

TortoiseMerge

**Una herramienta de diferencias/
fusión para Windows**

Versión 1.6.16

**Stefan Küng
Lübbe Onken
Simon Large**

TortoiseMerge: Una herramienta de diferencias/fusión para Windows: Versión 1.6.16

por Stefan Küng, Lübbe Onken, y Simon Large

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

publicado 2011/01/21 21:21:17 (r20750)

Tabla de contenidos

Prefacio	vi
1. ¡TortoiseMerge es gratis!	vi
2. Reconocimientos	vi
1. Introducción	1
1.1. Descripción	1
1.2. Historia de TortoiseMerge	2
2. Conceptos Básicos	3
2.1. Viendo y fusionando diferencias	3
2.2. Editando conflictos	3
2.3. Aplicando parches	4
3. Usando TortoiseMerge	5
3.1. Modos de visión	5
3.1.1. Viendo / Fusionando	5
3.1.2. Aplicando parches	7
3.2. Botones y otros controles	7
3.3. Iconos de estado de línea	8
3.4. Fusionando / Editando conflictos	9
3.5. Abrir ficheros	10
3.5.1. Viendo / Fusionando	10
3.5.2. Aplicando parches	10
3.6. Configuración	11
3.6.1. Página de Configuración principal	11
3.6.2. Página de Configuración de colores	12
A. Atajos de teclado	14
A.1. Atajos de teclado	14
B. Automatizando TortoiseMerge	15
B.1. Opciones de Línea de Comandos de TortoiseMerge	15
Glosario	17

Lista de figuras

1.1. Conflicto de ficheros	1
2.1. Conflicto de ficheros	4
3.1. Vista de un panel	5
3.2. Vista de dos paneles	5
3.3. Vista de tres paneles	6
3.4. Lista de ficheros de parche	7
3.5. La barra de herramientas	7
3.6. El diálogo Abrir	10
3.7. La página de Configuración principal	11
3.8. La página de Configuración de colores	12

Lista de tablas

B.1. Lista de las opciones de línea de comandos disponibles	15
---	----

Prefacio



Tortoise Merge

Cuando trabaja en un proyecto, bien sea un proyecto de código abierto donde varias personas revisan su código, o un proyecto comercial donde vende un componente con su código fuente, habrá gente y clientes que pueden encontrar errores o quizás quieran mejorar su proyecto. Para hacerlo, ellos le enviarán parches que tendrá que revisar y decidir si desea aplicarlos.

TortoiseMerge le ayuda con ambas tareas:

- Revisando parches
- Aplicando los parches

TortoiseMerge también le ayuda a poner en orden los ficheros con conflictos, que pueden ser ocasionados si trabaja con un sistema de control de código fuente y otros están trabajando en los mismos ficheros que usted.

1. ¡TortoiseMerge es gratis!

TortoiseMerge es gratis. No tiene que pagar por él, y puede utilizarlo de la forma que desee. Se desarrolla bajo la licencia GP (GPL).

TortoiseMerge es un proyecto de Código Abierto. Esto significa que tiene acceso total de lectura del código fuente del programa. Puede verlo en este vínculo <http://code.google.com/p/tortoisesvn/source/browse/>. Se le pedirá que introduzca un nombre de usuario y una contraseña. El usuario es `guest`, y la contraseña debe dejarse en blanco. La versión más reciente (en la que estamos trabajando actualmente) se encuentra bajo `/trunk/`, y las versiones liberadas se encuentran bajo `/tags/`.

¡Donar!

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.

2. Reconocimientos

Stefan Küng

por el duro trabajo en TortoiseMerge

Lübbe Onken

por los bonitos iconos, el logotipo, detección de errores y cuidar de la documentación

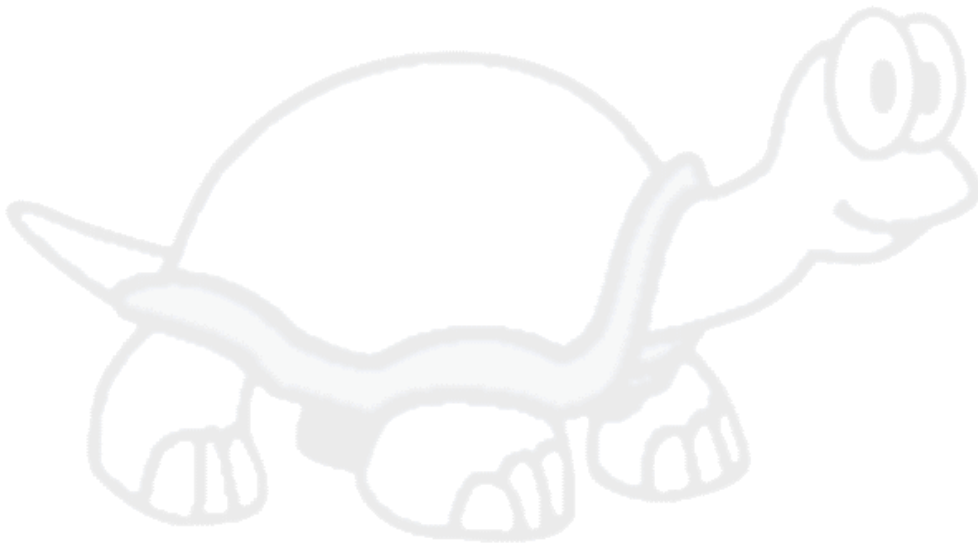
El proyecto Tigris Style

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



Capítulo 1. Introducción

1.1. Descripción

TortoiseMerge es una aplicación gratuita/de código abierto. Le permite ver las diferencias entre ficheros de texto, fusionar esos cambios e incluso revisar y aplicar ficheros de diff unificados, a menudo llamados *parches*.

Mientras trabaja con ficheros de texto, por ejemplo ficheros de código en su lenguaje de programación favorito, o ficheros html/xml para su documentación o sitio web, a menudo tendrá situaciones donde desee comparar versiones diferentes de esos ficheros. A veces obtiene una versión diferente de alguien, a veces querrá ver los cambios que ha hecho.

Si está trabajando con un sistema de control de versiones (por ejemplo Subversion) a veces obtendrá *conflictos* cuando actualice su copia de trabajo. Esto ocurre si alguien ha modificado las mismas partes que usted del fichero en el que está trabajando. En esos casos tendrá que resolver esos conflictos manualmente. Aquí es donde TortoiseMerge puede ayudarle.

El siguiente diagrama le muestra las relaciones entre los ficheros involucrados en un conflicto:

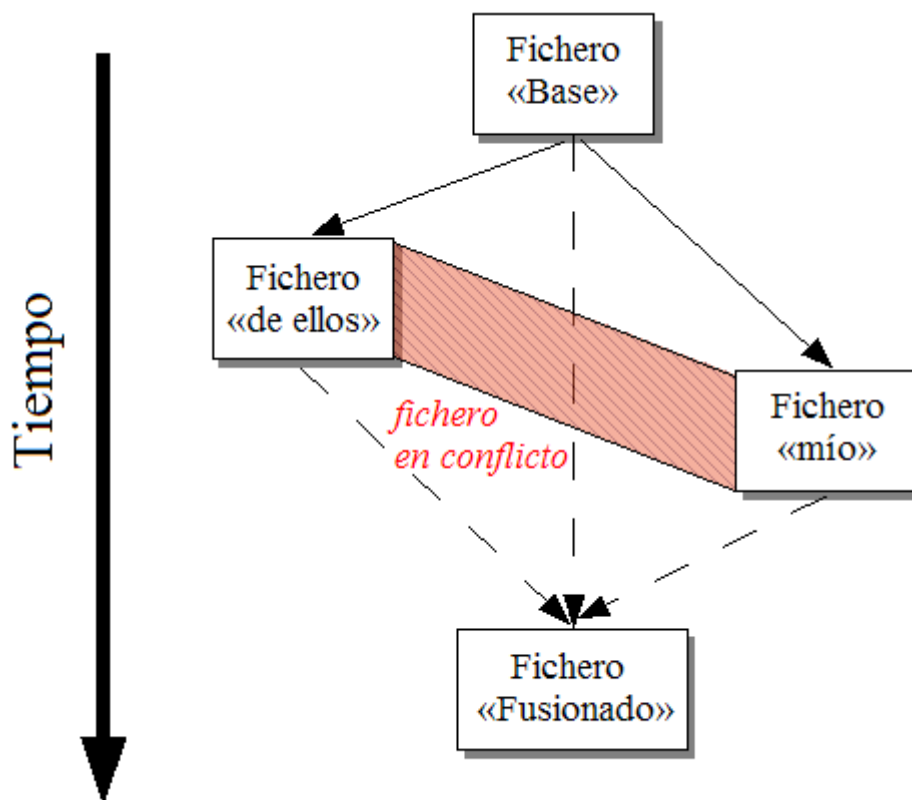


Figura 1.1. Conflicto de ficheros

Un conflicto también puede ocurrir si intenta aplicar un parche desactualizado. En estos casos, sin TortoiseMerge, tendría que pedir a la persona de la que obtuvo el parche una versión actualizada, o bien intentar aplicar el fichero de parche manualmente.

Pero ahora con TortoiseMerge ya no es necesario hacer eso - al menos no si está trabajando con un sistema de control de versiones como Subversion. TortoiseMerge parsea el fichero de parche buscando la revisión/versión sobre la que se creó el parche y automáticamente intenta obtener esa versión específica

de su control de código. Luego aplica el parche y le deja revisar tanto los cambios de usted en el fichero como los de los otros. Luego puede decidir qué cambios desea mantener.

1.2. Historia de TortoiseMerge

Mientras trabajamos en *TortoiseSVN* [<http://tortoisesvn.tigris.org>] hemos visto que cuando los usuarios nos enviaban parches para mejorar el proyecto o para arreglar errores, nos llevaba mucho tiempo aplicar esos parches. Y la mayor parte del tiempo esos parches estaban desactualizados y no se podían aplicar con la herramienta de línea de comandos patch <http://unxutils.sourceforge.net/> porque nosotros estábamos trabajando en esos ficheros y a veces ya habíamos cambiado esos ficheros.

Esta era también la razón principal por la que TortoiseSVN no tenía implementada una opción TortoiseSVN → Crear Parche: preferíamos que los usuarios nos enviaran los ficheros completos porque así era más fácil aplicar los parches.

Así que gastamos mucho tiempo buscando en internet una herramienta GUI que pudiera aplicar ficheros de parche, pero todo lo que encontramos es que dicha herramienta no existía. Para acortar un poco una larga historia: decidimos que si nadie más podía proporcionarnos dicha herramienta tendríamos que escribirla nosotros. Ahí fue cuando nació TortoiseMerge.

Dado que TortoiseSVN ya utiliza la librería de Subversion, que contiene un buen motor de diferencias, era natural que utilizáramos esa librería en vez del diff de GNU.

Capítulo 2. Conceptos Básicos

TortoiseMerge tiene diferentes usos:

- Viendo las diferencias entre dos ficheros y fusionando los cambios en uno de ellos, o eliminando cambios no deseados.
- Editando conflictos entre cambios locales y cambios en el repositorio de Subversion tras una actualización.
- Aplicando ficheros de parche. Esta es la razón por la cual se creó TortoiseMerge :)

2.1. Viendo y fusionando diferencias

En este modo está comparando dos ficheros. El fichero del panel izquierdo se considera el fichero original (a veces referido como *de ellos*) y el fichero en el panel derecho es el fichero modificado (a veces referido como *mío*).

Puede efectuar cambios simples basados-en-línea al fichero en el panel derecho, entre otros:

- Revirtiendo líneas cambiadas a como se muestran en el panel izquierdo.
- Utilizando una combinación de ambos bloques, bien el *de ellos* antes del *mío*, o bien el *mío* antes que el *de ellos*.

También puede editar el fichero de salida tal y como lo haría en un editor de texto. Las líneas que cambie de esta forma se marcarán utilizando un icono de un lápiz. Por favor, tenga en cuenta que si desea hacer alguno de los cambios basados en línea o bloque descritos anteriormente, es mejor que lo haga al principio: una vez que empiece a editar el fichero, TortoiseMerge no podrá mantener la relación de las líneas con los ficheros originales.

2.2. Editando conflictos

Esto a veces se denomina fusión a tres bandas, y se muestra en tres paneles. Aunque realmente hay cuatro ficheros involucrados. El fichero que se muestra es el fichero base común, el último ancestro común de los dos ficheros que ahora están en conflicto. Las relaciones entre los tres ficheros se explican en el diagrama siguiente:

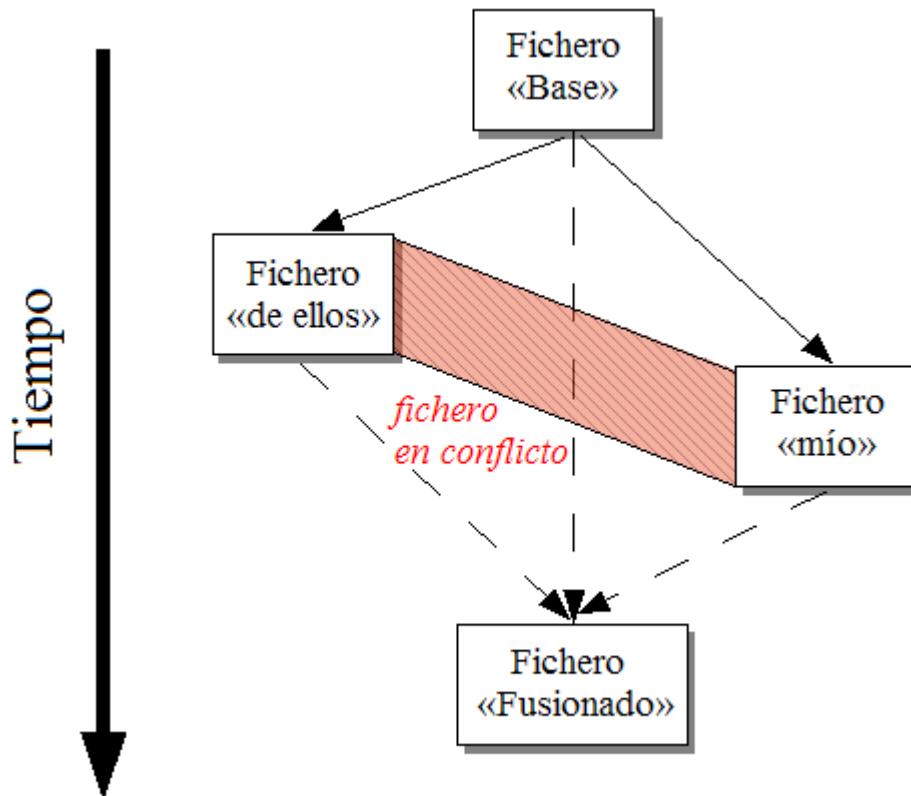


Figura 2.1. Conflicto de ficheros

El fichero base representa la versión más antigua del fichero, desde donde *Usted* y *Otros* empezaron a hacer cambios. *Mío* representa el fichero base con todos los cambios que *usted* ha hecho, y *De otros* es el fichero con todos los cambios que *cualquier otro* ha hecho al fichero. El panel izquierdo le muestra los cambios en *de otros* respecto a su fichero base, y el panel derecho le muestra los cambios en *mío* relativos al fichero base. El panel inferior es el fichero resultado que es donde intenta resolver los conflictos.

En la vista de resolución de conflictos, puede elegir utilizar bloques *míos*, o *de ellos*, o ambos. Pero en este caso, los cambios se muestran en el panel inferior.

2.3. Aplicando parches

Un fichero de parche es un fichero de diff unificado de Subversion, que contiene la información necesaria para aplicar cambios a un conjunto de ficheros. El parche puede haber sido proporcionado por otro desarrollador para que pueda ver sus cambios y posiblemente confirmarlos en el repositorio. O puede haber sido generado internamente por TortoiseSVN cuando compara dos carpetas. Esto ocurre cuando compara una revisión del repositorio con una carpeta de copia de trabajo, o si compara dos revisiones diferentes del repositorio, o si compara dos rutas de repositorio diferentes.

En cada caso TortoiseMerge mostrará una pequeña ventana mostrando los ficheros incluidos en el parche. Al hacer doble click en uno de estos ficheros, se obtiene el fichero relevante y se aplican los cambios. El panel izquierdo muestra el contenido original del fichero y el panel derecho muestra cómo queda tras aplicar el parche.

Puede editar el fichero en el panel derecho exactamente igual que lo haría en el modo de comparación/edición.

Capítulo 3. Usando TortoiseMerge

3.1. Modos de visión

TortoiseMerge tiene tres modos de vista principales: un panel, dos paneles y tres paneles. La vista de uno o dos paneles se utiliza para ver cambios, y la vista de tres paneles se utiliza para resolver conflictos.

3.1.1. Viendo / Fusionando

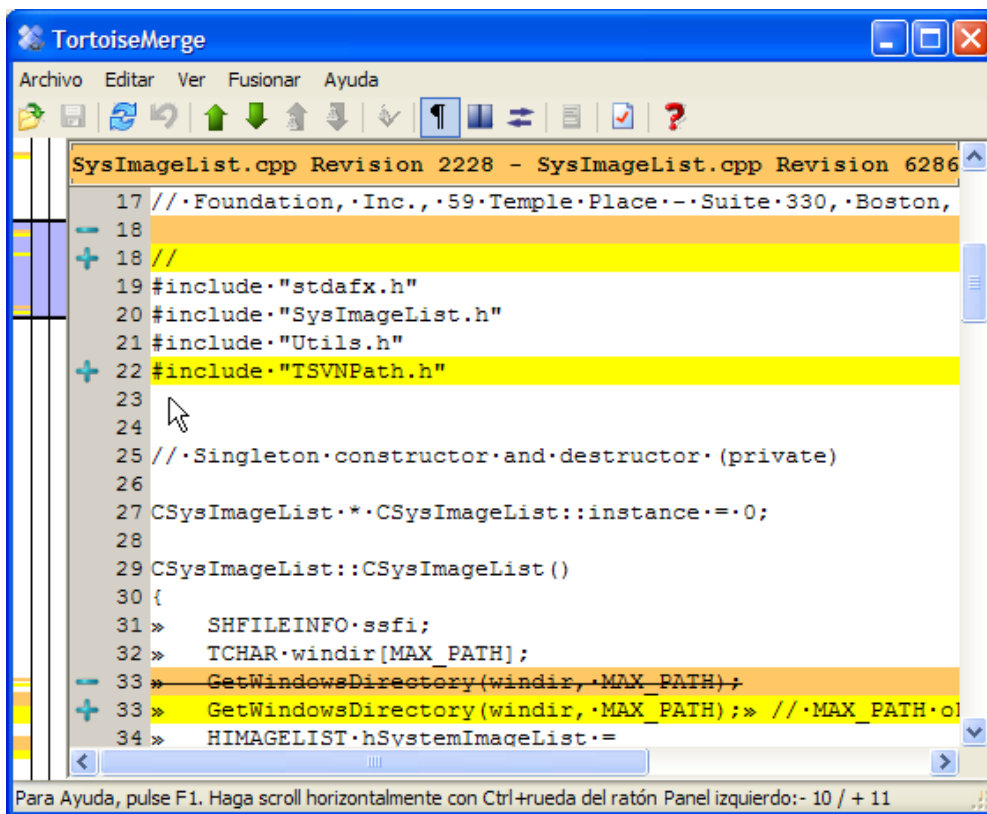


Figura 3.1. Vista de un panel

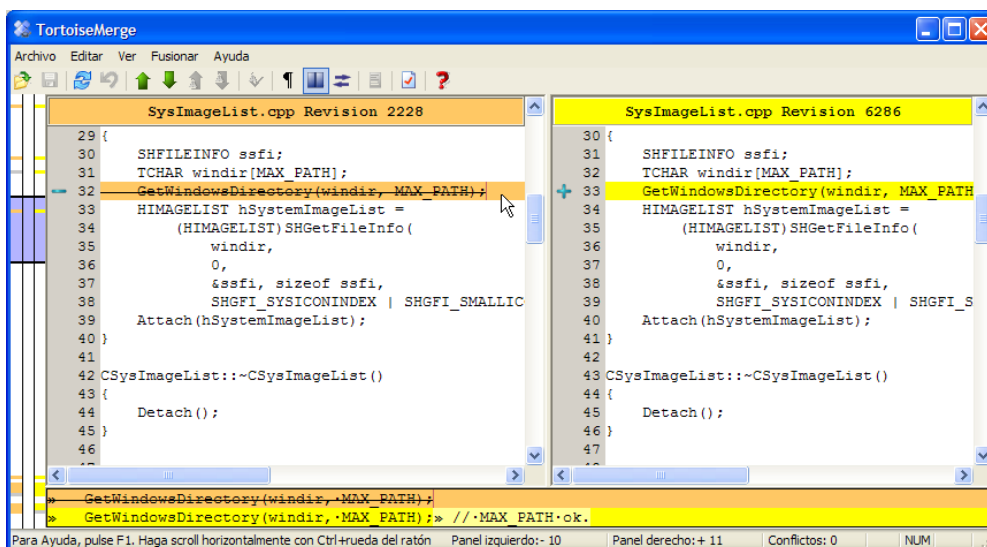


Figura 3.2. Vista de dos paneles

La vista de dos paneles tiene algunas funcionalidades que no están disponibles en la vista de un panel:

- Los cambios dentro de las líneas modificadas se muestran con colores diferentes. Las partes añadidas en la cadena se muestran con un color claro, aunque por supuesto puede configurar los colores utilizados aquí. Las partes eliminadas se indican con una línea vertical marrón oscura en la cadena. Observe la captura de pantalla anterior para ver cómo funciona.
- Reorganizar código a menudo significa un montón de cambios en los espacios en blanco (espacios, tabuladores, nuevas líneas) pero no cambios reales en el código. Por ejemplo, divide una línea muy larga en varias líneas, o a veces junta varias líneas en una única.

Dichos cambios se marcan con un círculo blanco en la parte izquierda de las vistas. Si observa dicho círculo blanco, inmediatamente sabrá que no se ha cambiado realmente nada ahí y no tendrá que seguir comprobando el bloque cambiado.

- En la vista de dos paneles, es posible la edición manual del fichero en el panel derecho. Esos cambios se marcarán con un símbolo de lápiz.

Si quiere comparar/fusionar tres ficheros, TortoiseMerge le mostrará las diferencias en una vista de tres paneles. Esta vista también se utiliza si necesita resolver ficheros en conflicto.

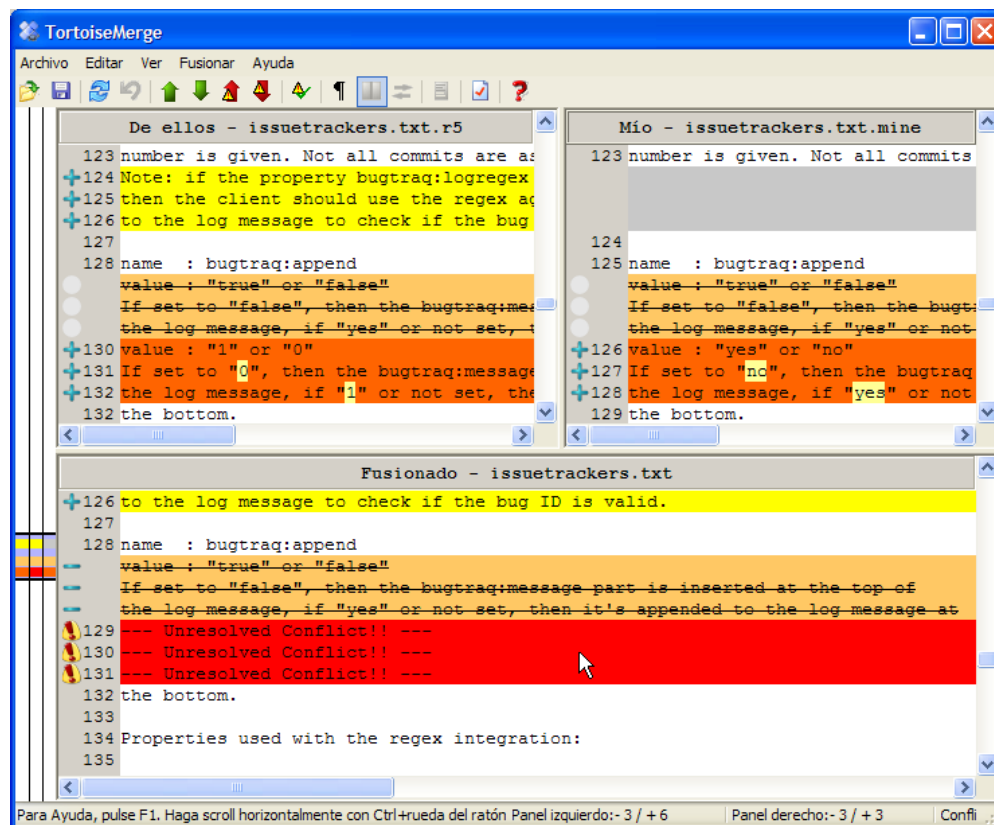


Figura 3.3. Vista de tres paneles

El panel izquierdo le muestra las diferencias entre el fichero *De otros* y el fichero *Base*, mientras que el panel derecho le muestra las diferencias entre el fichero *Mío* y el fichero *Base*. El panel inferior le muestra el resultado de fusionar los ficheros *Base*, *M* y *el De otros* con posibles conflictos.

Si pasa el ratón por encima del título del panel, una etiqueta de ayuda le mostrará los nombres de ficheros utilizados para las diferencias en cada panel.

3.1.2. Aplicando parches

Cuando TortoiseMerge termine de parsear el fichero de parche, le mostrará una pequeña ventana con todos los ficheros que tienen cambios según el fichero de parche.

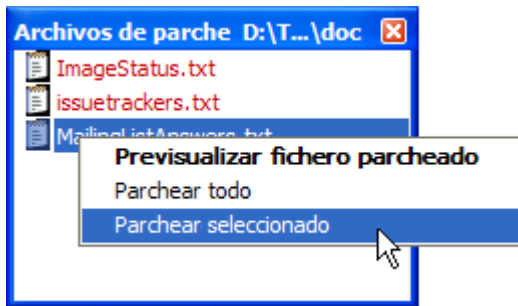


Figura 3.4. Lista de ficheros de parche

Si el nombre del fichero se muestra en negro, el parche se puede aplicar sin problemas. Eso significa que el fichero no está desactualizado según el parche. Sin embargo, si el nombre del fichero se muestra en rojo, entonces el parche no se puede aplicar directamente porque ya ha cambiado ese fichero.

Pero no se preocupe. Si trabaja con Subversion puede recuperar una versión antigua del fichero y resolver el conflicto. De hecho, TortoiseMerge automáticamente intenta obtener la versión antigua del fichero y le enseña las diferencias entre *Base* (la versión antigua del fichero), *Mío* (su fichero tal y como está ahora en su copia de trabajo) y *De otros* (la versión antigua con el parche ya aplicado). Entonces puede resolver los posibles conflictos manualmente.

La ventana de ficheros de parche tiene un menú contextual que le permite previsualizar el efecto del parche en el fichero actual (aplicar sin guardar), aplicar y guardar los cambios para el fichero seleccionado, o aplicar y guardar los cambios a *todos* los ficheros de la lista. La acción para el doble-click es previsualizar.

Dependiendo de si TortoiseMerge puede aplicar el parche directamente o bien obteniendo la versión anterior del fichero antes, le mostrará una vista de dos paneles (el parche se puede aplicar directamente) o una vista de tres paneles (se obtuvo una versión anterior del fichero).

3.2. Botones y otros controles



Figura 3.5. La barra de herramientas

Bajo la barra de menú está la barra de herramientas. Ésta le proporciona fácil acceso a la mayoría de las herramientas más utilizadas, que son

- **Abrir** accede al diálogo de abrir fichero. ¿Qué esperaba?
- **Grabar** almacena cualquier cambio que haya realizado.
- **Recargar** descarta cualquier cambio que haya hecho en TortoiseMerge y recarga los ficheros desde el disco.
- **Deshacer** implementa una función deshacer multinivel, lo que le permite revertir cualquier cambio que haya hecho con TortoiseMerge.
- **Siguiente/Anterior Diferencia** salta directamente al siguiente punto donde difieren los ficheros. Esto le ahorra el tener que hacer scroll por todas las partes sin cambios de los ficheros.

Cuando hace esto en el modo de diferencias de 2 paneles, la barra de detalles en la parte inferior de la pantalla le mostrará las diferencias dentro de la primera línea, sin importar sobre dónde esté el ratón.

- **Siguiente/Anterior Conflicto** salta directamente al siguiente punto donde las diferencias de los ficheros están en conflicto y debe resolverse.
- **Usar el suyo/Usar el mío** selecciona si desea usar el bloque del panel izquierdo o derecho. Lea la siguiente sección para más información sobre cómo editar cambios y conflictos.
- **Suyo antes del mío/Mío antes del suyo** se utiliza si desea combinar los cambios en un fichero en conflicto. Lea la siguiente sección para más información sobre cómo editar cambios y conflictos.
- **Marcar como Resuelto** Cuando ha resuelto todos los conflictos en un fichero y grabados los cambios, puede marcarlo como resuelto desde TortoiseMerge, en vez de volver al explorador y marcarlo como resuelto desde allí.
- **Mostrar espacios en blanco** cambia la opción para mostrar los caracteres de espacio y tabulador como símbolos, para que pueda distinguir los cambios en los espacios en blanco.
- **Diff en línea por-palabra** cambia el método utilizado para mostrar diferencias en una línea. En el modo por-palabra, cuando se cambia un carácter, se marca toda la palabra. En el modo por-carácter, se marca cada cambio individual.

El modo por-palabra es el modo por defecto y en la mayoría de los casos funciona mejor. por ejemplo, si cambia el nombre de una variable, se marca el nombre completo en el modo por-palabra, mientras que el modo por-carácter tiene a mostrarle un montón de caracteres añadidos y borrados, lo que no tiene mucho sentido. Sin embargo, algunos contenidos no forman realmente palabras, como por ejemplo ficheros hexadecimales, y en esos casos el modo por-carácter funciona mejor.

- **Cambiar Vista** selecciona entre la vista de un panel y de dos paneles. Está deshabilitado para la vista de 3 paneles.
- **Cambiar Vistas Izquierda y Derecha** En la vista de dos paneles, intercambia las ventanas izquierda y derecha. Cuando está comparando un fichero en 2 árboles diferentes, en vez de comparar 2 revisiones del mismo fichero, TortoiseSVN no sabe cuál debería ir en el panel izquierdo como versión “original/base”. Si lo hace mal, puede cambiarlo para verlo más fácilmente.
- **Show/Hide Patch List** Cuando TortoiseMerge tiene una lista de parches, este botón le permite ocultarla (para que pueda ver ambas ventanas), o mostrarla (para que pueda revisar la lista de ficheros).
- **Configuración** le lleva al diálogo de opciones de TortoiseMerge.

En la parte izquierda de la ventana está una barra de lugar. Esto proporciona una rápida referencia visual de dónde están los cambios en el fichero. La barra tiene tres columnas. La columna izquierda se refiere al panel izquierdo, la columna derecha al panel derecho, y la columna central al panel inferior (si está presente). En la vista de un solo panel sólo se utiliza la columna izquierda. La barra de lugar se puede utilizar también como una barra para mover todas las ventanas simultáneamente.

Bajo la ventana inferior está la barra de estado. Esta barra muestra el número de líneas añadidas y borradas en el **De otros** y **Mío**, y el número de conflictos que quedan por resolver. También puede ver qué estilo de fin de línea se está utilizando, qué codificación (ASCII, UTF-8, etc) ha asumido TortoiseMerge, y si hay una marca-de-orden-de-bytes (BOM) en los ficheros UTF-8. La parte izquierda de la barra de estado se utiliza para mostrar pistas cuando pasa el cursor por encima de un control.

3.3. Iconos de estado de línea

Las líneas cambiadas se marcan con un icono para indicar qué tipo de cambio ocurrió.



Se añadió una línea.



Se eliminó una línea.



Se deshizo un cambio revirtiendo al contenido original de la línea.



Esta línea sólo contiene cambios en los espacios en blanco. Donde aparezcan líneas consecutivas marcadas, el párrafo puede haberse reformateado, causando que las palabras se cambien a las líneas adyacentes.



Se ha editado la línea a mano, utilizando TortoiseMerge como editor de texto.



Esta línea está en conflicto.



Esta línea está en conflicto, pero el efecto está oculto por la configuración de los espacios en blanco o de los finales de línea.

3.4. Fusionando / Editando conflictos

TortoiseMerge no sólo le *muestra* las diferencias entre los ficheros, sino que también le permite resolver conflictos o aplicar cambios.

Si está en la vista de dos paneles, sólo podrá editar el fichero en el panel derecho (Mío). Para aplicar cambios hechos en el fichero izquierdo (De otros), haga click con el botón derecho en las líneas cambiadas y seleccione Menú Contextual → Utilizar bloque de texto “de otros”. Entonces se añadirán los cambios desde el fichero de la izquierda al de la derecha.

A veces en realidad querrá utilizar ambos bloques de texto, y el menú contextual también le ofrece Menú contextual → Utilizar ambos bloques de texto (este primero) y Menú contextual → Utilizar ambos bloques de texto (este el último).

También puede editar el fichero de salida tal y como lo haría en un editor de texto. Las líneas que cambie de esta forma se marcarán utilizando un icono de un lápiz. Por favor, tenga en cuenta que si desea hacer alguno de los cambios basados en línea o bloque descritos anteriormente, es mejor que lo haga al principio: una vez que empiece a editar el fichero, TortoiseMerge no podrá mantener la relación de las líneas con los ficheros originales.

Si está en la vista de tres paneles (a veces llamada *vista de fusión*) sólo podrá editar el fichero en la vista inferior (Fusionado). Como en la vista de dos paneles, haga click con el botón derecho en las líneas con conflictos y seleccione Menú Contextual → Utilizar bloque de texto “de otros” o Menú Contextual → Utilizar bloque de texto “mío”. Además, si quiere *ambos* bloques, puede seleccionar Menú Contextual → Utilizar bloque de texto “mío” antes del “de ellos” o Menú Contextual → Utilizar bloque de texto “de ellos” antes del “mío”. Dependiendo del comando que haya seleccionado, se utilizarán los cambios en el fichero Fusionado resultante.

A veces un fichero se marcará como en conflicto en Subversion, pero cuando lo vea utilizando TortoiseMerge no se muestra ningún conflicto. Esto puede ser debido a su selección de la forma en la que se tratan los espacios en blanco. Si elige ignorar los cambios en el fin de línea o en los espacios en blanco, esas líneas se marcarán utilizando el icono *Conflicto-Ignorado*. Para resolver el conflicto aún necesitará elegir qué versión desea utilizar.

A veces el conflicto es demasiado difícil de resolver simplemente eligiendo entre de ellos y mío y necesita editarlo manualmente. TortoiseMerge le permite hacer primero y de forma visual todos los cambios fáciles y luego grabar el fichero, dejando sólo los cambios difíciles para que los resuelva a mano.



Importante

Tenga en cuenta que si utiliza TortoiseMerge sobre los mismos ficheros otra vez, cualquier cambio en su copia de trabajo, tanto si se hizo por TortoiseMerge o editándolo a mano, *se descartará* y el fichero aparecerá tal y como lo hizo cuando se inició por primera vez la edición de conflictos.

3.5. Abrir ficheros

Cuando inicia TortoiseMerge sin ningún argumento en la línea de comandos tendrá que abrir los ficheros manualmente, utilizando Archivo → Abrir.

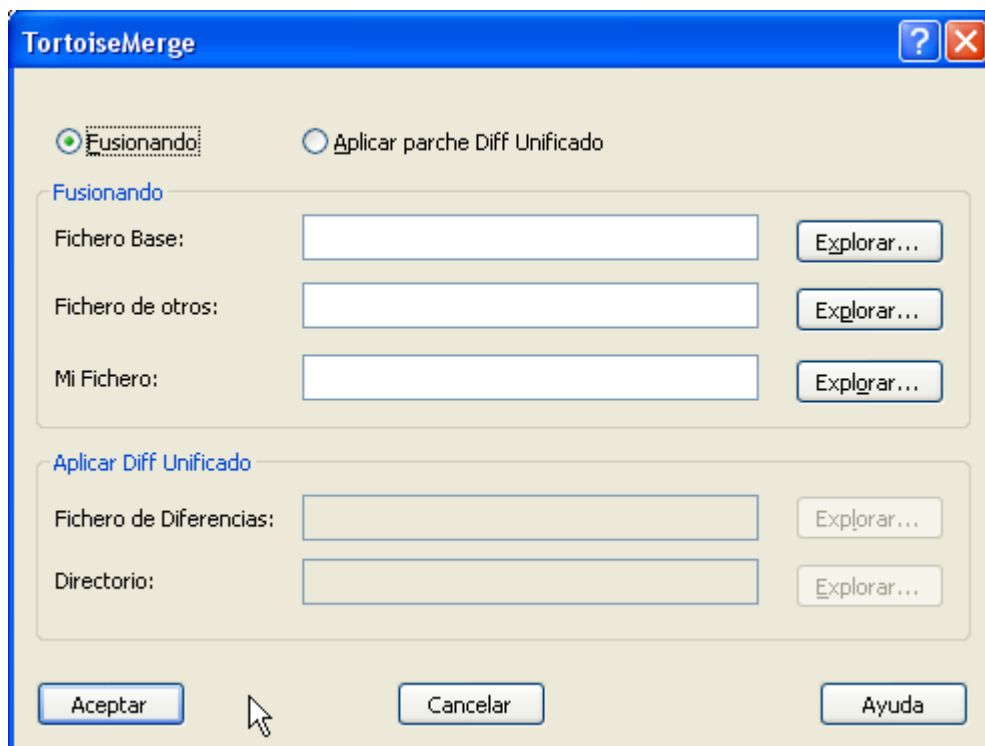


Figura 3.6. El diálogo Abrir

Lo primero que tiene que hacer es decidir si sólo quiere comparar/fusionar ficheros, o si quiere aplicar un fichero de parche. Dependiendo de su elección, se activan las casillas de texto y los botones de navegación correspondientes.

3.5.1. Viendo / Fusionando

Si quiere comparar/fusionar ficheros necesita establecer al menos dos de las tres posibles rutas para Base, Mío and De otros. Si sólo selecciona dos ficheros, TortoiseMerge le mostrará las diferencias entre esos dos ficheros, en dos paneles o en un único panel.

Si quiere comparar/fusionar tres ficheros, TortoiseMerge le mostrará las diferencias en una vista de tres paneles. Esta vista se utiliza normalmente si necesita resolver ficheros en conflicto. El fichero de salida no se nombra en este caso y tendrá que utilizar Archivo → Guardar como... para guardar el resultado.

3.5.2. Aplicando parches

Si desea aplicar un fichero de parche, tiene que especificar tanto la ruta al fichero de parche en si mismo como la ruta a la carpeta donde debe aplicarse el fichero de parche.



Atención

Actualmente, sólo se admiten ficheros de parche en el *Formato de Diff Unificado*. Y sólo los ficheros que se hayan generado desde una copia de trabajo de Subversion. ¡No se admiten actualmente otros formatos como los ficheros de parche generados por CVS!

3.6. Configuración

3.6.1. Página de Configuración principal

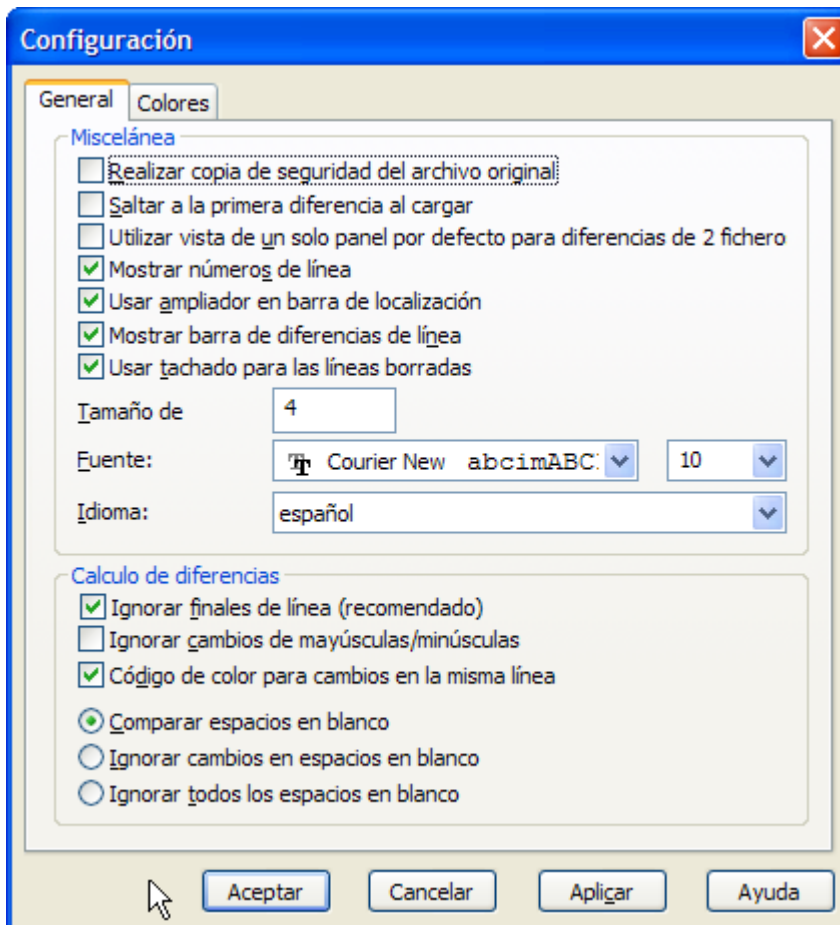


Figura 3.7. La página de Configuración principal

Muchas de estas opciones son autoexplicativas, pero hay algunos pocos puntos que necesitan clarificarse.

Realizar copia de seguridad del archivo original renombra el fichero original en la copia de trabajo a nombre .bak antes de guardar la versión modificada.

Ignorar finales de línea oculta los cambios que se deben únicamente a la diferencia en el estilo de las terminaciones de las líneas.

Ignorar cambios de mayúsculas/minúsculas oculta los cambios que se deben únicamente a cambiar mayúsculas por minúsculas o viceversa dentro del texto. Esto puede ser útil con aplicaciones como Visual Basic, que cambia las mayúsculas en las variables sin avisar.

Colorear cambios dentro de las líneas utiliza el mismo esquema de la barra de diferencias de línea para mostrar los cambios detallados de las líneas dentro de los paneles de visualización principales. Las líneas que han sufrido muchas modificaciones no se mostrarán utilizando este método ya que los resultados

son simplemente confusos. En este caso la línea se mostrará como reemplazada (línea antigua eliminada, línea nueva añadida).

Comparar espacios en blanco muestra todos los cambios en la indentación y en los espacios dentro de las líneas como si fueran líneas añadidas/eliminadas (que es como Subversion trata estos cambios).

Ignorar cambios en espacios en blanco oculta los cambios que se deben únicamente a cambiar la cantidad de espacios en blanco, por ejemplo la indentación o cambios de tabuladores por espacios. Añadir un espacio en blanco donde antes no había, o quitar un espacio en blanco completamente se seguirán mostrando como cambios.

Ignorar todos los espacios en blanco oculta todos los cambios que sólo se deben a espacios en blanco.

3.6.2. Página de Configuración de colores

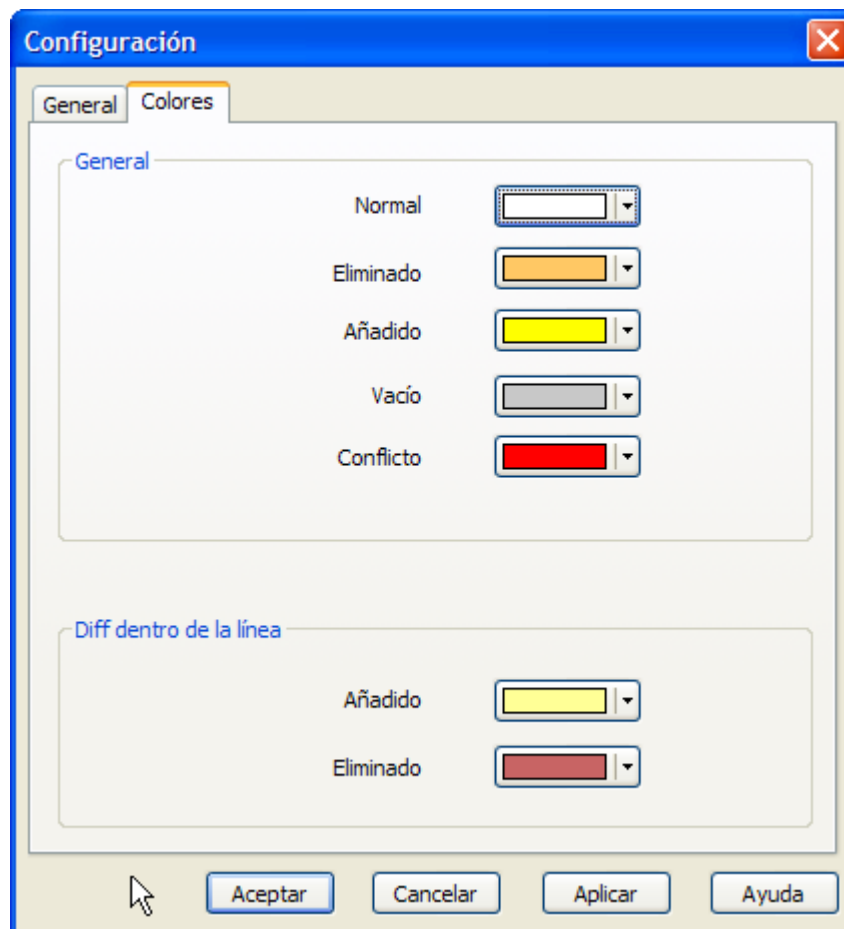


Figura 3.8. La página de Configuración de colores

Este diálogo le permite seleccionar la codificación de colores de fondo que se utilizan para remarcar los diferentes cambios en las líneas.

Normal

Todas las líneas que no tienen cambios, o cuyos cambios están ocultos.

Añadido

Líneas que han sido añadidas.

Borrado

Líneas que han sido borradas.

Modificado

Líneas que tienen relativamente pocos cambios, y se muestran utilizando diferencias en línea. Este color se utiliza para la parte *sin cambios* del contenido de la línea. Las partes modificadas se mostrarán utilizando los colores de añadido y eliminado en-línea que se describen a continuación. Si **Colorear cambios dentro de las líneas** no está seleccionado, este color no se utilizará, y las líneas con cambios siempre se mostrarán como reemplazos.

En conflicto

Donde la misma línea se ha cambiado en ambos ficheros.

Conflicto resuelto

Cuando la misma línea se ha cambiado en ambos ficheros, y ahora ya ha seleccionado qué versión debería utilizarse.

Vacío

Donde las líneas se han añadido en el panel contrario, y no existe dicha línea en este panel.

Texto añadido en línea

Cuando se muestran diferencias en-línea, el texto añadido se muestra en este color.

Texto eliminado en línea

Cuando se muestran diferencias en-línea, el texto eliminado se muestra en este color.

Espacios en blanco varios

Los caracteres usados para indicar espacios en blanco se muestran en un color distinto del texto normal.

Apéndice A. Atajos de teclado

Lista de atajos y comandos de teclado

A.1. Atajos de teclado

- Ctrl-Q
Sale del programa
- Ctrl-C
Copia el texto seleccionado al portapapeles
- Ctrl-F
Abre el diálogo Buscar para búsquedas de texto
- Ctrl-O
Abre los ficheros para ver las diferencias/fusionar
- Ctrl-S
Guardar los cambios
- Ctrl-Mayús-S
Guardar como...
- F7
Ir a la siguiente diferencia
- Mayús-F7
Ir a la diferencia previa
- F8
Ir al siguiente conflicto
- Mayús-F8
Ir al conflicto previo
- Ctrl-D
Cambia entre la vista de un panel y la vista de dos paneles
- Ctrl-R
Recarga el fichero y revierte todos los cambios hechos
- Ctrl-T
Cambia entre mostrar espacios en blanco o no
- Ctrl-rueda del ratón
Mueve el display a la izquierda/derecha

Apéndice B. Automatizando TortoiseMerge

TortoiseMerge puede iniciarse con parámetros en la línea de comandos para evitar pasar por el diálogo Abrir y seleccionar los ficheros allí. Esto es útil si desea utilizar TortoiseMerge desde otra aplicación.

B.1. Opciones de Línea de Comandos de TortoiseMerge

La mayoría de las opciones necesitan información adicional como una ruta o alguna otra cadena. En estos casos, añade un ':' a la opción y ponla la cadena/ruta tras él. Por ejemplo:

```
/base:"c:\carpeta\mi fichero base.txt"
```

Comando	Descripción
/?	Muestra un diálogo con los parámetros de la línea de comandos más importantes.
/help	Lo mismo que ?.
/base	Especifica el fichero <i>base</i> utilizado en las diferenciaciones en tres sentidos. Este es el ancestro común de los ficheros que están siendo diferenciados, aunque no se muestra en una ventana diferente. Para las diferenciaciones en dos sentidos, este es el fichero izquierdo.
/basename	El nombre del fichero base. Esto se muestra en el título de la vista en vez de la ruta del fichero. Para diferenciaciones de tres sentidos se muestra en un texto de ayuda para el título de la vista.
/theirs	Especifica el fichero <i>de ellos</i> utilizado en las diferenciaciones de tres sentidos, mostrado en el panel izquierdo.
/theirsname	El nombre del fichero de ellos. Esto se muestra en el título de la vista en vez de la ruta del fichero.
/mine	Especifica el fichero <i>mío</i> utilizado en las diferenciaciones en tres sentidos, mostrado en el panel derecho. Para las diferenciaciones en dos sentidos, este es el fichero derecho.
/minename	El nombre del fichero <i>mío</i> . Esto se muestra en el título de la vista en vez de la ruta del fichero.
/merged	Especifica el fichero <i>fusionado</i> resultante utilizando en las diferenciaciones en tres sentidos. Esta es la ruta del fichero donde se guardará el resultado de la resolución de la fusión/conflicto. Si no se establece, TortoiseMerge le preguntará al usuario dónde almacenar el resultado.
/mergedname	El nombre del fichero fusionado. Esto se muestra en el título de la vista en vez de la ruta del fichero.
/patchpath	La ruta donde se debería aplicar un parche. Si no establece esta ruta, entonces TortoiseMerge intentará encontrar la ruta por sí mismo, de forma que encaje con las rutas en el fichero de parche, pero eso puede llevarle <i>mucho tiempo</i> .
/patchoriginal	El nombre del fichero original a parchear. Utilizado para el título de la vista.
/patchpatched	El nombre del fichero parcheado resultante. Utilizado para el título de la vista.
/diff	La ruta del fichero de parche/diferencia a aplicar a un directorio.

Comando	Descripción
/oneway	Fuerza a TortoiseMerge a iniciar la vista de un sólo sentido en vez de la vista que el usuario especificó en la configuración.
/reversedpatch	Cambia la vista derecha e izquierda de los dos ficheros especificados para diferenciar.
/createunifieddiff	Crea un fichero de diff unificado (fichero de parche) de dos ficheros especificados con /origfile:"ruta_al_fichero_original" y /modifiedfile:"ruta_al_fichero_modificado". La ruta de destino se establece con /outfile:"ruta_al_fichero_de_parche_resultante". Si /outfile no se establece, se muestra un diálogo de guardar fichero para que el usuario pueda elegir el lugar donde guardar el fichero de parche. Nota: si se establece /createunifieddiff, todos los demás parámetros se ignoran.

Tabla B.1. Lista de las opciones de línea de comandos disponibles

También puede proporcionar nombres de ficheros simples en la línea de comando por compatibilidad con otros programas diff. En esta forma simplificada la línea de comandos es

```
TortoiseMerge RutaFicheroBase RutaMiFichero [ RutaSuFichero ]
```

Si se proporcionan dos ficheros se compararán entre ellos. Si se proporcionan tres ficheros, el primero se trata como fichero BASE, y los otros dos se comparan con él en una diferenciación en tres sentidos.

Glosario

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.
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.
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.
Bloqueo	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.
Cambiar	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 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.
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.
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”.
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.

Diff	Abreviatura de “Mostrar diferencias”. Muy útil cuando desea ver exactamente qué cambios se han hecho.
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 su historia.
Exportar	Este comando produce una copia de una carpeta versionada, igual que una copia de trabajo pero sin los directorios locales .svn.
FSFS	Un sistema de ficheros propietario de Subversion que se utiliza como soporte de los repositorios. Se puede utilizar en unidades compartidas de red. Se utiliza por defecto para los repositorios creados a partir de la versión 1.2.
Fusionar	<p>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.</p> <p>Las fusiones ocurren automáticamente cuando actualiza su copia de trabajo. También puede fusionar cambios específicos desde otras ramas utilizando el comando Fusionar de TortoiseSVN.</p>
GPO	Objeto de política de grupo
Historial	Muestra la historia de las revisiones de un fichero o carpeta. También se conoce como “Registro”.
Importar	Comando de Subversion para importar una jerarquía de carpetas completa en el repositorio en una única revisión.
Limpieza	Citando 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 los bloqueos y conseguir que su copia de trabajo vuelva a un estado usable 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.
Obtener	Un comando de Subversion que crea una copia local de trabajo en un directorio vacío al descargar los ficheros versionados desde el repositorio.
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 permite añadir metainformación versionada - llamadas “propiedades” para cada uno de sus directorios y ficheros

versionados. Cada propiedad tiene un nombre y un valor, como una entrada del registro. Subversion tiene algunas propiedades especiales que utiliza internamente, como `svn:eol-style`. TortoiseSVN también `svn:logminsize`. Puede a

Propiedad de revisión (revprop)	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.
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”.
Registro	Muestra la historia de las revisiones de un fichero o carpeta. También se conoce como “Historial”.
Relocalizar	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. 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.
repositorio	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.
Resolver	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.
Revertir	Subversion maneja una copia local “prístina” de cada fichero tal y como estaban cuando actualizó su copia de trabajo por última vez. Si ha hecho cambios y decide que quiere deshacerlos, puede utilizar el comando “revertir” para volver a la copia prístina.
Revisión	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.

	En otro sentido, una revisión puede referirse al conjunto de cambios que se hicieron cuando esa revisión se creó.
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 hizo la última operación de obtener, actualizar o confirmar. La revisión BASE normalmente no equivale a la revisión HEAD.
revisión HEAD	La última revisión de un fichero o una carpeta en el <i>repositorio</i> .
SVN	Una abreviatura muy usada para Subversion. El nombre del protocolo personalizado de Subversion que utiliza el servidor de repositorios "svnserve".