

# Funcionamiento del entorno integrado NetBeans - Git

---

## Índice

- [Introducción](#)
- [Inicializar un repositorio de Git](#)
- [Clonación de un repositorio de Git](#)
- [Agregar archivos a un repositorio de Git](#)
- [Editar archivos](#)
  - [Visualización de cambios en el editor de fuentes](#)
  - [Visualización de información sobre el estado del archivo](#)
    - [Insignias y codificación de colores](#)
    - [Etiquetas de estado de archivo](#)
    - [Vista de control de versiones de Git](#)
    - [Comparación de revisiones de archivos](#)
  - [Deshaciendo cambios](#)
- [Confirmación de fuentes en un repositorio](#)
- [Trabajar con ramas](#)
  - [Creando ramas](#)
  - [El registro de salida](#)
  - [Fusión de ramas](#)
  - [Eliminar una rama](#)
- [Trabajar con repositorios remotos](#)
  - [Fetching](#)
  - [Pulling](#)
  - [Pushing](#)

## Introducción

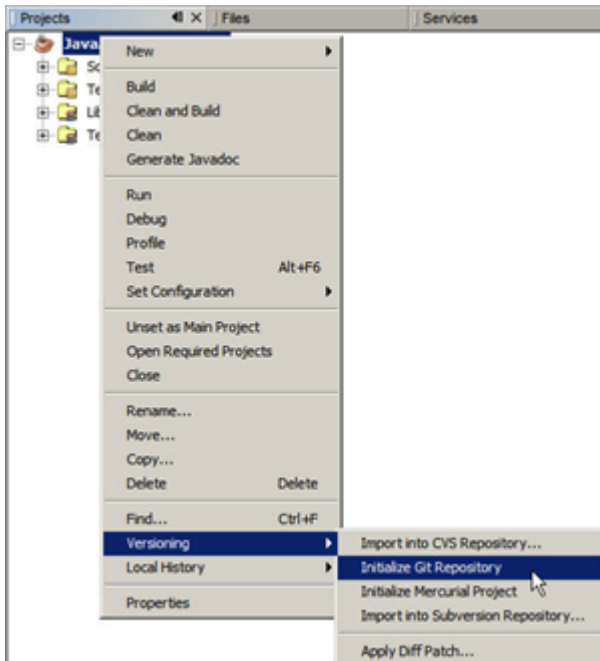
Apache NetBeans IDE proporciona soporte para el sistema de control de versiones de Git. Las funciones de Git del IDE le permiten realizar tareas de control de versiones directamente desde sus proyectos y código dentro del IDE. Este documento demuestra cómo realizar tareas de control de versiones en el IDE al guiarlo a través del flujo de trabajo estándar al usar Git.

Git es un sistema de control de versiones distribuido, gratuito y de código abierto, diseñado para manejar todo, desde proyectos pequeños a muy grandes, con velocidad y eficiencia. Cada clon de Git es un repositorio completo con historial completo y capacidades completas de seguimiento de revisiones, que no depende del acceso a la red o de un servidor central. La ramificación y la fusión son rápidas y fáciles de realizar. Git se utiliza para el control de versiones de archivos, al igual que herramientas como Mercurial, Subversion, CVS y Perforce.

## Inicializar un repositorio de Git

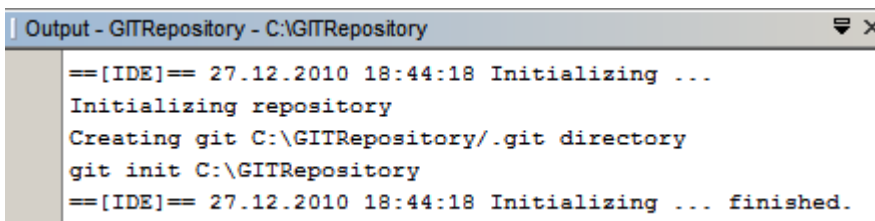
Para inicializar un repositorio de Git a partir de archivos existentes que aún no están en control de código fuente, debe completar los siguientes pasos:

1. En la ventana Proyectos, seleccione un proyecto sin versión y haga clic con el botón derecho en el nombre del proyecto.
2. En el menú contextual, seleccione **Versioning** > **Initialize GitRepository**(alternativamente, en el menú principal, seleccione **Team** > **Git** > **Initialize**).

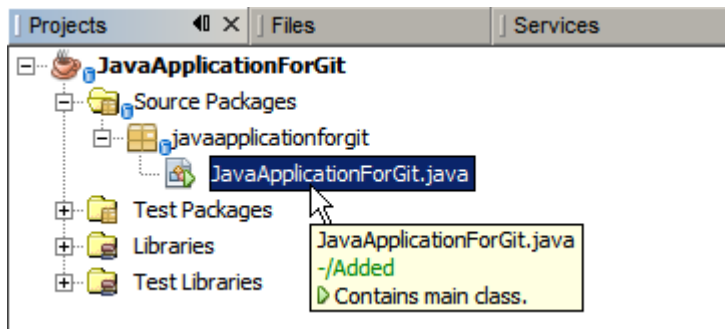


3. Especifique la ruta al repositorio en el que va a almacenar sus archivos versionados en el **Initialize Git Repository** cuadro de diálogo o haga clic en Examinar y busque el directorio requerido.
4. Haga clic en Aceptar.

Se crea una subcarpeta **.git** en la carpeta que especificó en el paso 1 anterior (su carpeta de proyecto NetBeans de forma predeterminada), que es su repositorio de Git donde se almacenan todos los datos de las instantáneas de su proyecto. Git comienza a crear versiones de todos los archivos en la carpeta que especificó. Puede abrir **Window** > **Output** para ver el informe de la IDE sobre el progreso de la creación del repositorio bajo el directorio de trabajo local.



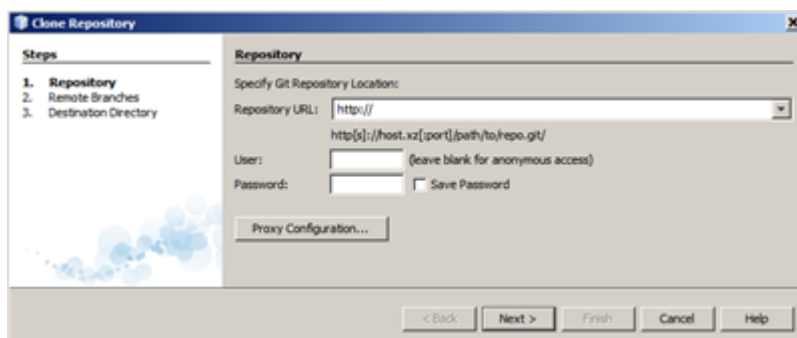
Todos los archivos del proyecto están marcados **Add** en su árbol de trabajo. Para ver el estado de un archivo, coloque el cursor sobre el nombre del archivo en la ventana Proyectos. El estado del archivo en el árbol de trabajo se muestra en verde a la derecha de la barra, como se muestra en la siguiente imagen.



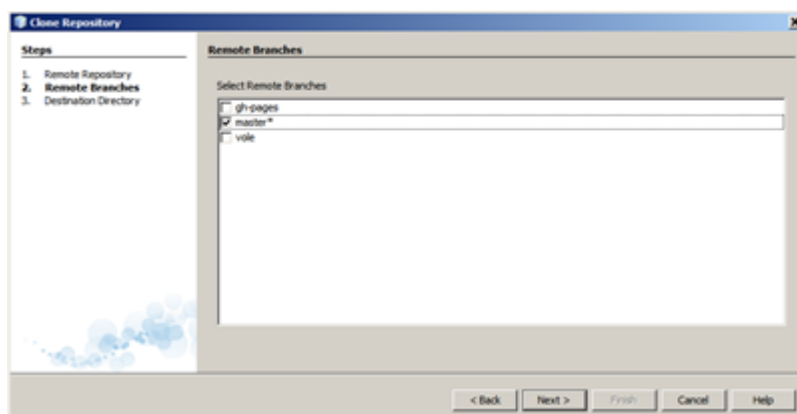
## Clonación de un repositorio de Git

Para obtener una copia de un repositorio de Git ya existente, debe clonarlo. Asegúrese de conocer la URL del repositorio de Git antes de iniciar el asistente de clonación del repositorio en el IDE.

Elija **Team > Git > Clone** desde el menú principal. Aparece el asistente Clonar repositorio.

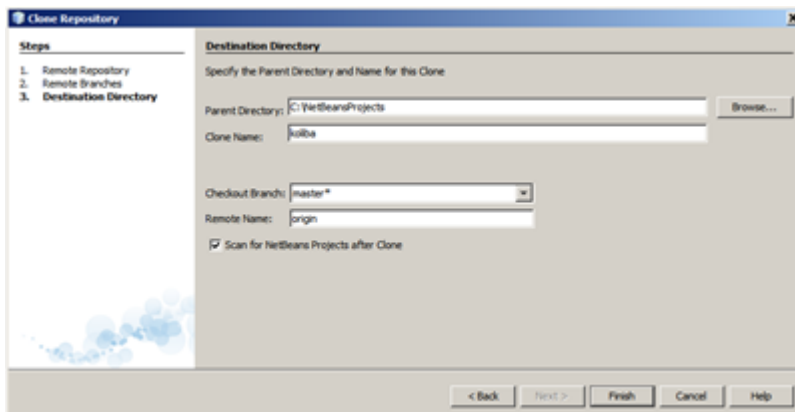


1. En el apartado Repositorio, especifique la ruta a la ubicación del repositorio Git, el nombre de usuario y la contraseña/token (puede guardarlos para el futuro si es necesario).
2. (Opcional) Haga clic en Configuración de proxy para mostrar el cuadro de diálogo Opciones y establecer la configuración del servidor proxy. Haga clic en Aceptar cuando haya terminado.
3. Haga clic en Siguiente para pasar al siguiente paso del asistente.
4. En la página de Ramas Remotas, seleccione las ramas del repositorio que se van a buscar (descargar) en su repositorio local. Haga clic en Siguiente.



5. En la página Directorio de destino, especifique lo siguiente:
  - En el campo **Parent directory**, la ruta al directorio destinado al repositorio clonado en su disco duro (alternativamente, haga clic en el botón Examinar y navegue hasta el directorio). El campo Directorio principal está precargado con la ruta al NetBeansProjects directorio predeterminado donde se almacenan todos los proyectos de NetBeans.

- En el campo **Clone Name**, el nombre de la carpeta local donde se clonará el proyecto original. De forma predeterminada, el nombre de clonación se completa con el nombre real del repositorio de Git.
- En el campo **Checkout Branch**, seleccione la rama que se desprotegerá en el árbol de trabajo.
- En el campo **Remote Name**, el nombre que representa el repositorio original que se está clonando. origin es el alias predeterminado del repositorio que se está clonando. Es un valor recomendado.
- La casilla de **Scan form NetBeans Projects after clone** seleccionada para activar el análisis posterior justo después de que finalice la clonación. (El complemento busca proyectos de NetBeans en los recursos clonados y ofrece abrir los proyectos encontrados).





6. Haga clic en Finalizar. Después de clonar un repositorio de Git, .git se crea la carpeta de metadatos dentro de la carpeta que seleccionó en el asistente.

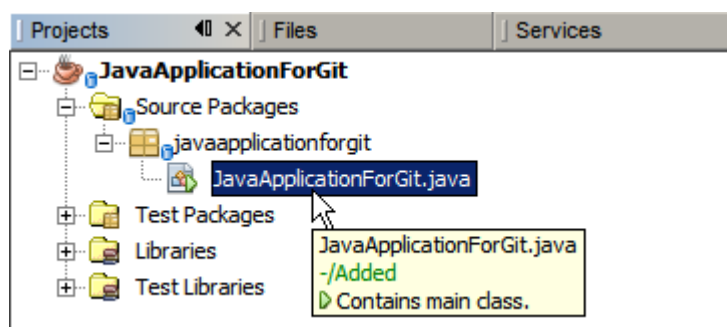
## Agregar archivos a un repositorio de Git

Para comenzar a rastrear un nuevo archivo y también para realizar cambios en un archivo ya rastreado en el repositorio de Git, debe agregarlo al repositorio.

Al agregar archivos a un repositorio de Git, el IDE compone y guarda instantáneas de su proyecto primero en el índice. Después de realizar la confirmación, el IDE guarda esas instantáneas en HEAD. El IDE le permite elegir entre los dos flujos de trabajo descritos en la siguiente tabla.

Descripción del flujo de trabajo	Agregue explícitamente archivos nuevos o modificados al Índice y luego confirme solo aquellos que están en etapas en el Índice al HEAD	Omita la adición de archivos nuevos o modificados al índice y confirme los archivos necesarios directamente en HEAD.
Pasos para seguir el flujo de trabajo	<p>1. En la ventana Proyectos, haga clic con el botón derecho en el archivo que desea agregar. [inicio = 2]. En el menú contextual, elija Git &gt; Add. Esto agrega el contenido del archivo al índice antes de enviarlo. [inicio = 3]. En la ventana Proyectos, haga clic con el botón derecho en el archivo que desea confirmar. [inicio = 4]. En el cuadro de diálogo Confirmar, seleccione el  botón de alternancia Cambios entre HEAD e Índice. Esto muestra la lista de archivos que ya están almacenados. [inicio = 5]. Confirme los archivos como se describe en la sección Confirmación de fuentes en un repositorio.</p>	<p>1. En la ventana Proyectos, haga clic con el botón derecho en el archivo que desea confirmar.[inicio = 2]. En el menú contextual, elija Git &gt; Commit. [inicio = 3]. En el cuadro de diálogo Commit, seleccione el  botón de alternancia Seleccionar los cambios entre HEAD y Working Tree. Esto muestra la lista de archivos que no están almacenados.[inicio = 4]. Confirme los archivos como se describe en la sección Confirmación de fuentes en un repositorio.</p>

El estado del archivo en HEAD se muestra en verde a la izquierda de la barra, como se muestra en la siguiente imagen.



## Editar archivos

Una vez que tenga un proyecto con versión de Git abierto en el IDE, puede comenzar a realizar cambios en las fuentes. Al igual que con cualquier proyecto abierto en NetBeans IDE, puede abrir archivos en el Editor de código fuente haciendo doble clic en sus nodos, tal como aparecen en las ventanas del IDE (por ejemplo, Proyectos (Ctrl-1), Archivos (Ctrl-2), Favoritos (Ctrl-3) ventanas).

Al trabajar con archivos de origen en el IDE, hay varios componentes de la interfaz de usuario a su disposición, que ayudan tanto en la visualización como en el funcionamiento de los comandos de control de versiones:

- Visualización de cambios en el editor de fuentes
- Visualización de información sobre el estado del archivo
- Deshaciendo cambios

### Visualización de cambios en el editor de fuentes

Cuando abre un archivo versionado en el editor de código fuente del IDE, puede ver los cambios en tiempo real que ocurren en su archivo a medida que lo modifica contra la versión base del repositorio de Git.

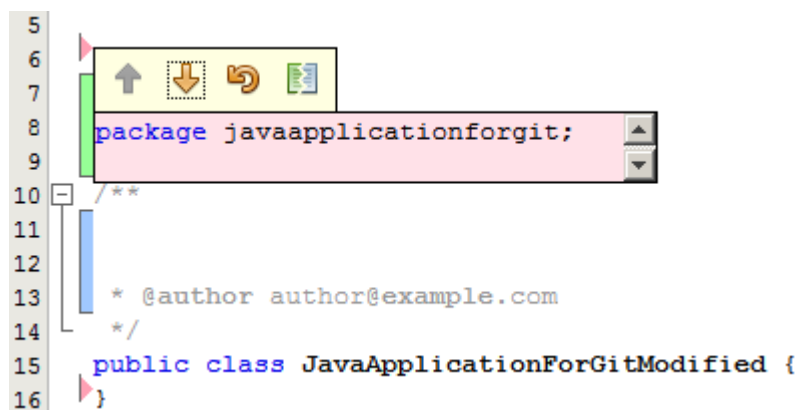
Mientras trabaja, el IDE utiliza códigos de colores en los márgenes del Editor de fuentes para transmitir la siguiente información:

- **Azul.** Indica líneas que se han modificado desde la revisión anterior.
- **Verde.** Indica líneas que se han agregado desde la revisión anterior.
- **Rojo.** Indica líneas que se han eliminado desde la revisión anterior.

El margen izquierdo del editor de fuentes muestra los cambios que ocurren línea por línea. Cuando modifica una línea determinada, los cambios se muestran inmediatamente en el margen izquierdo.



Puede hacer clic en una agrupación de colores en el margen para llamar a los comandos de control de versiones. Por ejemplo, la siguiente imagen muestra los widgets disponibles al hacer clic en un icono rojo que indica que las líneas se han eliminado de su copia local:



El margen derecho del editor de fuentes le proporciona una descripción general que muestra los cambios realizados en su archivo en su totalidad, de arriba a abajo. La codificación de colores se genera inmediatamente cuando realiza cambios en su archivo.

```
public String getDescription() {
    // This description will be displayed in the dialog,
    // hard-coded = ugly, should be done via I18N
    return "Text documents (*.txt)";
}
}
```

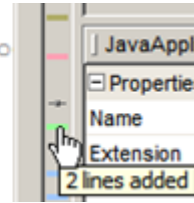


**Nota :** Puede hacer clic en un punto específico dentro del margen para llevar su cursor en línea inmediatamente a esa ubicación en el archivo. Para ver la cantidad de líneas afectadas, pase el mouse sobre los íconos de colores en el margen derecho:


```

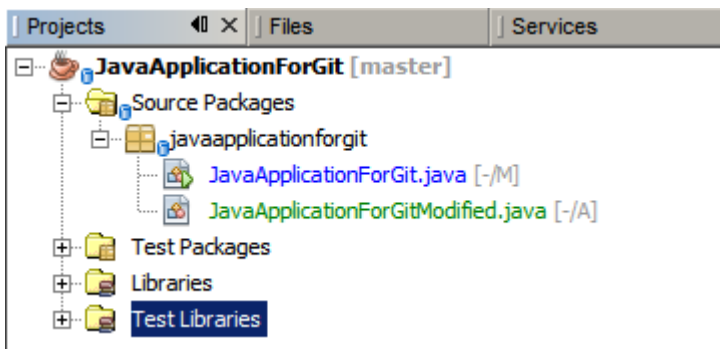
public String getDescription() {
    // This description will be displayed in the dialo
    // hard-coded = ugly, should be done via I18N
    return "Text documents (*.txt)";
}

```



## Visualización de información sobre el estado del archivo

Cuando trabaja en las vistas Proyectos (Ctrl-1), Archivos (Ctrl-2), Favoritos (Ctrl-3) o Control de versiones, el IDE proporciona varias funciones visuales que ayudan a ver la información de estado de sus archivos. En el siguiente ejemplo, observe cómo la insignia (  ), El color del nombre del archivo y la etiqueta de estado adyacente coinciden entre sí para brindarle una manera simple pero efectiva de realizar un seguimiento de la información de versiones en sus archivos:





Las insignias, la codificación de colores, las etiquetas de estado de los archivos y, quizás lo más importante, el visor de diferencias de Git contribuye a su capacidad para ver y administrar de manera efectiva la información de versiones en el IDE.

## Insignias y codificación de colores






Las insignias se aplican a los nodos de proyectos, carpetas y paquetes y le informan sobre el estado de los archivos contenidos dentro de ese nodo:

La siguiente tabla muestra el esquema de color utilizado para las insignias:

componente de interfaz de usuario	Descripción
Insignia azul 	Indica la presencia de archivos que se han modificado, agregado o eliminado en su árbol de trabajo. Para los paquetes, esta insignia se aplica solo al paquete en sí y no a sus subpaquetes. Para proyectos o carpetas, la insignia indica cambios dentro de ese elemento o cualquiera de las subcarpetas contenidas.
Insignia roja 	Marca proyectos, carpetas o paquetes que contienen archivos conflictivos. Para los paquetes, esta insignia se aplica solo al paquete en sí y no a sus subpaquetes. Para proyectos o carpetas, la insignia indica conflictos dentro de ese elemento o cualquiera de las subcarpetas contenidas.

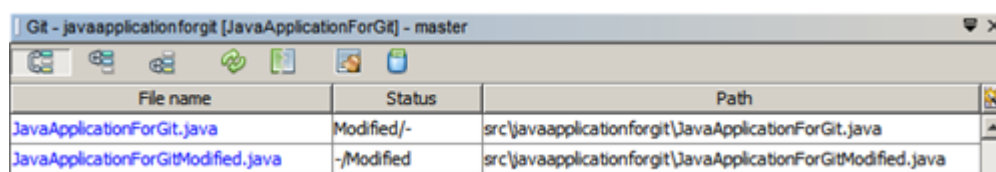
## Etiquetas de estado de archivo

La codificación de colores se aplica a los nombres de los archivos para indicar su estado actual en el repositorio:

color	Ejemplo	Descripción
sin color específico(negro)	 Main.java	Indica que el archivo no tiene cambios.
Azul	 Main.java	Indica que el archivo se ha modificado localmente.
Verde	 Main.java	Indica que el archivo se ha agregado localmente.
Rojo	 Main.java	Indica que el archivo tiene un conflicto de combinación.
Gris	 Main.java	Indica que Git ignora el archivo y no se incluirá en los comandos de control de versiones (por ejemplo, Actualizar y Confirmar). Los archivos no se pueden ignorar si están versionados.

## Vista de control de versiones de Git


La vista de control de versiones de Git le proporciona una lista en tiempo real de todos los cambios realizados en los archivos dentro de una carpeta seleccionada de su árbol de trabajo local. Se abre de forma predeterminada en el panel inferior del IDE, enumerando los archivos agregados, eliminados o modificados. Para abrir la vista Control de versiones, seleccione un archivo o carpeta con versión (por ejemplo, de la ventana Proyectos, Archivos o Favoritos) y elija **Git > Show Changes** en el menú contextual, o elija **Team > Show Changes** en el menú principal. La siguiente ventana aparece en la parte inferior del IDE:



File name	Status	Path
JavaApplicationForGit.java	Modified/-	src\javaapplicationforgit\JavaApplicationForGit.java
JavaApplicationForGitModified.java	-/Modified	src\javaapplicationforgit\JavaApplicationForGitModified.java

De forma predeterminada, la vista Control de versiones muestra una lista de todos los archivos modificados dentro del paquete o carpeta seleccionados en su Árbol de trabajo. Usando los botones de la barra de herramientas, puede elegir mostrar la lista de archivos que tienen diferencias entre Índice y HEAD, Árbol de trabajo e Índice o Árbol de trabajo y HEAD. También puede hacer clic en los encabezados de las columnas sobre los archivos enumerados para ordenar los archivos por nombre, estado o ubicación.

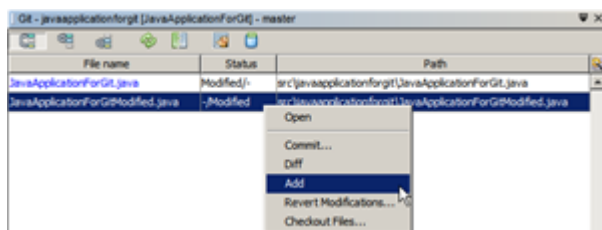
La barra de herramientas de la vista Control de versiones también incluye botones que le permiten invocar las tareas de Git más comunes en todos los archivos que se muestran en la lista. La siguiente tabla enumera los comandos de Git disponibles en la barra de herramientas de la vista Versionado:

Icono	Nombre	Función
	Cambios entre HEAD y Working Tree	Muestra una lista de archivos que ya están preparados o solo modificados / creados y aún no preparados.



Icono	Nombre	Función
	Cambios entre HEAD e Index	Muestra una lista de archivos almacenados.
	Cambios entre el Index y Working Tree	Muestra archivos que tienen diferencias entre sus estados de árbol de trabajo y por etapas.
	Actualizar estados	Actualiza el estado de los archivos y carpetas seleccionados. Los archivos que se muestran en la vista Control de versiones se pueden actualizar para reflejar cualquier cambio que se haya realizado externamente.
	Diferencia abierta	Abre el visor de diferencias que le proporciona una comparación en paralelo de sus copias locales y las versiones que se mantienen en el repositorio.
	Revertir modificaciones	Muestra el cuadro de diálogo Revertir modificaciones.
	Cometer cambios	Muestra el Commit cuadro de diálogo.

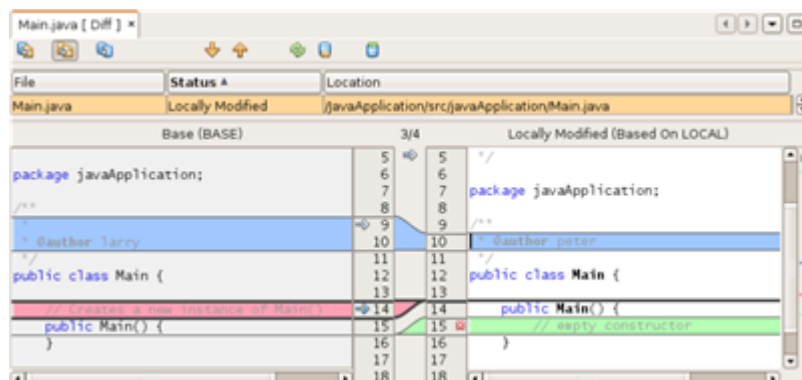
Puede acceder a otros comandos de Git en la vista Control de versiones seleccionando una fila de la tabla que corresponda a un archivo modificado y eligiendo un comando en el menú contextual:



## Comparación de revisiones de archivos

Comparar versiones de archivos es una tarea común cuando se trabaja con proyectos versionados. El IDE le permite comparar revisiones mediante el comando Diff:


1. Seleccionar un archivo o carpeta versionada (por ejemplo, de la Projects, Files)
2. Elija **Team > Diff > Diff to HEAD** desde el menú principal. Se abre un visor gráfico de diferencias para los archivos seleccionados y las revisiones en la ventana principal del IDE. El visor de diferencias muestra dos copias en paneles uno al lado del otro. La copia más actual aparece en el lado derecho, por lo que está comparando una revisión del repositorio con su árbol de trabajo, el árbol de trabajo se muestra en el panel derecho:






El visor de diferencias utiliza la misma codificación de colores que se utiliza en otros lugares para mostrar los cambios de control de versiones. En la captura de pantalla que se muestra arriba, el bloque verde indica contenido que se ha agregado a la revisión más actual. El bloque rojo indica que el contenido de la revisión anterior se ha eliminado de la última. El azul indica que se han producido cambios dentro de las líneas resaltadas.

La barra de herramientas del Visor de diferencias también incluye botones que le permiten invocar las tareas de Git más comunes en todos los archivos que se muestran en la lista. La siguiente tabla enumera los comandos de Git disponibles en la barra de herramientas del Visor de diferencias:

Icono	Nombre	Función
	Cambios entre HEAD y Working Tree	Muestra una lista de archivos que ya están preparados o solo modificados / creados y aún no preparados.
	Cambios entre HEAD e Index	Muestra una lista de archivos almacenados.
	Cambios entre el Index y Working Tree	Muestra archivos que tienen diferencias entre sus estados de árbol de trabajo y por etapas.
	Actualizar estados	Actualiza el estado de los archivos y carpetas seleccionados. Los archivos que se muestran en la vista Control de versiones se pueden actualizar para reflejar cualquier cambio que se haya realizado externamente.
	Diferencia abierta	Abre el visor de diferencias que le proporciona una comparación en paralelo de sus copias locales y las versiones que se mantienen en el repositorio.
	Revertir modificaciones	Muestra el cuadro de diálogo Revertir modificaciones.
	Cometer cambios	Muestra el Commit cuadro de diálogo.
	Ir a la siguiente diferencia	Muestra la siguiente diferencia en el archivo.

Icono	Nombre	Función
	Ir a la diferencia anterior	Muestra la diferencia anterior en el archivo.

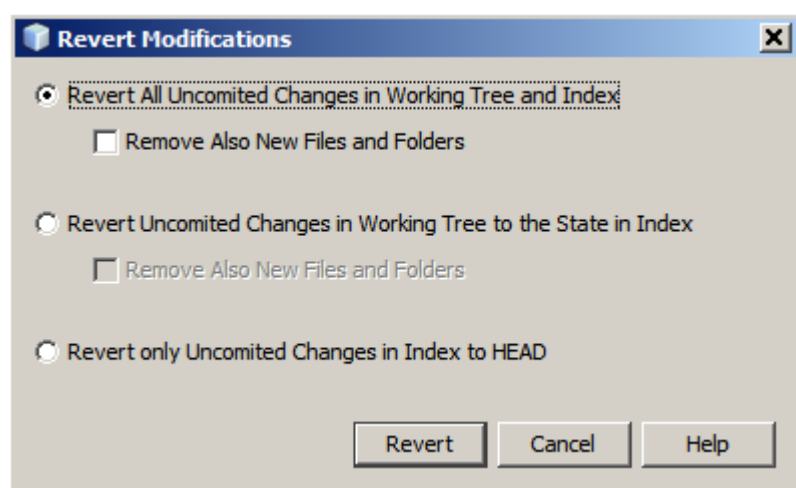
Si está realizando una diferencia en su copia local en el árbol de trabajo, el IDE le permite realizar cambios directamente desde el visor de diferencias. Para hacerlo, puede colocar el cursor en el panel derecho del visor de diferencias y modificar su archivo en consecuencia; de lo contrario, utilice los iconos en línea que se muestran junto a cada cambio resaltado:

Icono	Nombre	Función
	Reemplazar	Inserta el texto resaltado en su copia del Árbol de trabajo.
	Mueve todo	Revierte toda la copia del árbol de trabajo local.
	Eliminar	Elimina el texto resaltado de la copia del Árbol de trabajo local.

## Deshaciendo cambios

Para deshacerse de los cambios locales realizados en los archivos seleccionados en su Árbol de trabajo y reemplazar esos archivos con los del Índice o HEAD:

1. Seleccionar un archivo o carpeta versionada (por ejemplo, de Projects, Files)
2. Elija **Team > Revert Modifications** en el menú principal. Aparece el cuadro de diálogo:

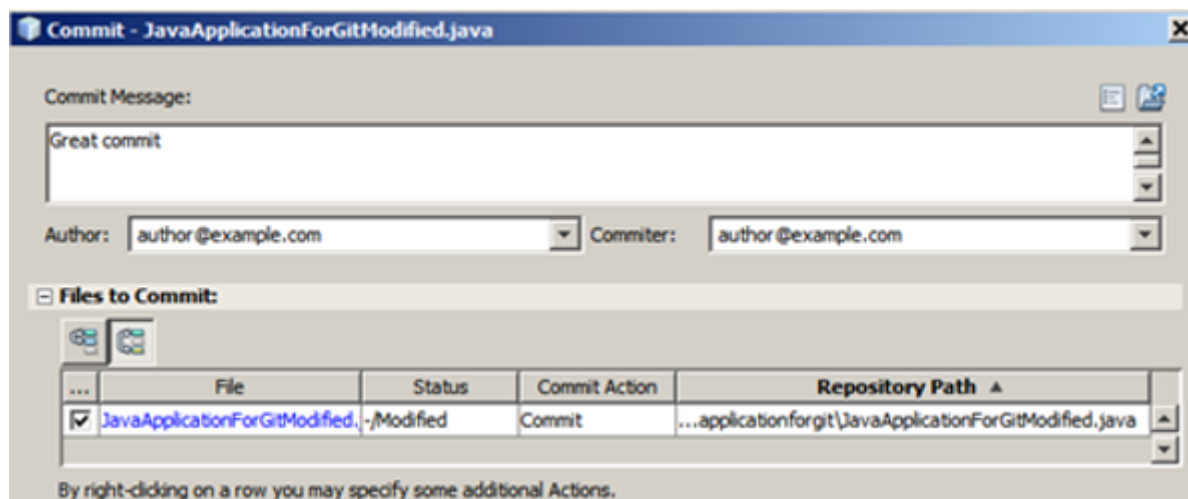


3. Especifique opciones adicionales (por ejemplo, Revert only Uncommitted Changes in Index to HEAD).
4. Haga clic en Revertir. El IDE reemplaza los archivos seleccionados por los especificados.

## Confirmación de fuentes en un repositorio

Para enviar archivos al repositorio de Git:

1. En la ventana de proyectos, haga clic con el botón derecho en los archivos que desea confirmar.
2. En el menú contextual, elija **Git > Commit**. Aparece el cuadro de diálogo:



El cuadro de diálogo contiene los siguientes componentes:

- **Commit Message** área de texto destinada a describir el cambio que se está cometiendo
- **Author y Committer** listas desplegables que permiten diferenciar entre quienes realizaron el cambio y quienes cometieron físicamente el expediente si fuera necesario.
- **Files to Commit** sección que enumera:
  - todos los archivos modificados,
  - todos los archivos que se han eliminado en el árbol de trabajo (localmente),
  - todos los archivos nuevos (es decir, archivos que aún no existen en el repositorio de Git),
  - todos los archivos cuyo nombre ha cambiado.

Aquí están disponibles dos botones de alternancia que cambian el modo en el que se realizará la confirmación real:

Icono	Nombre	Función
	Cambios entre HEAD y Working Tree	Muestra una lista de archivos que ya están preparados o solo modificados / creados y aún no preparados.
	Cambios entre HEAD e Index	Muestra una lista de archivos almacenados.

**Nota:** Para especificar aquí **si excluir archivos individuales de la confirmación**, deseleccione la casilla de verificación en la primera columna llamada **Commit** o haga clic con el botón derecho en una fila de archivo en la columna **Commit Action** y elija **Exclude from commit** en el menú emergente. Para mostrar **el Visor de diferencias** aquí, haga clic con el botón derecho en una fila de archivo en la columna **Commit Action** y elija **Diff** en el menú emergente.

## Trabajar con ramas

El soporte Git del IDE le permite mantener diferentes versiones de una base de código completa utilizando ramas.

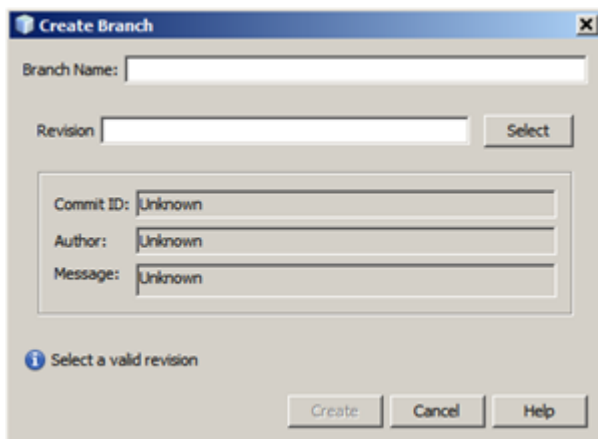
Al trabajar con ramas en el IDE, se admiten las siguientes acciones:

- Creando ramas
- El registro de salida
- Fusión de ramas
- Eliminando

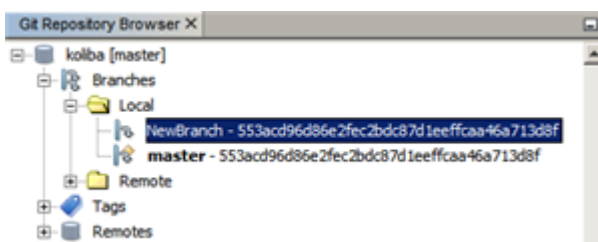
## Creando ramas

Para crear una rama local, si desea trabajar en una versión separada de su sistema de archivos con fines de estabilización o experimentación sin perturbar el tronco principal, complete los siguientes pasos:

1. En la ventana Proyectos o Archivos, elija un proyecto o carpeta del repositorio en el que desea crear la rama.
2. En el menú principal, elija **Teams> Branch / Tag> Create Branch**. Como alternativa, haga clic con el botón derecho en el proyecto o carpeta con versión y elija **Git> Branch / Tag> Create Branch** en el menú emergente. Aparece el cuadro de diálogo:



3. En el campo **Branch Name**, ingrese el nombre de la sucursal que se está creando.
4. En el campo **Revision** Escriba una Revision específica del elemento seleccionado ingresando un ID de confirmación, una rama existente o un nombre de etiqueta o presione Seleccionar para ver la lista de revisiones mantenidas en el repositorio.
5. (Opcional) En el cuadro de diálogo **Select** de revisión, expanda Ramas y elija la rama requerida, especifique el ID de confirmación en la lista adyacente y presione Seleccionar.
6. Revise la información de los campos Commit ID, Author, Message específica de la revisión desde la que se está ramificando y haga clic en Crear. La rama se agrega a la **Branches/Local** carpeta del repositorio de Git.

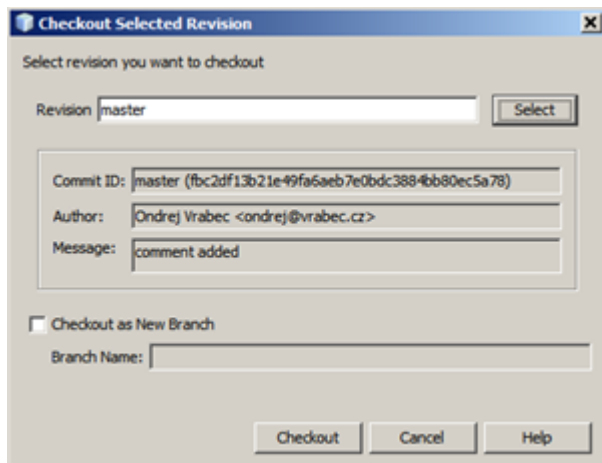


## El registro de salida

Si necesita editar archivos en una rama que ya existe, puede consultar la rama para copiar los archivos a su Árbol de trabajo.

Para verificar una revisión, realice lo siguiente:

1. Elija **Teams> Checkout> Checkout Revision** en el menú principal. Aparece el cuadro de diálogo:



2. Especifique **la revisión** requerida ingresando un ID de confirmación, una rama existente o un nombre de etiqueta en el campo Revisión o presione Seleccionar para ver la lista de revisiones mantenidas en el repositorio.
3. Omita si no presionó Seleccionar en el paso anterior. En el cuadro de diálogo Seleccionar revisión, expanda Ramas y elija la rama requerida, especifique el ID de confirmación en la lista adyacente si es necesario y presione Seleccionar.

Si la revisión especificada se refiere a una confirmación válida que no está marcada con un nombre de rama, su HEAD se desconecta y ya no está en ninguna rama.

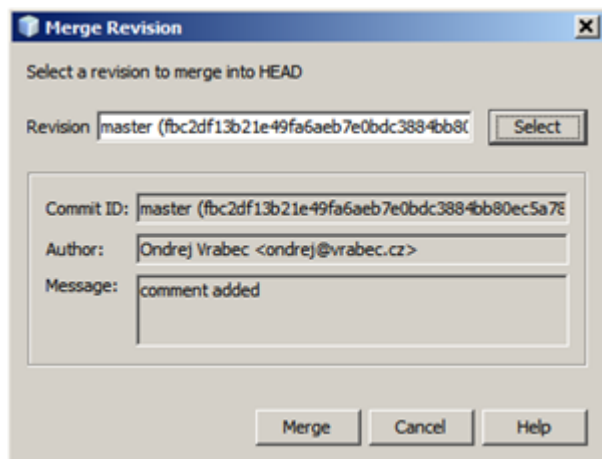
4. Revise la información de los campos Commit ID, Author, Message específica de la revisión que se está extrayendo.
5. Para crear una nueva rama a partir de la revisión extraída, elija la opción **checkout as New Branch** e ingrese el nombre en el campo Nombre de la rama.
6. Presione **Checkout** para verificar la revisión. Los archivos del árbol de trabajo y del índice se actualizan para que coincidan con la versión de la revisión especificada.

Si desea cambiar sus archivos a una rama que ya existe (por ejemplo, a una confirmación que no está en la parte superior de una de sus ramas), puede usar el comando **Teams> Git> Branch> Switch To Branch**, especifique la rama en el cuadro de diálogo Cambiar a rama seleccionada, compruébelo como una nueva rama (opcionalmente) y presione Cambiar.

## Fusión de ramas

Para transferir modificaciones de una revisión del repositorio al Árbol de trabajo, haga lo siguiente:

1. Elija **Teams> Branch / Tag> Merge Revision** en el menú principal. Aparece el cuadro de diálogo:



2. Especifique la revisión requerida ingresando un ID de confirmación, una rama existente o un nombre de etiqueta en el campo Revisión o presione Seleccionar para ver la lista de revisiones mantenidas en el repositorio.
3. Omita si no presionó Seleccionar en el paso anterior. En el cuadro de diálogo Seleccionar revisión, expanda Ramas y elija la rama requerida, especifique el ID de confirmación en la lista adyacente si es necesario y presione Seleccionar.
4. Presione Fusionar. Se realiza una combinación de tres vías entre la rama actual, el contenido de su Árbol de trabajo y la rama especificada.

Si se produce un conflicto de fusión, el archivo en conflicto se marca con una insignia roja para indicarlo.

Después de fusionar, aún debe confirmar los cambios para que se agreguen al HEAD.

## Eliminar una rama

Para eliminar una rama local innecesaria, complete los siguientes pasos:

1. Elija **Teams> Repository Browser** en el menú principal.
2. Y elija la rama que se eliminará.

La rama debe estar inactiva, es decir, no estar actualmente registrada en el Árbol de trabajo.

3. Haga clic con el botón derecho en la rama seleccionada y elija Eliminar rama en el menú emergente.
4. En el cuadro de diálogo Eliminar rama, presione Aceptar para confirmar la eliminación de la rama. La rama se elimina del repositorio local y del navegador del repositorio de Git.

## Trabajar con repositorios remotos

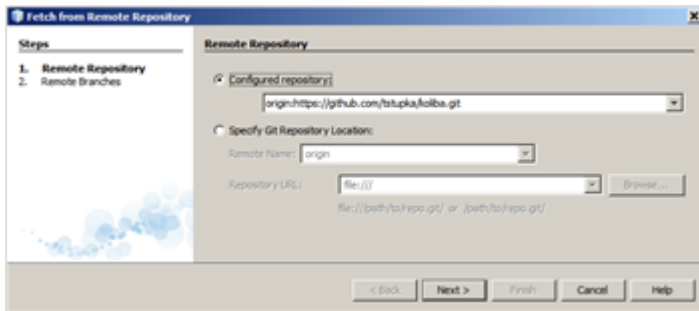
Cuando trabaja con otros desarrolladores, necesita compartir su trabajo, lo que implica buscar, enviar y extraer datos desde y hacia repositorios remotos alojados en Internet o en la red.

### Fetching

La recuperación obtiene los cambios del repositorio remoto original que aún no tiene. Nunca cambia ninguna de sus ramas locales. La recuperación obtiene todas las ramas de repositorios remotos, que puede fusionar en su rama o simplemente inspeccionar en cualquier momento.

Para obtener las actualizaciones, haga lo siguiente:

1. Elija **Team > Remote > Fetch**. Aparece el asistente Obtener desde el repositorio remoto.

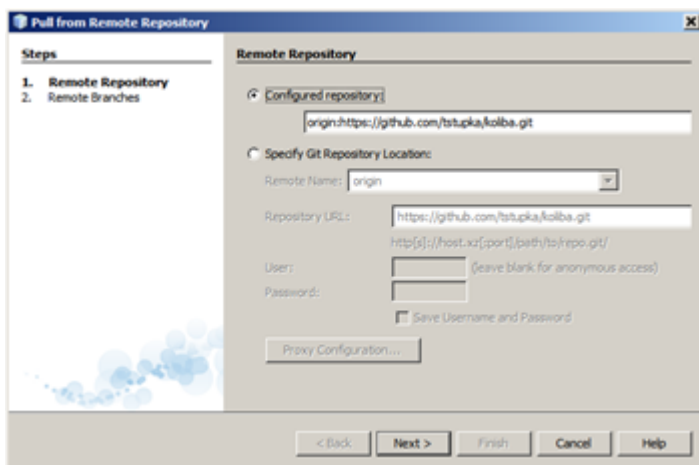


2. En la página Repositorio remoto del asistente, seleccione el repositorio configurado (para usar la ruta al repositorio configurado anteriormente) o la opción Especificar ubicación del repositorio Git (para definir la ruta a un repositorio remoto al que aún no se ha accedido, su nombre, configuración de inicio de sesión, contraseña y proxy si es necesario) y haga clic en Siguiente.
3. En la página Ramas remotas del asistente, elija las ramas desde las que buscar los cambios y haga clic en Finalizar. Se crea una copia local de una rama remota. La rama seleccionada se actualiza en el directorio Branches > Remote en el navegador del repositorio de Git. A continuación, las actualizaciones obtenidas se pueden fusionar en una rama local.

## Pulling

Al extraer algunas actualizaciones de un repositorio de Git remoto, los cambios se obtienen de él y se fusionan en el HEAD actual de su repositorio local. Para realizar la extracción, completa los siguientes pasos:

1. Elija **Team > Remote > Pull**. Aparece el asistente Extraer del repositorio remoto.



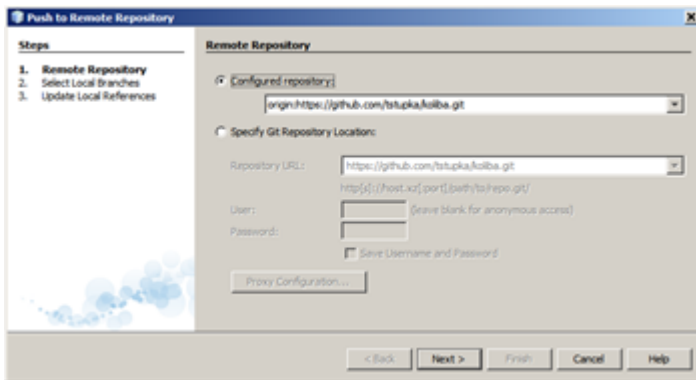
2. En la página Repositorio remoto del asistente, seleccione el repositorio configurado (para usar la ruta al repositorio configurado anteriormente) o la opción Especificar ubicación del repositorio Git (para definir la ruta a un repositorio remoto al que aún no se ha accedido, su nombre, e nombre de usuario y contraseña si es necesario) y haga clic en Siguiente.
3. En la página Ramas remotas del asistente, elija las ramas de las que se extraerán los cambios y haga clic en Finalizar. Su repositorio local está sincronizado con el repositorio de origen.

## Pushing

Para contribuir con cambios desde su repositorio local de Git a un repositorio público de Git, realice los siguientes pasos:



1. Elija **Team > Remote > Push**. Aparece el asistente Enviar a repositorio remoto.



2. En la página Repositorio remoto del asistente, seleccione el repositorio configurado (para usar la ruta al repositorio configurado anteriormente) o la opción Especificar ubicación del repositorio Git (para definir la ruta a un repositorio remoto al que aún no se ha accedido, su nombre, e nombre de usuario y contraseña si es necesario) y haga clic en Siguiente.
3. En la página Seleccionar ramas locales, elija las ramas a las que enviar sus ediciones y haga clic en Siguiente.
4. En la página Actualizar referencias locales, elija las ramas que se actualizarán en el directorio remoto de su repositorio local y haga clic en Finalizar. La rama de repositorio remoto especificada se actualiza con el estado más reciente de su rama local.