

# Git

Alessandro

7 de agosto de 2011

## 1 O que é Git?

Git é um sistema de controle de versão distribuído, desenvolvido originalmente por Linus Torvalds para manter o kernel do Linux, com foco na eficiência e utilização em grandes projetos. Um característica importante do Git é que para realizar um *commit* não é necessário estar conectado ao repositório distribuído, ou seja, a sincronização remota só é necessária no momento que consideramos adequada. Em outras palavras, no Git, controlamos as alterações nos arquivos localmente, e posteriormente, informamos ao repositório “master” (remoto) sobre todas elas.

## 2 O que é um Sistema de Controle de Versão (VCS)?

Um sistema de controle de versão é um software responsável pelo controle e gerenciamento de diferentes versões na implementação de um arquivo, guardando o que foi alterado, quando e o responsável pela alteração.

O controle de versão é basicamente dividido em dois segmentos: o repositório, que é responsável pelo armazenamento das informações e a área de trabalho. O repositório é responsável pelas informações relacionadas ao projeto. Nele será guardado todo o histórico de desenvolvimento do projeto e registrado todo e qualquer modificação realizada nos arquivos do projeto. Já a área de trabalho, corresponde a um espaço privado à cada desenvolvedor para realização das modificações nos arquivos.

## 3 O que é preciso para utilizar o Git?

1. Instalar;

2. Criar um repositório local;
3. Criar um repositório remoto;
4. Comitar no repositório local;
5. Mandar ou recuperar as alterações do repositório local para o *master* (remoto).

## 3.1 Instalação

Temos versão do Git para os principais sistemas operacionais Linux, Mac e Windows.

### 3.1.1 Instalação no Linux

Para instalar o Git nas distribuições derivadas do Debian, como o Ubuntu, é necessário executar a instrução no console (terminal) do computador.

```
sudo aptitude install git-core git-doc git-web git-gui gitk git-email
```

### 3.1.2 Instalação no Windows

Para instalar o Git no Windows é necessário executar as seguintes ações:

1. Realizar o download no endereço: <http://goo.gl/Mhjbw>;
2. Após concluir o download, execute o programa de instalação e siga as instruções de acordo com sequência das figuras [1-8].

### 3.1.3 Criando um repositório local

Para criar um repositório, acesse o diretório a ser realizado o controle de versão e digite o comando: **git init**. O resultado prático da execução deste comando é um diretório oculto *.git* que contém todos os arquivos necessários para o repositório Git.

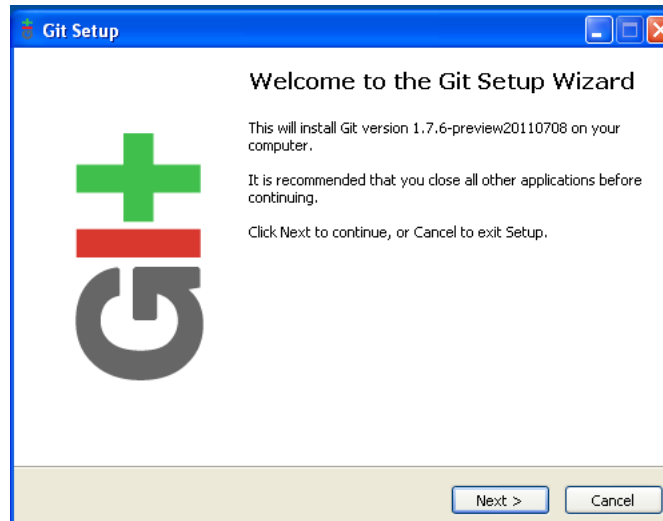


Figura 1: Tela de Instalação do Git.

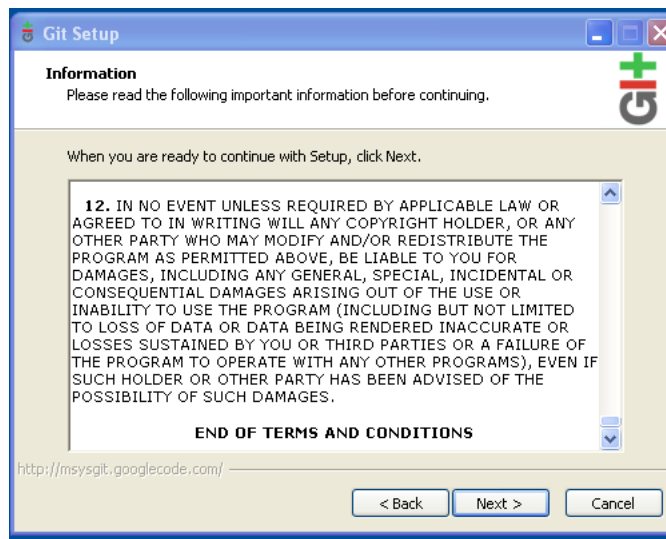


Figura 2: Termo de uso.

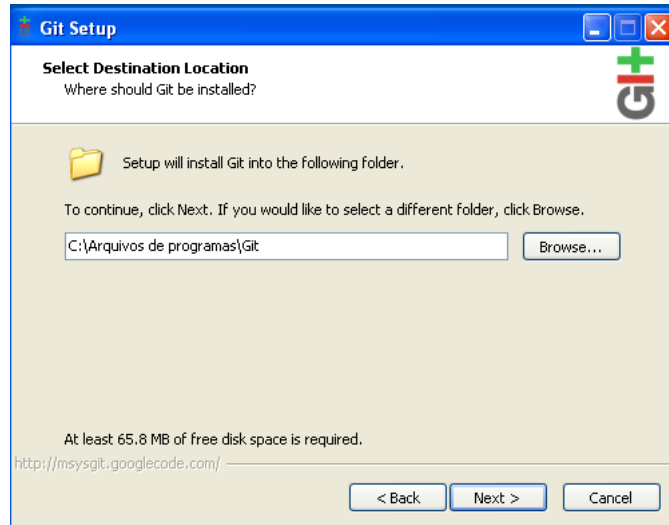


Figura 3: Diretório de instalação.

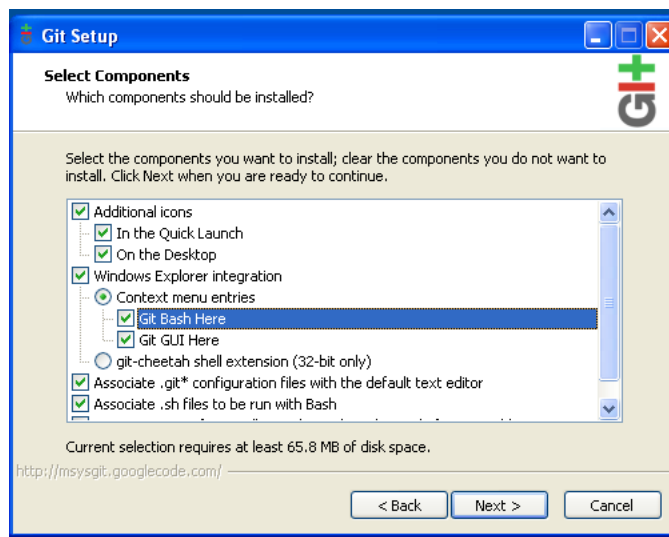


Figura 4: Componentes a serem instalados.

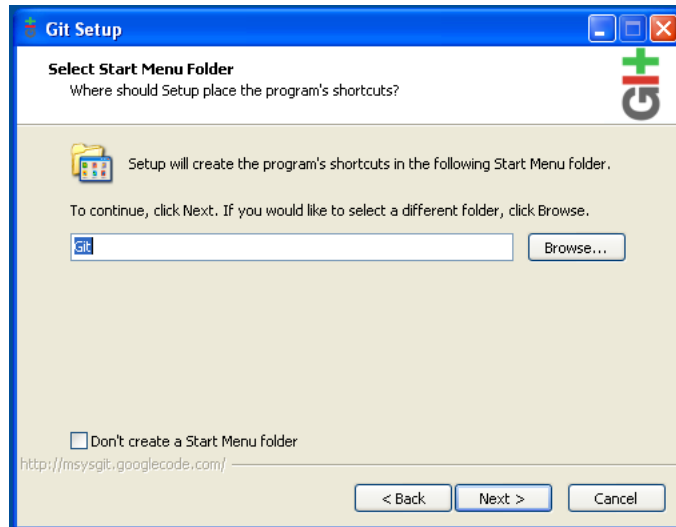


Figura 5: Inclusão do menu de inicialização

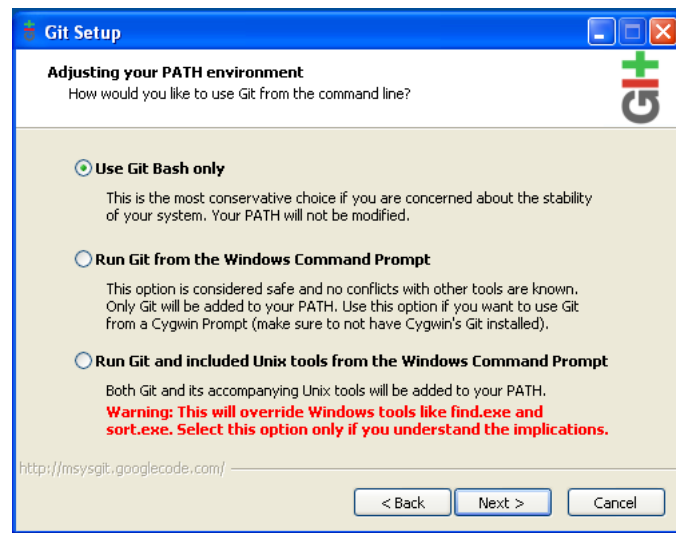


Figura 6: Configuração do ambiente de linha de comando.

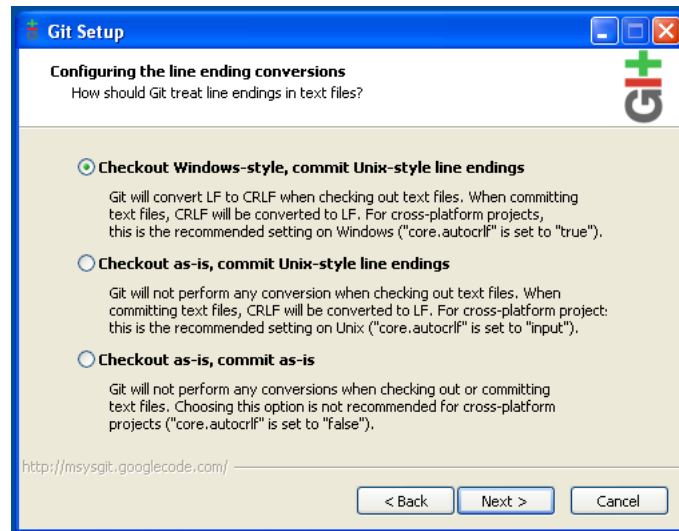


Figura 7: Configuração do padrão de quebra de linha.

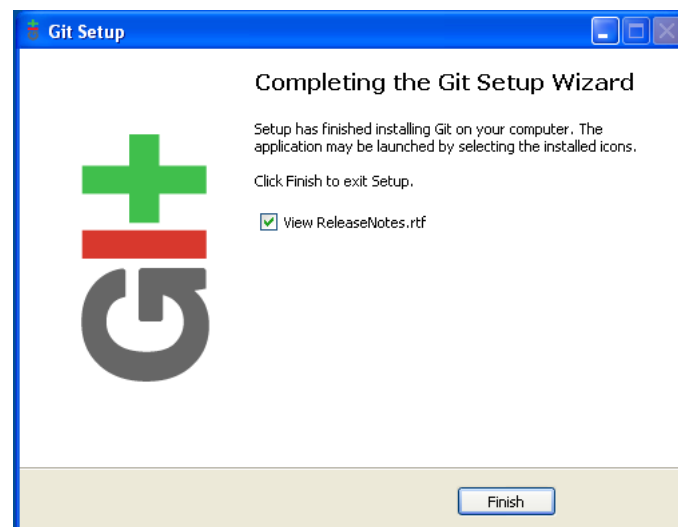


Figura 8: Conclusão da instalação.

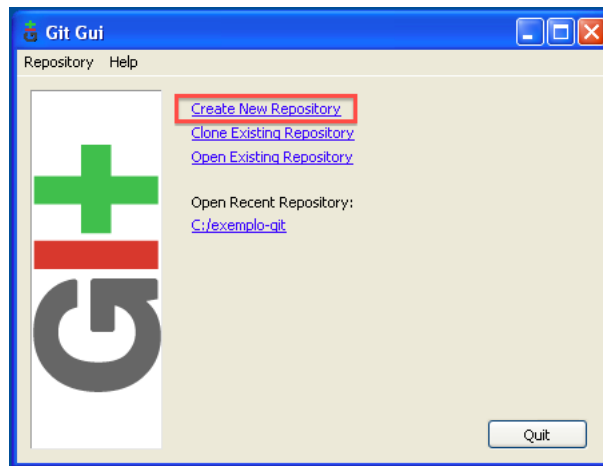


Figura 9: Git GUI.

1. No Windows podemos criar o repositório via linha de comando (Git Bash) ou pela interface gráfica (Git GUI). Vamos criar utilizando a interface gráfica (Git GUI) através do **Menu Iniciar** ⇒ **Git GUI** (Figura 9).
2. Clique na opção “*Create New Repository*” (Figura 10);

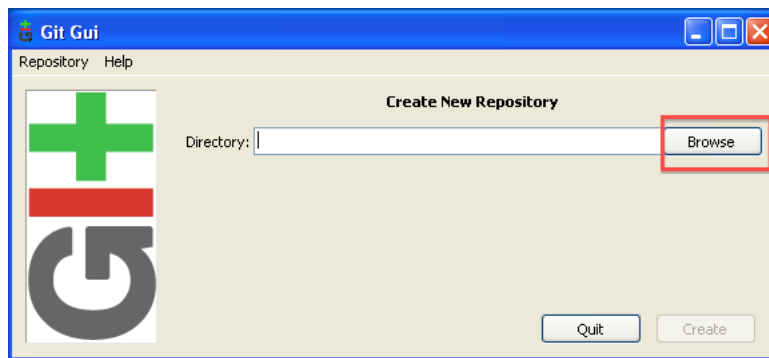


Figura 10: Criando um repositório Git.

3. Escolha o diretório a ser criado o repositório Git (Figura 11);
4. Por fim, clique em **Create** (Figura 11 e será apresentada a Figura 12).

Na figura 12, temos quatro seções, a saber:

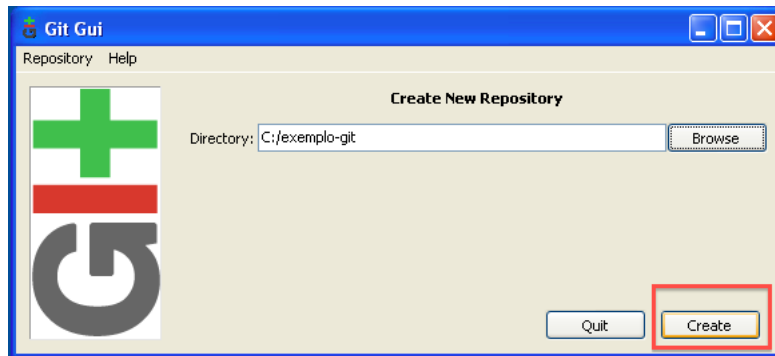


Figura 11: Diretório a ser criado o repositório Git.

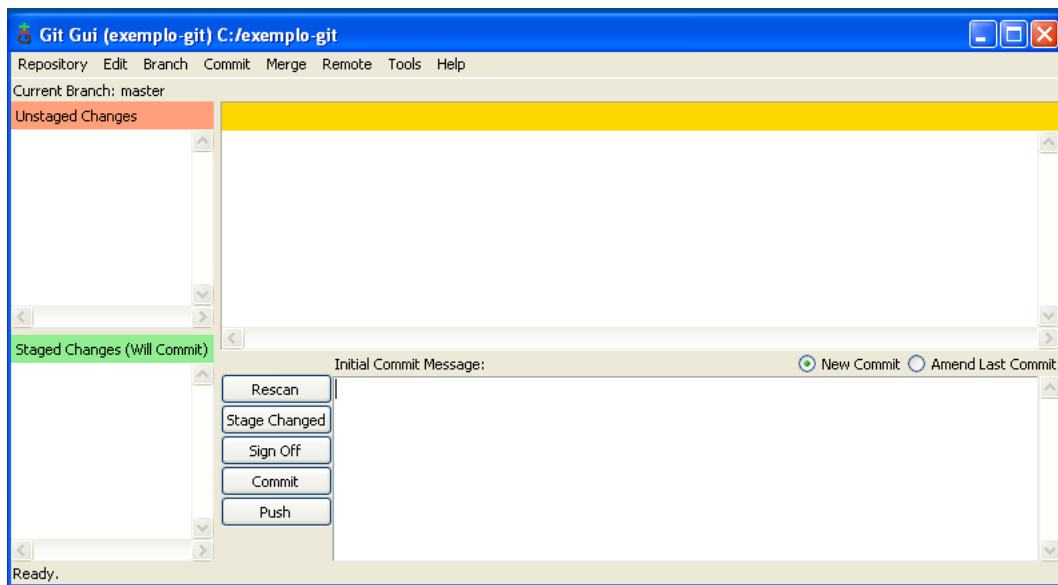


Figura 12: Visualização do repositório.



1. Unstaged Changes: Relaciona os arquivos estão no diretório sem no entanto, terem sido adicionados ao repositório.
2. Staged Changes (Will Commit): Relaciona os arquivos que estão prontos para serem **comitados**;
3. Changed Files (amarela): Fornece informações sobre o(s) arquivo(s) selecionado(s);
4. Initial Commit Message: Mensagem a ser utilizada no commit do(s) arquivo(s).

Os arquivos que se encontram no estado de “*Unstaged Changes*” para serem comitados, primeiro precisam ser adicionados (**git add**). Esse procedimento é realizado através do menu: Commit → Stage to Commit (Ctrl + T). Após adicionar o(s) arquivo(s), precisamos comitá-lo(s). Informe as mudanças realizadas (o que está sendo comitado) e clique no botão **Commit**.

### 3.1.4 Criar um repositório remoto

Há várias opções de repositórios de códigos remotos que disponibilizam espaço gratuito. O mais conhecido deles é o Github (<http://github.com>). O Github é o serviço mais utilizado para repositórios Git, funcionando até mesmo como uma rede social de programadores.

#### Criando uma conta no Github

1. Para criar um repositório no Github é preciso ter uma conta. Se você ainda não tem, acesse o endereço <https://github.com/signup/free> e crie uma.
2. Adicione a tua chave pública. Para isso, é necessário primeiro criar uma. A ferramenta Git GUI nos possibilita criarmos uma facilmente.
  - (a) Volte a ferramenta Git GUI e acesse o menu: Help → Show SSH Key;
  - (b) Clique no botão **Generate Key** e na tela seguinte informe uma senha (passphrase);
  - (c) Após confirmar a senha, clique no botão **Copy to Clipboard**.
3. Agora que já temos a nossa chave SSH pública, precisamos informá-la ao Github.

4. Acesse <https://github.com/account/ssh> e clique no link **Add another public key**.
5. Informe um título para identificar a chave, cole o texto no campo **Key** e clique em **Add key**.

### **Criando um repositório no Github**

1. Procure pelo botão de novo repositório (**New Repository**).
2. Informe um nome (project name), descrição (description) e clique em **Create Repository**.
3. Copie a URI do repositório remoto criado para adicioná-lo ao repositório local. Exemplo de URI: `git@github.com:alessandrofleite/ltp2.git`.

### **Adicionar um repositório remoto**

1. Volte a ferramenta Git GUI e acesse o menu: Remote → Add ...;
2. Informe o nome do repositório e a URI (Location) do repositório.
3. Clique em **Add**.
4. Informe a senha (passphrase).

#### **3.1.5 Mandar ou recuperar as alterações do repositório local para o *master* (remoto)**

1. Para enviar as alterações ao repositório remoto, volte a ferramenta Git GUI e acesse o menu: Remote → Push ... (Ctrl + P). Neste instante, todas as alterações realizadas no repositório local serão enviadas ao repositório no Github e estarão disponíveis para visualização pública.
2. O inverso, do repositório do Github para o local é realizado através do menu: Remote → Fetch from → (nome do repositório remoto).

## 4 Principais Comandos do Git

1. **clone**: Utilizado para a criação de uma cópia de trabalho/repositório. Conhecido como checkout nos VCS's centralizados;
2. **add**: Adiciona arquivos modificados/novos ao seu repositório. Será preciso realizar um commit nestes arquivos;
3. **commit**: Utilizado para informar sobre alterações ao repositório, criando uma nova revisão. Apenas os arquivos que foram “adicionados” serão submetidos ao repositório;
4. **update**: Atualiza sua área de trabalho com as alterações do repositório;
5. **pull**: “Puxa” todas as alterações realizadas no repositório master;
6. **push**: Informa ao repositório master sobre as alterações realizadas no repositório local.