**Manual de Git: Comandos Essenciais e Uso Prático**

Git é um sistema de controle de versão distribuído amplamente utilizado no desenvolvimento de software. Este manual fornecerá uma visão geral dos comandos mais importantes e suas funções, além de algumas boas práticas para o uso do Git.

**1. Inicializando um Repositório Local**

**git init**  
Inicializa um novo repositório Git na pasta atual. Cria a estrutura básica necessária para o versionamento.

**2. Criando e Trabalhando com Branches**

**git branch -M main**  
Cria uma nova branch chamada "main" e define-a como a branch principal do repositório. Se você já estiver em outra branch, este comando também muda para a branch "main".

**git checkout -b nome-da-branch**  
Cria uma nova branch e automaticamente muda para ela.

**git branch**  
Lista todas as branches existentes no repositório.

**git checkout nome-da-branch**  
Muda para uma branch específica que já foi criada anteriormente.

**3. Conectando a um Repositório Remoto**

**git remote add origin <url>**  
Conecta o repositório local a um repositório remoto (como GitHub, GitLab, etc.). <url> deve ser substituído pela URL do repositório remoto.

**git remote -v**  
Exibe as URLs associadas ao repositório remoto.

**4. Sincronizando Repositórios Local e Remoto**

**git pull origin main**  
Atualiza o repositório local com as mudanças do repositório remoto. O parâmetro main pode ser substituído pelo nome de qualquer outra branch.

**git fetch**  
Baixa as atualizações do repositório remoto, mas não aplica as mudanças automaticamente. Você pode visualizar as atualizações antes de mesclar.

**5. Adicionando Arquivos ao Controle de Versão**

**git add .**  
Adiciona todos os arquivos novos e modificados para serem monitorados pelo Git.

**git add <arquivo>**  
Adiciona um arquivo específico ao controle de versão.

**6. Commitando as Mudanças**

**git commit -m "comentário"**  
Cria um commit com uma mensagem descritiva sobre as alterações feitas. Um commit é um "snapshot" do estado atual do repositório.

**git commit --amend**  
Modifica o último commit (útil para alterar a mensagem ou adicionar novos arquivos que foram esquecidos).  
git commit --amend -m "Mensagem corrigida do commit anterior"

**7. Mesclando Branches**

**git merge nome-da-branch**  
Mescla as alterações de uma branch específica (ex.: teste) na branch atual. Geralmente, você faz merge em main ou develop após o desenvolvimento.  
git merge teste

**8. Enviando as Alterações para o Repositório Remoto**

**git push origin main**  
Envia as alterações da branch atual para o repositório remoto. O main pode ser substituído por qualquer outra branch.  
git push origin main

**git push -u origin main**  
Define a branch main como a padrão para futuros pushes, eliminando a necessidade de especificar origin main em cada push.  
git push -u origin main

**9. Revertendo e Desfazendo Alterações**

**git reset --soft HEAD~1**  
Reverte o último commit, mas mantém as alterações no diretório de trabalho para que possam ser editadas e re-commitadas.  
git reset --soft HEAD~1

**git reset --hard HEAD~1**  
Reverte o último commit e descarta todas as alterações feitas no diretório de trabalho.  
git reset --hard HEAD~1

**git revert <commit>**  
Cria um novo commit que desfaz as alterações feitas no commit especificado, preservando o histórico.  
git revert 1234abcd

**10. Histórico e Logs**

**git log**  
Mostra o histórico de commits do repositório.  
git log

**git log --oneline**  
Exibe o histórico de commits de forma compacta, mostrando apenas o hash e a mensagem de cada commit.  
git log --oneline

**11. Verificando o Status**

**git status**  
Exibe o status do repositório, mostrando arquivos modificados, não rastreados, e prontos para commit.

**12. Visualizando Diferenças**

**git diff**  
Mostra as diferenças entre os arquivos no diretório de trabalho e os arquivos rastreados.

**git diff --staged**  
Exibe as diferenças entre os arquivos adicionados (stage) e o último commit.

**13. Clonando Repositórios**

**git clone <url>**  
Faz o download de uma cópia de um repositório remoto para o seu computador local.  
git clone https://github.com/usuario/repo.git

**14. Removendo Arquivos do Controle de Versão**

**git rm <arquivo>**  
Remove um arquivo do repositório e do diretório de trabalho.  
git rm arquivo.txt

**git rm --cached <arquivo>**  
Remove um arquivo do controle de versão, mas o mantém no diretório de trabalho.  
git rm --cached arquivo.txt