Sumário

[Descrição do documento 2](#_Toc379450239)

[Instalando o GIT 3](#_Toc379450240)

[Configurando o GIT 4](#_Toc379450241)

[Listando as configurações 4](#_Toc379450242)

[Nome de usuário e email 4](#_Toc379450243)

[Ferramentas de diff e merge 4](#_Toc379450244)

[Editor de texto 4](#_Toc379450245)

[Cores do GIT 5](#_Toc379450246)

[Criando aliases 5](#_Toc379450247)

[Clonando um repositório 6](#_Toc379450248)

[Workflow Git 7](#_Toc379450249)

# Descrição do documento

Este documento tem por objetivo descrever o procedimento de instalação e configuração básica do Git em máquinas windows. Além disso, ele define o workflow e boas práticas no uso do Git em projetos desenvolvidos pela equipe do Núcleo de Sistemas da DPE.

# Instalando o GIT

Aqui será explicada apenas a instalação em Windows. O seguinte tutorial deve ser seguido, ele pertence à documentação oficial do git:

<http://git-scm.com/book/en/Getting-Started-Installing-Git#Installing-on-Windows>

Depois de instalado, conforme as instruções do tutorial, deve-se realizar as configurações globais básicas conforme os itens seguintes.

# Configurando o GIT

## Listando as configurações

O parâmetro '--global': as configurações se aplicam a todos os repositórios daquele usuário. Pode-se usar '--system' para configurações de todos os repositórios de todos os usuários do sistema, ou não passar qualquer parâmetro, assim o git config irá entender que a configuração se aplicar apenas ao repositório corrente (por isso deve estar dentro de um repositório ou executar este comando sem parâmetros).

|  |
| --- |
| git config --global --list |

## Nome de usuário e email

Com o prompt de comando aberto (é recomendado o uso do Git Bash ou Cygwin) digite os seguintes comandos:

|  |
| --- |
| git config --global user.name "Seu nome"  git config --global user.email seuemail@defensoria.rs.gov.br |

## Ferramentas de diff e merge

O Git para Windows vem com o editor de texto, a ferramenta de diff pré-configuradas. São elas vim e vimdiff.

Estas ferramentas podem ser mudadas, no caso do vimdiff, que é uma ferramenta de difícil manuseio, esta mudança é bastante recomendada.

Existem muitas outras opções de ferramentas e cada um pode configurar a que julgar melhor (tortoisemerge, p4pmerge, kdiff3, emerge, meld, openDiff, etc).

|  |
| --- |
| DiffMerge:  Para usuários do MsysGit, seguir este tutorial:  <http://sourcegear.com/diffmerge/webhelp/sec__git__windows__msysgit.html>  Para usuários do Cygwin, seguir este tutorial:  <http://sourcegear.com/diffmerge/webhelp/sec__git__windows__cygwin.html> |

## Editor de texto

Com o prompt de comando aberto (é recomendado o uso do Git Bash ou Cygwin) digite os seguintes comandos.

No lugar de <EDITOR\_NAME> coloque o nome/caminho da ferramenta de sua preferência.

|  |
| --- |
| git config --global core.editor <EDITOR\_NAME> |

## Cores do GIT

Isto irá deixar as saídas dos seus comandos Git com cores, o que ajuda na leitura.

|  |
| --- |
| git config --global color.ui true |

## Criando aliases

O GIT oferece algumas facilidades para o seu uso em linha de comando. Uma delas é a possibilidade de se criar alias para comandos mais usados ou muito complexos (que possuam vários parâmetros).

Neste link podes-se encontrar uma explicação mais completa sobre aliases (inclusive a nível de shell, caso a sua shell suporte):

* <http://githowto.com/aliases>

Abaixo segue uma lista dos mais usados:

|  |
| --- |
| git config --global alias.co checkout  git config --global alias.ci commit  git config --global alias.st status  git config --global alias.br branch  git config --global alias.hist 'log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short' |

# Clonando um repositório

O processo de clonar um repositório nada mais é que criar uma cópia de algum repositório remoto ou até mesmo local.

Para mais informações sobre este comando:

* <http://git-scm.com/docs/git-clone>

Exemplo:

|  |
| --- |
| O comando abaixo irá clonar o repositório remoto "linux.git" na pasta "my-linux".  A URL irá definir o protocolo de transporte (ssh, git, http, https, ftp, ftps e rsync) o endereço do servidor e o caminho até o repositório (caminho até o arquivo ".git").  git clone git://git.kernel.org/pub/scm/linux.git my-linux |

# Workflow Git

|  |
| --- |
| 1. Criar um branch local, apartir do master, e fazer a implementação desejada:  # branch atual: master  git checkout -b meu\_branch  2. Após terminar a implementação, amassar todos os commits parciais em um commit que será integrado.  # branch atual: meu\_branch  git rebase –i  3. Atualizar o branch origin/master com o código mais atual do remoto:  # branch atual: meu\_branch  git fetch  4. Atualizar o código do seu branch de desenvolvimento com o código trazido do remoto:  # branch atual: meu\_branch  git rebase origin/master  5. Neste momento poderão ocorrer conflitos, caso eles ocorram, resolver os conflitos com a ferramenta configurada.  # branch atual: meu\_branch  git mergetool  6. Após resolvidos todos os conflitos, finalizar o rebase:  # branch atual: meu\_branch   git rebase --continue  5. fazer o push para o branch master do origin:  # branch atual: meu\_branch   git push origin meu\_branch:máster |