

Comandos básicos de GIT

OBJETIVO

Este tutorial tem como objetivo listar alguns comandos do software de controle de versões GIT.

DEFINIÇÕES

GIT - O Git é um projeto de código aberto maduro e com manutenção ativa desenvolvido em 2005 por Linus Torvalds, o famoso criador do kernel do sistema operacional Linux. Um número impressionante de projetos de software depende do Git para controle de versão, incluindo projetos comerciais e de código-fonte aberto.

CONFIGURAÇÕES

Configurar usuário global: `git config --global "Nome do usuário"`

Configurar e-mail global: `git config --global user.email "seuemail@provedor.com.br"`

Retornar o status de configuração específica: `git config user.name` //Deverá ser retornando o nome de usuário global configurado no GIT

Retornar o status de todas as configurações: `git config --list` //Deverá ser retornado todas as configurações

COMANDOS PARA O REPOSITÓRIO

Inicializar repositório: `git init`

Verificar status dos arquivos: `git status`

Adicionar arquivo para ser rastreado: `git add arquivo`

Adicionar todos os arquivos: `git add .`

Adicionar um novo commit: `git commit -m "informações sobre o commit"`

Adicionar um novo commit já adicionando os arquivos alterados: `git commit -am "informações sobre o commit"`

Listar commits realizados: `git log`

Listar commits com suas ramificações ou branches: `git log --graph`

Verificar alterações específicas de um commit: git show comitespecifico

Verificar alterações nos arquivos antes de fazer um novo commit: git diff

Verificar quais arquivos foram alterados antes de fazer um novo commit: git diff --name-only

Retornar um arquivo para o ultimo commit realizado antes de adicionar para staged com o comando git add: git checkout arquivo

Retornar um arquivo para o ultimo commit realizado após adicionar para staged com o comando git add: git reset HEAD arquivo, em seguida git checkout arquivo

Remover um commit: git reset --soft #hash anterior#, mantem os arquivos staged. git reset --mixed #hash anterior#, não mantem os arquivos staged é preciso adicionar novamente com git add. git reset --hard #hash anterior#, apaga qualquer alteração feita após o commit. **ESTES COMANDO DEVEM SER USADOS COM MUITO CUDADO, POIS SE O PROJETO JÁ ESTIVER EM UM REPOSITARIO EXTERNO HAVERÃO MUITAS DIFICULDADES PARA A SINCRONIZAÇÃO.**

Clonar de um repositório: git clone endereço-do-repositório

Criar uma nova branch: git checkout -b nome-da-branch

Verificar qual a branch atual: git brach

RETORNO

- branch atual

branch

Navegar entre as branches: git checkout branch-desejada

Apagar uma branch: git branch -D branch

Apagar uma branch em repositório remoto: git push origin: branch

Criar uma tag: git tag -a 1.0.0 -m "Informações da tag"

Subir tags: git push origin master --tags

Reverter mudanças para aproveitar posteriormente: git revert #hash#

Apagar tags: git tag -d 1.0.0

Apagar tag em repositório remoto: git push origin: 1.0.0

Revision #3

Created 3 May 2024 00:38:42 by Jose Mauro Mendes Junior

Updated 4 May 2024 18:54:41 by Jose Mauro Mendes Junior