

# Usando Suporte Git no NetBeans IDE

O NetBeans IDE fornece o suporte ao cliente de controle de versão GIT. O Suporte ao Git do IDE permite executar tarefas de controle de versão diretamente de seu projeto dentro do IDE. Este documento demonstra como executar tarefas básicas de controle de versão no IDE orientando você pelo fluxo de trabalho padrão ao usar o software de controle de versão.

O Git é um sistema de controle de versão gratuito e de código-fonte aberto, projetado para tratar de tudo de um projeto pequeno a muito grande com rapidez e eficiência. Cada clone do Git é um repositório com todos os recursos com histórico completo e recursos completos de rastreamento da revisão, não dependente de acesso à rede ou de um servidor central. A ramificação e a mesclagem são rápidas e fáceis de serem feitas. O Git é usado para o controle de versão de arquivos, de forma similar a ferramentas como Mercurial, Subversion, CVS, Perforce etc.

## Conteúdo



- [Inicializando um Repositório Git](#)
- [Clonando um Repositório Git](#)
  - [Clonando um Repositório a partir do GitHub por meio do Protocolo SSH](#)
- [Adicionando Arquivos em um Repositório Git](#)
- [Editando Arquivos](#)
  - [Exibindo Alterações no Editor de Código-fonte](#)
  - [Exibindo Informações de Status do Arquivo](#)
  - [Comparando Revisões de Arquivos](#)
  - [Revertendo Alterações](#)
- [Confirmando Códigos-fonte em um Repositório](#)
- [Trabalhando com Ramificações](#)
  - [Criando uma Ramificação](#)
  - [Fazendo Check-out de uma Ramificação](#)
  - [Mesclando](#)
  - [Deletando uma Ramificação](#)
- [Trabalhando com Repositórios Remotos](#)
  - [Extraindo](#)
  - [Extraindo](#)
  - [Expandindo](#)
- [Resumo](#)
- [Consulte Também](#)

## Requisitos

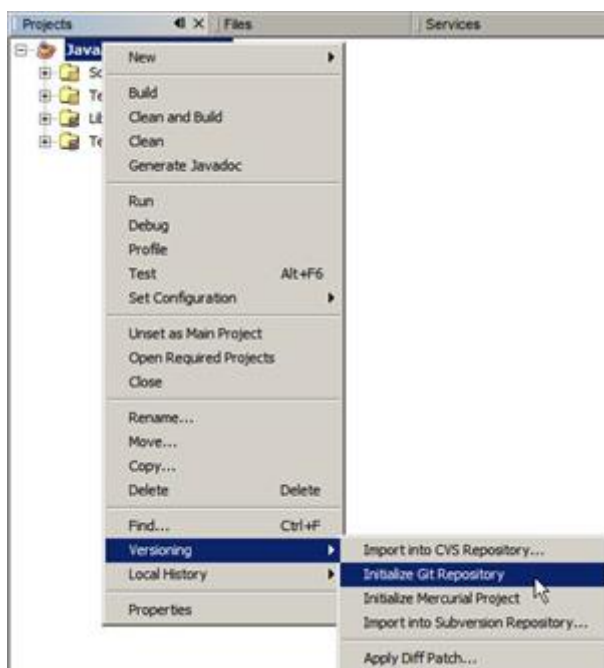
Para concluir este tutorial, você precisa dos seguintes recursos e softwares.

| Software ou Recurso                        | Versão Necessária |
|--|-------------------|
| <a href="#">NetBeans IDE</a>               | Versão 7.4 ou 8.0 |
| <a href="#">Java Development Kit</a> (JDK) | Versão 7 ou 8     |

## Inicializando um Repositório Git

Para inicializar um repositório Git de arquivos existentes que ainda não estão sob o controle de versão, é preciso completar as seguintes etapas:

1. Na janela Projetos, selecione um projeto sem controle de versão e clique com o botão direito do mouse no nome do projeto.
2. No menu de contexto, escolha `Controle de Versão > Inicializar o Repositório Git` (alternativamente, no menu principal escolha `Equipe > Git > Inicializar`).



3. Especifique o caminho para o repositório em que você armazenará seus arquivos com controle de versão na caixa de diálogo `Inicializar Repositório do Git` ou clique em `Procurar` e navegue para o diretório necessário.
4. Clique em `OK`.

Uma subpasta `.git` é criada na pasta especificada na etapa 3 acima (sua pasta do projeto NetBeans por default), que é seu repositório Git onde todos os dados dos snapshots de seu projeto são armazenados. O Git inicia o controle de versão de todos os arquivos na pasta que você especificou.

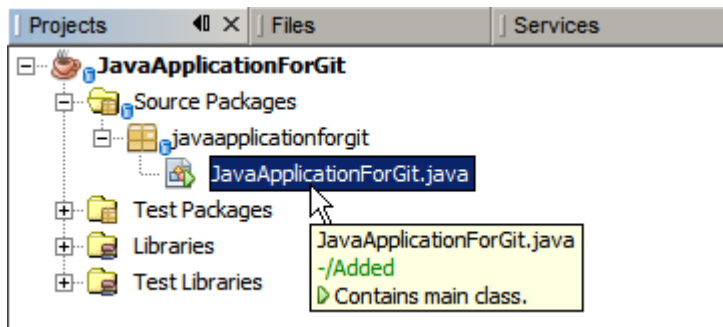
Você pode abrir `Janela > Saída > Saída` para exibir o relatório do IDE sobre o andamento da criação do repositório sob seu diretório local de trabalho.

```

Output - GITRepository - C:\GITRepository
==[IDE]== 27.12.2010 18:44:18 Initializing ...
Initializing repository
Creating git C:\GITRepository\.git directory
git init C:\GITRepository
==[IDE]== 27.12.2010 18:44:18 Initializing ... finished.

```

Todos os projetos são marcados como **Adicionados** em sua Árvore de Trabalho. Para exibir o **status** de um arquivo, coloque o cursor sobre o nome do arquivo na janela Projetos. O status do arquivo na Árvore de Trabalho é exibido em verde na direita da barra, como mostrado na figura a seguir.

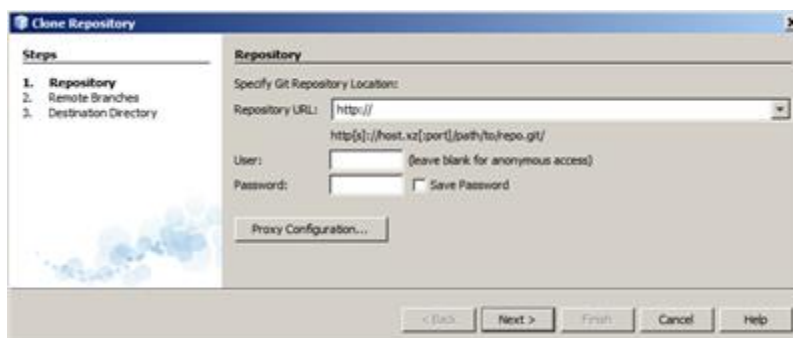


Após ter inicializado o repositório Git, você [adiciona](#) arquivos ou [confirma-os](#) diretamente no repositório Git.

## Clonando um Repositório Git

Para obter uma cópia de um repositório Git já existente, é preciso cloná-lo. Assegure que você conheça o URL do repositório Git antes de iniciar o assistente Clonar Repositório no IDE.

1. Escolha **Equipe > Git > Clonar** no menu principal. O Assistente Clonar Repositório é Exibido.



2. Na página Repositório, especifique o caminho da localização do repositório Git, o nome de usuário e senha (você pode salvá-los para uso futura se necessário).
3. (Opcional) Clique em Configuração de Proxy para exibir a caixa de diálogo Opções e especificar as definições do servidor proxy. Clique em OK quando finalizar.
4. Clique em Próximo para ir para a próxima etapa do assistente.
5. Na página Ramificações Remotas, selecione as ramificações de repositório a serem extraídas (por download) para seu repositório local. Clique em Próximo.
6. Na página Diretório de Destino, especifique o seguinte:
  - o No campo Diretório-Pai, o caminho do diretório desejado para o repositório clonado em seu disco rígido (alternativamente, clique no botão Procurar e navegue para o diretório).

O campo Diretório-Pai é preenchido com o caminho para o diretório default `NetBeansProjects` onde todos os projetos do NetBeans são armazenados.

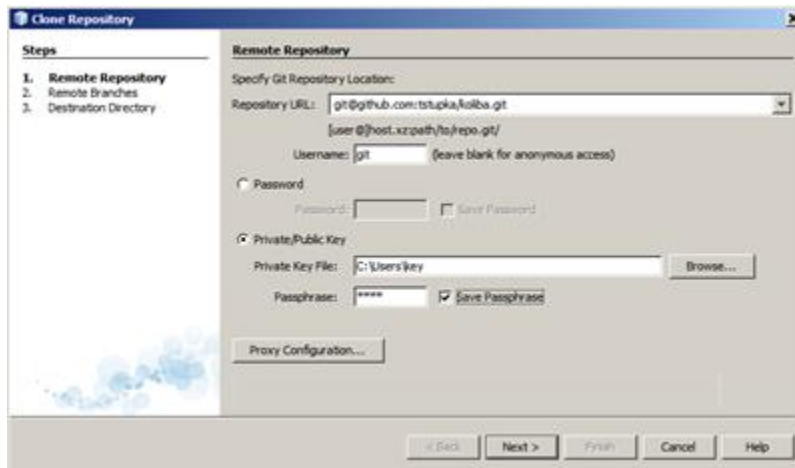
- No campo Nome do Clone, o nome da pasta local onde o projeto original será clonado. Por default, o Nome do Clone é preenchido com o nome real do repositório Git.
  - No campo Efetuar Check-out da Ramificação, selecione a ramificação cujo check-out deverá se feito na árvore de trabalho.
  - No campo Nome Remoto, o nome que representa o repositório original sendo clonado. `origin` é o apelido default do repositório sendo clonado. Esse é o valor recomendado.
  - Deixe selecionada a caixa de seleção Varrer por projetos do NetBeans após clonar para ativar a pós-varredura assim que a clonagem finalizar. (O plug-in pesquisa projetos NetBeans nos recursos clonados e se oferece para abrir os projetos encontrados).
7. Clique em Finalizar.
- Após o repositório do Git ser clonado, a pasta de metadados `.git` é criada dentro da pasta selecionada no assistente.

## Clonando um repositório a partir do GitHub por meio do Protocolo SSH

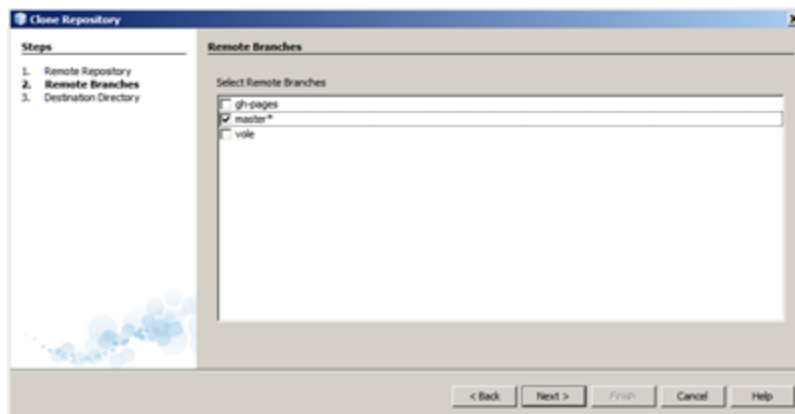
Para clonar um repositório por meio do GitHub usando o Protocolo SSH, faça o seguinte:

**Observação:** é preciso ter uma conta do GitHub e ser membro do projeto para poder clonar por meio de SSH.

1. Escolha `Equipe > Git > Clonar` no menu principal. O Assistente Clonar Repositório é Exibido.
  2. Na página Repositório Remoto do assistente Clonar Repositório, especifique o caminho para o repositório requerido no campo URL do Repositório, por exemplo,  
`git@github.com:tstupka/koliba.git`.
  3. Verifique se `git` está especificado no campo de texto Nome de Usuário.
  4. Selecione a opção Chave privada/pública.
  5. **(Ignorar se usar o agente SSH ou Pageant para acesso SSH automatizado para o servidor Git.)** Conclua as seguintes etapas para acessar o servidor Git usando a chave SSH privada e uma frase-senha:
    - a. Especifique o caminho para o arquivo da chave, por exemplo `C:\Users\key`.
- Observação:** o formato da chave privada `openssh` é necessário. Chaves geradas por PuTTYgen para Microsoft Windows deve ser convertido para o formato `openssh` antes de usá-los no IDE.
- b. Insira a senha para o arquivo da chave, por exemplo `abcd`.
    - c. (Opcional) Selecione a opção Salvar Senha, se necessário.
  6. **(Aplica-se se usar o agente SSH ou Pageant para acesso de SSH automatizado para o servidor Git.)** Deixe os campos Arquivo de Chave Privada e Frase-Senha vazios para obter acesso autenticado do IDE para o servidor Git, por meio de agente SSH ou Pageant configurado.
  7. (Opcional) Clique em Configuração de Proxy para exibir a caixa de diálogo Opções e especificar as definições do servidor proxy. Clique em OK quando finalizar.



8. Clique em Próximo.
9. Na página Ramificações Remotas, selecione as ramificações de repositório a serem extraídas (por download) para seu repositório local; por exemplo, `master`.

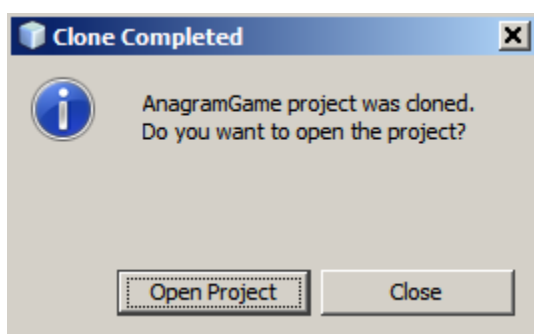


10. Clique em Próximo.
11. Na página Diretório de Destino, especifique o seguinte:
  - No campo Diretório-Pai, o caminho do diretório desejado para o repositório clonado em seu disco rígido (alternativamente, clique no botão Procurar e navegue para o diretório). O campo Diretório-Pai é preenchido com o caminho para o diretório default `NetBeansProjects` onde todos os projetos do NetBeans são armazenados.
  - No campo Nome do Clone, o nome da pasta local onde o projeto original será clonado. Por default, o Nome do Clone é preenchido com o nome real do repositório Git.
  - No campo Efetuar Check-out da Ramificação, selecione a ramificação cujo check-out deverá se feito na árvore de trabalho.
  - No campo Nome Remoto, o nome que representa o repositório original sendo clonado. `origin` é o apelido default do repositório sendo clonado. Esse é o valor recomendado.
  - Deixe selecionada a caixa de seleção Varrer por projetos do NetBeans após clonar para ativar a pós-varredura assim que a clonagem finalizar. (O plug-in pesquisa projetos NetBeans nos recursos clonados e se oferece para abrir os projetos encontrados).



12. Clique em Finalizar.

Após o repositório ter sido clonado, a mensagem Clonagem Concluída é exibida.





13. Escolha a opção desejada.

## Adicionando Arquivos em um Repositório Git

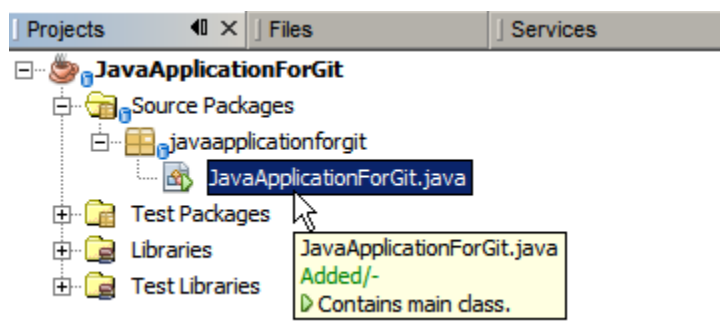
Para começar a rastrear um novo arquivo e também colocar em armazenamento temporário as alterações de um arquivo já rastreado no repositório do Git, é preciso adicioná-lo no repositório.

Quando você adiciona arquivos em um repositório do Git, o IDE compõe e salva snapshots primeiro de seu projeto no Índice. Após você executar a confirmação, o IDE salva estes snapshots no HEAD. O IDE permite escolher entre os dois fluxos de trabalho descritos na tabela a seguir.

| Descrição do Fluxo de Trabalho       | Adicione explicitamente arquivos novos ou modificados ao Índice e confirme somente os arquivos em armazenamento temporário no Índice para o HEAD  | Ignore a adição de arquivos novos ou modificados para o Índice e confirme os arquivos requeridos diretamente para o HEAD  |
|--------------------------------------|---|---|
| Etapas a Seguir no Fluxo de Trabalho | <ol style="list-style-type: none"> <li>1. Na janela Projetos, clique com o botão direito do mouse no arquivo que deseja adicionar.</li> <li>2. No menu de contexto, escolha <code>Git &gt; Adicionar</code>.<br/>Esse procedimento</li> </ol> | <ol style="list-style-type: none"> <li>1. Na janela Projetos, clique com o botão direito do mouse no arquivo que deseja confirmar.</li> <li>2. No menu de contexto, escolha <code>Git &gt; Confirmar</code>.</li> </ol> |

- adiciona o conteúdo do arquivo ao Índice antes de confirmá-lo.
3. Na janela Projetos, clique com o botão direito do mouse no arquivo que deseja confirmar.
4. Na caixa de diálogo Confirmar, selecione as alterações entre o botão de alternância HEAD e índice (  ). Isso exibe uma lista dos arquivos que já estão em armazenamento temporário.
5. Confirme os arquivos como descrito na seção [Confirmando Código-fonte em um Repositório](#) abaixo.
3. Na caixa de diálogo Confirmar, marque Selecionar as Alterações entre o botão de alternância Índice e Árvore de Trabalho (  ). Isso exibe uma lista dos arquivos que não estão em armazenamento temporário.
4. Confirme os arquivos como descrito na seção [Confirmando Código-fonte em um Repositório](#) abaixo.

**Observação:** o [status](#) do arquivo no HEAD é exibido em verde na esquerda da barra, como mostrado na figura a seguir.



A ação funciona de forma recursiva se chamada em pastas enquanto respeita a estrutura de conteúdo de pasta plana do NetBeans IDE.

## Editando Arquivos

Após ter um projeto com controle de versão Git aberto no IDE, é possível começar a fazer alterações nos códigos-fonte. Como acontece com qualquer projeto aberto no NetBeans IDE, é possível abrir os arquivos no Editor de Código-fonte clicando duas vezes em seus nós, conforme eles aparecem nas janelas do IDE (ou seja, janelas Projetos (Ctrl-1), Arquivos (Ctrl-2) e Favoritos (Ctrl-3)).




Quando você trabalha com códigos-fonte no IDE, há vários componentes de IU a sua disposição, o que ajuda a exibir e usar os comandos de controle de versão:

- [Exibindo Alterações no Editor de Código-fonte](#)
- [Exibindo Informações de Status do Arquivo](#)

- [Revertendo Alterações](#)

## Exibindo alterações no Editor de Código-fonte

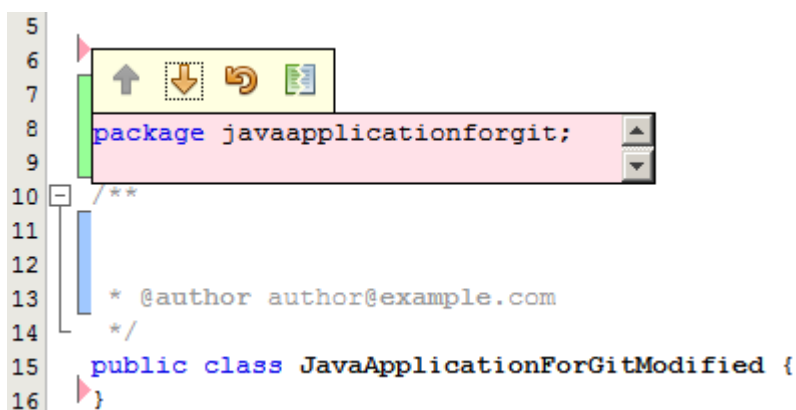
Quando abre um arquivo com controle de versão no Editor de Código-fonte do IDE, você pode ver alterações em tempo real acontecendo em seu arquivo, conforme ele é modificado em relação à versão base do repositório do Git. Conforme você trabalha, o IDE usa a codificação de cor nas margens do Editor de Código-fonte para fornecer as seguintes informações:

- Azul** (  ) Indica as linhas alteradas desde a revisão anterior.
- Verde** (  ) Indica as linhas adicionadas desde a revisão anterior.
- Vermelho** (  ) Indica as linhas removidas desde a revisão anterior.

A margem esquerda do Editor de Código-fonte mostra as alterações que ocorrem linha por linha. Quando você modifica uma determinada linha, as alterações são imediatamente mostradas na margem esquerda.



**Observação:** você pode clicar em um agrupamento colorido na margem para chamar todos os comandos de controle de versão. Por exemplo, a figura abaixo mostra os widgets disponíveis quando você clica em um ícone vermelho, indicando que linhas foram removidas de sua cópia local:



A margem direita do Editor de Código-fonte fornece uma visão geral que exibe as alterações feitas no arquivo como um todo, do início ao fim. A codificação de cor é gerada imediatamente quando você altera o arquivo.



```

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

```

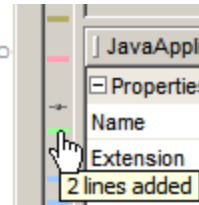


**Observação:** você pode clicar em um ponto específico na margem para trazer imediatamente o seu cursor em linha para esse local no arquivo. Para exibir o número de linhas afetadas, passe o mouse sobre os ícones coloridos na margem direita:


```

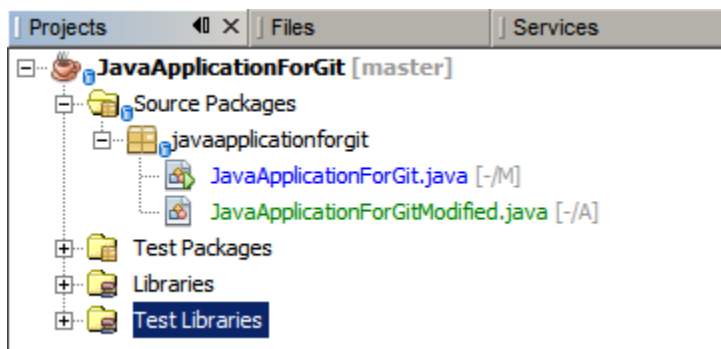
public String getDescription() {
    // This description will be displayed in the dialo
    // hard-coded = ugly, should be done via I18N
    return "Text documents (*.txt)";
}
}

```



## Exibindo Informações de Status do Arquivo

Quando você está trabalhando nas janelas Projetos (Ctrl-1), Arquivos (Ctrl-2), Favoritos (Ctrl-3) ou nas views de Controle de Versão, o IDE fornece algumas funcionalidades visuais que ajudam a exibir as informações de status sobre seus arquivos. No exemplo a seguir, observe como o emblema (por exemplo, , cor do nome do arquivo, e label de status adjacente, todos coincidem com outros para fornecer a você uma maneira efetiva e simples para controlar informações sobre seus arquivos de controle de versão:



Os emblemas, a codificação de cor, as legendas de status do arquivo e, talvez o mais importante, o Visualizador de Diferenciação Git, ajudam você a exibir e gerenciar as informações de controle de versão no IDE.

- [Emblemas e Codificação de Cor](#)
- [Labels de Status do Arquivo](#)
- [View Controle de Versão do Git](#)






## Emblemas e Codificação de Cor

Os emblemas são aplicados ao projeto, pasta e nós do pacote e informam o status dos arquivos contidos nesse nó:

A tabela a seguir exibe o esquema de cores usado nos emblemas:

| Componente de IU            | Descrição  |
|-----------------------------|--|
| <b>Emblema Azul</b> (🔵)     | Indica a presença de arquivos que foram modificados, adicionados ou deletados de sua árvore de trabalho. No caso de pacotes, esse emblema se aplica somente ao pacote em si, e não aos seus subpacotes. Para projetos ou pastas, o emblema indica as alterações nesse item, ou em qualquer uma das subpastas contidas. |
| <b>Emblema Vermelho</b> (🔴) | Marca projetos, pastas ou pacotes que contêm arquivos <i>conflitantes</i> . No caso de pacotes, esse emblema se aplica somente ao pacote em si, e não aos seus subpacotes. Para projetos ou pastas, o emblema indica os conflitos nesse item ou em qualquer uma das subpastas contidas.                                |

A codificação de cor é aplicada aos nomes de arquivo para indicar seu status atual em relação ao repositório:

| Cor                               | Exemplo   | Descrição  |
|-----------------------------------|---|--|
| <b>Sem cor específica (preto)</b> |  Main.java | Indica que o arquivo não tem alterações.   |
| <b>Azul</b>                       |  Main.java | Indica que o arquivo foi modificado localmente.  |
| <b>Verde</b>                      |  Main.java | Indica que o arquivo foi adicionado localmente.  |
| <b>Vermelho</b>                   |  Main.java | Indica que o arquivo tem conflito de mesclagem.  |
| <b>Cinza</b>                      |  Main.java | Indica que o arquivo é ignorado pelo Git e não será incluído nos comandos de controle de versão (ou seja, Atualizar e Confirmar). Os arquivos não podem ser ignorados se estiverem sob controle de versão. |

## Labels de Status do Arquivo

O IDE exibe dois valores de status para um arquivo:

- Um status que descreve as diferenças entre os arquivos na Árvore de Trabalho e no estado de Índice,
- Um status que descreve as diferenças entre os arquivos no estado de Índice e a confirmação atual no HEAD.

Os labels de status do arquivo fornecem uma indicação textual do status dos arquivos com controle de versão nas janelas do IDE:

| Label do Status | Significado                 |
|-----------------|-----------------------------|
| -               | Não modificado              |
| A               | Adicionado                  |
| U               | Atualizado mas não mesclado |
| M               | Modificado                  |
| D               | Deletado                    |

|          |           |
|----------|-----------|
| <b>I</b> | Ignorado  |
| <b>R</b> | Renomeado |

Por default, o IDE exibe as informações de status (novo, modificado, ignorado etc.) e de pastas em texto cinza à direita dos arquivos, à medida que eles são listados em janelas.



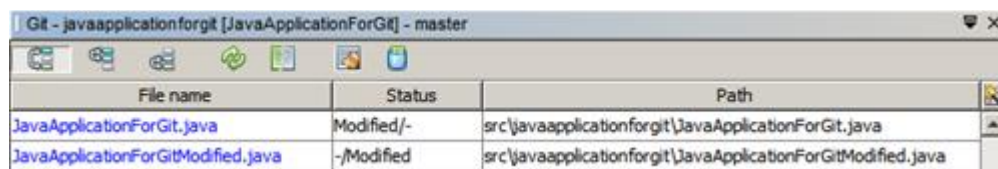
Os arquivos com a funcionalidade de conflito de mesclagem têm o status de não mesclado que é normalmente anotado em vermelho até que os arquivos não sejam resolvidos por sua ação explícita. O label de de status para arquivos não mesclados depende do cenário (por exemplo, A/A - não mesclado, ambos adicionados).

Os labels de status do arquivo podem ser ativados e desativados, ao selecionar **Exibir > Mostrar Labels de Controle de Versão** no menu principal.

## View Controle de Versão do Git

A view Controle de Versão do Git fornece uma lista em tempo real de todas as alterações feitas nos arquivos em uma pasta selecionada de sua árvore de trabalho local. Ela é aberta por default no painel inferior do IDE, listando os arquivos adicionados, deletados ou modificados.

Para abrir a view Controle de Versão, selecione um arquivo ou pasta com controle de versão (ou seja, na janela Projetos, Arquivos ou Favoritos) e escolha **Git > Mostrar Alterações** no menu do botão direito do mouse ou escolha **Equipe > Mostrar Alterações** no menu principal. A janela seguinte aparece na parte inferior do IDE:



Por default, a view Controle de Versão exibe uma lista de todos os arquivos modificados no pacote ou pasta selecionada na Árvore de Trabalho. Usando o botões da barra de ferramentas, é possível optar por exibir a lista de arquivos que têm diferenças entre o Índice e o HEAD, entre a Árvore de Trabalho e o Índice ou entre a Árvore de Trabalho e o HEAD. É possível clicar nos cabeçalhos das colunas acima dos arquivos listados para classificá-los por nome, status ou local.

A barra de ferramentas da view Controle de Versão também inclui botões que permitem que você chame as tarefas Git mais comuns em todos os arquivos exibidos na lista. A tabela a seguir lista os comandos do Git disponíveis na barra de ferramentas da view Controle de Versão:

| Ícone | Nome  | Função   |
|-------|---|--|
|       | <b>Alterações entre HEAD e Árvore de Trabalho</b> | Exibe uma lista de arquivos que já estão em armazenamento temporário ou somente modificados/criados e ainda não estão em armazenamento temporário. |
|       | <b>Alterações entre HEAD e Índice</b>             | Exibe uma lista de arquivos que estão em armazenamento temporário.   |



### Alterações entre o Índice e a Árvore de Trabalho

Exibe os arquivos que têm diferenças entre seus estados de armazenamento temporário e na Árvore de Trabalho.



### Atualizar Status

Atualiza o status dos arquivos e pastas selecionados. Os arquivos exibidos na view Controle de Versão podem ser atualizados para refletir quaisquer alterações que possam ter sido feitas externamente.



### Abrir Diferença

Abre o Visualizador de Diferenciação que fornece uma comparação lado a lado de suas cópias locais e das versões mantidas no repositório.



### Reverter Modificações

Exibe a caixa de diálogo [Reverter Modificações](#).



### Confirmar Alterações

Exibe a caixa de diálogo [Confirmar](#).

Você pode acessar outros comandos do Git na view Controle de Versão, selecionando uma linha da tabela que corresponde a um arquivo modificado, e escolhendo um comando do menu com o botão direito do mouse:



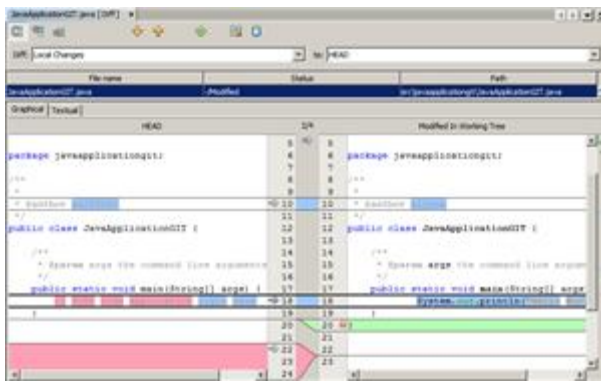
## Comparando Revisões de Arquivos

Comparar versões de arquivo é uma tarefa comum ao se trabalhar com projetos com controle de versão. O IDE permite comparar versões ao usar o comando Diferenciar:

**Observação:** vários modos de comparação - Diff To HEAD, Diff To Tracked e Diff To - estão disponíveis no IDE.

1. Selecione um arquivo ou pasta com versão (por exemplo, na janela Projetos, Arquivos ou Favoritos).
2. Escolha **Team > Diff > Diff to HEAD** no menu principal.

Um Visualizador gráfico de Diferenciação é aberto para os arquivos selecionados na janela principal do IDE. O Visualizador de Diferenciação exibe duas cópias nos painéis lado a lado. Quanto mais atual a cópia aparecer no lado direito, ou seja, você está comparando uma revisão no repositório contra sua árvore de trabalho, a árvore de trabalho é exibida no painel direito:






O Visualizador de Diferenciação usa a mesma [codificação de cor](#) usada para exibir alterações de controle de versão. Na captura de tela exibida acima, o bloco verde indica o conteúdo que foi adicionado à revisão mais atual. O bloco vermelho indica que o conteúdo da revisão anterior foi removido da última revisão. Azul indica que as alterações ocorreram na(s) linha(s) realçada(s).

**Observação:** outras revisões podem ser selecionadas para listas drop-down `Diff` e `to` abaixo da barra de ferramentas Visualizador de Diferenciação.

A barra de ferramentas Visualizador de Diferenciação também inclui botões que permitem chamar as tarefas mais comuns do Git em todos os arquivos exibidos na lista. A tabela a seguir lista os comandos do Git disponíveis na barra de ferramentas do Visualizador de Diferenciação:

| Ícone | Nome  | Função   |
|-------|---|--|
|       | <b>Alterações entre HEAD e Árvore de Trabalho</b>       | Exibe uma lista de arquivos que já estão em armazenamento temporário ou somente modificados/criados e ainda não estão em armazenamento temporário.                                       |
|       | <b>Alterações entre HEAD e Índice</b>                   | Exibe uma lista de arquivos que estão em armazenamento temporário.   |
|       | <b>Alterações entre o Índice e a Árvore de Trabalho</b> | Exibe os arquivos que têm diferenças entre seus estados de armazenamento temporário e na árvore de trabalho.   |
|       | <b>Ir para a Próxima Diferença</b>                      | Exibe a próxima diferença no arquivo.  |
|       | <b>Ir para a Diferença Anterior</b>                     | Exibe a diferença anterior no arquivo.   |
|       | <b>Atualizar Status</b>                                 | Atualiza o status dos arquivos e pastas selecionados. Os arquivos exibidos na janela de Controle de Versão podem ser atualizados para refletir quaisquer alterações feitas externamente. |
|       | <b>Reverter Modificações</b>                            | Exibe a caixa de diálogo <a href="#">Reverter Modificações</a> .   |
|       | <b>Confirmar Alterações</b>                             | Exibe a caixa de diálogo <a href="#">Confirmar</a> .   |

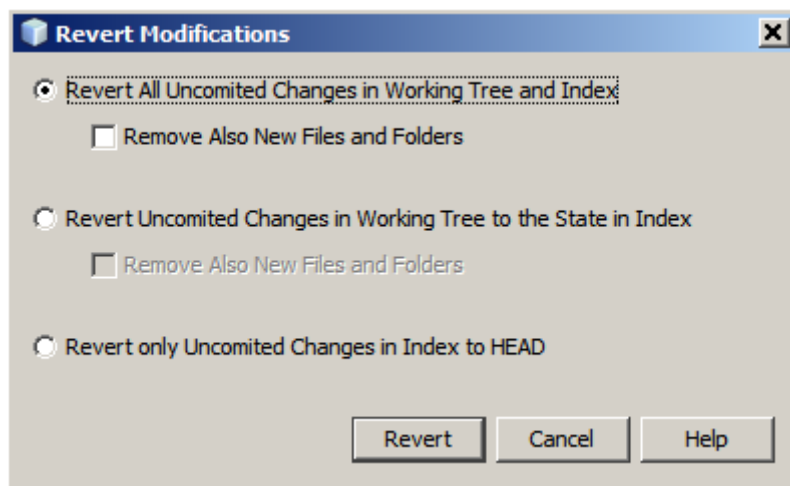
Se você estiver executando uma diferença em sua cópia de trabalho local, o IDE permite fazer alterações diretamente no Visualizador de Diferenciação. Para fazer isso, você pode colocar o cursor no painel direito do Visualizador de Diferenciação e modificar seu arquivo adequadamente, caso contrário, use os ícones in-line exibidos ao lado de cada alteração realçada:

| Ícone   | Nome               | Função  |
|---|--------------------|---|
|  | <b>Substituir</b>  | Insere o texto realçado em sua cópia da Árvore de Trabalho. |
|  | <b>Mover Todos</b> | Reverte toda a cópia local da Árvore de Trabalho.           |
|  | <b>Remover</b>     | Remove o texto realçado da cópia local da Área de Trabalho. |

## Revertendo Alterações

Para descartar as alterações locais feitas nos arquivos selecionados em sua árvore de trabalho e substituir os arquivos usando os que estejam no Índice ou HEAD:

1. Selecione um arquivo ou pasta com versão (por exemplo, na janela **Projetos**, **Arquivos** ou **Favoritos**).
2. Escolha **Equipe > Reverter Modificações** no menu principal.  
A caixa de diálogo **Reverter Modificações** é exibida.



3. Especificar opções adicionais (por exemplo, **Reverter Somente as Alterações Não Confirmadas no Índice para o HEAD**).
4. Clique em **Reverter**.

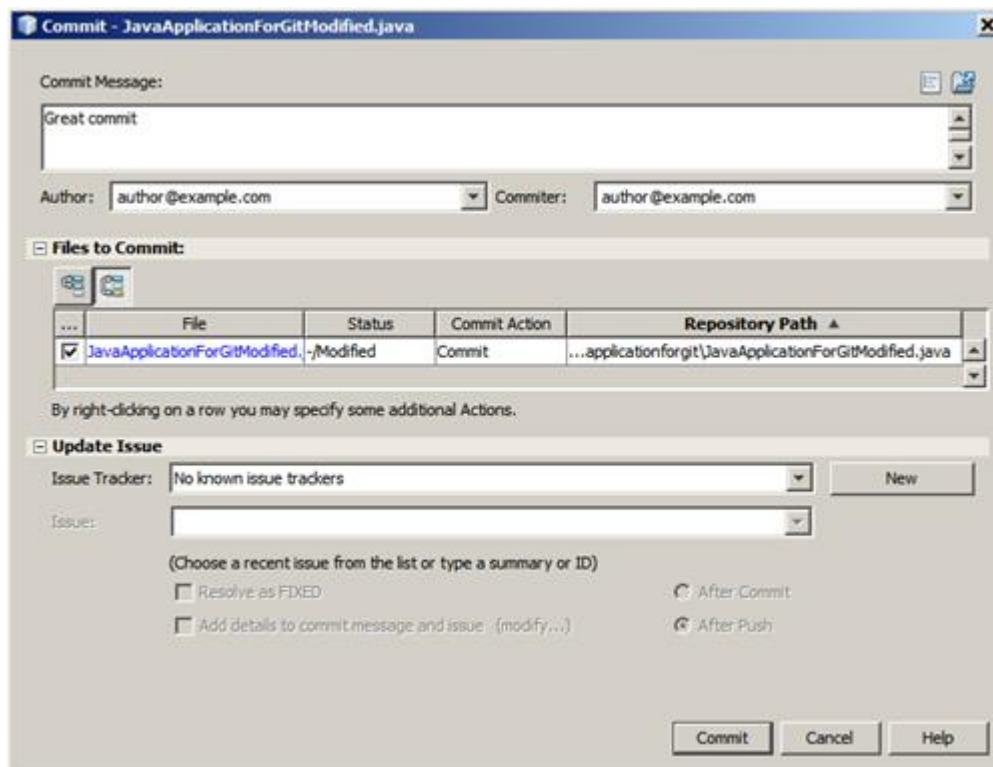
O IDE substitui os arquivos selecionados pelos usando os especificados na [etapa 3](#) acima.

## Confirmando Códigos-fonte em um Repositório

Para confirmar arquivos para o repositório Git:

1. Na janela `Projetos`, clique com o botão direito do mouse no arquivo que deseja confirmar.
2. No menu de contexto, escolha `Git > Confirmar`.



A caixa de diálogo `Confirmar` é exibida.



A caixa de diálogo `Confirmar` contém os seguintes componentes:



- A área de texto `Mensagem de Confirmação` descreve a alteração que está sendo confirmada
- As listas drop-down `Autor` e `Confirmador` permitem distinguir entre quem fez a alteração e quem confirmou fisicamente o arquivo, se necessário.
- A seção `Arquivos a serem Confirmados` que lista:
  - todos os arquivos modificados,
  - todos os arquivos que foram deletados da `Árvore de Trabalho` (localmente),
  - todos os arquivos novos (ou seja, arquivos que ainda não existem no repositório `Git`),
  - todos os arquivos que você renomeou.

Dois botões de alternância que alternam o modo no qual a confirmação deve ser executada estão aqui disponíveis:

| Componente de IU  | Nome                                       | Descrição  |
|---|--|--|
|  | Alterações entre HEAD e Índice             | Exibe uma lista de arquivos que estão em armazenamento temporário.   |
|  | Alterações entre HEAD e Árvore de Trabalho | Exibe uma lista de arquivos que já estão em armazenamento temporário ou somente modificados/criados e ainda não estão em armazenamento temporário. |

- **Observação:** para especificar se os arquivos individuais devem ser excluídos da confirmação, cancele a seleção da caixa de seleção na primeira coluna denominada Confirmar ou clique com o botão direito do mouse em uma linha de arquivo na coluna Ação de Confirmação e escolha Excluir da confirmação no menu pop-up. Para exibir aqui o Visualizador de Diferenciação, clique com o botão direito do mouse em uma linha de arquivo na coluna Ação de Confirmação e escolha Diferenciar no menu pop-up.
- A seção Atualizar Problema acompanha os problemas relativos à alteração que está sendo confirmada.

**Observação:** é preciso instalar o plug-in JIRA ou Subversion para iniciar o rastreamento de problemas no IDE.

3. Digite uma mensagem de confirmação na área de texto Mensagem de Confirmação. Alternativamente, você pode executar um dos seguintes procedimentos:
  - clique no ícone Mensagens Recentes () localizado no canto superior direito para ver e selecionar de uma lista de mensagens que você usou anteriormente existente.
  - Clique no ícone Carregar Modelo () localizado no canto superior direito para selecionar um modelo de mensagem.
4. Depois de especificar ações para arquivos individuais, clique em Confirmar. O IDE executa a confirmação e armazena seus snapshots no repositório. A barra de status do IDE, localizada na parte inferior direita da interface, é exibida conforme a ação de confirmação ocorre. Com uma confirmação bem-sucedida, os emblemas de controle de versão desaparecem nas janelas Projetos, Arquivos e Favoritos e a codificação de cor dos arquivos confirmados volta a ser preta.



## Trabalhando com Ramificações

O suporte ao Git do IDE permite manter diferentes versões de toda uma base de códigos usando ramificações.

Quando você trabalha com ramificações no IDE, as seguintes ações são suportadas:

- [Criando](#)
- [Fazendo check-out](#)
- [Mesclando](#)
- [Deletando](#)

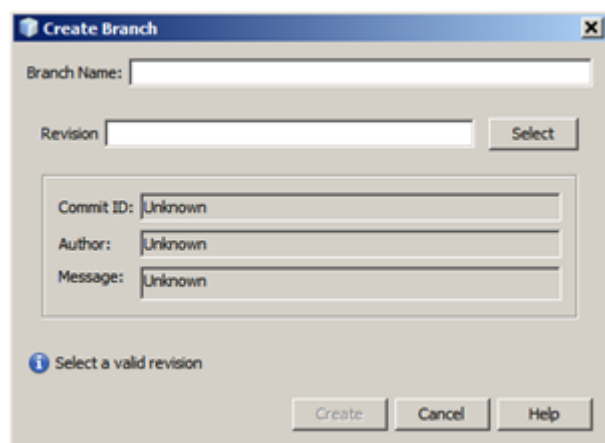
### Criando uma Ramificação

Para criar uma ramificação local, caso deseje trabalhar em uma versão separada de seu sistema de arquivos para fins de estabilização ou experimento sem perturbar o tronco principal, complete as seguintes etapas:

1. Na janela Projetos ou Arquivos, escolha um projeto ou pasta do repositório no qual deseja criar a ramificação.
2. No menu principal, escolha Equipe > Ramificação/Tag > Criar Ramificação.

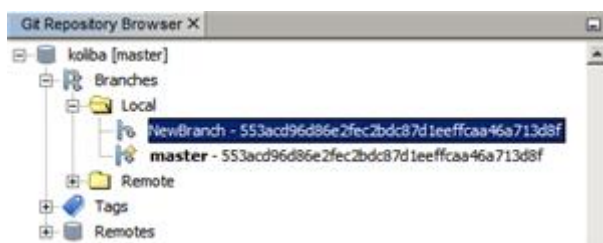
**Observação:** como alternativa, clique com o botão direito do mouse no projeto ou pasta com controle de versão e escolha Git > Ramificação/Tag > Criar Ramificação no menu pop-up.

A caixa de diálogo Criar Ramificação é exibida.



3. No campo Nome da Ramificação, insira o nome da ramificação que está sendo criada.
4. Especifique a revisão necessária ao inserir um ID de confirmação, ramificação existente, ou nome da tag no campo Revisão, ou pressione Selecionar para exibir uma lista das revisões mantidas no repositório.
5. (Opcional) Na caixa de diálogo Selecionar Revisão, expanda Ramificações e escolha a ramificação necessária, especifique o ID de confirmar na lista adjacente e pressione Selecionar.
6. Reveja as informações dos campos ID da Confirmação, Autor e Mensagem específicos da revisão sendo ramificada.

A ramificação é adicionada à pasta Ramificações/Local do repositório Git.

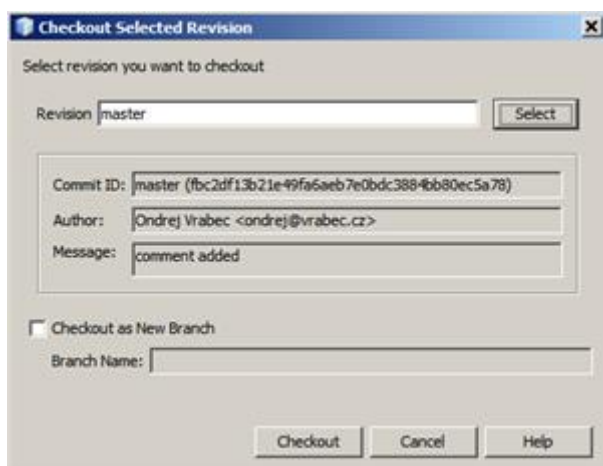


## Fazendo Check-out

Se precisar editar arquivos em uma ramificação existente, você pode fazer check-out da ramificação para copiar os arquivos para sua Árvore de Trabalho.

Para fazer check-out de uma versão, proceda da seguinte forma:

1. Escolha Equipe > Check-out > Fazer Check-out da Revisão no menu principal. A caixa de diálogo Fazer Check-out da Revisão Selecionada é exibida.



2. Especifique a revisão necessária ao inserir um ID de confirmação, uma ramificação existente ou o nome da tag no campo Revisão, ou pressione Selecionar para exibir uma lista das revisões mantidas no repositório.
3. Ignore se você não pressionou Selecionar na etapa anterior. Na caixa de diálogo Selecionar Revisão, expanda Ramificações e escolha a ramificação necessária, especifique o ID de confirmação na lista adjacente, se necessário, e pressione Selecionar.

**Observação:** se a revisão especificada se refere a uma confirmação válida que não esteja marcada com um nome de ramificação, seu HEAD se torna desanexado e você não mais está em nenhum ramificação.

4. Reveja as informações dos campos ID de Confirmação, Autor e Mensagem específicos da revisão cujo check-out está sendo feito.
5. Para criar uma nova ramificação da versão submetida a check-out, escolha a opção Efetuar Check-out como uma Nova Ramificação e insira o nome no campo Nome da Ramificação.
6. Pressione Check-out para fazer check-out da revisão.  
Os arquivos na Árvore de Trabalho e no Índice são atualizados para coincidirem com a versão na revisão especificada.

**Observação:** caso deseje alternar seus arquivos para uma ramificação existente (por exemplo, para uma confirmação que não esteja no topo de uma de suas ramificações), você pode usar o comando Equipe > Git > Ramificação > Alternar para a Ramificação, especificar a ramificação na caixa de diálogo Alternar

para a Ramificação Seleccionada, fazer check-out como uma nova ramificação (opcionalmente) e pressionar Alternar.

O IDE suporta check-out contextual de arquivos, pastas ou projetos no momento seleccionados no IDE. Para fazer check-out de alguns arquivos (e não de uma ramificação), siga as seguintes etapas:

1. Escolha Equipe > Check-out > Fazer Check-out de Arquivos no menu principal. A caixa de diálogo Fazer Check-out dos Caminhos Seleccionados é exibida.

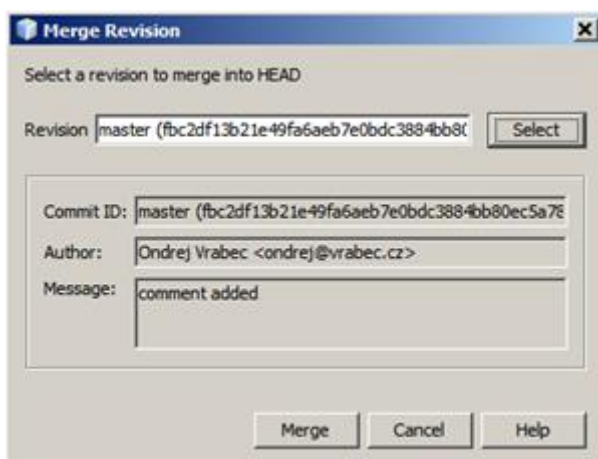


2. Escolha a opção Atualizar Índice com Entradas da Revisão Seleccionada. Caso seleccionado, o Índice é atualizado com o estado da revisão seleccionada antes do check-out do arquivo (ou seja, os arquivos seleccionados na Árvore de Trabalho e Índice são atualizados).
3. Especifique a revisão necessária ao inserir um ID de confirmação, uma ramificação existente ou o nome da tag no campo Revisão, ou pressione Seleccionar para exibir uma lista das revisões mantidas no repositório.
4. Ignore se você não pressionou Seleccionar na etapa anterior. Na caixa de diálogo Seleccionar Revisão, expanda Ramificações e escolha a ramificação necessária, especifique o número da revisão na lista adjacente se necessário e pressione Seleccionar.
5. Pressione Check-out para concluir o check-out.

## Mesclando

Para portar as modificações de uma revisão do repositório para a Árvore de Trabalho, faça o seguinte:

1. Escolha Equipe > Ramificação/Tag > Mesclar Revisão no menu principal. A caixa de diálogo Mesclar Revisão é exibida.



2. Especifique a revisão necessária ao inserir um ID de confirmação, uma ramificação existente ou o nome da tag no campo Revisão, ou pressione Selecionar para exibir uma lista das revisões mantidas no repositório.
3. Ignore se você não pressionou Selecionar na etapa anterior. Na caixa de diálogo Selecionar Revisão, expanda Ramificações e escolha a ramificação necessária, especifique o ID de confirmação na lista adjacente, se necessário, e pressione Selecionar.
4. Pressione Mesclar.  
Uma mesclagem de três vias entre a ramificação atual, o conteúdo de sua Árvore de Trabalho e a ramificação especificada é feita.

**Observação:** caso ocorra um conflito de mesclagem, o arquivo conflitante é marcado com um [emblema vermelho](#) para indicar isso.

**Observação:** depois da mesclagem, ainda é necessário [confirmar](#) as alterações para que sejam adicionadas ao HEAD.

## Deletando uma Ramificação

Para deletar uma ramificação local desnecessária, conclua as seguintes etapas:

1. Escolha Equipe > Browser de Repositório no menu principal.
2. No Browser de Repositório Git, escolha a ramificação a ser deletada.

**Observação:** a ramificação precisa estar inativa, ou seja, não deve ter tido check-out na Árvore de Trabalho.

3. Clique com o botão direito do mouse na ramificação selecionada e escolha Deletar Ramificação a partir do menu pop-up.
4. Na caixa de diálogo Deletar Ramificação, pressione OK para confirmar a exclusão da ramificação. A ramificação é removida do repositório local, assim como do Browser de Repositório Git.

## Trabalhando com Repositórios Remotos

Quando você trabalha com outros desenvolvedores precisa compartilhar seu trabalho, o que envolve extrair, inserir e obter dados em relação aos repositórios remotos hospedados na Internet ou em uma rede.

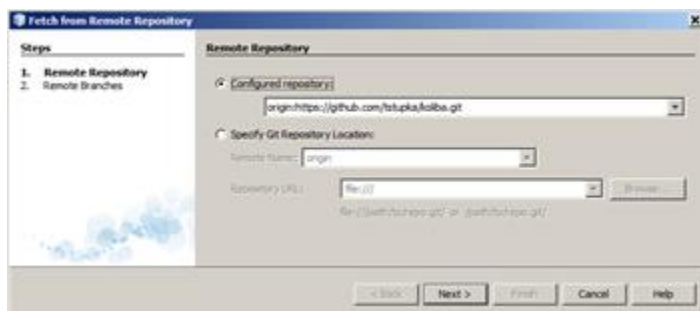
- [Extraindo](#)
- [Extraindo](#)
- [Expandindo](#)

### Extraindo

A extração recebe as alterações do repositório remoto original que você ainda não tem. Ela nunca altera quaisquer de suas ramificações locais. A extração recebe todas as ramificações dos repositórios remotos, que você pode mesclar com sua ramificação ou apenas inspecionar a qualquer momento.

Para extrair as atualizações, faça o seguinte:

1. Escolha Equipe > Remoto > Extrair.  
O assistente Extrair do Repositório Remoto é exibido.



- Na página Repositório Remoto do assistente, selecione a opção Repositório configurado (para usar o caminho para o repositório anteriormente configurado) ou a opção Especificar Local do Repositório Git (para definir o caminho para um repositório remoto que ainda não foi acessado, seu nome, e login e senha, e qualquer configuração de proxy necessária) e clique em Próximo.
- Na página Ramificações Remotas do assistente, escolha as ramificações para obter as alterações e clique em Finalizar.

Uma cópia local da ramificação remota é criada. A ramificação selecionada é atualizada no diretório Ramificações > Remotas no Browser de Repositório Git.

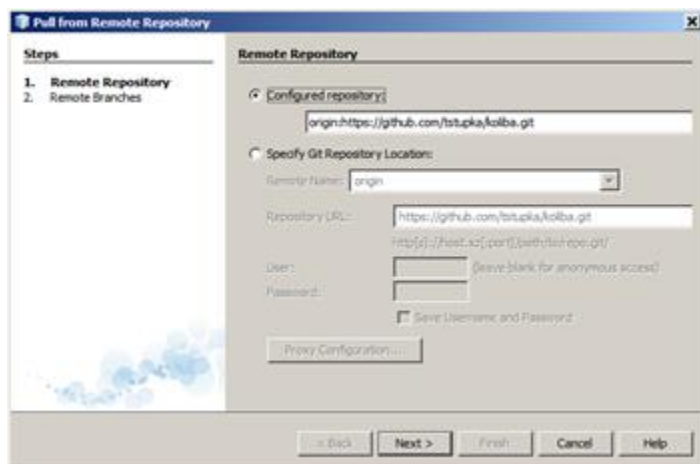
Em seguida, as atualizações extraídas podem ser mescladas em uma ramificação local.

## Extraindo

Durante a obtenção de algumas atualizações de um repositório Git remoto, as alterações são extraídas dele mesmo e mescladas no HEAD atual de seu repositório local.

Para executar a extração, conclua as seguintes etapas:

- Escolha Equipe > Remoto > Extrair.  
O assistente Extrair do Repositório Remoto é exibido.



- Na página Repositório Remoto do assistente, selecione a opção Repositório configurado (para usar o caminho do repositório anteriormente configurado) ou Especificar a Localização do Repositório Git: (para definir o caminho de um repositório remoto que ainda não foi acessado, seu nome, login e senha, se necessário) e clique em Próximo.
  - Na página Ramificações Remotas do assistente, escolha as ramificações a serem extraídas com as alterações e clique em Finalizar.
- Seu repositório local é sincronizado com o repositório de origem.

## Expandindo

Para contribuir com alterações de seu repositório Git local para um repositório Git local, execute as seguintes etapas:

1. Escolha Equipe > Remoto > Expandir.  
O assistente Expandir para o Repositório Remoto é exibido.



2. Na página Repositório Remoto do assistente, selecione a opção Repositório configurado (para usar o caminho do repositório anteriormente configurado) ou Especificar a Localização do Repositório Git: (para definir o caminho de um repositório remoto que ainda não foi acessado, seu nome, login e senha, se necessário) e clique em Próximo.
3. Na página Selecionar Ramificações Locais, escolha as ramificações de onde serão extraídas suas edições e clique em Próximo.
4. Na página Atualizar Referências Locais, escolha as ramificações a serem atualizadas no diretório Remoto de seu repositório local e clique em Finalizar.  
A ramificação do repositório remoto especificada é atualizada com o estado mais recente de sua ramificação local.

## Resumo

Este tutorial demonstrou como executar tarefas de controle de versão básicas no IDE ao guiá-lo pelo fluxo de trabalho padrão ao usar o suporte Git do IDE. Ele mostrou como configurar um projeto com controle de versão e como executar tarefas básicas em arquivos com controle de versão e apresentando-o algumas das novas funcionalidades do Git incluídos no IDE.

[Enviar Feedback neste Tutorial](#)

## Consulte Também

Para ver o material relacionado, consulte os seguintes documentos:

- [Usando o Suporte de ClearCase no NetBeans IDE](#)
- [Usando Suporte ao Subversion no NetBeans IDE](#)
- [Usando Suporte ao Mercurial no NetBeans IDE](#)
- [Usando Suporte ao CVS no NetBeans IDE](#)
- [Aplicando Controle de Versão às Aplicações com Controle de Versão](#) em *Desenvolvendo Aplicações com o NetBeans IDE*