número de la última revisión del rango que desea fusionar. A menudo ésta será la revisión **HEAD**, aunque no necesita serlo - puede querer fusionar únicamente una única revisión.

Si hay otras personas que puedan estar confirmando cambios, entonces tenga cuidado al utilizar la revisión **HEAD**. Puede que no se refiera a la revisión que usted cree si alguien más ha hecho una confirmación después de su última actualización.

5. Pulse **Aceptar** para completar la fusión.

La fusión ya está completa. Es una buena idea observar la fusión y ver si está como se esperaba. Las fusiones son normalmente bastante complicadas. A menudo aparecen conflictos si la rama se ha desviado mucho del tronco.

Cuando haya comprobado los cambios y vaya a confirmar esta revisión, su mensaje de registro debería incluir *siempre* los números de revisión que han sido portados en la fusión. Si desea aplicar otra fusión más tarde, necesitará saber lo que ya ha fusionado, porque no querrá portar un cambio más de una vez. Desafortunadamente, Subversion no almacena la información de fusiones. Para más información sobre esto, consulte *Tracking Merges Manually* [<http://svnbook.red-bean.com/en/1.1/ch04s03.html#svn-ch-4-sect-3.2.1>] en el *Libro de Subversion* [<http://svnbook.red-bean.com/>]

El manejo de las ramas es importante. Si desea mantener esta rama actualizada respecto al tronco,

debería asegurarse de fusionar a menudo para que la rama y el tronco no se alejen mucho. Por supuesto, sigue debiendo evitar fusiones repetidas de cambios, como se explicó anteriormente.



Importante

Subversion no puede fusionar un fichero con una carpeta, y viceversa - sólo carpetas con carpetas y ficheros con ficheros. Si pulsa en un fichero y abre el diálogo de fusión, entonces deberá darle una ruta a un fichero en ese diálogo. Si selecciona una carpeta y saca el diálogo, debe especificar una URL de una carpeta para la fusión.

5.17.2. Fusionando dos árboles diferentes

Este método cubre el caso cuando ha realizado una rama de funcionalidad como se discute en el libro de Subversion. Todos los cambios del tronco han sido portados a la rama de la funcionalidad, semana a semana, y ahora que la funcionalidad está completa desea fusionarla de nuevo en el tronco. Dado que ha mantenido la rama de la funcionalidad sincronizada con el tronco, las últimas versiones de la rama y del tronco serán absolutamente iguales excepto por sus cambios en la rama. En este caso especial, fusionaría comparando la rama con el tronco.

Para fusionar la rama de la característica en el tronco necesita ir a una copia de trabajo del tronco. Seleccione TortoiseSVN → Fusionar.. desde el menú contextual.

1. En el campo **Desde**: introduzca la URL completa de la carpeta del *tronco* (trunk). Esto puede sonar erróneo, pero recuerde que el tronco es el punto de inicio al que desea añadir los cambios de la rama. También puede pulsar ... para navegar el repositorio.
2. Dado que está comparando dos árboles diferentes, asegúrese de que la casilla **Usar la URL "Desde:"** no está marcada.
3. En el campo **Hasta**: introduzca la URL completa de la carpeta con la rama de la característica.
4. En los dos campos **Desde la Revisión** y **Hasta la Revisión**, introduzca el último número de revisión en el que se sincronizaron los dos árboles. Si está seguro de que nadie más está haciendo confirmaciones puede utilizar HEAD en ambos casos. Si hay alguna posibilidad de que alguien haya hecho una confirmación desde esa sincronización, utilice el número de revisión específico para evitar perder cambios más recientes.

También puede utilizar **Mostrar Registro** para seleccionar la revisión. Tenga en cuenta que en este caso no está seleccionando un rango de revisiones, por lo que la revisión que seleccione es la que aparecerá realmente en el campo **Revisión**.

5. Pulse **Aceptar** para completar la fusión.

En este caso no necesitará la rama de la característica de nuevo porque la característica ya está fusionada en el tronco. La rama de la característica es redundante y puede ser borrada del repositorio si se necesita.

5.17.3. Previsualizando los resultados de la fusión

Si no está seguro sobre la operación de fusión, puede querer previsualizar qué ocurrirá antes de que le permita cambiar su copia de trabajo. Hay tres botones adicionales para ayudarle.

Diff Unificado crea el fichero de diferencias (recuerde que la fusión se basa en la diferenciación) y le muestra qué líneas se cambiarán en los ficheros de su copia de trabajo. Este fichero de diferencias unificado (fichero de parche) no siempre es fácil de leer fuera de contexto, pero para cambios a pequeña escala a menudo es útil porque le muestra todos los cambios de golpe.

Diff le muestra una lista de ficheros cambiados. Haga doble click en cualquiera de los ficheros listados para iniciar el visor de diferencias. Al contrario que el diff unificado, esto le muestra los cam-

bios con su detalle contextual completo. Al igual que el diff unificado, los cambios que ve aquí son los cambios entre las revisiones Desde: y A:. No le muestra cómo afectará ese cambio a su copia de trabajo.

Simular realiza la operación de fusión, pero *no* modifica la copia de trabajo en absoluto. Le muestra una lista de ficheros que se fusionarán con una fusión real, y especifica esas áreas donde ocurrirán conflictos.

5.17.4. Ignorando a los ancestros

La mayoría del tiempo desea fusionar para tener en cuenta el historial de un fichero, para que se fusionen los cambios relativos a un ancestro en común. A veces puede necesitar fusionar ficheros que quizás estén relacionados, pero no en su repositorio. Por ejemplo, puede haber importado las versiones 1 y 2 de una biblioteca de terceros en dos directorios separados. Aunque están relacionados de forma lógica, Subversion no tiene conocimiento de ello porque sólo ve los ficheros que ha importado. Si intenta fusionar las diferencias entre estos dos ficheros verá un borrado completo seguido de un añadir completo. Para hacer que Subversion sólo utilice las diferencias basadas-en-ruta en vez de diferencias basadas-en-historial, seleccione la casilla **Ignorar ancestros**. Puede leer más sobre este tema en el libro de Subversion; *Noticing or Ignoring Ancestry* [<http://svnbook.red-bean.com/en/1.1/ch04s03.html#svn-ch-4-sect-3.3.4>]

5.18. Bloqueando

Subversion generalmente trabaja mejor sin bloqueos, utilizando los métodos “Copiar-Modificar-Fusionar” que se describieron anteriormente en el Sección 2.2.3, “La solución copiar-modificar-fusionar”. Sin embargo, hay algunas pocas situaciones en las que puede querer implementar alguna forma de política de bloqueo.

- Está utilizando ficheros “no fusionables”, por ejemplo, ficheros de imagen. Si dos personas cambian el mismo fichero, la fusión no es posible, así que alguno de ellos perderá sus cambios.
- Su compañía ha utilizado en el pasado un SCV bloqueante, y la dirección ha decidido que “bloquear es lo mejor”.

Primero necesita asegurarse de que su servidor de Subversion está actualizado al menos a la versión 1.2. Las versiones anteriores no tienen ningún soporte de bloqueos. Si está usando el acceso file:/// , sólo necesita actualizar sus clientes, por supuesto.

5.18.1. Cómo trabaja el bloqueo en Subversion

Por defecto, nada se bloquea y todo el mundo que tiene acceso de confirmación puede confirmar cambios a cualquier fichero en cualquier momento. Los demás actualizarán sus copias de trabajo periódicamente y los cambios en el repositorio se fusionarán con los cambios locales.

Si *Obtiene un Bloqueo* en un fichero, entonces sólo usted puede confirmar ese fichero. Las confirmaciones de los demás se bloquearán hasta que quite el bloqueo. Un fichero bloqueado no puede ser modificado de ninguna forma en el repositorio, por lo que no puede ser siquiera borrado o renombrado, excepto por el dueño del bloqueo.

Sin embargo, los demás usuarios no necesariamente saben que usted ha obtenido un bloqueo. A menos que ellos comprueben el estado de bloqueo regularmente, la primera vez que sabrán sobre él es cuando sus confirmaciones fallen, lo que muchas veces no es muy útil. Para hacer más fácil el manejo de bloqueos, hay una nueva propiedad de Subversion `svn:needs-lock`. Cuando se establece esta propiedad (a cualquier valor) en un fichero, siempre que el fichero se obtiene o actualiza, la copia local se deja como sólo-lectura *a menos* que la copia de trabajo tenga un bloqueo para ese fichero. Esto actúa como una advertencia de que no debería modificar el fichero a menos que obtenga un bloqueo. Los ficheros que están versionados y marcados como sólo-lectura se marcan con un icono de sobreimpresión especial en TortoiseSVN para indicar que necesita obtener un bloqueo antes de editar.

Los bloqueos se graban con el lugar de la copia local y también con el propietario. Si tiene varias

copias de trabajo (en casa, en el trabajo) entonces sólo puede obtener un bloqueo en *una* de esas copias de trabajo.

Si uno de sus compañeros de trabajo obtiene un bloqueo y luego se va de vacaciones sin quitarlo, ¿qué hace? Subversion proporciona un método para forzar bloqueos. Quitar un bloqueo que tiene otra persona se define como *Romper* el bloqueo, y obtener forzosamente un bloqueo que tiene otro se define como *Robar* el bloqueo. Naturalmente estas no son cosas que hacer a la ligera si desea continuar siendo amigo de sus compañeros de trabajo.

Los bloqueos se guardan en el repositorio, y se crea un token de bloqueo en su copia local de trabajo. Si hay una discrepancia, por ejemplo si alguien ha roto el bloqueo, el token local se convierte en inválido. El repositorio siempre es la referencia definitiva.

5.18.2. Obteniendo un bloqueo

Seleccione el fichero o ficheros en su copia de trabajo para los que desee obtener un bloqueo, y seleccione el comando TortoiseSVN → Obtener Bloqueo....

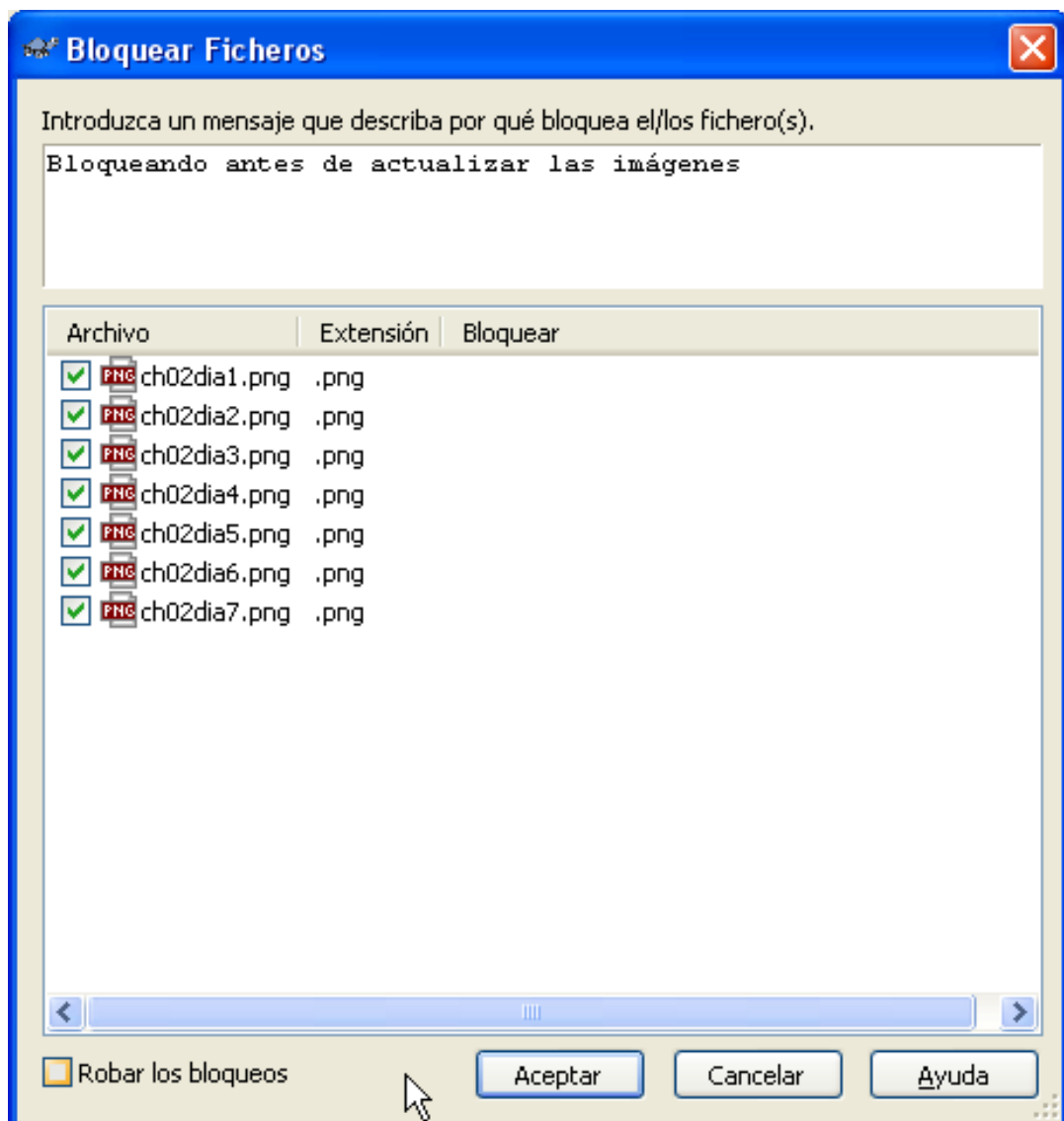


Figura 5.33. El diálogo Bloquear

Aparece un diálogo, que le permite introducir un comentario, para que los demás vean por qué ha bloqueado el fichero. El comentario es opcional, y de momento sólo se utiliza con repositorios basa-

dos en Svnserve. Si (y *sólo* si) necesita robar el bloqueo de alguien, marque la casilla **Robar bloqueo**, y luego pulse **Aceptar**.

Si selecciona una carpeta y luego utiliza TortoiseSVN → **Obtener Bloqueo...** se abrirá el diálogo de bloquear con *todos* los ficheros en *todas* las subcarpetas seleccionados para bloquearlos. Si realmente quiere bloquear una jerarquía completa, ésta es la forma de conseguirlo, pero puede convertirse en persona no grata para sus compañeros de trabajo si les bloquea todo el proyecto. Utilícelo con cuidado...

5.18.3. Quitando un Bloqueo

Para asegurarse de que no se olvida de quitar un bloqueo que no necesita más, los ficheros bloqueados se muestran en el diálogo de confirmar y se seleccionan por defecto. Si continúa con la confirmación, los bloqueos que tenga sobre los ficheros seleccionados se quitan, incluso si los ficheros no se han modificado. Si no desea quitar el bloqueo en algunos ficheros, puede desmarcarlos (si no están modificados). Si desea mantener los bloqueos en un fichero que ha modificado, tiene que habilitar la casilla **Mantener bloqueos** antes de confirmar sus cambios.

Para quitar un bloqueo manualmente, seleccione el fichero o ficheros de su copia de trabajo para los que desee quitar el bloqueo, y luego seleccione el comando TortoiseSVN → **Quitar Bloqueo**. No hay nada más que introducir por lo que TortoiseSVN contactará con el repositorio y quita los bloqueos. También puede utilizar este comando en una carpeta para quitar todos los bloqueos recursivamente.

5.18.4. Comprobando el estado de los bloqueos

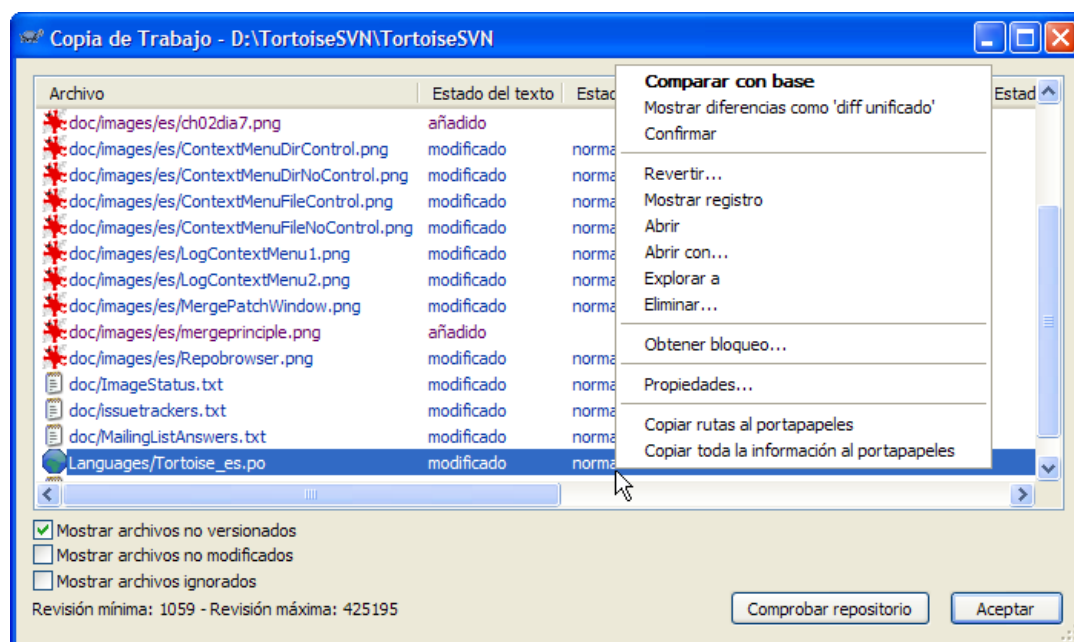


Figura 5.34. El diálogo Comprobar modificaciones

Para ver qué bloqueos tiene usted y los de los demás, puede utilizar el comando TortoiseSVN → **Comprobar Modificaciones...** Los tokens de bloqueos que se tienen localmente se muestran inmediatamente. Para comprobar los bloqueos que tienen los demás (y para ver si alguno de sus bloqueos se han roto o han sido robados) necesita pulsar **Comprobar Repositorio**.

Desde el menú contextual aquí, también puede obtener y quitar bloqueos, además de romper y robar bloqueos que tienen otros.



Evite romper y robar bloqueos

Si ha roto o robado el bloqueo de otro sin decírselo, puede causar una pérdida potencial de trabajo. Si está trabajando con tipos de ficheros no fusionables y roba el bloqueo de otro, una vez que usted quite el bloqueo ellos son libres de confirmar sus cambios y sobrescribir los suyos. Subversion no pierde datos, pero ha perdido la protección para trabajo en equipo que el bloqueo le proporcionaba.

5.18.5. Haciendo ficheros no-bloqueados como sólo-lectura

Como se menciona arriba, la manera más efectiva de utilizar los bloqueos es establecer la propiedad `svn:needs-lock` en los ficheros. Consulte Sección 5.15, “Configuración del proyecto” para obtener instrucciones sobre cómo establecer propiedades. Los ficheros con esta propiedad establecida siempre se obtendrán y actualizarán con la marca de sólo-lectura a menos que su copia de trabajo tenga un bloqueo.



Como recordatorio, TortoiseSVN utiliza un icono de sobreimpresión especial para indicarlo.

Si opera una política donde todos los ficheros han de ser bloqueados, puede que encuentre más fácil utilizar la característica de Subversion auto-props para establecer la propiedad automáticamente cada vez que añada nuevos ficheros. Para más información, lea Sección 5.26, “Configuración de TortoiseSVN”.

5.18.6. Los scripts ganchos de bloqueo

Cuando crea un nuevo repositorio con Subversion 1.2 o superior, se crean cuatro plantillas en el directorio `hooks` del repositorio. Éstos se llaman antes y después de obtener un bloqueo, y antes y después de quitar un bloqueo.

Es una buena idea instalar un script gancho `post-lock` y `post-unlock` en el servidor que envíe un email indicando el fichero que se ha bloqueado. Poniendo este script en su sitio, todos sus usuarios serán informados si alguien bloquea/quita el bloqueo de un fichero. Puede encontrar un script gancho de ejemplo `hooks/post-lock.tmpl` en su carpeta del repositorio.

También puede utilizar ganchos para no permitir romper o robar bloqueos, o quizás para limitarlo a un administrador designado para ello. O quizás desea enviar un email al propietario cuando uno de sus bloqueos se haya roto o robado.

Para más información, lea la Sección 4.3, “Scripts gancho”.

5.19. Creando y aplicando parches

En proyectos de código abierto (como éste), todos tienen acceso de lectura al repositorio, y cualquiera puede hacer una contribución al proyecto. ¿Así que cómo se controlan esas contribuciones? Si cualquiera pudiera confirmar cambios, el proyecto estaría permanentemente inestable y probablemente roto de forma permanente. En esta situación, el cambio se maneja mediante un fichero de *parche* enviado al equipo de desarrollo, que tienen acceso de escritura. Ellos pueden revisar el parche antes, y luego o bien confirmarlo en el repositorio o devolvérselo al autor.

Los ficheros de parche son simplemente ficheros de diff unificados que muestran las diferencias entre su copia de trabajo y la revisión base.

5.19.1. Creando un fichero parche

Primero necesita hacer y *probar* sus cambios. Luego en vez de utilizar el comando TortoiseSVN → Confirmar... sobre la carpeta padre, seleccione TortoiseSVN → Crear Parche...

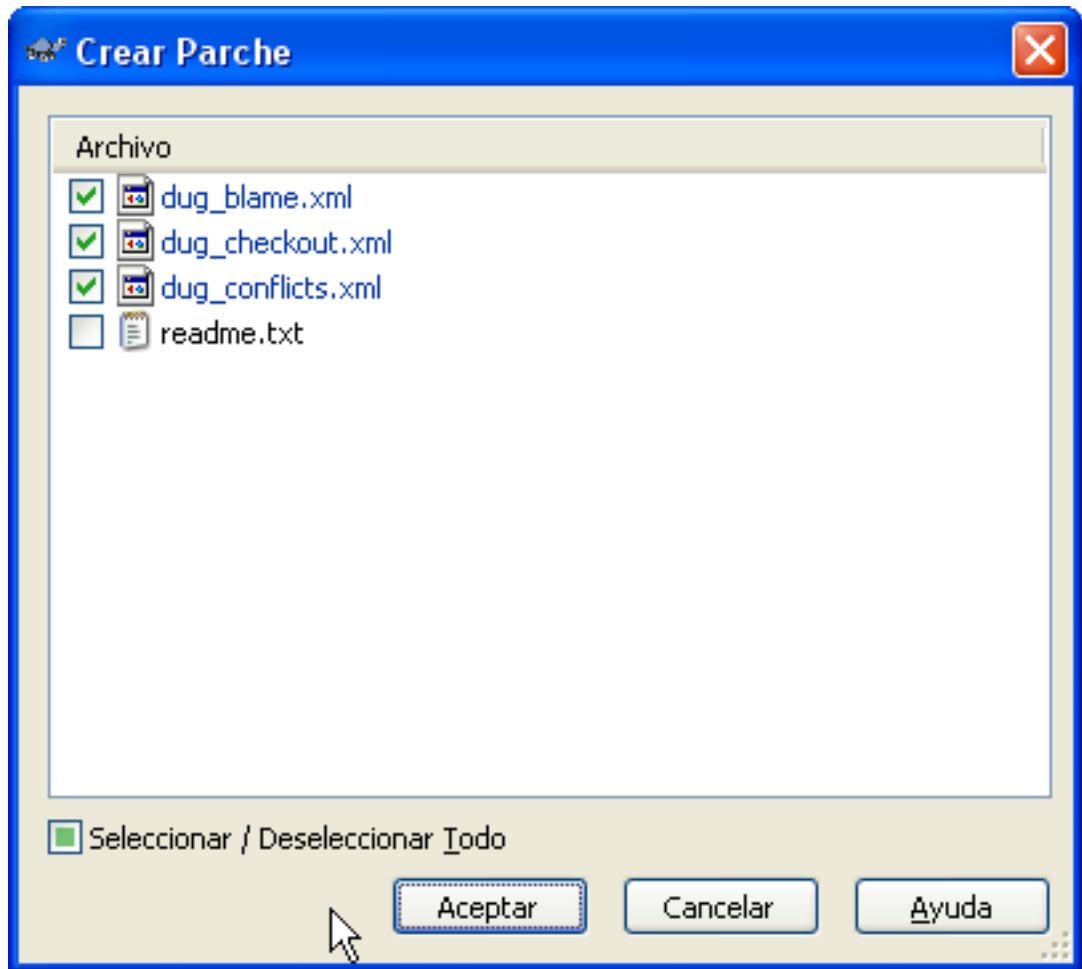


Figura 5.35. El diálogo Crear parche

ahora puede seleccionar los ficheros que desea incluir en el parche, del mismo modo a como lo haría con una confirmación completa. Esto producirá un único fichero que contendrá un resumen de todos los cambios que ha hecho a los ficheros seleccionados desde la última vez que se actualizó desde el repositorio.

Las columnas en este diálogo pueden ser personalizadas de la misma forma que las columnas en el diálogo Comprobar modificaciones. Para más detalles, lea Sección 5.7.3, “Estado local y remoto”.

Puede producir parches separados que contengan cambios a diferentes conjuntos de ficheros. Por supuesto, si crea un fichero de parche, hace algunos cambios más a los *mismos* ficheros y luego crea otro parche, el segundo fichero de parche incluirá *ambos* conjuntos de cambios.

Simplemente grabe el fichero utilizando un nombre de fichero de su elección. Los ficheros de parche pueden tener la extensión que desee, pero por convención se suele utilizar las extensiones `.patch` o `.diff` extension. Ya está preparado para enviar su fichero de parches.

5.19.2. Aplicando un fichero parche

Los ficheros de parches se aplican en su copia de trabajo. Esto debe hacerse desde el mismo nivel de carpetas que se utilizó para crear el parche. Si no está seguro de cuál es, mire la primera línea del fichero de parche. Por ejemplo, si el primer fichero en el que se trabajó era `doc/source/english/chapter1.xml` y la primera línea en el fichero de parche es `Index: english/chapter1.xml` entonces necesita aplicar el parche en la carpeta `english`. Sin embargo, suponiendo que esté en la copia de trabajo correcta, si ha seleccionado un nivel de carpeta erró-

neo, TSVN se dará cuenta y le sugerirá el nivel correcto.

Para aplicar un fichero de parche en su copia de trabajo, necesita al menos acceso de lectura al repositorio. La razón para esto es que el programa de fusión debe referenciar los cambios sobre la revisión contra la que se hicieron por el desarrollador remoto.

Desde el menú contextual de esa carpeta, pulse en TortoiseSVN → Aplicar Parche... Esto mostrará un diálogo de abrir fichero que le permitirá seleccionar el fichero de parche a aplicar. Por defecto sólo aparecen los ficheros `.patch` o `.diff`, pero puede elegir "Todos los ficheros".

Alternativamente, si el fichero de parche tiene una extensión `.patch` o `.diff`, puede hacer click con el botón derecho en él directamente y seleccionar TortoiseSVN → Aplicar Parche.... En este caso se le preguntará la ruta de la copia de trabajo.

Estos dos métodos le ofrecen formas diferentes de hacer lo mismo. Con el primer método selecciona la copia de trabajo y busca el fichero de parche. Con el segundo selecciona el fichero de parche y busca la copia de trabajo.

Una vez que haya seleccionado el fichero de parche y la ruta de la copia de trabajo, se ejecuta TortoiseMerge para fusionar los cambios del fichero de parche contra su copia de trabajo. Una pequeña ventana le muestra los ficheros que han sido cambiados. Haga doble click por turnos en cada uno de ellos, compruebe los cambios y grabe los ficheros fusionados.

El parche del desarrollador remoto ya ha sido aplicado a su copia de trabajo, por lo que necesita confirmar para que todos los demás accedan a los cambios desde el repositorio.

5.20. ¿Quién cambió qué línea?

A veces necesita saber no sólo qué líneas han cambiado, sino también exactamente quién cambió líneas específicas en un fichero. Entonces es cuando el comando TortoiseSVN → Autoría..., a veces conocido como comando de *anotar*, tiene su utilidad.

Este comando muestra, por cada línea en un fichero, su autor y la revisión en la que se cambió la línea.

5.20.1. Autoría de ficheros

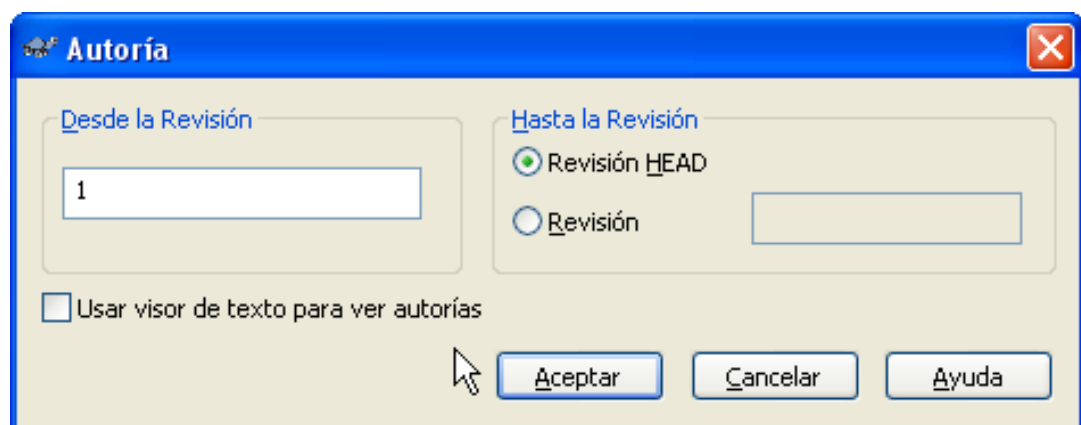


Figura 5.36. El diálogo Anotar / Autoría

Si no está interesado en cambios de revisiones anteriores puede establecer la revisión desde la cual debe empezar la autoría. Ponga este valor a 1 si desea ver la autoría de *cada* revisión.

Por defecto, el fichero de autoría se ve utilizando *TortoiseBlame*, que remarca las diferentes revisiones para hacerlas más fáciles de leer. Si desea imprimir o editar el fichero de autoría, seleccione Utilizar visor de texto para ver autorías

Una vez que pulse **Aceptar**, TortoiseSVN empieza a recoger la información para crear el fichero de autoría. Tenga ésto en cuenta: esto puede llevar varios minutos para completarse, dependiendo en cuánto haya cambiado el fichero y por supuesto de su conexión de red con el repositorio. Una vez que el proceso de autoría ha terminado, el resultado se escribe en un fichero temporal y puede ver los resultados.

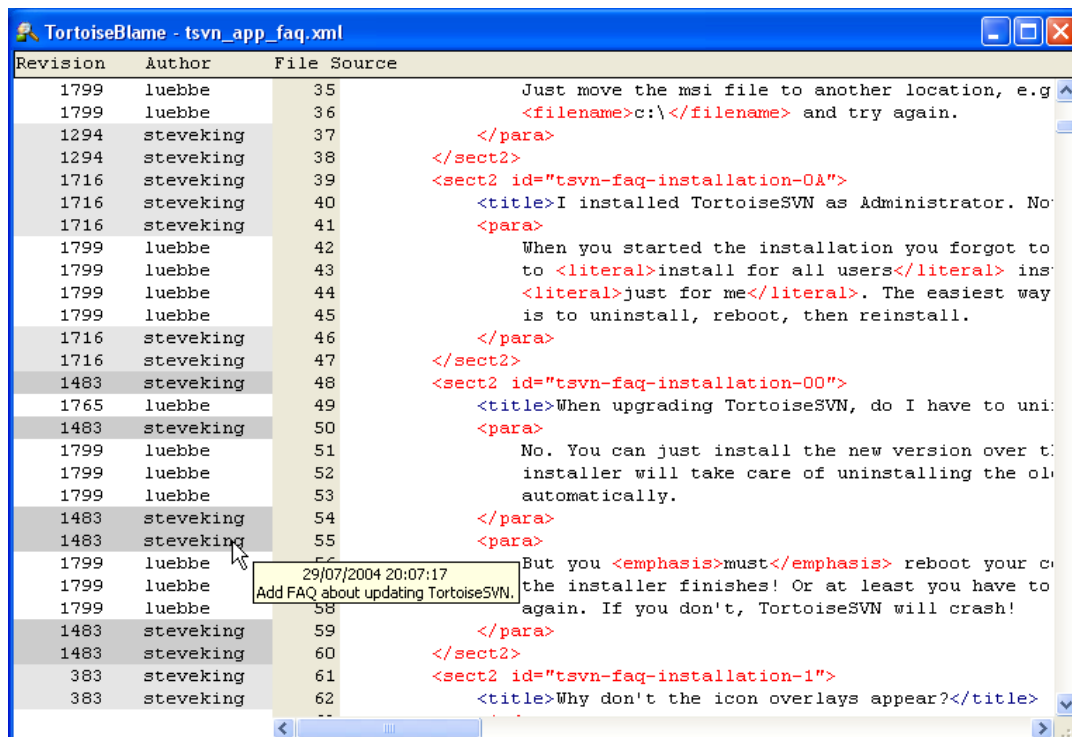


Figura 5.37. TortoiseBlame

TortoiseBlame, que se incluye con TortoiseSVN, hace más fáciles de leer los ficheros de autoría. Cuando pasa el ratón por encima de una línea en la columna de información de autoría, todas las líneas con la misma revisión se señalan con un fondo más oscuro. Las líneas de otras revisiones que fueron cambiadas por el mismo autor se señalan con un fondo claro. Los colores pueden no funcionar de forma muy eficiente si su pantalla está en el modo de 256 colores.

Si hace click en una línea, todas las líneas con la misma revisión se señalan, y las líneas de otras revisiones del mismo autor se señalan en un color más claro. Este señalado se mantiene, permitiéndole mover el ratón sin perder los señalados. Pulse en esa revisión de nuevo para desactivar el señalado.

Los comentarios de la revisión se muestran en una caja de ayuda cuando mueve el cursor encima de la columna de información de autoría.

Puede buscar dentro del informe de Autoría utilizando Edición → Buscar.... Esto le permite buscar por números de revisión, autores y por el contenido del fichero en si mismo. Los mensajes de registro no se incluyen en la búsqueda - deberá utilizar el diálogo de Mostrar Registro para buscar en ellos.

5.20.2. Autoría de las diferencias

Una de las limitaciones del informe de Autoría es que sólo muestra el fichero tal y como estaba en una revisión en concreto, y muestra la última persona que cambió cada línea. A veces querrá saber qué cambio se hizo, y también quién lo hizo. Lo que necesita aquí es una combinación de los informes de diferencias y autoría.

El diálogo del historial de revisiones incluye varias opciones que le permiten hacerlo.

Autoría de las revisiones

En el panel superior, seleccione dos revisiones, y luego seleccione **Menú Contextual → Ver autoría de las revisiones**. Esto obtendrá los datos de autoría de las dos revisiones, y luego utilizará el visor de diferencias para comparar los dos ficheros de autoría.

Autoría de las diferencias

Seleccione una revisión en el panel superior, luego seleccione un fichero en el panel inferior y seleccione **Menú contextual → Ver autoría de las diferencias**. Esto obtendrá los datos de autoría de la revisión seleccionada y la anterior, y utilizará el visor de diferencias para comparar los dos ficheros de autoría.

Comparar y ver autoría con la BASE de trabajo

Muestre el registro de un único fichero, y en el panel superior, seleccione una única revisión, y luego seleccione **Menú contextual → Comparar y ver autoría con la BASE de trabajo**. Esto obtendrá los datos de autoría de la revisión seleccionada, y también del fichero en la BASE de trabajo, y utilizará el visor de diferencias para comparar los dos ficheros de autoría.

5.21. El navegador de repositorios

A veces necesitará trabajar directamente en el repositorio, sin tener una copia de trabajo. Ésa es la razón por la que existe el *Navegador de Repositorios*. Igual que el Explorador y los iconos sobreimpresionados le permiten ver su copia de trabajo, el navegador de repositorios le permite ver la estructura y el estado del repositorio.

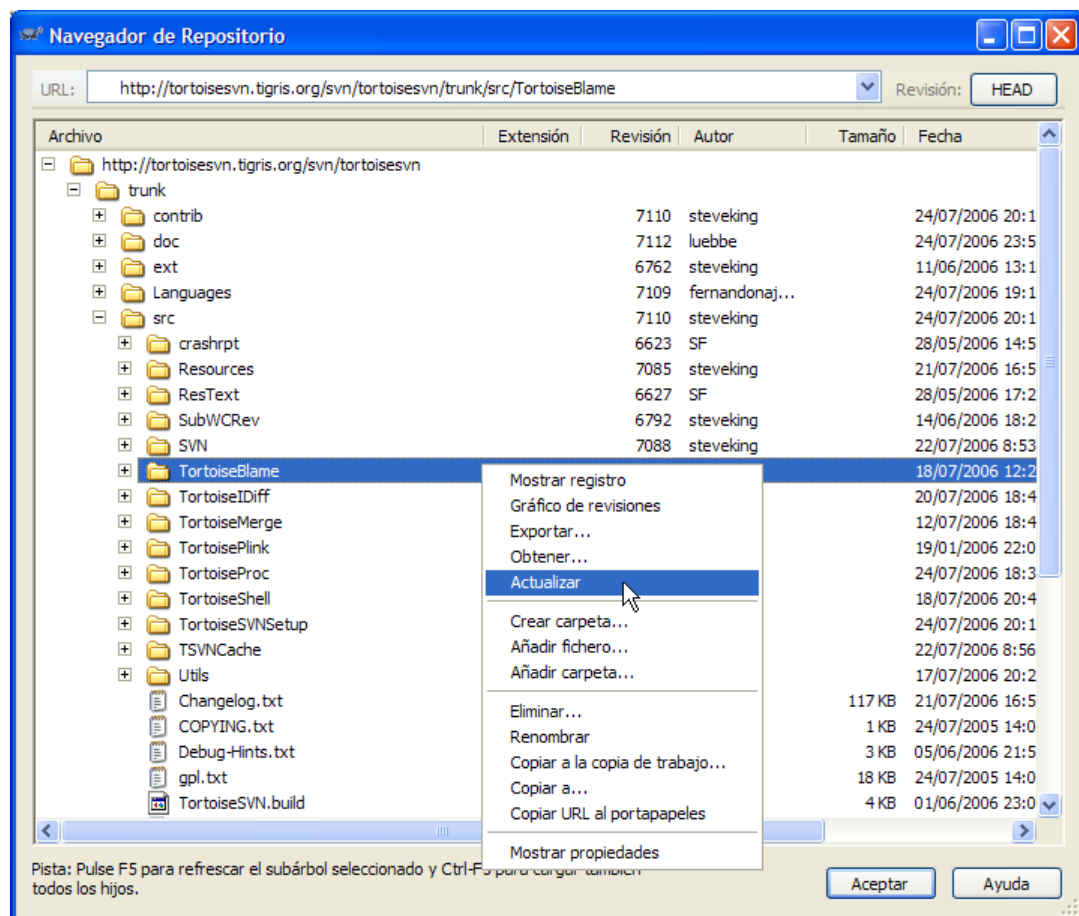


Figura 5.38. El navegador de repositorios

Con el navegador de repositorios puede ejecutar comandos como copiar, mover, renombrar... direc-

tamente en el repositorio.

En la parte superior de la Ventana del Navegador de Repositorios puede introducir la URL del repositorio y la revisión a la que desea navegar. Navegar a una versión más antigua es útil si por ejemplo desea recuperar un fichero borrado anteriormente. Use el comando **Menú Contextual → Copiar a...** para hacerlo e introduzca la ruta completa de la copia de trabajo donde desea recuperar su fichero borrado.

La ventana principal del navegador se parece mucho a cualquier otro navegador en árbol. Puede pulsar en las cabeceras para cambiar el orden. Si hace click con el botón derecho en la cabecera **Fichero** puede activar o desactivar la **Ordenación Numérica**. Esto proporciona un manejo más inteligente de números que el que proporciona la ordenación ASCII, y es útil para obtener etiquetas de números de versión en el orden correcto. El estado por defecto de esta opción puede cambiarse en el diálogo de configuración.

Si selecciona dos ítems, puede ver las diferencias bien como diff unificado, o bien como una lista de ficheros que pueden diferenciarse visualmente utilizando la herramienta de diferencias por defecto. Esto puede ser particularmente útil para comparar dos etiquetas y ver qué ha cambiado.

Si selecciona dos etiquetas que se han copiado desde la misma raíz (típicamente, `/trunk/`), puede utilizar **Menú Contextual → Mostrar Registro...** para ver la lista de revisiones entre los dos puntos etiquetados.

También puede utilizar el navegador de repositorios para operaciones de arrastrar-y-soltar. Si arrastra una carpeta desde el Explorador al navegador de repositorios, se importará esa carpeta en el repositorio. Tenga en cuenta que si arrastra múltiples ítems, se importarán en confirmaciones separadas.

Si desea mover un ítem en el repositorio, simplemente arrastre con el botón izquierdo el ítem a su nueva localización. Si desea crear una copia en vez de mover el ítem, arrastre con el botón izquierdo manteniendo la tecla Control pulsada. Cuando copia, el cursor tiene un símbolo “más” en él, tal y como lo hace en el Explorador.

Si desea copiar/mover un fichero o una carpeta a otro lugar y darle también otro nombre a la vez, puede arrastrar con el botón derecho o arrastrar con el botón derecho pulsando Control el ítem en vez de arrastrarlo con el botón izquierdo. En ese caso, aparece un diálogo para renombrar en el que puede introducir un nuevo nombre para el fichero o la carpeta.

Siempre que haga cambios en el repositorio utilizando uno de estos métodos, se le presentará un diálogo para introducir un mensaje de registro. Si ha arrastrado algo por error, esta es su oportunidad para cancelar la acción.

A veces cuando intenta abrir una ruta obtendrá un mensaje de error en vez de los detalles del ítem. Esto puede ocurrir si ha especificado una URL inválida, o si no tiene permisos de acceso, o si hay algún otro problema en el servidor. Si necesita copiar este mensaje para incluirlo en un email, simplemente haga click con el botón derecho sobre él y utilice **Menú Contextual → Copiar mensaje de error al portapapeles**, o simplemente utilice CTRL+C.

5.22. Gráficos de revisión

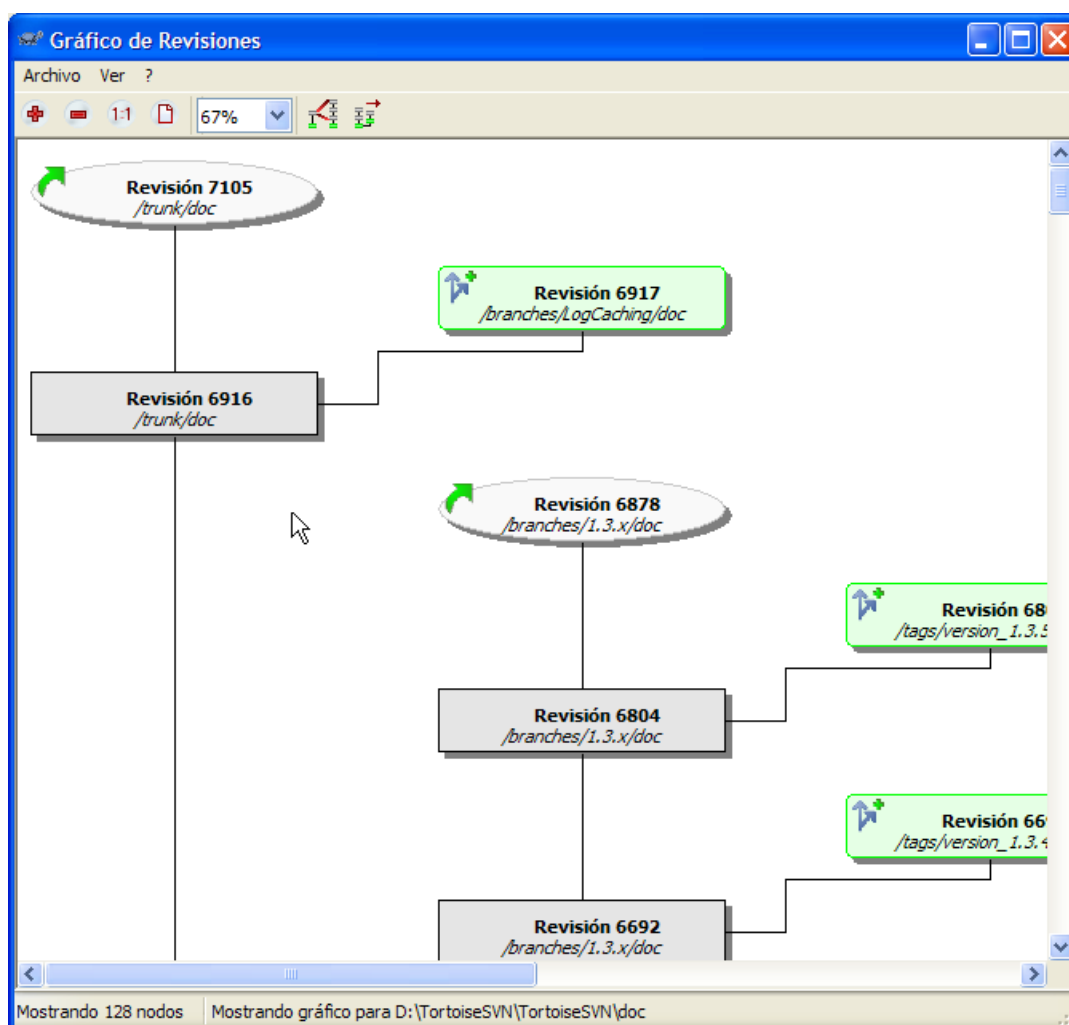


Figura 5.39. Un gráfico de revisiones

A veces necesita saber desde qué revisión del tronco se tomaron las ramas y las etiquetas, y la forma ideal de ver este tipo de información es en un gráfico o una estructura de árbol. Ahí es cuando necesita utilizar TortoiseSVN → Gráfico de Revisiones...

Este comando analiza la historia de las revisiones e intenta crear un árbol mostrando los puntos en los que se tomaron las copias, y cuando se borraron las ramas/etiquetas.



Importante

Para generar el gráfico, TortoiseSVN debe obtener todos los mensajes de registro desde la raíz del repositorio. No es necesario decir que esto puede llevar varios minutos incluso con un repositorio con unos pocos miles de revisiones, dependiendo de la velocidad del servidor, el ancho de la red, etc. Si intenta esto con algo como el proyecto Apache, que actualmente tiene más de 300.000 revisiones, puede estar esperando un buen rato.

El gráfico de revisiones muestra varios tipos de nodos:

Fichero/carpeta añadida

Los ítems que se han añadido, o se crearon copiándolos de otros ficheros/carpetas se muestran

utilizando un rectángulo redondeado.

Fichero/carpeta eliminada

Los ítems borrados, por ejemplo una rama que no se necesita más, se muestran utilizando un octágono (rectángulo con las esquinas cortadas).

Revisión punta de la rama

Si una rama (o tronco o etiqueta) ha sido modificado desde el último nodo de rama, este hecho se muestra utilizando una elipse. Esto significa que en el gráfico siempre se muestra la última revisión de cada rama.

Fichero/carpeta normal

Todos los demás ítems se muestran utilizando un rectángulo normal.

Tenga en cuenta que por defecto el gráfico sólo muestra los puntos en los que se añadieron o borraron ítems. Mostrar cada revisión de un proyecto podría generar un gráfico enorme para los casos no triviales. Si realmente desea ver *todas* las revisiones en las que se hicieron cambios, hay una opción para hacerlo en el menú **Ver** y en la barra de herramientas.

También hay una opción para organizar el gráfico *por ruta*. Esto intenta ordenar las ramas desde las etiquetas. Las rutas que no contienen modificaciones después de la copia se suponen que son etiquetas y se aplian en una única columna. Las ramas (que contienen modificaciones tras su creación) tienen cada una su propia columna, para que pueda ver cómo se desarrolla la rama.

La fecha de la revisión, el autor y los comentarios se muestran en una caja de ayuda cuando se mueve el ratón encima de una caja de revisión.

Si selecciona dos revisiones (utilice Ctrl + click con el botón izquierdo), puede utilizar el menú contextual para mostrar las diferencias entre esas revisiones. Puede elegir mostrar las diferencias en los puntos de creación de las ramas, pero normalmente querrá mostrar las diferencias en los puntos finales de las ramas, esto es, en la revisión HEAD.

Puede ver las diferencias como un fichero de Diff Unificado, que le muestra todas las diferencias en un fichero único con contexto mínimo. Si opta por **Menú Contextual → Comparar Revisiones** se le presentará una lista de ficheros cambiados. Haga doble click en un nombre de fichero para obtener ambas revisiones y compararlas utilizando una herramienta de diferencias visual.

Si hace click con el botón derecho en una revisión, puede utilizar **Menú Contextual → Mostrar registro** para ver la historia.



Atención

Dado que Subversion no puede proporcionar toda la información necesaria, se necesita una cierta cantidad de interpretación, que a veces puede dar resultados extraños. En cualquier caso, la salida para el tronco generalmente da resultados útiles.

5.23. Exportando una copia de trabajo de Subversion

A veces querrá una copia de su árbol de trabajo sin ninguno de esos directorios `.svn`, por ejemplo para crear un fichero comprimido de sus fuentes, o para exportarlo a un servidor web. En vez de hacer una copia y borrar todos esos directorios `.svn` manualmente, TortoiseSVN ofrece el comando **TortoiseSVN → Exportar...**. Si ejecuta este comando en su copia de trabajo, se le preguntará por el lugar donde guardar la copia de trabajo *limpia* sin las carpetas `.svn`. Por defecto, sólo se exportan los ficheros versionados, pero puede utilizar la casilla **Exportar también los ficheros no versionados** para incluir cualquier otro fichero sin versionar que exista en su copia de trabajo y no en el repositorio. Las referencias externas definidas por `svn:externals` se pueden omitir si se necesita.

Otra forma de hacer una exportación es arrastrar con el botón derecho una carpeta con una copia de trabajo a otro lugar y elegir **Menú contextual → SVN Exportar aquí** o **Menú contextual → SVN Exportar todo aquí**. La segunda opción incluirá también los ficheros sin versionar.

Y por supuesto también puede exportar directamente desde el repositorio. Utilice el Navegador de repositorios para navegar al árbol relevante en su repositorio, y luego utilice Menú contextual → Exportar.

5.24. Relocalizando una copia de trabajo

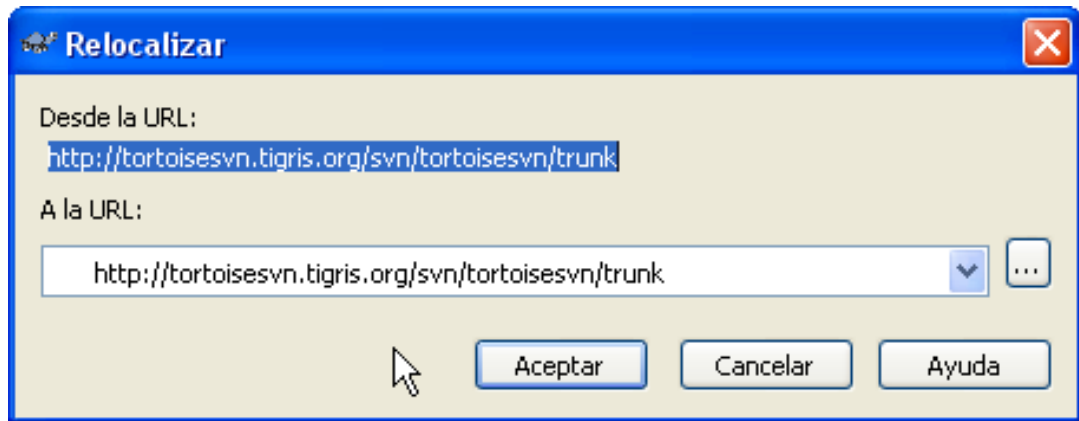


Figura 5.40. El diálogo Relocalizar

Si su repositorio ha cambiado por algún motivo de lugar (IP/URL); o quizás está estancado y no puede confirmar y no quiere obtener de nuevo su copia de trabajo de la nueva localización y mover todos sus datos cambiados a la copia de trabajo nueva: el comando TortoiseSVN → Relocalizar es lo que está buscando. Básicamente hace muy poco: escanea todos los ficheros "entries" en las carpetas .svn y cambia la URL de las entradas al nuevo valor.



Aviso

Esta es una operación que se utiliza realmente poco. El comando relocalizar se utiliza sólo si la URL de la raíz del repositorio ha cambiado. Estas son algunas posibles razones:

- La dirección IP del servidor ha cambiado.
- El protocolo ha cambiado (por ejemplo, de http:// a https://).
- La ruta raíz del repositorio ha cambiado en la configuración del servidor.

De otra forma, necesita relocalizar cuando su copia de trabajo se refiere al mismo lugar en el mismo repositorio, pero es el propio repositorio el que se ha movido.

Esto no se aplica si:

- Quiere moverse a un repositorio de Subversion diferente. En ese caso debería realizar una obtención limpia desde la nueva localización del repositorio.
- Quiere cambiar a una rama o a un directorio diferente dentro del mismo repositorio. Para hacer eso debería utilizar TortoiseSVN → Cambiar.... Para más información, lea Sección 5.16.2, "Obtener o cambiar...".

Si utiliza relocalizar en alguno de los casos anteriores, *corromperá su copia de trabajo* y obtendrá muchos mensajes de error inexplicables cuando actualice, confirme, etc. Una vez que ésto ha ocurrido, el único arreglo es hacer una obtención limpia.

5.25. Integración con sistemas de control de errores / seguimiento de incidencias

Es muy común en el Desarrollo de Software que los cambios se refieran a un ID de error o incidencia específico. A los usuarios de los sistemas de control de errores (seguimiento de incidencias) les gustaría asociar los cambios que hacen en Subversion con un ID específico en su programa de seguimiento de incidencias. La mayoría de programas de seguimiento de incidencias proporcionan un script gancho pre-commit que parsea el mensaje de log para encontrar el ID del error con el que se asocia la confirmación. Esto es de alguna forma propenso a errores, porque se basa en el que usuario escriba el mensaje de registro de forma correcta para que el script gancho pre-commit pueda parsearlo correctamente.

TortoiseSVN puede ayudar al usuario de dos formas:

1. Cuando el usuario introduce un mensaje de registro, puede añadirse una línea bien definida incluyendo el número de incidencia asociada a la confirmación. Esto reduce el riesgo de que el usuario introduzca el número de incidencia que las herramientas de control de errores no puedan parsear correctamente.

O TortoiseSVN puede remarcar la parte del mensaje de registro introducido que el programa de seguimiento de incidencias reconoce. De esta forma el usuario sabrá que el mensaje de registro introducido puede parsearse correctamente.

2. Cuando el usuario navega por los mensajes de registro, TortoiseSVN crea un enlace desde cada ID de error en el mensaje de registro, que lanza el navegador con la incidencia mencionada.

Puede integrar una Herramienta de Control de Errores de su elección con TortoiseSVN. Para hacerlo, necesita definir algunas propiedades, que empiezan por `bugtraq:`. Deben establecerse en Carpetas: (Sección 5.15, “Configuración del proyecto”)

Hay dos formas de integrar TortoiseSVN con los programas de seguimiento de incidencias. Una se basa en simples cadenas de texto, la otra se basa en expresiones regulares. Las propiedades que se utilizan en ambos casos son:

`bugtraq:url`

Set this property to the url of your bugtracking tool. It must be properly URI encoded and it has to contain `%BUGID%`. `%BUGID%` is replaced with the Issuenumber you entered. This allows TortoiseSVN to display a link in the log dialog, so when you are looking at the revision log you can jump directly to your bugtracking tool. You do not have to provide this property, but then TortoiseSVN shows only the issuenumber and not the link to it, e.g the TortoiseSVN project is using `http://issues.tortoisesvn.net/?do=details&id=%BUGID%`

`bugtraq:warnifnoissue`

Establezca esta propiedad a `true` si desea que TortoiseSVN le avise cuando deje en blanco el campo de texto del número de incidencia. Los valores válidos son `true/false`. Si no se define, se supone `false`.

En el caso simple, TortoiseSVN muestra al usuario un campo de entrada separado donde se puede introducir un ID del error. En ese caso se añade una línea adicional al final o al principio del mensaje de registro que el usuario introduce.

`bugtraq:message`

Esta propiedad activa el Sistema de Control de Errores en el modo campo de entrada. Si se establece esta propiedad, TortoiseSVN le preguntará por el número de incidencia cuando confirme sus cambios. Se utiliza para añadir una línea al final del mensaje de registro. Debe contener `%BUGID%`, que se reemplaza por el número de la incidencia al confirmar. Esto asegura que su registro de confirmación contiene una referencia al número de incidencia que siempre está en un formato consistente y que puede ser parseado por su herramienta de Control de Erro-

res para asociar el número de incidencia con una confirmación particular. Por ejemplo, el proyecto TortoiseSVN utiliza `Issue : %BUGID%`, pero esto depende de su herramienta.

bugtraq:append

Esta propiedad define si el ID del error se añade (`true`) al final del mensaje de registro o se inserta (`false`) al inicio del mensaje de registro. Los valores válidos son `true/false`. Si no se define, se asume `true`, para que los proyectos ya existentes no se rompan.

bugtraq:label

Éste es el texto que se muestra por TortoiseSVN en el diálogo de confirmar para etiquetar el cuadro de texto en el que introduce el número de incidencia. Si no se establece, aparecerá `Bug-ID / Issue-Nr:`. Tenga en cuenta, sin embargo, que la ventana no se cambiará de tamaño para acomodar esta etiqueta, por lo que debería mantener el tamaño de la etiqueta en menos de 20-25 caracteres.

bugtraq:number

Si se establece a `true` sólo se permitirán números en el campo de número de incidencia. La coma es una excepción, para que pueda introducir varios números separados por comas. Los valores válidos son `true/false`. Si no se define, se asumirá `true`.

En el caso de utilizar expresiones regulares, TortoiseSVN no muestra un campo de entrada separado sino que marca, del mensaje de registro que el usuario introduce, la parte que reconoce el programa de seguimiento de incidencias. Esto se realiza mientras el usuario introduce el mensaje de registro. ¡Esto también significa que el ID del error puede introducirse en cualquier parte dentro de un mensaje de registro! Este método es mucho más flexible, y es el que se utiliza en el propio proyecto TortoiseSVN.

bugtraq:logregex

Esta propiedad activa el Sistema de Control de Errores en modo Regex. Contiene una o dos expresiones regulares, separadas por un salto de línea. Si `s[li]incidencia(?:s)?#?(d+)` Si se especifican dos expresiones, la primera expresiSi desea encontrar todos los patrones "incidencia #XXX" e "incidencia #890, #789" dentro de un mensaje de registro, podr[li]incidencia #?(d+)(,? ?#(d+)) y la segunda expresi(d+) Si no esthttp://www.regular-expressions.info/.*

Si se establecen las dos propiedades `bugtraq:message` y `bugtraq:logregex`, `logregex` tiene preferencia.



Sugerencia

¡Incluso si no tiene un programa de seguimiento de incidencias con un gancho pre-commit parseando sus mensajes de log, puede seguir utilizando ésto para convertir las incidencias mencionadas en su mensajes de registro en vínculos!

Algunas propiedades `tsvn:` necesitan un valor `true/false` (verdad/mentira). TSVN también entiende `yes` (sí) como un sinónimo de `true` y `no` como un sinónimo de `false`.



Estableciendo las propiedades en las carpetas

Estas propiedades deben estar establecidas en carpetas para que el sistema funcione. Cuando confirma un fichero o una carpeta, se leen las propiedades de esa carpeta. Si no se encuentran allí las propiedades, TortoiseSVN las buscará hacia arriba en el árbol de carpetas para encontrarlas, hasta que llega a una carpeta sin versionar, o se encuentra la raíz del árbol (por ejemplo, `C:\`). Si puede estar seguro de que cada usuario obtiene sólo desde por ejemplo `trunk/` y no desde alguna subcarpeta, entonces es suficiente establecer las propiedades en `trunk/`. Si no puede estar seguro, debería establecer las propiedades recursivamente en cada subcarpeta. Una propiedad establecida hacia las profundidades de la jerarquía del proyecto tiene preferencia sobre las propie-

dades establecidas en niveles más altos (más cerca de `trunk/`).

Para las propiedades `tsvn`: sólo puede utilizar la casilla **Recurso** para establecer la propiedad en todas las subcarpetas de la jerarquía, sin establecera en todos los ficheros.

La integración del seguimiento de incidencias no se restringe a TortoiseSVN; puede utilizarse con cualquier cliente de Subversion. Para más información, lea toda la *Especificación de Integración con los Seguidores de Incidencias* [<http://tortoisesvn.tigris.org/svn/tortoisesvn/trunk/doc/issuetrackers.txt>].

5.26. Configuración de TortoiseSVN

Para averiguar para qué sirven cada una de las diferentes opciones, deje el puntero del ratón un segundo sobre el cuadro de texto/casilla... y aparecerá un útil mensaje de ayuda.

5.26.1. Configuración general

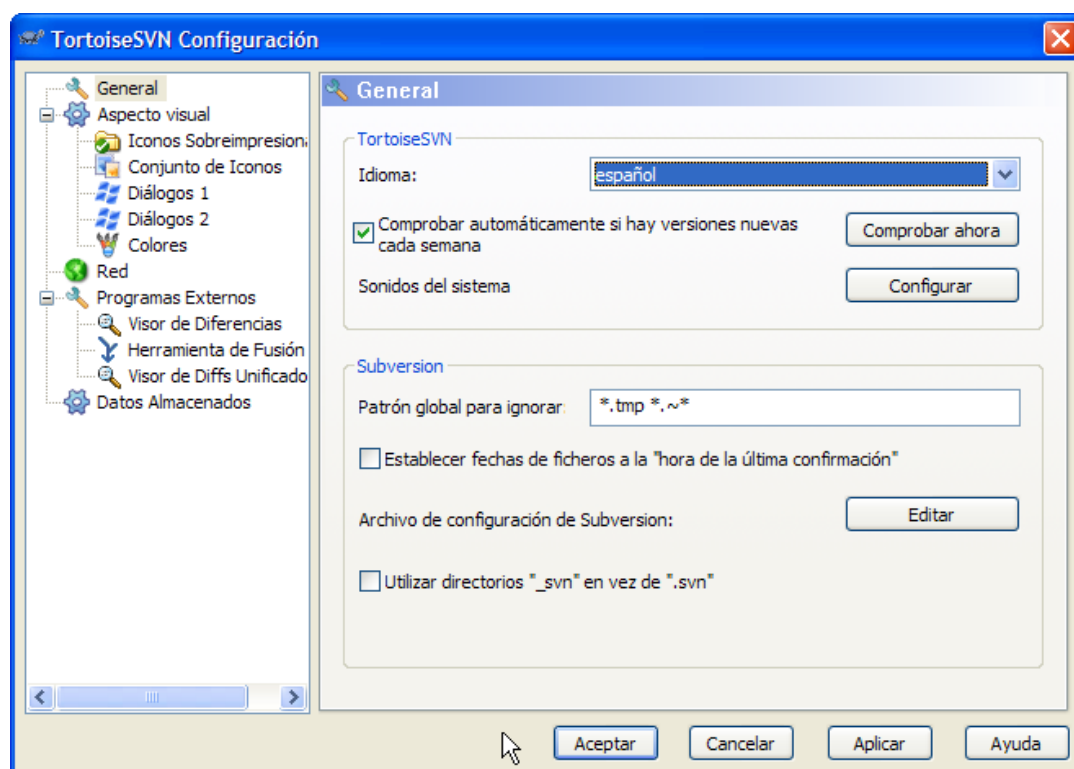


Figura 5.41. El diálogo de Configuración, página General

Este diálogo le permite especificar su idioma preferido, y las configuraciones específicas de Subversion.

Idioma

Selecciona el idioma del interfaz de usuario. ¿Qué otra cosa esperabas?

Comprobar automáticamente si hay nuevas versiones todas las semanas

Si se marca, TortoiseSVN contactará su sitio de descarga una vez a la semana para ver si hay disponible una versión más reciente del programa. Utilice el botón **Comprobar ahora** si desea una respuesta inmediata. No se descargará la versión más reciente; simplemente recibirá un diálogo de información diciéndole que hay una versión nueva disponible.

Sonidos del sistema

TortoiseSVN tiene tres sonidos personalizados que se instalan por defecto.

- Error
- Información
- Atención

Puede seleccionar sonidos diferentes (o quitarlos completamente) utilizando el Panel de Control de Windows. **Configurar** es un acceso directo al Panel de Control.

Patrón global de ignorar

Los patrones globales de ignorar se utilizan para evitar que aparezcan ficheros no versionados, por ejemplo, en el diálogo de confirmación. Los ficheros que concuerden con los patrones también se ignoran en las importaciones. Ignore ficheros o directorios escribiendo sus nombres o extensiones. Los patrones se separan por espacios, por ejemplo, `*/bin */obj *.bak *.~?? *.jar *. [Tt]mp`. Las dos primeras entradas se refieren a directorios, las otras cuatro a ficheros. Estos patrones utilizan expansión de comodines en los nombres de los ficheros. Para más información, lea Sección 5.11.1, "Expansión de comodines en los nombres de los ficheros de la lista de ignorados".

Tenga en cuenta que los patrones de ignorar que especifique aquí también afectarán a otros clientes de Subversion que se ejecuten en su PC, incluyendo el cliente de línea de comandos.



Atención

Si utiliza el fichero de configuración de Subversion para establecer un patrón `global-ignores`, prevalecerá sobre las configuraciones que haga aquí. El fichero de configuración de Subversion se accede utilizando **Editar** como se describe a continuación.

Este patrón de ignorar afectará a todos sus proyectos. No se versiona, por lo que no afectará a otros usuarios. En el lado opuesto, puede utilizar también la propiedad versionada `svn:ignore` para excluir ficheros o directorios del control de versiones. Para más información, lea Sección 5.11, "Ignorando ficheros y directorios".

Establecer las fechas de los ficheros a la "hora de la última confirmación"

Esta opción le indica a TortoiseSVN que establezca las fechas de los ficheros al momento de la última confirmación cuando se hace una obtención o una actualización. Si no, TortoiseSVN utiliza la fecha actual. Si está desarrollando software generalmente es mejor que utilice la fecha actual porque los sistemas de compilación generalmente se fija en las fechas de los ficheros para determinar qué ficheros necesitan compilarse. Si utiliza "hora de la última confirmación" y revierte a una revisión anterior del fichero, su proyecto puede que no se compile como lo esperaba.

Fichero de configuración de Subversion

Utilice **Editar** para editar el fichero de configuración de Subversion directamente. Algunas opciones no pueden ser modificadas directamente por TortoiseSVN, y en cambio necesitan establecerse aquí. Para más información sobre el fichero `config` de Subversion, vea el *Manual de Subversion* [<http://svnbook.red-bean.com/en/1.1/ch07.html#svn-ch-7-sect-1.3.2>]. La sección sobre *Configuración Automática de Propiedades* [<http://svnbook.red-bean.com/en/1.1/ch07s02.html#svn-ch-7-sect-2.4>] es especialmente interesante, y eso es lo que se configura aquí. Tenga en cuenta que Subversion puede leer la información de configuración desde varios lugares, y necesita saber cuál tiene prioridad. Para obtener más información sobre esto, lea *Configuración y el Registro de Windows* [<http://svnbook.red-bean.com/en/1.1/ch07.html#svn-ch-7-sect-1.2>].

Utilizar directorios "_svn" en vez de ".svn"

VS.NET cuando se utiliza con proyectos web no puede manejar las carpetas `.svn` que Subversion utiliza para almacenar su información interna. Esto no es un error de Subversion. El error

está en VS.NET y las extensiones de Frontpage que utiliza. Lea Sección 5.26.7, “Carpetas de trabajo de Subversion” para averiguar más sobre este problema.

Si desea cambiar el comportamiento de Subversion y TortoiseSVN, puede utilizar esta casilla para establecer la variable de entorno que lo controla.

Debería tener en cuenta que cambiar esta opción no convertirá automáticamente sus copias de trabajo para que usen ese nuevo directorio administrativo. Deberá hacerlo utilizando un script (vea nuestro FAQ) o simplemente obteniendo una copia de trabajo nueva.

5.26.2. Configuración del aspecto visual

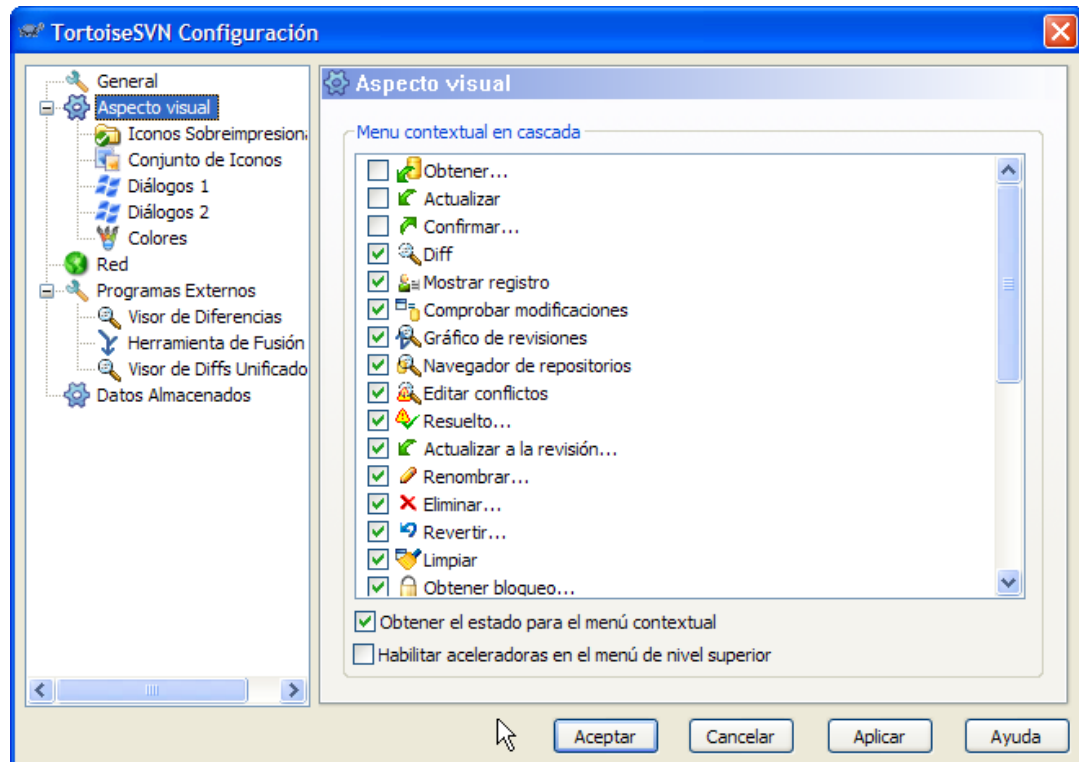


Figura 5.42. El diálogo Configuración, página de Aspecto visual

Esta página le permite especificar qué entradas del menú contextual de TortoiseSVN se mostrarán en el menú contextual principal, y cuales aparecerán en el submenú TortoiseSVN. Por defecto, la mayoría de los ítems están marcados y aparecen en el submenú.

Si tiene un número muy elevado de ficheros en sus carpetas de copias de trabajo, puede llevarle mucho tiempo al menú contextual para aparecer desde que hace click con el botón derecho en una carpeta. Esto ocurre porque Subversion obtiene el estado de todos los ficheros cuando se le pide el estado de una carpeta. Para evitar este retraso puede desmarcar la casilla **Obtener el estado para el menú contextual**. Sin embargo, deberá tener en cuenta que el menú contextual de las carpetas no siempre será correcto, y que puede incluir ítems que no deberían estar allí realmente. Por ejemplo, verá TortoiseSVN → **Mostrar Registro** para una carpeta **Añadida**, lo que no funcionará porque la carpeta aún no está en el repositorio.

La opción **Habilitar aceleradoras en el menú de nivel superior** tiene tres estados:

Desmarcado (por defecto)

En este estado los ítems del menú los dibuja TortoiseSVN. No se muestran teclas aceleradoras.

Marcado

Esto activa las teclas aceleradoras para los comandos de TortoiseSVN, pero por supuesto puede haber conflictos con las aceleradoras de cualquier otro programa en el menú contextual del Explorador. Si presiona la tecla del atajo varias veces, se moverá entre los diferentes ítems del menú contextual que comparten el atajo. En este estado, los ítems del menú se pintan por Windows, lo que hace que los iconos no se vean muy bien.

Indeterminado

En este modo las teclas aceleradoras están activas, y los ítems del menú se dibujan en modo sólo texto sin iconos.

5.26.2.1. Configuración de los iconos sobreimpresionados

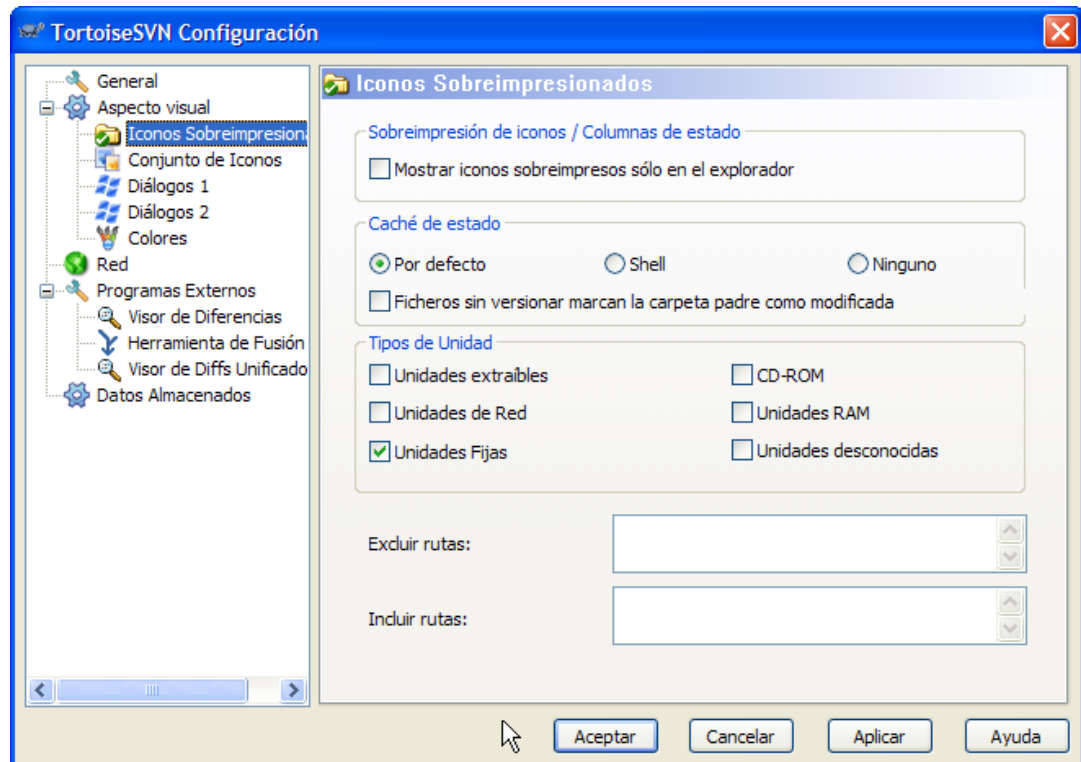


Figura 5.43. El diálogo Configuración, página de Aspecto visual

Esta página le permite elegir los ítems para los que TortoiseSVN mostrará iconos sobreimpresionados. Las unidades de red pueden ser muy lentas, por lo que por defecto los iconos no se muestran en las copias de trabajo creadas en unidades de red. Puede incluso deshabilitar todos los iconos sobreimpresionados, pero ¿cuál es la gracia?

Las unidades USB Flash parecen ser un caso especial en el que el tipo de unidad lo identifica el propio dispositivo. Algunas aparecen como discos duros, y otros como discos extraíbles.

Por defecto, los iconos sobreimpresionados aparecerán tanto en todos los diálogos abrir/guardar como en el Explorador de Windows. Si desea que aparezcan *únicamente* en el Explorador de Windows, marque **Mostrar sobreimpresiones sólo en el Explorador** box.

Dado que lleva bastante tiempo obtener el estado de una copia de trabajo, TortoiseSVN utiliza una caché para almacenar el estado de forma que el explorador no se vea demasiado atosigado cuando se muestren las sobreimpresiones. Puede elegir qué tipo de caché deberá utilizar TortoiseSVN de acuerdo a sus sistema y al tamaño de las copias de trabajo aquí:

Por defecto

Cachea toda la información de estado en un proceso separado (TSVNCache.exe). Ese proce-

so observa todas las unidades en busca de cambios y obtiene el estado de nuevo si se modifican ficheros dentro de una copia de trabajo. El proceso se ejecuta con la menor prioridad posible para que el rendimiento de otros programas no se resientan por su culpa. Eso también significa que la información de estado no está en *tiempo real* sino que puede llevar unos pocos segundos hasta que las sobreimpresiones cambien.

Ventaja: las sobreimpresiones le muestran el estado de forma recursiva, es decir, si un fichero muy dentro de su copia de trabajo se modifica, todas las carpetas padre hasta la raíz de su copia de trabajo mostrarán la sobreimpresión de modificación. Y dado que el proceso puede enviar notificaciones al shell, las sobreimpresiones del árbol de la izquierda usualmente también cambiarán.

Desventaja: el proceso se ejecuta constantemente, incluso si no está trabajando en sus proyectos. Además utiliza alrededor de 10-50 MB de RAM dependiendo del número y tamaño de sus copias de trabajo.

Shell

El cacheo se realiza directamente dentro de la dll de la extensión del shell, pero sólo para la carpeta actualmente visible. Cada vez que navega a otra carpeta, se obtiene de nuevo la información de estado.

Ventaja: sólo necesita muy poca memoria (alrededor de 1 MB de RAM) y puede mostrar el estado en *tiempo real*.

Desventaja: Dado que sólo se hace caché de una carpeta, las sobreimpresiones no muestran el estado recursivamente. Para copias de trabajo grandes, mostrar una carpeta en el explorador puede llevar más tiempo que con la caché por defecto. Además, la columna tipo-mime no está disponible.

Ninguno

Con esta configuración, TortoiseSVN no obtiene ningún estado en el Explorador. Por esa causa, los ficheros no tendrán sobreimpresiones y las carpetas sólo tendrán una sobreimpresión 'normal' si están versionadas. No se mostrará ninguna otra sobreimpresión, y tampoco estarán disponibles las columnas extra.

Ventaja: no utiliza memoria adicional en absoluto y no ralentiza de ninguna forma el Explorador mientras se navega.

Desventaja: La información de estado de los ficheros y carpetas no se muestra en el Explorador. Para ver si sus copias de trabajo están modificadas, tendrá que utilizar el diálogo "Comprobar modificaciones".

Si selecciona la opción por defecto, también puede elegir marcar las carpetas como modificadas si contienen ítems sin versionar. Esto puede ser útil para recordarle que ha creado nuevos ficheros que aún no están versionados.

Excluir Rutas se usa para decirle a TortoiseSVN en qué rutas *no* debe mostrar iconos sobreimpresionados ni columnas de estado. Esto es útil si tiene algunas copias de trabajo muy grandes que contienen únicamente librerías que no cambian en absoluto y por tanto no necesitan las sobreimpresiones. Por ejemplo:

`f:\desarrollo\SVN\Subversion` desactivará las sobreimpresiones *sólo* en esa ruta concreta. Todavía podrá ver las sobreimpresiones en todos los ficheros y carpetas dentro de esa carpeta.

`f:\desarrollo\SVN\Subversion*` deshabilitará las sobreimpresiones en *todos* los ficheros y carpetas cuya ruta empiece por `f:\desarrollo\SVN\Subversion`. Esto significa que no verá sobreimpresiones para ningún fichero y carpetas debajo de esa ruta.

Lo mismo se aplica a **Incluir rutas**. Excepto que para esas rutas las sobreimpresiones se muestran incluso si están deshabilitadas para ese tipo de unidad en concreto, o por una ruta de exclusión especificada más arriba.

TSVNCache.exe también utiliza estas rutas para restringir su escaneo. Si desea que sólo mire en de-

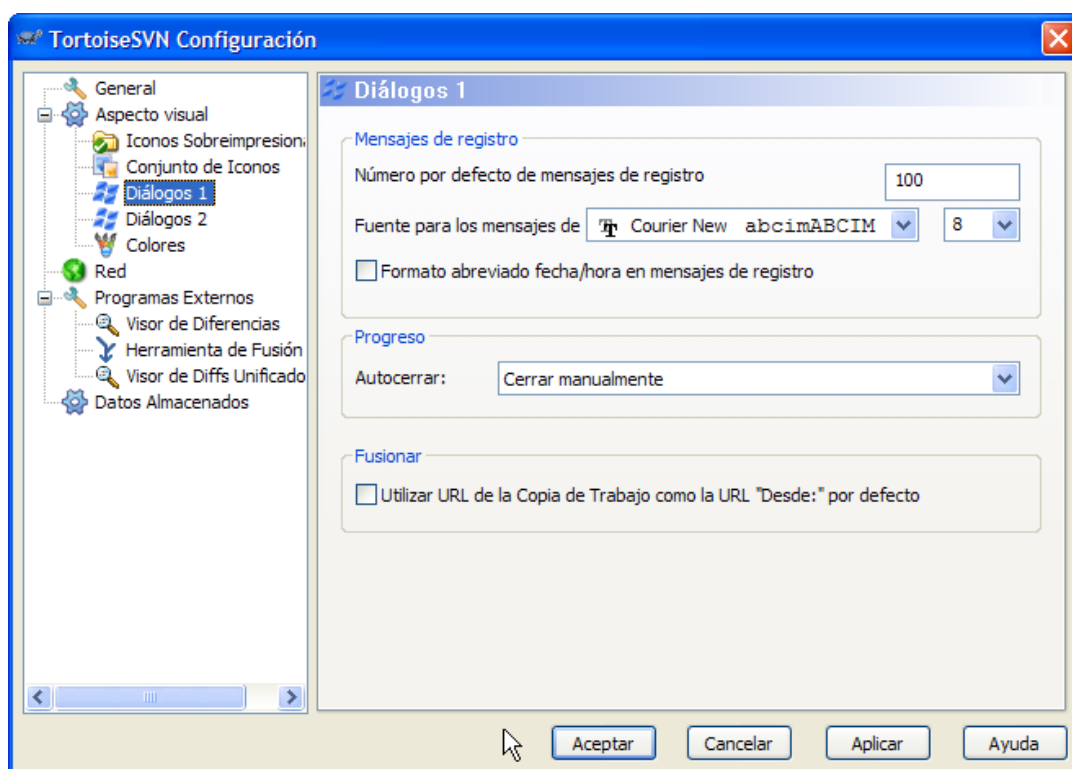


Figura 5.45. El diálogo Configuración, página de Diálogos 1

Este diálogo le permite configurar a su gusto algunos de los diálogos de TortoiseSVN.

Número por defecto de mensajes de registro

Limita el número de mensajes de registro que TortoiseSVN obtiene la primera vez que seleccione TortoiseSVN → Mostrar Registro Útil para conexiones lentas a servidores. Siempre puede utilizar Obtener Todos o Sigüientes 100 para obtener más mensajes.

Fuente para los mensajes de registro

Selecciona la fuente y el tamaño de la letra que se utiliza para mostrar el propio mensaje de registro en el panel del medio del diálogo Registro de Revisiones, y cuando se componen los mensajes de registro en el diálogo Confirmar.

Formato abreviado de fecha/hora en los mensajes de registro

Si los mensajes estándar largos toman mucho espacio en la pantalla utilice la forma corta.

Diálogo de progreso

TortoiseSVN puede cerrar automáticamente todos los diálogos de progreso cuando la acción se termina sin error. Esta configuración le permite seleccionar las condiciones para el cierre de diálogos. La configuración por defecto (recomendada) es **Cerrar manualmente** que le permite revisar todos los mensajes y comprobar qué ha pasado. Sin embargo, puede decidir que desea ignorar algunos tipos de mensajes y hacer que el cuadro de diálogo se cierre automáticamente si no hay cambios críticos.

Auto-cerrar si no hay fusiones, adiciones o eliminaciones significa que el diálogo de progreso se cerrará si ha habido actualizaciones simples, pero si algún cambio del repositorio se ha fusionado con los suyos, o si se añadió o borró algún fichero, el diálogo seguirá abierto. También seguirá abierto si hubo algún conflicto o error durante la operación.

Auto-cerrar si no hay fusiones, adiciones o eliminaciones para operaciones locales significa que el diálogo de progreso se cerrará igual que para Auto-cerrar si no hay fusiones, adiciones o eliminaciones, pero sólo para operaciones locales como añadir ficheros o revertir

cambios. Para las operaciones remotas el diálogo seguirá abierto.

Auto-cerrar si no hay conflictos relaja el criterio y cerrará el diálogo incluso si ha habido fusiones, adiciones o borrados. Sin embargo, si hay algún conflicto o error, el diálogo continuará abierto.

Auto-cerrar si no hay errores siempre cierra el diálogo, incluso si hay conflictos. La única condición que deja el diálogo abierto es una condición de error, que ocurre cuando Subversion no puede terminar una tarea. Por ejemplo, una actualización falla si el servidor es inaccesible, o una confirmación falla cuando la copia de trabajo no está actualizada.

Utilizar URL de la Copia de Trabajo como la URL "Desde:" por defecto

En el diálogo de fusión, el comportamiento por defecto es que la URL Desde: se mantenga entre fusiones. Sin embargo, algunas personas quieren realizar fusiones desde diferentes puntos en su jerarquía, y les resulta más cómodo empezar con la URL de la copia de trabajo actual. Esto puede luego cambiarse para referirse a una ruta paralela en otra rama.

5.26.2.4. Configuración de diálogos de TortoiseSVN 2

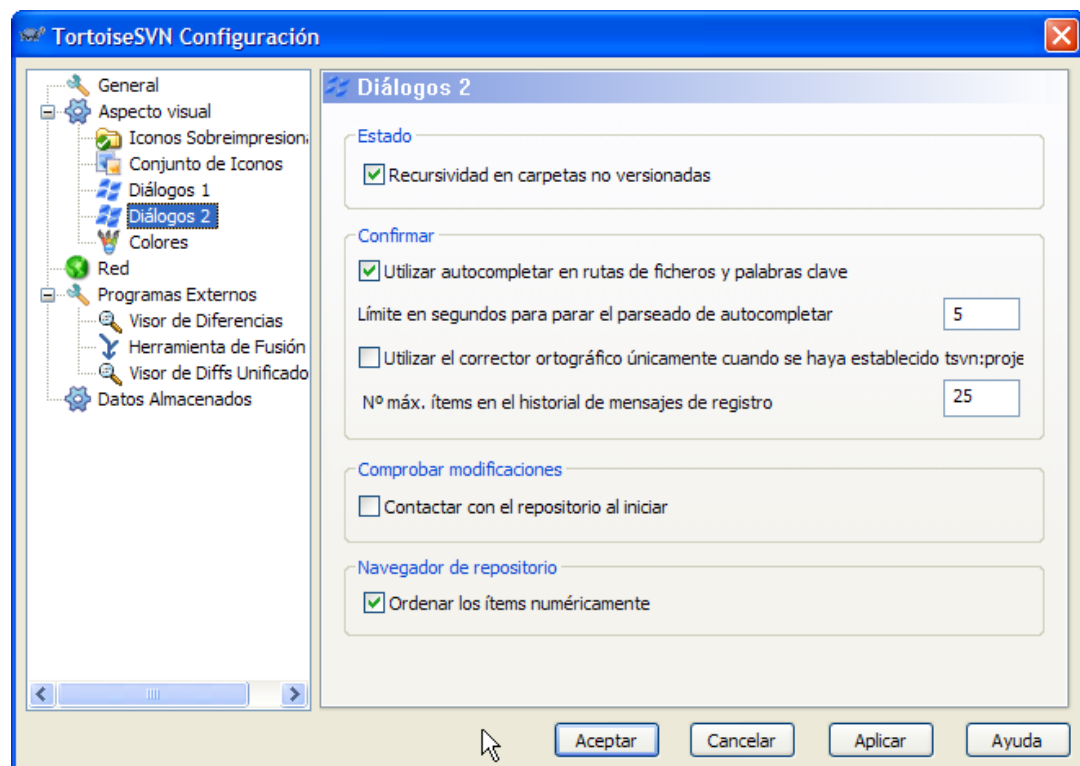


Figura 5.46. El diálogo Configuración, página de Diálogos 2

Recursividad en carpetas no versionadas

Si se marca esta casilla (por defecto está marcada), entonces siempre que se muestre el estado de una carpeta sin versionar en los diálogos de Añadir, Confirmar o Comprobar Modificaciones, se muestran también sus ficheros y carpetas hijos. Si desmarca esta casilla, sólo se muestra la carpeta padre sin versionar, lo que reduce el desorden en esos diálogos. En ese caso, si selecciona una carpeta sin versionar para Añadir, se añadirá recursivamente.

Utilizar autocompletar en rutas de ficheros y palabras clave

El diálogo confirmar incluye una ayuda para parsear la lista de nombres de ficheros que se van a confirmar. Cuando teclee las primeras 3 letras de un ítem en la lista, aparecerá la caja de autocompletar, y puede pulsar Enter para completar el nombre del fichero. Seleccione la caja para

habilitar esta opción.

Tiempo límite en segundos para parar el parseado de autocompletar

El parseador de autocompletar puede ser bastante lento si hay muchos ficheros grandes que comprobar. Este tiempo límite evita que el diálogo de confirmación se congele durante mucho tiempo. Si está perdiendo información de autocompletar importante, puede extender el tiempo límite.

Utilizar el corrector ortográfico únicamente cuando se haya establecido `tsvn:projectlanguage`

Si no desea utilizar el corrector ortográfico para todas las confirmaciones, marque esta casilla. Aun así, el corrector ortográfico se activará cuando las propiedades del proyecto así lo indiquen.

Número máximo de ítems para mantener en el historial de mensajes de registro.

TortoiseSVN almacena los últimos 25 mensajes de registro que haya introducido para cada repositorio. Aquí puede parametrizar la cantidad. Si tiene muchos repositorios diferentes, puede querer reducirlo para evitar llenar su registro.

Re-abrir el diálogo de confirmación después de una confirmación fallida

Cuando una confirmación falla por alguna razón (la copia de trabajo necesita actualizarse, un gancho pre-confirmación rechaza la confirmación, error de red, etc), puede seleccionar esta opción para mantener el diálogo de confirmación abierto y listo para intentarlo de nuevo. Sin embargo, debe tener en cuenta de que esto puede provocar problemas. Si el fallo significa que debe actualizar su copia de trabajo, y esa actualización conlleva conflictos, deberá resolver esos conflictos primero.

Contactar con el repositorio al iniciar

El diálogo Comprobar Modificaciones comprueba la copia de trabajo por defecto, y sólo contacta con el repositorio cuando pulse el botón **Comprobar repositorio**. Si siempre quiere comprobar el repositorio, puede utilizar esta opción para que esa acción siempre ocurra automáticamente.

Ordenar los ítems numéricamente

El navegador de repositorios puede utilizar un algoritmo de ordenación más inteligente que maneja rutas que contienen números mejor que una ordenación puramente ASCII. Esto a veces es útil para obtener las etiquetas con números de versión en el orden correcto. Esta opción controla el tipo de ordenación que se utiliza por defecto.

5.26.2.5. Configuración de colores de TortoiseSVN

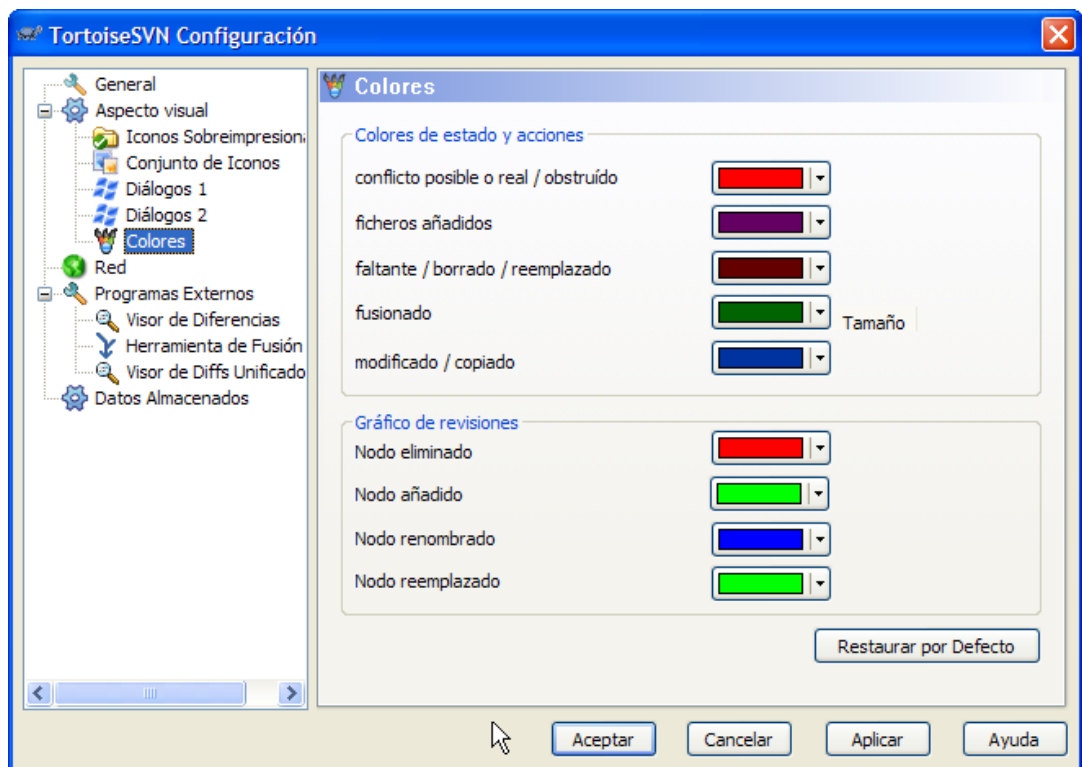


Figura 5.47. El Diálogo Configuración, Página de Colores

Este diálogo le permite configurar a su gusto los colores de los textos utilizados en los diálogos de TortoiseSVN.

Conflicto posible o real / obstruido

Ha ocurrido un conflicto durante la actualización, o puede ocurrir durante la fusión. La actualización está obstruida por una carpeta o un fichero sin versionar que ya existe en su copia de trabajo, y tiene el mismo nombre que uno versionado.

Este color se utiliza también para los mensajes de error en los diálogos de progreso.

Ficheros añadidos

Ítems añadidos al repositorio.

Faltante / borrado / reemplazado

Ítems borrados del repositorio, faltantes en la copia de trabajo, o borrados de la copia de trabajo y reemplazados con otro fichero del mismo nombre.

Fusionado

Cambios del repositorio que se han fusionado satisfactoriamente con su copia de trabajo sin crear ningún conflicto.

Modificado / copiado

Añadido con historia, o rutas copiadas en el repositorio. También se utiliza en el diálogo de registro para las entradas que incluyen ítems copiados.

Nodo eliminado

Un ítem que ha sido eliminado del repositorio.

Nodo añadido

Un ítem que ha sido añadido al repositorio, mediante una operación añadir, copiar o mover.

Nodo renombrado

Un ítem que ha sido renombrado dentro del repositorio.

Nodo reemplazado

El ítem original ha sido borrado y un nuevo ítem con el mismo nombre le reemplaza.

5.26.3. Configuración de red

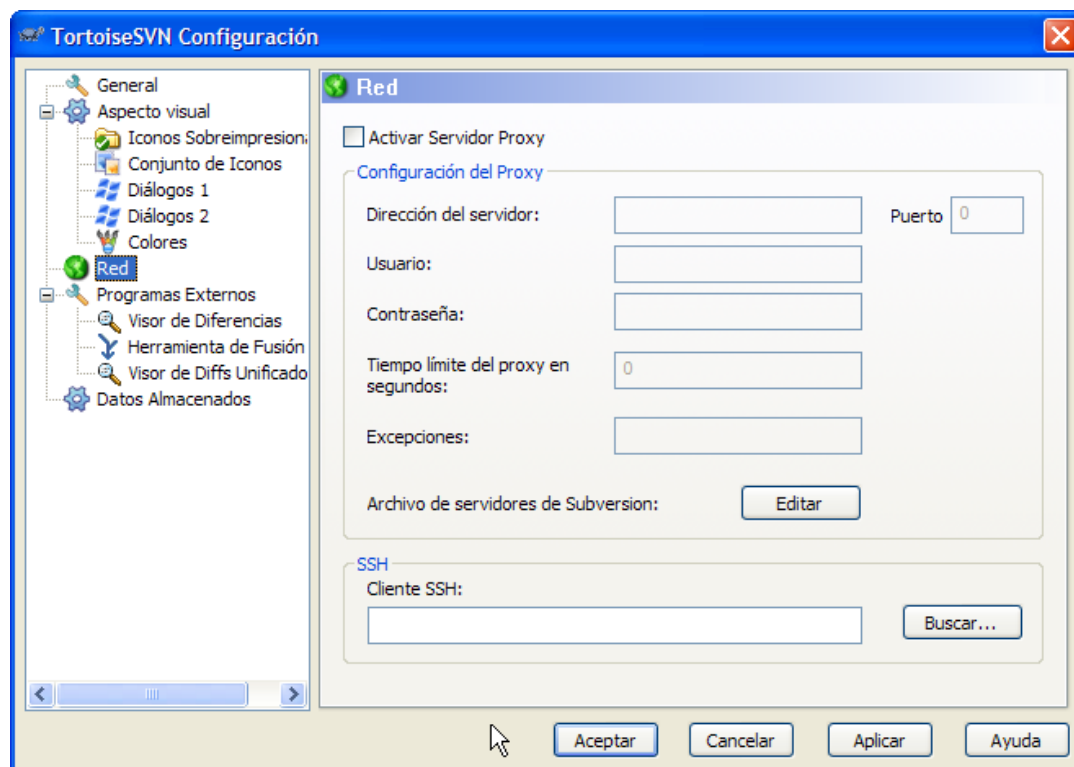


Figura 5.48. El diálogo Configuración, página de Red

Aquí puede configurar su servidor proxy, si necesita uno para atravesar el cortafuegos de su compañía.

Si necesita preparar configuraciones de proxy por cada repositorio, necesitará utilizar el fichero `servers` de Subversion para configurar esto. Utilice **Editar** para llegar allí directamente. Consulte el *Manual de Subversion* [<http://svnbook.red-bean.com/en/1.1/ch07.html#svn-ch-7-sect-1.3.1>] para ver los detalles sobre cómo utilizar este fichero.

También puede especificar qué programa debe utilizar TortoiseSVN para establecer una conexión segura a un repositorio `svn+ssh`. Le recomendamos que utilice `TortoisePlink.exe`. Ésta es una versión del popular programa `Plink`, y se incluye con TortoiseSVN, pero está compilado como una aplicación sin ventanas, por lo que no verá una ventana DOS molestando cada vez que se autentifica.

Un efecto colateral de no tener una ventana es que no hay sitio para que vayan los mensajes de error, por lo que si la autentificación falla simplemente obtendrá un mensaje diciendo algo como “Unable to write to standard output” (“No se puede escribir en la salida estándar”). Por esta razón le recomendamos que primero pruebe su configuración con el `Plink` estándar. Cuando todo funcione, puede utilizar TortoisePlink con exactamente los mismos parámetros.

5.26.4. Configuración de programas externos

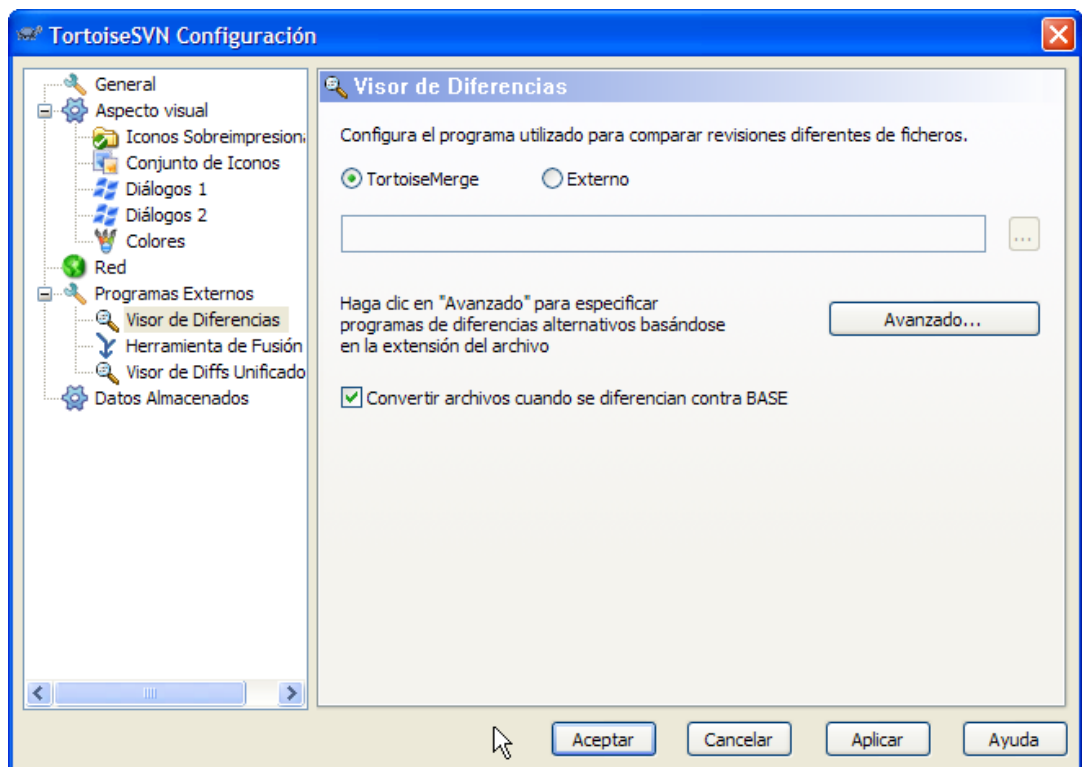


Figura 5.49. El diálogo Configuración, página de Visor de diferencias

Aquí puede definir sus propios programas de diferencias/fusión que TortoiseSVN debería utilizar. La configuración básica es utilizar TortoiseMerge que se instala junto con TortoiseSVN.

En Sección 5.9.4, “Herramientas externas de diferencias/fusión” encontrará una lista de algunos de los programas externos de diferencias / fusión que la gente está utilizando con TortoiseSVN.

5.26.4.1. Visor de diferencias

Se puede utilizar un programa externo de diferencias para comparar diferentes revisiones de ficheros. El programa externo necesitará obtener los nombres de los ficheros de la línea de comando, junto con otras opciones de la línea de comandos. TortoiseSVN usa sustitución de parámetros con prefijo %. Cuando se encuentra uno de estos se sustituirá por el valor apropiado. El orden de los parámetros dependerá del programa de Diferencias que utilice.

%base

El fichero original sin sus cambios

%bname

La ventana de título para el fichero base

%mine

Su propio fichero, con sus cambios

%yname

El título de la ventana para su fichero

Los títulos de las ventanas no son nombres de ficheros puros. TortoiseSVN lo trata como un nombre para mostrar y crea los nombres según eso. Por ejemplo, si está haciendo diferencias entre un fichero en la revisión 123 y un fichero en su copia de trabajo, los nombres pueden ser nombre-de-fichero : revision 123 y nombre-de-fichero : copia de trabajo

Por ejemplo, con ExamDiff Pro:

```
C:\Ruta-A\ExamDiff.exe %base %mine
```

o con KDiff3:

```
C:\Ruta-A\kdiff3.exe %base %mine --L1 %bname --L2 %yname
```

o con WinMerge:

```
C:\Ruta-A\WinMerge.exe -e -ub -dl %bname -dr %yname %base %mine
```

o con Araxis:

```
C:\Ruta-A\compare.exe /max /wait /title1:%bname /title2:%yname
                        %base %mine
```

Si utiliza la propiedad `svn:keywords` para expandir palabras clave, y en particular la `revision` de un fichero, entonces puede haber una diferencia entre ficheros que se debe únicamente al valor actual de esta palabra clave. También puede ocurrir que si utiliza `svn:eol-style = native` el fichero BASE tenga finales de líneas puros LF mientras que sus ficheros tendrán finales de líneas CR-LF. TSVN normalmente oculta esas diferencias automáticamente ya que primero parsea el fichero BASE para expandir las palabras clave y los finales de líneas antes de realizar la operación de diferenciar. Sin embargo, esto puede tomar bastante tiempo con ficheros grandes. Si se desmarca **Convertir ficheros cuando se diferencia contra BASE**, TSVN se saltará el preprocesado de los ficheros.

5.26.4.2. Herramienta de fusión

Un programa de fusión externo que se utiliza para resolver ficheros en conflicto. La sustitución de parámetros se utiliza de la misma forma que el Programa de Diferencias.

`%base`

El fichero original sin ningún cambio, ni suyo ni de otros

`%bname`

La ventana de título para el fichero base

`%mine`

Su propio fichero, con sus cambios

`%yname`

El título de la ventana para su fichero

`%theirs`

El fichero tal cual estaba en el repositorio

`%tname`

El título de la ventana del fichero en el repositorio

`%merged`

El fichero en conflicto, el resultado de la operación de fusión

`%mname`

El título de la ventana para el fichero fusionado

Por ejemplo, con Perforce WinMerge:

```
C:\Ruta-A\P4WinMrg.exe %base %theirs %mine %merged
```

o con KDiff3:

```
C:\Ruta-A\kdiff3.exe %base %mine %theirs -o %merged
```

o con Araxis:

```
C:\Ruta-A\compare.exe /max /wait /3 /title1:%tname /title2:%bname  
/title3:%ynome %theirs %base %mine %merged /a2
```

5.26.4.3. Configuración avanzada de diferencias / fusiones

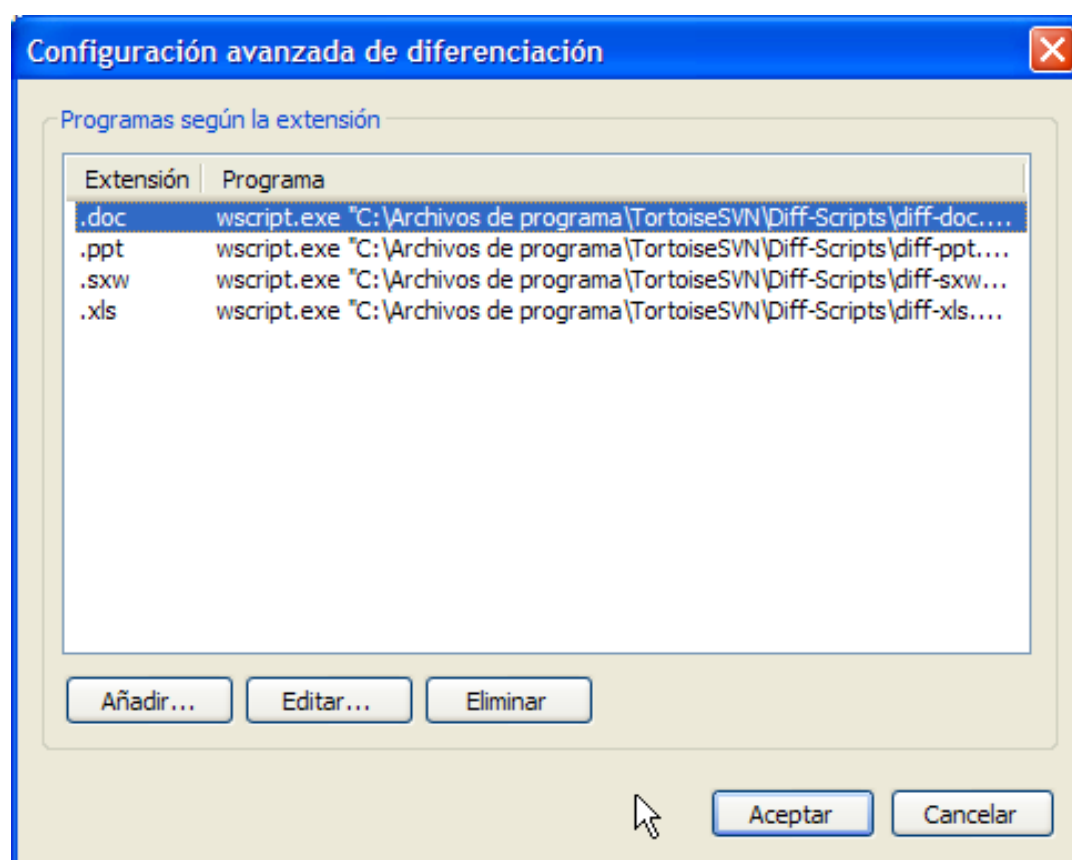


Figura 5.50. El diálogo Configuración, diálogo de Diferencias/Fusión avanzadas

En las configuraciones avanzadas, puede definir un programa de diferencias y fusión diferente por cada extensión de fichero. Por ejemplo, podría asociar Photoshop como el Programa de “Diferencias” para ficheros .jpg :-)

Necesita especificar sólo la extensión sin el punto o el comodín. Utilice BMP para describir ficheros de bitmaps de Windows, *no* * .BMP.

5.26.4.4. Visor para ficheros de diff unificado

Un programa visor de ficheros diff unificado (ficheros de parche). No se necesitan parámetros. La opción Por defecto es para buscar un programa asociado a los ficheros .diff, y si no, para fiche-

ros .txt. Si no tiene un visor para los ficheros .diff, seguramente obtendrá el Bloc de notas.

El programa Bloc de notas original de Windows no se comporta bien con los ficheros que no tienen terminaciones de líneas CR-LF estándar. Dado que la mayoría de ficheros diff unificados tienen terminaciones de líneas LF puras, no se ven bien en el Bloc de Notas. Sin embargo, puede descargar un reemplazo del Bloc de Notas *Notepad2* [<http://www.flos-freeware.ch/notepad2.html>] que no sólo enseña las terminaciones de líneas correctamente, sino que también añade código de colores a las líneas añadidas y eliminadas.

5.26.5. Datos de configuración almacenados

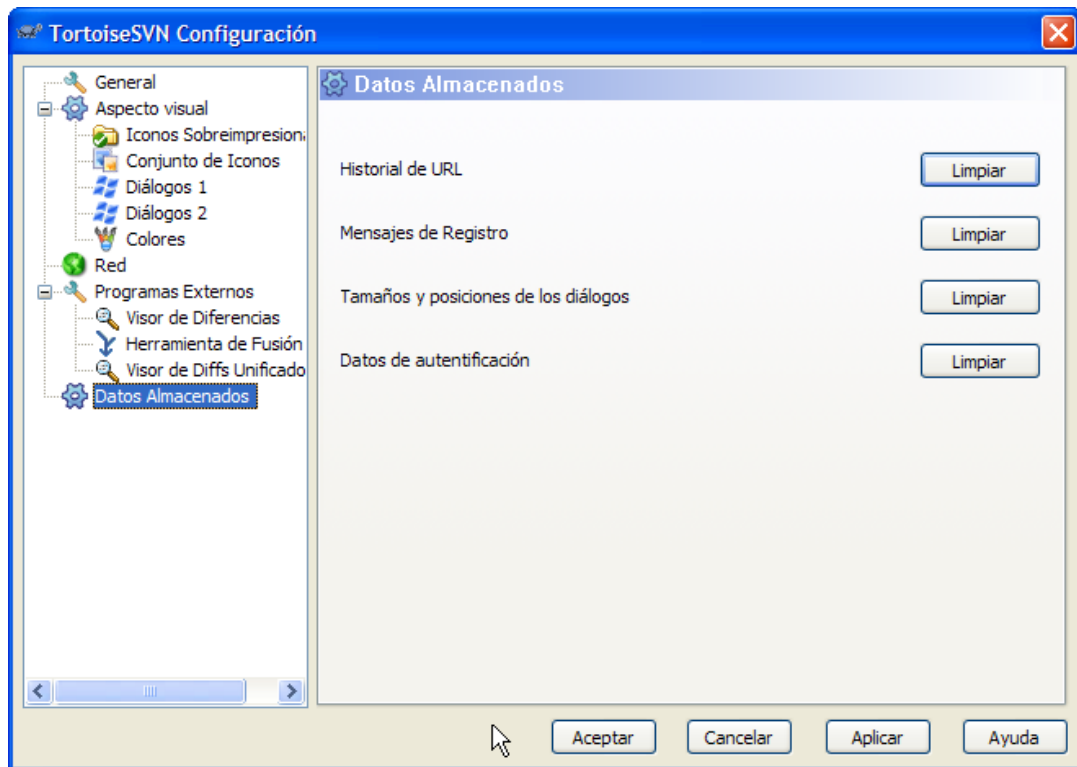


Figura 5.51. El diálogo Configuración, página de Datos almacenados

Por su comodidad, TortoiseSVN almacena muchas de las opciones que utiliza, y recuerda dónde ha estado últimamente. Si desea limpiar esa caché de datos, puede hacerlo aquí.

Historia de URL

Siempre que obtiene una copia de trabajo, fusione cambios o utilice el navegador de repositorios, TortoiseSVN guarda un registro de las últimas URLs utilizadas y las ofrece en una caja desplegable. A veces esa lista se llena de URLs desfasadas por lo que puede ser útil limpiarla periódicamente.

Mensajes de registro

TortoiseSVN almacena los mensajes de registro de confirmación que introduce. Estos se almacenan por cada repositorio, por lo que si accede a muchos repositorios la lista puede crecer bastante.

Tamaños y posiciones de los diálogos

Muchos diálogos almacenan el tamaño y la posición en pantalla que tenían la última vez.

Datos de autenticación

Cuando se autentifica con un servidor de Subversion, el usuario y la contraseña se cachean localmente para que no tenga que seguir introduciéndolos. Puede querer limpiar esto por razones

de seguridad, o porque quiere acceder al repositorio bajo un nombre de usuario diferente ... ¿sabe John que está utilizando su PC?

Si desea eliminar los datos de autenticación de un único servidor en particular, lea Sección 5.1.5, “Autenticación” para obtener instrucciones sobre cómo encontrar los datos cacheados.

5.26.6. Configuraciones del registro

Hay algunas configuraciones poco utilizadas que están disponibles únicamente editando directamente el registro.

Configuración

Puede especificar un lugar diferente para el fichero de configuración de Subversion utilizando la clave del registro HKCU\Software\TortoiseSVN\ConfigDir. Esto afectará a todas las operaciones de TortoiseSVN.

Icono de la caché en la barra de tareas

Para añadir en la barra de tareas un icono de la caché para el programa TSVNCache, cree una entrada DWORD con valor 1 en HKCU\Software\TortoiseSVN\CacheTrayIcon. Esto realmente sólo es útil para desarrolladores dado que le permite terminar el programa de forma normal.

5.26.7. Carpetas de trabajo de Subversion

VS.NET cuando se utiliza con proyectos web no puede manejar las carpetas `.svn` que Subversion utiliza para almacenar su información interna. Esto no es un error de Subversion. El error está en VS.NET y las extensiones de Frontpage que utiliza.

Desde la versión 1.3.0 de Subversion y TortoiseSVN, puede establecer la variable de entorno `SVN_ASP_DOT_NET_HACK`. Si se establece esa variable, Subversion utilizará carpetas `_svn` en vez de carpetas `.svn`. Debe reiniciar el shell para que esa variable de entorno tenga efecto. Normalmente, eso significa reiniciar su PC. Para hacerlo más sencillo, ahora puede hacer esto desde la página de configuración general utilizando una simple casilla - lea Sección 5.26.1, “Configuración general”.

For more information, and other ways to avoid this problem in the first place, check out the article about this in our *FAQ* [<http://tortoisesvn.net/node/aspdotnethack>].

5.27. Último paso

Lista de deseos

Incluso aunque TortoiseSVN y TortoiseMerge son gratuitos, puede ayudar a los desarrolladores enviándoles parches y tomando un rol activo en el desarrollo. También puede ayudar a alegrarnos las horas interminables que gastamos delante de nuestros ordenadores.

Nos encanta escuchar música mientras trabajamos en TortoiseSVN. Y dado que gastamos muchas horas en el proyecto necesitamos *mucha* música. Por tanto, hemos preparado algunas listas de deseos con nuestros CDs y DVDs de música favoritos: <http://tortoisesvn.tigris.org/donate.html> Por favor mire también la lista de personas que han contribuido al proyecto enviando parches o traducciones.

Capítulo 6. El programa SubWCRev

SubWCRev es un programa de consola para Windows que puede utilizarse para leer el estado de una copia de trabajo local y opcionalmente realizar sustituciones de palabras clave en un fichero plantilla. A menudo se utiliza como parte del proceso de compilación como una forma de incorporar información de la copia de trabajo en el objeto que está construyendo. Típicamente, se puede utilizar para incluir el número de revisión en un diálogo “Acerca de”.

6.1. La línea de comandos de SubWCRev

SubWCRev lee el estado de Subversion de todos los ficheros en una copia de trabajo, excluyendo los externos por defecto. Apunta el número de revisión de confirmación más alto que encuentra, y la fecha de esa confirmación. También apunta si hay modificaciones locales en la copia de trabajo, o revisiones de actualización mezcladas. El número de revisión, el rango de revisiones de actualización y el estado de las modificaciones se muestra por la salida estandar stdout.

SubWCRev.exe se llama desde la línea de comandos o desde un script, y se controla utilizando parámetros de la línea de comandos.

```
SubWCRev WorkingCopyPath [SrcVersionFile DstVersionFile] [-nmdfe]
```

WorkingCopyPath es la ruta a la copia de trabajo que se va a comprobar. Sólo puede utilizar SubWCRev en copias de trabajo, no directamente en el repositorio. La ruta puede ser absoluta o relativa al directorio actual.

Si quiere que SubWCRev realice sustituciones de palabras clave, para que se graben los campos como la revisión del repositorio y la URL a un fichero de texto, debe proporcionar un fichero plantilla SrcVersionFile y un fichero de salida DstVersionFile que contendrá la versión sustituida de la plantilla.

Hay un número de modificadores opcionales que afectan la forma en la que SubWCRev trabaja. Si utiliza más de una, debe especificarlos como un único grupo, por ejemplo, -nm y no -n -m.

Cam-biar	Descripción
-n	Si se especifica esta opción, SubWCRev terminará con ERRORLEVEL 7 si la copia de trabajo tiene modificaciones locales. Esto puede utilizarse para evitar compilaciones cuando hay presentes cambios sin confirmar.
-m	Si se especifica esta opción, SubWCRev terminará con ERRORLEVEL 8 si la copia de trabajo contiene revisiones mezcladas. Esto puede utilizarse para evitar compilaciones de una copia de trabajo parcialmente actualizada.
-d	Si se especifica esta opción, SubWCRev terminará con ERRORLEVEL 9 si el fichero de destino ya existe.
-f	Si se especifica esta opción, SubWCRev incluirá la revisión del último cambio de las carpetas. El comportamiento por defecto es utilizar únicamente los ficheros cuando se obtienen los números de revisión.
-e	Si se especifica esta opción, SubWCRev examinará los directorios que se han incluido con svn:externals, pero sólo si son del mismo repositorio. El comportamiento por defecto es ignorar los externos.

Tabla 6.1. Lista de opciones de línea de comandos disponible

6.2. Sustitución de palabras clave

Si se especifican ficheros de origen y destino, SubWCREv copia el fichero origen al destino, realizando las siguientes sustituciones de palabras clave:

Palabra clave	Descripción
\$WCREV\$	Se reemplaza con la revisión de confirmación más alta de la copia de trabajo.
\$WCDATE\$	Se reemplaza con la fecha y hora de la confirmación de la revisión de confirmación más alta de la copia de trabajo. Para evitar confusiones, se utiliza el formato internacional, es decir, yyyy-mm-dd hh:mm:ss
\$WCNOW\$	Se reemplaza con la fecha y hora actual. Esto puede utilizarse para indicar la fecha de compilación. Se utiliza el formato internacional como se describe arriba.
\$WCRANGE\$	Se reemplaza con el rango de revisiones de actualización de la copia de trabajo. Si la copia de trabajo está en un estado consistente, esto será una única revisión. Si la copia de trabajo contiene revisiones mezcladas, bien por estar desactualizada, o bien porque se ha realizado deliberadamente una actualización a una revisión concreta, se mostrará el rango de la forma 100:200.
\$WCMIXED?TText:FText\$	Se reemplaza por TText si hay revisiones de actualización mezcladas, o por FText si no las hay.
\$WCMODS?TText:FText\$	Se reemplaza por TText si hay modificaciones locales, o por FText si no las hay.
\$WCURL\$	Se reemplaza con la URL del repositorio de la copia de trabajo pasada a SubWCREv.

Tabla 6.2. Lista de opciones de línea de comandos disponible

6.3. Ejemplo de palabras clave

El siguiente ejemplo le muestra cómo se sustituyen las palabras clave de un fichero plantilla en el fichero resultado.

```
// Test file for SubWCREv

char *Revision = "$WCREV$";
char *Modified = "$WCMODS?Modified:Not modified$";
char *Date = "$WCDATE$";
char *RevRange = "$WCRANGE$";
char *Mixed = "$WCMIXED?Mixed revision WC:Not mixed$";
char *URL = "$WCURL$";

#if $WCMODS?1:0$
#error Source is modified
#endif

// EndOfFile
```

Tras ejecutar SubWCREv.exe, el fichero de salida será similar a este:

```
// Test file for SubWCREv

char *Revision = "3701";
char *Modified = "Modified";
char *Date = "2005/06/15 11:15:12";
char *RevRange = "3699:3701";
char *Mixed = "Mixed revision WC";
char *URL = "http://tortoisesvn.tigris.org/svn/tortoisesvn/trunk/src/SubWCREv"
```

```
#if 1
#error Source is modified
#endif

// EndOfFile
```

Apéndice A. Preguntas más frecuentes (FAQ)

Because TortoiseSVN is being developed all the time it is sometimes hard to keep the documentation completely up to date. We maintain an *interactive online FAQ* [<http://tortoisesvn.net/faq>] which contains a selection of the questions we are asked the most on the TortoiseSVN mailing lists [<dev@tortoisesvn.tigris.org>](mailto:dev@tortoisesvn.tigris.org) and [<users@tortoisesvn.tigris.org>](mailto:users@tortoisesvn.tigris.org).

We also maintain a *project Issue Tracker* [<http://issues.tortoisesvn.net>] which tells you about some of the things we have on our To-Do list, and bugs which have already been fixed. If you think you have found a bug, or want to request a new feature, check here first to see if someone else got there before you.

Si tiene una pregunta que no está respondida en ningún otro lado, el mejor lugar para preguntar es en la lista de correo.

Apéndice B. ¿Cómo...?

Este apéndice contiene soluciones a los problemas o preguntas que puede tener mientras utiliza TortoiseSVN.

B.1. Mover/copiar muchos ficheros de golpe

Se pueden mover/copiar ficheros de uno en uno utilizando TortoiseSVN → Renombrar.... Pero si desea mover/copiar muchos ficheros, esta forma es muy lenta y trabajosa.

La forma recomendada es arrastrar con el botón derecho los ficheros a su nuevo destino. Simplemente haga click con el botón derecho en los ficheros que desea mover o copiar sin soltar el botón del ratón. Luego arrastre los ficheros a su nuevo destino y suelte el botón del ratón. Aparecerá un menú contextual donde puede elegir Menú contextual → Copiar en Subversion aquí o Menú contextual → Mover en Subversion aquí.

B.2. Obligar a los usuarios a introducir un mensaje de registro

Hay dos formas de evitar que los usuarios puedan confirmar con un mensaje de registro vacío. Uno es específico de TortoiseSVN, el otro funciona con todos los clientes de Subversion, pero necesita acceso directo al servidor.

B.2.1. Script gancho en el servidor

Si tiene acceso directo al servidor del repositorio, puede instalar un script gancho pre-commit que rechace todas las confirmaciones con mensajes de registro cortos o vacíos.

En la carpeta del repositorio en el servidor, hay una subcarpeta `hooks` que contiene algunos scripts ganchos de ejemplo que puede utilizar. El fichero `pre-commit.tmpl` contiene un script de ejemplo que rechaza las confirmaciones si no se ha introducido un mensaje de registro, o si ese mensaje es demasiado corto. El fichero también tiene comentarios sobre cómo instalar/utilizar este script. Simplemente siga las instrucciones de ese fichero.

Este método es la forma recomendada si sus usuarios también utilizan otros clientes de Subversion además de TortoiseSVN. La parte negativa es que la confirmación se rechaza por el servidor, y por tanto los usuarios obtendrán un mensaje de error. El cliente no puede saber antes de realizar la confirmación que se rechazará. Si desea que TortoiseSVN deshabilite el botón **Aceptar** mientras el mensaje de registro sea demasiado corto, utilice el método descrito a continuación.

B.2.2. Propiedades del proyecto

TortoiseSVN utiliza propiedades para controlar algunas de sus funciones. Una de esas propiedades es la propiedad `tsvn:logminsize`.

Si establece esa propiedad en una carpeta, TortoiseSVN deshabilitará el botón **Aceptar** en todos los diálogos de confirmación hasta que el usuario introduzca un mensaje de registro con al menos la longitud especificada en la propiedad.

Para obtener información más detallada sobre esas propiedades de proyectos, por favor lea Sección 5.15, “Configuración del proyecto”

B.3. Actualizar los ficheros seleccionados desde el repositorio

Normalmente actualizará su copia de trabajo utilizando TortoiseSVN → Actualizar. Pero si sólo desea obtener algunos ficheros nuevos que un colega ha añadido sin fusionar ningún cambio en los demás ficheros a la vez, necesitará otra forma de actuación.

Utilice TortoiseSVN → Comprobar Modificaciones. y pulse en Comprobar repositorio para ver lo que se cambió en el repositorio. Seleccione los ficheros que desea actualizar localmente, y utilice

el menú contextual para actualizar sólo esos ficheros.

B.4. Deshacer revisiones en el repositorio

B.4.1. Utilice el diálogo Registro de revisiones

La manera más fácil de revertir los cambios de una revisión en concreto, o de un rango de revisiones, es utilizar el diálogo de registro de revisiones. Este es también el método a utilizar si desea descartar cambios recientes y hacer que una revisión antigua se convierta en la nueva HEAD.

1. Seleccione el fichero o la carpeta en la que desea revertir los cambios. Si desea revertir todos los cambios, esta debería ser la carpeta más alta.
2. Seleccione TortoiseSVN → Mostrar Registro para mostrar una lista de revisiones. Puede que necesite utilizar Obtener Todas o Sigüientes 100 para mostrar la o las revisiones que le interesen.
3. Seleccione la revisión que desea revertir. Si desea deshacer un rango de revisiones, seleccione la primera y pulse la tecla Mayúsculas mientras selecciona la última. Tenga en cuenta que para múltiples revisiones, el rango debe ser seguido sin huecos. Haga click con el botón derecho en la o las revisiones seleccionadas, y luego seleccione Menú Contextual → Revertir cambios de esta revisión.
4. O si desea hacer que una revisión antigua se convierta en la nueva revisión HEAD, haga click con el botón derecho en la revisión seleccionada(s), luego seleccione Menú Contextual → Revertir a esta revisión. Esto descartará *todos* los cambios que se hicieron después de la revisión seleccionada.

Ha revertido los cambios dentro de su copia de trabajo. Compruebe los resultados, y luego confirme los cambios.

B.4.2. Utilice el diálogo Fusionar

Para deshacer un rango grande de revisiones, puede utilizar el diálogo Fusionar. El método anterior utiliza la fusión por detrás; este método la utiliza explícitamente.

1. En su copia de trabajo seleccione TortoiseSVN → Fusionar.
2. En el campo Desde: introduzca la URL completa de la carpeta de la rama o la etiqueta que contiene los cambios que desea revertir en su copia de trabajo. Debería aparecer como la URL por defecto.
3. En el campo Desde la Revisión introduzca el número de revisión en la que está actualmente. Si está seguro de que nadie más está haciendo cambios, puede utilizar la revisión HEAD.
4. Asegúrese de que la casilla Usar la URL "Desde:" esté marcada.
5. En el campo Hasta la Revisión introduzca el número de revisión a la que desea revertir, esto es, justo la *anterior* a la primera revisión que va a ser revertida.
6. Pulse Aceptar para completar la fusión.

Ha revertido los cambios dentro de su copia de trabajo. Compruebe los resultados, y luego confirme los cambios.

B.4.3. Utilice svndumpfilter

Dado que TortoiseSVN nunca pierde datos, sus revisiones “deshechas” aún existen como revisiones intermedias en el repositorio. Sólo se ha cambiado la revisión HEAD a su estado previo. Si desea hacer que las revisiones desaparezcan completamente de su repositorio, borrando toda traza de que

alguna vez existieran, deberá utilizar medidas más extremas. A menos de que haya una buena razón para hacer esto, *no se recomienda*. Una razón posible sería que alguien haya confirmado un documento confidencial en un repositorio público.

La única forma de quitar datos de un repositorio es utilizar la herramienta de línea de comandos de Subversion `svnadmin`. Puede encontrar una descripción sobre cómo funciona en el *Libro de Subversion* [<http://svnbook.red-bean.com/en/1.1/ch05s03.html#svn-ch-5-sect-3.1.3>].

B.5. Comparar dos revisiones de un fichero

Si desea comparar dos revisiones en la historia de un fichero, por ejemplo las revisiones 100 y 200 del mismo fichero, utilice TortoiseSVN → Mostrar Registro para ver la historia de revisiones de ese fichero. Seleccione las dos revisiones que desea comparar y luego utilice Menú Contextual → Comparar Revisiones.

Si desea comparar el mismo fichero en dos árboles diferentes, por ejemplo el troncal y una rama, puede utilizar el navegador de repositorios para abrir ambas ramas, seleccionar el fichero en ambos lugares, y utilizar Menú Contextual → Comparar Revisiones.

Si desea comparar dos árboles para ver lo que ha cambiado, por ejemplo la troncal y una versión etiquetada, puede utilizar TortoiseSVN → Gráfico de Revisiones Seleccione los dos nodos a comparar, y luego utilice Menú Contextual → Comparar Revisiones HEAD. Esto le mostrará una lista de los ficheros cambiados, y luego podrá seleccionar los ficheros individuales para ver los cambios en detalle. Alternativamente, puede utilizar Menú Contextual → Diff Unificado de las Revisiones HEAD para ver un resumen de todas las diferencias, con un contexto mínimo.

B.6. Incluir un sub-proyecto común

A veces querrá incluir otro proyecto en su copia de trabajo, quizás el código de alguna librería. No quiere hacer un duplicado de ese código en su repositorio, porque perdería la conexión con el código original (y mantenido). O quizás tiene varios proyectos que comparten el núcleo de código. Hay al menos 3 formas de lidiar con esto.

B.6.1. Utilice `svn:externals`

Establece la propiedad `svn:externals` para una carpeta en su proyecto. Esta propiedad contiene una o más líneas; cada línea tiene el nombre de una subcarpeta que quiere utilizar como la carpeta de obtención del código común, y la URL del repositorio que desea obtener ahí. Para obtener todos los detalles lea Sección 5.2.4, “Proyectos referenciados”.

Confirme la nueva carpeta. Ahora, cuando actualice, Subversion tirará una copia de ese proyecto desde su repositorio en su copia de trabajo. Las subcarpetas se crearán automáticamente si es necesario. Cada vez que actualice su copia de trabajo, también recibirá la última versión de todos los proyectos externos.

Si los proyectos externos están en el mismo repositorio, cualquier cambio que haga allí se incluirá en la lista de confirmación cuando confirme su proyecto principal.

Si los proyectos externos están en repositorios diferentes, cualquier cambio que haga en el proyecto externo se notificará cuando confirme el proyecto principal, pero tendrá que confirmar esos cambios externos de forma separada.

B.6.2. Utilice una copia de trabajo anidada

Cree una nueva carpeta dentro de su proyecto para contener el código común, pero no la añada a Subversion

Seleccione TortoiseSVN → Obtener en la nueva carpeta, y obtenga una copia del código común en ella. Ahora tiene anidada una copia de trabajo separada dentro de su copia de trabajo principal.

Las dos copias de trabajo son independientes. Cuando confirme los cambios de la padre, los cambios en la copia de trabajo anidada se ignoran. De la misma forma, cuando actualice el padre, la co-

pia de trabajo anidada no se actualiza.

B.6.3. Utilice una ruta relativa

Si utiliza el mismo código principal común en varios proyectos, y no quiere tener varias copias de trabajo de él en cada proyecto que lo utilice, puede obtenerlo en un lugar separado que se relacione con todos los otros proyectos que lo utilicen. Por ejemplo:

```
C:\Proyectos\Proy1
C:\Proyectos\Proy2
C:\Proyectos\Proy3
C:\Proyectos\Comun
```

y refiérase al código común con una ruta relativa, por ejemplo `..\..\Comun\DSPcore`.

Si sus proyectos están dispersos en lugares sin relación entre sí puede utilizar una variante de esto, que es poner el código común en un lugar y utilizar la sustitución de letras de unidades para mapear esa ruta a algo que puede codificar en sus proyectos, por ejemplo, obtenga el código común en `D:\Documentos\Framework` o `C:\Documents and Settings\{login}\Mis Documentos\Framework` y luego utilice

```
SUBST X: "D:\Documentos\Framework"
```

para crear el mapeo de unidades utilizado en su código fuente. Su código puede entonces utilizar rutas absolutas.

```
#include "X:\superio\superio.h"
```

Este método sólo funcionará en un entorno de todo-PCs, y necesitará documentar los mapeos de unidades requeridos para que su equipo sepa dónde están esos ficheros misteriosos. Este método debería utilizarse estrictamente en entornos de desarrollo cerrados, y no se recomienda para su uso generalizado.

B.7. Crear un acceso directo a un repositorio

Si necesita abrir el navegador de repositorios frecuentemente en un lugar en concreto, puede crear un acceso directo en el escritorio utilizando el interfaz de automatización de TortoiseProc. Simplemente cree un nuevo acceso directo y ponga como destino:

```
TortoiseProc.exe /command:repobrowser /path:"url/al/repositorio" /notempfile
```

Por supuesto necesitará incluir la URL real del repositorio.

B.8. Ignorar ficheros que ya están versionados

Si accidentalmente añadió algunos ficheros que deberían haber sido ignorados, ¿cómo puede sacarlos del control de versiones sin perderlos? Quizás tiene su propio fichero de configuración IDE que no es parte del proyecto, pero que le costó bastante tiempo ajustar a su gusto.

Si aún no ha confirmado los ficheros añadidos, todo lo que debe hacer es utilizar TortoiseSVN → Revertir... para deshacer la operación. Entonces debería añadir los ficheros a la lista de ignorados para que no se añadan de nuevo por error.

Si los ficheros ya están en el repositorio, tendrá que trabajar un poco más.

1. Mueva el fichero a algún lugar seguro, fuera de su copia de trabajo.
2. TortoiseSVN → Confirmar la carpeta padre. TortoiseSVN verá que el fichero falta y podrá

marcarlo para que sea borrado del repositorio.

3. Mueva el fichero de nuevo a su lugar original.
4. Añada el fichero a la lista de ignorados para que no vuelva a tener este problema en el futuro.

Si necesita eliminar una carpeta/jerarquía completa del control de versiones, el procedimiento vuelve a ser diferente.

1. TortoiseSVN → Exportar la carpeta a algún lugar seguro, fuera de su copia de trabajo.
2. TortoiseSVN → Eliminar la carpeta de su copia de trabajo.
3. TortoiseSVN → Confirmar la carpeta eliminada para quitarla del repositorio.
4. Mover la carpeta exportada de nuevo a su lugar original en su copia de trabajo.
5. Añadir la carpeta a la lista de ignorados para que no tenga el mismo problema de nuevo.

Apéndice C. Administradores

Este apéndice contiene soluciones a los problemas o preguntas que pueda tener cuando es responsable de distribuir TortoiseSVN a múltiples ordenadores clientes.

C.1. Instalar TortoiseSVN utilizando políticas de grupo

El instalador de TortoiseSVN viene en un fichero MSI, lo que significa que no debería tener problemas añadiendo ese fichero MSI a las políticas de grupo de su controlador de dominio.

Puede encontrar una buena guía sobre cómo hacerlo en el artículo de la base de datos de conocimiento 314934 de Microsoft: <http://support.microsoft.com/?kbid=314934>.

Las versiones 1.3.0 y posteriores de TortoiseSVN deben instalarse desde *Configuración del ordenador* y no bajo *Configuración del usuario*. Esto es así porque esas versiones necesitan las nuevas dlls CRT y MFC, que sólo pueden ser distribuidas *por ordenador* y no *por usuario*. Si realmente desea instalar TortoiseSVN por usuario, primero debe instalar el paquete MFC y CRT versión 8 de Microsoft en cada ordenador en el que quiera instalar TortoiseSVN por usuario.

C.2. Redirigir la comprobación de actualización

TortoiseSVN comprueba si hay una nueva versión disponible cada pocos días. Si hay una nueva versión disponible, aparece un diálogo informando al usuario sobre ello.

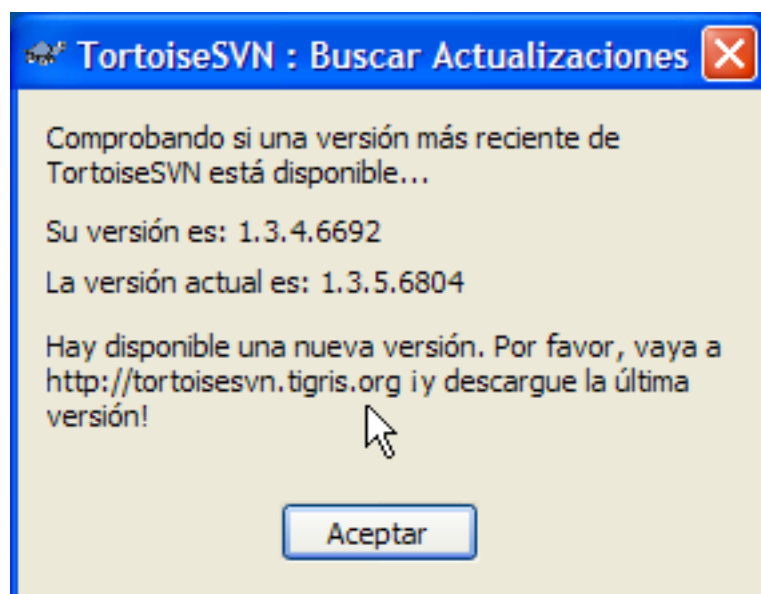


Figura C.1. El diálogo de Actualización

Si es responsable de muchos usuarios en su dominio, puede querer que sus usuarios utilicen sólo versiones que haya aprobado y que no siempre se instalen la última versión. Probablemente no querrá que aparezca el diálogo de actualización para que sus usuarios no vayan y se actualicen inmediatamente.

Las versiones 1.4.0 y superiores de TortoiseSVN le permiten redirigir esa comprobación de actualización a su servidor de la intranet. Puede utilizar la clave de registro HK-CU\Software\TortoiseSVN\UpdateCheckURL (valor de cadena) a una URL que apunte a un fichero de texto en su intranet. Ese fichero de texto debe tener el siguiente formato:

```
1.4.1.6000
```

```
¡Hay una nueva versión de TortoiseSVN disponible para descargar!  
http://192.168.2.1/downloads/TortoiseSVN-1.4.1.6000-svn-1.4.0.msi
```

La primera línea en ese fichero es la cadena de la versión. Debe asegurarse de que concuerde con la cadena de versión exacta del paquete de instalación de TortoiseSVN. La segunda línea es un texto personalizado, que se muestra en el diálogo de actualización. Puede escribir ahí lo que desee. Pero tenga en cuenta que el espacio en el diálogo de actualización es limitado. ¡Los mensajes demasiado largos se truncarán! La tercera línea es la URL del nuevo paquete de instalación. Esta URL se abre cuando el usuario hace click en la etiqueta del mensaje personalizado en el diálogo de actualización. También puede llevar al usuario a una página web en vez de al fichero msi directamente. La URL se abre con el navegador web predeterminado, por lo que si especifica una página web, esa página se abre y se muestra al usuario. Si especifica el paquete MSI, el navegador le pedirá al usuario guardar el fichero MSI localmente.

C.3. Estableciendo la variable de entorno SVN_ASP_DOT_NET_HACK

Desde la versión 1.4.0, el instalador de TortoiseSVN ya no ofrece al usuario la opción de establecer la variable de entorno SVN_ASP_DOT_NET_HACK, dado que eso ha causado muchos problemas y confusiones a algunos usuarios que siempre instalan *todo* sin importar si saben para lo que sirve.

Pero esa opción sólo está oculta para el usuario. Aún puede forzar al instalador de TortoiseSVN para que ponga esa variable de entorno estableciendo la propiedad ASPDOTNETHACK a TRUE. Por ejemplo, puede iniciar el instalador así:

```
msiexec /i TortoiseSVN-1.4.0.msi ASPDOTNETHACK=TRUE
```

Apéndice D. Automatizando TortoiseSVN

Dado que todos los comandos de TortoiseSVN se controlan a través de parámetros de línea de comandos, puede automatizarlo con scripts batch o iniciando comandos y diálogos específicos desde otros programas (por ejemplo, desde su editor de textos favorito).



Importante

Recuerde que TortoiseSVN es un cliente GUI, y esta guía de automatización le muestra cómo conseguir que aparezcan los diálogos de TortoiseSVN para recolectar información del usuario. Si desea escribir un script que no requiera entradas del usuario, debería utilizar el cliente de línea de comandos oficial de Subversion.

D.1. Comandos de TortoiseSVN

El programa de interfaz gráfico de TortoiseSVN se llama `TortoiseProc.exe`. Todos los comandos se especifican con el parámetro `/command:abcd` donde `abcd` es el nombre obligatorio del comando. La mayoría de estos comandos necesitan al menos un argumento de ruta, que se proporciona con `/path:"alguna\ruta"`. En la siguiente tabla el comando se refiere al parámetro `/command:abcd` y la ruta se refiere al parámetro `/path:"alguna\ruta"`.

Como algunos comandos pueden tomar una lista de rutas de destino (por ejemplo, si se confirman varios ficheros específicos) el parámetro `/path` puede admitir varias rutas, separadas con un carácter `*`.

Como TortoiseSVN utiliza ficheros temporales para pasar múltiples argumentos entre la extensión del Shell y el programa principal, *debe* añadir el parámetro `/notempfile`. ¡Si no lo hace, el comando no funcionará y el fichero que pase en el parámetro `/path` se borrará!

El diálogo de progreso que se utiliza para las confirmaciones, actualizaciones y muchos más comandos normalmente se queda abierto después de que el comando termina, hasta que el usuario pulsa el botón **Aceptar**. Esto puede cambiarse marcando la opción correspondiente en el diálogo de configuración. Pero utilizando esa opción se cerrará el diálogo de progreso, sin importar si se inició el comando desde el fichero batch o desde el menú contextual de TortoiseSVN.

Para especificar una localización diferente del fichero de configuración, utilice el parámetro `/configdir:"ruta\al\directorio\de\configuración"`. Esto tomará preferencia sobre la ruta por defecto, incluida cualquier configuración en el registro.

Para cerrar el diálogo de progreso al final de un comando automáticamente sin utilizar la configuración permanente, puede pasar el parámetro `/closeonend`.

- `/closeonend:0` no cierra el diálogo automáticamente
- `/closeonend:1` cierra automáticamente si no hay errores
- `/closeonend:2` cierra automáticamente si no hay errores ni conflictos
- `/closeonend:3` cierra automáticamente si no hay errores, conflictos ni fusiones
- `/closeonend:4` cierra automáticamente si no hay errores, conflictos ni fusiones para operaciones locales

The table below lists all the commands which can be accessed using the `TortoiseProc.exe` command line. As described above, these should be used in the form `/command:abcd`. In the table, the /

command prefix is omitted to save space.

Comando	Descripción
:about	Muestra el diálogo Acerca de... También se muestra si no se pasa ningún comando.
:log	Abre el diálogo de registro. La ruta especifica el fichero o la carpeta para el que se debe mostrar el registro. Se pueden establecer tres opciones adicionales: /revstart:xxx, /revend:xxx y /strict
:checkout	Abre el diálogo de obtener. /path especifica el directorio de destino y /url especifica la URL desde la que obtener.
:import	Abre el diálogo de importar. La ruta especifica el directorio con los datos a importar.
:update	Actualiza la copia de trabajo en /path hasta HEAD. Si se especifica la opción /rev se muestra un diálogo para preguntar al usuario a qué revisión debe actualizarse.
:commit	Abre el diálogo de confirmación. La ruta especifica el directorio de destino o la lista de ficheros a confirmar. También puede especificar la opción /logmsg para pasar un mensaje de registro predefinido al diálogo de confirmación. O, si no desea pasar el mensaje de registro en la línea de comandos, utilice /logmsgfile:path, donde ruta apunta a un fichero que contiene el mensaje de registro. Para rellenar la caja de texto del ID del error (en caso de que haya preparado adecuadamente la integración con el sistema de control de errores), puede utilizar /bugid:"el ID del error aquí".
:add	Añade los ficheros en /path al control de versiones.
:revert	Revierte las modificaciones locales de una copia de trabajo. La /path dice qué items deben revertirse.
:cleanup	Hace limpieza tras operaciones interrumpidas o abortadas, y desbloquea la copia de trabajo en /path.
:resolve	Marca un fichero en conflicto especificado en /path como resuelto. Si se especifica /noquestion, la resolución se realiza sin preguntar al usuario primero si realmente debe hacerse.
:repocreate	Crea un repositorio en /path
:switch	Abre el diálogo cambiar. La ruta especifica el directorio de destino.
:export	Exporta la copia de trabajo en /path a otro directorio. Si /path apunta a un directorio sin versionar, aparecerá un diálogo pidiendo una URL para exportar al directorio en /path.
:merge	Abre el diálogo de fusión. La ruta especifica el directorio de destino.
:copy	Abre el diálogo rama/etiqueta. El /path es la copia de trabajo desde la que se creará la rama/etiqueta.
:settings	Abre el diálogo de configuración.
:remove	Elimina el/los fichero/s en /path del control de versiones.
:rename	Renombra el fichero en /path. El nuevo nombre del fichero se pregunta con un diálogo. Para evitar la pregunta sobre renombrar ficheros similares en un paso, utilice /noquestion.
:diff	Inicia el programa externo de diferencias especificado en la configuración de TortoiseSVN. /path especifica el primer fichero. Si se establece la opción /path2, se inicia el programa de diferencias con esos dos ficheros. Si se omite /path2, se hace la diferenciación entre el fichero de /path y su BASE.
:conflicteditor	Inicia el editor de conflictos especificado en la configuración de TortoiseSVN con los ficheros correctos para el fichero en conflicto de /path.
:relocate	Abre el diálogo relocar. /path especifica la ruta de la copia de tra-

Comando	Descripción
	bajo a relocalizar.
:help	Abre el fichero de ayuda.
:repostatus	Abre el diálogo comprobar modificaciones. La ruta especifica el directorio de la copia de trabajo.
:repobrowser	Inicia el diálogo del navegador de repositorios, apuntando a la URL de la copia de trabajo dada en /path o /path apunta directamente a una URL. Puede utilizarse una opción adicional /rev:xxx para especificar la revisión que el navegador de repositorios debe mostrar. Si se omite /rev:xxx, se supone HEAD.
:ignore	Añade todo lo especificado en /path a la lista de ignorados, es decir, añade esos ficheros a la propiedad svn:ignore.
:blame	Abre el diálogo de autoría para el fichero especificado en /path. Si se establecen las opciones startrev y endrev, entonces no se muestra el diálogo pidiendo el rango de autoría sino que se toman los valores de las revisiones desde esas opciones.
:cat	Graba un fichero de una URL o una copia de trabajo dada en /path al lugar dado en /savepath:ruta. La revisión se da en /revision:xxx. Esto puede utilizarse para obtener un fichero con una revisión concreta.
:createpatch	Crea un fichero de parte para la ruta dada en /path.
:revisiongraph	Muestra el gráfico de revisiones para la ruta dada en /path.
:lock	Bloquea un fichero. Se muestra el diálogo 'bloquear' para que el usuario pueda introducir un comentario para el bloqueo. /path
:rebuildiconcache	Reconstruye la caché de iconos de Windows. Utilice esto sólo en el caso de que los iconos de Windows se hayan corrompido. Como efecto secundario de esto (y no puede evitarse) es que los iconos del escritorio se re-colocan. /noquestion

Tabla D.1. List of available commands and options

Ejemplos (que deben introducirse en una única línea):

```
TortoiseProc.exe /command:commit /path:"c:\svn_wc\file1.txt*c:\svn_wc\file2.txt"
/logmsg:"mensaje de registro de prueba" /notempfile /closeonend
```

```
TortoiseProc.exe /command:update /path:"c:\svn_wc\" /notempfile /closeonend
```

```
TortoiseProc.exe /command:log /path:"c:\svn_wc\file1.txt"
/revstart:50 /revend:60 /notempfile /closeonend
```

Apéndice E. Referencia cruzada del interface de línea de comandos

A veces este manual hace referencia a la documentación principal de Subversion, que describe Subversion en términos del Interfaz de Línea de Comandos (CLI). Para ayudarle a entender lo que TortoiseSVN hace tras el escenario, hemos compilado una lista mostrando los comandos CLI equivalentes a cada operación en la Interfaz de Usuario de TortoiseSVN.

Nota

Incluso aunque hay equivalentes en CLI para lo que TortoiseSVN hace, recuerde que TortoiseSVN *no* llama al CLI sino que utiliza la librería de Subversion directamente.

Si cree que ha encontrado un error en TortoiseSVN, puede que le pidamos que intente reproducirlo utilizando el CLI, para que podamos distinguir los problemas de TSVN de los problemas de Subversion. Esta referencia le dice qué comando debe probar.

E.1. Convenciones y reglas básicas

En las descripciones que aparecen a continuación, la URL con el lugar de un repositorio se muestra simplemente como URL, que corresponde por ejemplo a `http://tortoisesvn.tigris.org/svn/tortoisesvn/trunk`. La ruta de la copia de trabajo se muestra simplemente como RUTA, que corresponde por ejemplo a `C:\TortoiseSVN\trunk`.



Importante

Dado que TortoiseSVN es una Extensión del Shell de Windows, no es capaz de utilizar la noción de directorio de trabajo actual. Todas las rutas de las copias de trabajo se deben especificar con rutas absolutas, no relativas.

Algunos ítems son opcionales, y a menudo se controlan con casillas o botones de opciones en TortoiseSVN. Estas opciones se muestran [entre corchetes] en las definiciones de la línea de comandos.

E.2. Comandos de TortoiseSVN

E.2.1. Obtener

```
svn checkout [-N] [--ignore-externals] [-r rev] URL RUTA
```

Si se marca **Sólo obtener la carpeta superior**, utilice la opción `-N`.

Si se marca **Omitir externos**, utilice la opción `--ignore-externals`.

Si está obteniendo una revisión en concreto, especifíquela detrás de la URL utilizando la opción `-r`.

E.2.2. Actualizar

```
svn info URL_de_la_copia_de_trabajo
svn update [-r rev] RUTA
```

A día de hoy, la actualización no es una operación atómica en Subversion. Por tanto, TortoiseSVN primero encuentra la revisión HEAD del repositorio, y luego actualiza todos los ítems a ese número de revisión en concreto, evitando crear así una copia de trabajo mezclada.

Si sólo se selecciona un único ítem para actualizar, o si los ítems seleccionados no son todos del mismo repositorio, TortoiseSVN simplemente actualiza hasta HEAD.

Aquí no se utilizan opciones en la línea de comandos. **Actualizar a la Revisión** también implementa el comando actualizar, pero ofrece más opciones.

E.2.3. Actualizar a la revisión

```
svn info URL_de_la_copia_de_trabajo
svn update [-r rev] [-N] [--ignore-externals] RUTA
```

Si se marca **Sólo obtener la carpeta superior**, utilice la opción `-N`.

Si se marca **Omitir externos**, utilice la opción `--ignore-externals`.

E.2.4. Confirmar

En TortoiseSVN, el diálogo de confirmación utiliza varios comandos Subversion. La primera fase es una comprobación del estado que determina los ítems de su copia de trabajo que potencialmente pueden ser confirmados. Puede revisar la lista, comparar los ficheros contra BASE y seleccionar los ítems que desea incluir en la confirmación.

```
svn status -v RUTA
```

Si se marca **Mostrar archivos no versionados**, TortoiseSVN le mostrará también todos los ficheros y carpetas no versionados en su jerarquía de la copia de trabajo, teniendo en cuenta las reglas de ignorar. Esta característica en concreto no tiene equivalente directo en Subversion, ya que el comando `svn status` no entra en carpetas no versionadas.

Si selecciona cualquier fichero o carpeta no versionado, esos ítems primero se añadirán a su copia de trabajo.

```
svn add RUTA...
```

Cuando pulsa **Aceptar**, tiene lugar la confirmación en Subversion. Si ha dejado todas las casillas de selección de ficheros sin tocar tal y como aparecieron, TortoiseSVN utilizará una única confirmación recursiva en la copia de trabajo. Si ha deseleccionado algún fichero, entonces se debe utilizar una confirmación no-recursiva (`-N`), y se debe especificar individualmente cada ruta en la línea de comandos de la confirmación.

```
svn commit -m "Mensaje de registro" [-N] [--no-unlock] RUTA...
```

`Mensaje de registro` aquí representa los contenidos del cuadro de texto **Mensaje de Registro**. Puede estar vacío.

Si se marca **Mantener Bloqueos**, utilice la opción `--no-unlock`.

E.2.5. Diff

```
svn diff RUTA
```


Si utiliza Diff desde el menú contextual principal, está comparando un fichero modificado contra su revisión BASE. La salida del comando CLI anterior también hace esto y produce la salida en formato diff unificado. Sin embargo, esto no es lo que TortoiseSVN utiliza. TortoiseSVN utiliza TortoiseMerge (o un programa para comparar de su elección) para mostrar las diferencias visualmente entre ficheros de texto completos, por lo que no hay equivalente directo en el CLI.

También puede comparar 2 ficheros cualquiera utilizando TortoiseSVN, estén o no bajo el control de versiones. TortoiseSVN simplemente alimenta con los dos ficheros al programa de diferencias y le deja que él encuentre dónde están las diferencias.

E.2.6. Mostrar registro

```
svn log -v -r 0:N --limit 100 [--stop-on-copy] RUTA
o
svn log -v -r M:N [--stop-on-copy] RUTA
```

Por defecto, TortoiseSVN intenta recoger 100 mensajes de registro utilizando el método `--limit`. Si la configuración establece que utilice las APIs antiguas, se usa la segunda forma para obtener los mensajes de registro de 100 revisiones del repositorio.

Si se marca **Parar en copia/renombrado**, utilice la opción `--stop-on-copy`.

E.2.7. Comprobar modificaciones

```
svn status -v RUTA
o
svn status -u -v RUTA
```

La comprobación de estado inicialmente sólo muestra su copia de trabajo. Si pulsa en **Comprobar repositorio**, se comprueba también el repositorio para ver qué ficheros cambiarían tras una actualización, lo que necesita la opción `-u`.

Si se marca **Mostrar archivos no versionados**, TortoiseSVN le mostrará también todos los ficheros y carpetas no versionados en su jerarquía de la copia de trabajo, teniendo en cuenta las reglas de ignorar. Esta característica en concreto no tiene equivalente directo en Subversion, ya que el comando `svn status` no entra en carpetas no versionadas.

E.2.8. Gráfico de revisión

El gráfico de revisiones es una característica única de TortoiseSVN. No hay equivalente en el cliente de línea de comandos.

Lo que hace TortoiseSVN es

```
svn info URL_de_la_copia_de_trabajo
svn log -v URL
```

donde URL es la *raíz* del repositorio, y luego analiza los datos devueltos.

E.2.9. Navegador de repositorios

```
svn info URL_de_la_copia_de_trabajo
svn list [-r rev] -v URL
```

Puede utilizar `svn info` para determinar la raíz del repositorio, que es el nivel superior mostrado en el navegador de repositorios. No puede navegar más arriba de ese nivel. Además, este comando devuelve toda la información sobre bloqueos que se muestra en el navegador de repositorios.

La llamada `svn list` le mostrará los contenidos de un directorio, cuando se proporciona una URL y una revisión.

E.2.10. Editar conflictos

Este comando no tiene equivalente CLI. Invoca TortoiseMerge o una herramienta externa de fusión/comparación en 3 paneles para mirar los ficheros en conflictos y ver qué líneas hay que utilizar.

E.2.11. Resuelto

```
svn resolved RUTA
```

E.2.12. Renombrar

```
svn rename RUTA_ACTUAL NUEVA_RUTA
```

E.2.13. Eliminar

```
svn delete RUTA
```

E.2.14. Revertir

```
svn status -v RUTA
```

La primera fase es una comprobación del estado que determina los ítems en su copia de trabajo que pueden ser revertidos potencialmente. Puede revisar la lista, comparar los ficheros contra BASE y seleccionar los ítems que desea incluir en la operación de revertir.

Cuando pulsa Aceptar, tiene lugar la operación de revertir en Subversion. Si ha dejado todas las casillas de selección de ficheros sin tocar tal y como aparecieron, TortoiseSVN utilizará una única operación revertir recursivamente (`-R`) en la copia de trabajo. Si ha deseleccionado algún fichero, entonces se debe especificar individualmente cada ruta en la línea de comandos de la operación de revertir.

```
svn revert [-R] RUTA...
```

E.2.15. Limpieza

```
svn cleanup RUTA
```

E.2.16. Obtener bloqueo

```
svn status -v RUTA
```

La primera fase es una comprobación del estado que determina los ficheros en su copia de trabajo que pueden ser potencialmente bloqueados. Puede seleccionar los ítems que desea bloquear.

```
svn lock -m "Mensaje de bloqueo" [--force] RUTA...
```

Mensaje de bloqueo aquí representa los contenidos del cuadro de texto Mensaje de bloqueo. Puede estar vacío.

Si se marca Robar los bloqueos, utilice la opción `--force`.

E.2.17. Quitar bloqueo

```
svn unlock RUTA
```

E.2.18. Ramas / Etiqueta

```
svn copy -m "Mensaje de registro" URL URL
o
svn copy -m "Mensaje de registro" URL@rev URL@rev
o
svn copy -m "Mensaje de registro" RUTA URL
```

El diálogo Rama/Etiqueta hace una copia en el repositorio. Hay 3 botones de opciones:

- Revisión HEAD en el repositorio
- Revisión específica en el repositorio
- Copia de trabajo

que corresponden a las 3 variantes de línea de comandos anteriores.

Mensaje de registro aquí representa los contenidos del cuadro de texto Mensaje de Registro. Puede estar vacío.

E.2.19. Cambiar

```
svn info URL_de_la_COPIA_DE_TRABAJO
svn switch [-r rev] URL RUTA
```

E.2.20. Fusionar

```
svn merge [--dry-run] --force URL_Origen@revN URL_Destino@revM RUTA
```

Simular hace la misma fusión con la opción `--dry-run`.

```
svn diff URL_Origen@revN URL_Destino@revM
```

Diff unificado muestra la operación de diferenciación que se utilizará para hacer la fusión.

E.2.21. Exportar

```
svn export [-r rev] [--ignore-externals] URL RUTA_Exportación
```

Esta forma se utiliza cuando se accede desde una carpeta sin versionar, y la carpeta se utiliza como el destino.

La exportación de una copia de trabajo a otro lugar se hace sin utilizar las bibliotecas de Subversion, por lo que no hay línea de comandos equivalente.

Lo que TortoiseSVN hace es copiar todos los ficheros al nuevo destino mientras le muestra el progreso de la operación. Opcionalmente, pueden exportarse también los ficheros y/o carpetas sin versionar.

En ambos casos, si se marca **Omitir externos**, utilice la opción `--ignore-externals`.

E.2.22. Relocalizar

```
svn switch --relocate URL_Origen URL_Destino
```

E.2.23. Crear repositorio aquí

```
svnadmin create --fs-type fsfs RUTA
o
svnadmin create --fs-type bdb RUTA
```

E.2.24. Añadir

```
svn add RUTA...
```

Si había seleccionado una carpeta, en primer lugar TortoiseSVN la escanea recursivamente en busca de ítems que puedan ser añadidos.

E.2.25. Importar

```
svn import -m "Mensaje de registro" RUTA URL
```

Mensaje de registro aquí representa los contenidos del cuadro de texto Mensaje de Registro. Puede estar vacío.

E.2.26. Autoría

```
svn blame -r N:M -v RUTA
svn log -r N:M RUTA
```

Si utiliza TortoiseBlame para ver la información de autoría, se necesita también el fichero de registro para mostrar los mensajes de registro en un texto de ayuda. Si visualiza la autoría como un fichero de texto, no se necesita esta información.

E.2.27. Añadir a la lista de ignorados

```
svn propget svn:ignore RUTA > ficherotemporal
{editar los nuevos ítems a ignorar en el fichero ficherotemporal}
svn propset svn:ignore -F ficherotemporal RUTA
```

Dado que `svn:ignore` a menudo tiene un valor con varias líneas, aquí se muestra cómo se cambiaría utilizando un fichero de texto mejor que hacerlo directamente en la línea de comandos.

E.2.28. Crear parche

```
svn diff RUTA > fichero_parche
```

TortoiseSVN crea un fichero de parche en formato diff unificado comparando la copia de trabajo con su versión BASE.

E.2.29. Aplicar parche

Aplicar parches es un trabajo complicado a no ser que el parche y la copia de trabajo estén en la misma revisión. Afortunadamente, puede utilizar TortoiseMerge, que no tiene equivalente directo en Subversion.

Glosario

Añadir	Un comando de Subversion que se utiliza para añadir un fichero o un directorio a su copia de trabajo. Los nuevos ítems se añaden al repositorio cuando confirme.
revisión BASE	La revisión base actual de un fichero o una carpeta en su <i>copia de trabajo</i> . Esta es la revisión en la que estaba el fichero o la carpeta, cuando se realizó la última operación de obtener, actualizar o confirmar. La revisión BASE normalmente no es la misma que la revisión HEAD.
Autoría	Este comando es sólo para ficheros de texto, y anota cada línea para mostrar la revisión del repositorio en la que se cambió por última vez, y el autor que hizo ese cambio. Nuestra implementación GUI se llama TortoiseBlame y también le muestra la fecha y hora de la confirmación y el mensaje de registro cuando mueve el ratón por encima del número de revisión.
BDB	Base de datos Berkeley. Un soporte de base de datos muy probado para los repositorios, que no puede utilizarse en unidades compartidas de red. Por defecto para repositorios anteriores a la versión 1.2.
Rama	Un término de uso frecuente en los sistemas de control de versiones para describir lo que ocurre cuando el desarrollo se parte en un punto en concreto y sigue dos caminos separados. Puede crear una rama desde la línea principal de desarrollo para desarrollar una nueva funcionalidad sin hacer que la línea principal quede inestable. O puede hacer una rama de una versión estable en la que sólo hará correcciones de errores, mientras los nuevos desarrollos tienen lugar en el inestable tronco. En Subversion una rama se implementa como una “copia barata”.
Obtener	Un comando de Subversion que crea una copia local de trabajo en un directorio vacío al descargar los ficheros versionados desde el repositorio.
Limpiar	Copiando el libro de Subversion: “Limpia recursivamente la copia de trabajo, eliminando los bloqueos y continuando las operaciones sin terminar. Si alguna vez obtiene un error copia de trabajo bloqueada, ejecute este comando para eliminar bloqueos robados y conseguir que su copia de trabajo vuelva a un estado estable de nuevo.” Tenga en cuenta que en este contexto “bloqueado” se refiere a un bloqueo del sistema de ficheros local, no a bloqueos del repositorio.
Confirmar	Este comando de Subversion se utiliza para pasar los cambios desde su copia de trabajo local al repositorio, creando una nueva revisión en el repositorio.
Conflicto	Cuando los cambios desde el repositorio se mezclan con los cambios locales, a veces esos cambios ocurren en las mismas líneas. En este caso Subversion no puede decidir automáticamente qué versión utilizar, y se dice que el fichero está en conflicto. Tiene que editar el fichero manualmente y resolver el conflicto antes de que pueda confirmar más cambios.
Copiar	En un repositorio de Subversion puede crear una copia de un único fichero o un árbol completo. Estos se implementan como

	“copias baratas” que actúan un poco como un enlace al original en el sentido de que casi no ocupan espacio. Haciendo una copia preserva la historia del ítem en la copia, por lo que puede rastrear los cambios que se hicieron antes de que ocurriera la copia.
Eliminar	Cuando elimina un ítem versionado (y confirma el cambio), el ítem ya no existirá nunca más en el repositorio después de la revisión confirmada. Pero por supuesto aún existe en las revisiones previas en el repositorio, por lo que aún puede acceder a él. Si es necesario, puede copiar un ítem eliminado y “resucitarlo” por completo con historia.
Diferenciar	Abreviatura de “Mostrar diferencias”. Muy útil cuando desea ver exactamente qué cambios se han hecho.
Exportar	Este comando produce una copia de una carpeta versionada, igual que una copia de trabajo pero sin los directorios locales <code>.svn</code> .
FSFS	Sistema de ficheros FS. Un sistema de ficheros propietario de Subversion que se utiliza como soporte de los repositorios. Se puede utilizar en unidades compartidas de red. Por defecto para los repositorios a partir de la versión 1.2.
GPO	Objeto de política de grupo
revisión HEAD	La última revisión de un fichero o una carpeta en el <i>repositorio</i> .
Importar	Comando de Subversion para importar una jerarquía de carpetas completa en el repositorio en una única revisión.
Bloquear	Cuando obtiene un bloqueo de un ítem versionado, lo marca en el repositorio como no confirmable, excepto para la copia de trabajo desde la que se obtuvo el bloqueo.
Registro	Muestra la historia de las revisiones de un fichero o carpeta. También se conoce como “Historial”.
Historial	Muestra la historia de las revisiones de un fichero o carpeta. También se conoce como “Registro”.
Fusionar	El proceso por el cual los cambios del repositorio se añaden a su copia de trabajo sin interrumpir cualquier cambio que haya realizado ya localmente. A veces estos cambios no se pueden reconciliar automáticamente, y se dice que la copia de trabajo está en conflicto. Las fusiones ocurren automáticamente cuando actualiza su copia de trabajo. También puede fusionar cambios específicos desde otras ramas utilizando el comando Fusionar de TortoiseSVN.
Parche	Si una copia de trabajo tiene cambios únicamente en ficheros de texto, es posible utilizar el comando Diff de Subversion para generar un único fichero con el resumen de esos cambios en el formato de Diff unificado. Un fichero de este tipo a menudo se denomina “Parche”, y puede ser enviado por email a otro (o a una lista de correo) y aplicado en otra copia de trabajo. Alguien sin acceso de confirmación puede hacer cambios y enviar un fichero de parche para que lo aplique un confirmador autorizado. O si no está seguro de un cambio, puede enviar el parche a otros para que lo revisen.
Propiedad	Además de versionar sus directorios y ficheros, Subversion le

		<p>permite añadir metainformación versionada - llamadas “propiedades” para cada uno de sus directorios y ficheros versionados. Cada propiedad tiene un nombre y un valor, como un entrada del registro. Subversion tiene algunas propiedades especiales que utiliza internamente, como <code>svn:eol-style</code>. TortoiseSVN tambitsvn:logminsize. Puede a</p>
Relocalizar		<p>Si su repositorio se mueve, quizás porque lo ha movido a un directorio diferente en su servidor, o el nombre de dominio del servidor ha cambiado, necesitará “relocalizar” su copia de trabajo para que sus URLs de repositorio apunten al nuevo lugar.</p> <p>Nota: sólo debería utilizar este comando si su copia de trabajo apunta al mismo lugar en el mismo repositorio, pero es el propio repositorio el que se ha movido. En cualquier otra circunstancia probablemente necesita el comando “Cambiar” en su lugar.</p>
repositorio		<p>Un repositorio es un lugar central donde se almacenan y mantienen los datos. Un repositorio puede ser un lugar donde se encuentran múltiples bases de datos o ficheros para distribuirlos en una red, o un repositorio puede ser un lugar directamente accesible por el usuario sin tener que viajar por una red.</p>
Resolver		<p>Cuando los ficheros en una copia de trabajo se quedan en un estado de conflicto tras una fusión, estos conflictos deben arreglarse por un humano utilizando un editor (o quizás TortoiseMerge). Este proceso se denomina “Resolviendo conflictos”. Cuando finalice, marque los ficheros con conflictos como resueltos, y eso le permitirá confirmarlos.</p>
Revertir		<p>Subversion keeps a local “pristine” copy of each file as it was when you last updated your working copy. If you have made changes and decide you want to undo them, you can use the “revert” command to go back to the pristine copy.</p>
Revisión		<p>Cada vez que confirma un conjunto de cambios, crea una nueva “revisión” en el repositorio. Cada revisión representa el estado del árbol del repositorio en un cierto punto de su historia. Si desea volver en el tiempo para examinar el repositorio tal y como estaba en la revisión N.</p> <p>En otro sentido, una revisión puede referirse al conjunto de cambios que se hicieron cuando esa revisión se creó.</p>
Propiedad (revprop)	de revisión	<p>Al igual que los ficheros pueden tener propiedades, las revisiones del repositorio también. Algunas revprops especiales se añaden automáticamente cuando se crea la revisión; son: <code>svn:date</code> <code>svn:author</code> <code>svn:log</code> que representan la fecha/hora de la confirmación, el confirmador y el mensaje de registro, respectivamente. Estas propiedades pueden editarse, pero no están versionadas, por lo que cualquier cambio es permanente y no puede deshacerse.</p>
SVN		<p>Una abreviatura muy usada para Subversion.</p> <p>El nombre del protocolo personalizado de Subversion que utiliza el servidor de repositorios “svnserve”.</p>
Cambiar		<p>De la misma forma que “Actualizar-a-la-revisión” cambia la ventana temporal de una copia de trabajo para que quede como estuvo en un punto diferente de su historia, “Cambiar” cambia la ventana espacial de una copia de trabajo para que apunte a una parte diferente del repositorio. Es particularmente útil cuando</p>

	esté trabajando en troncos y ramas en los que sólo hay unos pocos ficheros diferentes. Puede cambiar su copia de trabajo entre los dos y sólo se transferirán los ficheros cambiados.
Actualizar	Este comando de Subversion incorpora los últimos cambios del repositorio a su copia de trabajo, fusionando cualquier cambio hecho por otros con los cambios locales en la copia de trabajo.
Copia de trabajo	Esta es su “caja de arena” local, el área donde trabaja con los ficheros versionados, y normalmente se encuentra en su disco duro local. Puede crear una copia de trabajo al “Obtener” desde un repositorio, y devolver sus cambios al repositorio utilizando “Confirmar”.

Índice

Símbolos

Índice de proyectos, 19

A

Acceso, 32
acceso directo, 123
actualizar, 49, 120
alabanza, 91
anotar, 91
Apache, 16
arrastrar-y-soltar, 36
autenticación, 37
Autenticación múltiple, 23
automatización, 127
Autorización, 21
autoría, 91
añadir, 67

B

bloqueando, 86

C

cambiar, 81
carpeta .svn, 115
carpeta _svn, 115
CLI, 130
cliente de línea de comandos, 130
Columnas del explorador, 53
comparar ficheros, 122
comparar revisiones, 65
comprobación de actualización, 125
configuración, 100
confirmar, 44
conflicto, 9, 50
control de errores, 98
control de versiones, 1
controlador de dominio, 22
controlando errores, 98
copia de seguridad, 32
copia de trabajo, 10
copiar, 36, 79, 93
corrector ortográfico, 3
Crear
 Cliente de línea de comandos, 31
 TortoiseSVN, 31

D

deshacer, 72, 121
desversionar, 123
diccionario, 3
diferenciando, 55
diferenciar, 63
diferenciar imágenes, 65
dominio de Windows, 22

E

editar registro/autor, 60
eliminar, 70, 70
enlace de obtener, 33
estado, 51, 53
estadísticas, 61
etiqueta, 79
expansión de comodines, 69
explorador, 1
exportar, 96
exportar cambios, 65
externos, 42
extracción de versión, 116

F

ficheros especiales, 41
ficheros temporales, 40
filtro, 60
fusionar, 82

G

ganchos, 33
GPO, 125
gráfico, 94

H

herramientas de diferenciación, 66
herramientas de fusión, 66
historial, 55

I

iconos, 51
ignorar, 68
ignorar global, 101
importar, 38
instalación, 3

L

Libro de Subversion, 5
limpieza, 73
línea de comandos, 127

M

manejador de arrastre, 36
mensaje de registro, 120
Menú contextual, 34
modificaciones, 53
mod_authz_svn, 17, 21
mover, 70
moviendo, 120

N

navegador, 93

O

obtener, 43

P

packs de idioma, 3
parche, 89

patrón de ignorar, 101
políticas de grupo, 125
propiedades, 74
proyectos comunes, 122

R

rama, 79
registro, 55, 115
relocalizar, 97
renombrar, 70
repositorio, 5, 38
repositorios externos, 42
revertir, 72
revisión, 13, 94
right drag, 36
rutas UNC, 32

S

seguimiento de incidencias, 98
server-side actions, 93
servidor proxy, 110
shell de Windows, 1
sobreimpresiones, 51
sonidos, 100
SSL, 24
SSPI, 22
SVNParentPath, 18, 19
SVNPath, 18
svnserve, 26
svnservice, 28
SVN_ASP_DOT_NET_HACK, 126

T

TortoiseIDiff, 65
traducciones, 3

W

WebDAV, 16