Aqui estão alguns dos comandos Git mais comuns e úteis que você pode usar:

1. **git init**: Inicializa um novo repositório Git.

2. **git clone [URL]:** Clona um repositório remoto para a sua máquina local.

3. **git status**: Mostra o estado atual do repositório, como arquivos modificados ou não rastreados.

4. **git add [arquivo] ou git add .:** Adiciona arquivos ao índice para serem incluídos no próximo commit.

5. **git commit -m "mensagem":** Cria um commit com uma mensagem descritiva.

6. **git push**: Envia as alterações feitas para um repositório remoto.

7**. git pull**: Baixa as alterações mais recentes de um repositório remoto e as mescla com a sua branch local.

8. **git log**: Exibe o histórico de commits.

9. **git branch**: Lista, cria ou exclui branches.

10. **git checkout [branch]**: Alterna entre branches ou restaura arquivos.

11. **git merge [branch]:** Mescla uma branch específica na branch atual.

12**. git diff**: Mostra as diferenças entre os arquivos modificados.

1. **git stash**: Salva temporariamente alterações para limpar seu espaço de trabalho, permitindo que você volte mais tarde.

2. **git stash pop**: Restaura as alterações armazenadas pelo `git stash`.

3. **git reset [arquivo]** ou **git reset --hard**: Remove arquivos do índice ou desfaz alterações no histórico de commits.

4. **git rm [arquivo]**: Remove arquivos do repositório e do sistema de arquivos.

5. **git fetch**: Atualiza as informações sobre o repositório remoto sem mesclar automaticamente as alterações.

6. **git rebase [branch]**: Reorganiza os commits de uma branch para criar um histórico mais linear.

7. **git tag [nome]**: Marca um ponto específico no histórico com um nome, geralmente usado para versões.

8. **git cherry-pick [commit ID]**: Aplica um commit específico em outra branch.

9. **git mv [arquivo antigo] [arquivo novo]**: Renomeia ou move arquivos no repositório.

10. **git bisect**: Ajuda a encontrar um commit com erro entre dois pontos no histórico.

Certamente! Aqui estão mais comandos úteis do Git para você explorar:

1. **git show [commit ID]**: Exibe informações detalhadas de um commit específico.

2. **git archive**: Cria um arquivo compactado (ZIP, TAR) com o conteúdo de um commit.

3. **git blame [arquivo]**: Mostra o autor de cada linha de um arquivo, útil para rastrear mudanças.

4. **git reflog**: Mostra o histórico de referências de HEAD, útil para recuperar commits perdidos.

5. **git config**: Configura opções globais ou específicas do Git, como nome e e-mail do usuário.

6. **git clean -f**: Remove arquivos não rastreados do diretório de trabalho.

7. **git revert [commit ID]**: Reverte as mudanças de um commit específico, criando um novo commit.

8. **git remote -v**: Lista os repositórios remotos e seus URLs.

9. **git grep [texto]**: Procura por texto nos arquivos versionados.

10. **git submodule**: Gerencia submódulos dentro de um repositório Git.

A lista completa de comandos Git é bastante extensa, mas aqui estão algumas categorias principais com exemplos de comandos:

Configuração

- **git config**: Configura opções globais ou específicas do repositório, como nome e e-mail do usuário.

Inicialização e Clonagem

- **git init**: Inicializa um novo repositório.

- **git clone [URL]**: Clona um repositório remoto.

Controle de Versão

- **git add**: Adiciona arquivos ao índice.

- **git commit**: Salva as alterações no histórico do repositório.

- **git status**: Mostra o estado atual do repositório.

Branches e Mesclagem

- **git branch**: Gerencia branches.

- **git checkout**: Alterna entre branches.

- **git merge**: Mescla branches.

Sincronização com Repositórios Remotos

- **git push**: Envia alterações para o repositório remoto.

- **git pull**: Baixa e mescla alterações do repositório remoto.

- **git fetch**: Atualiza informações do repositório remoto sem mesclar.

Histórico e Logs

- **git log**: Exibe o histórico de commits.

- **git show**: Mostra detalhes de um commit específico.

Reversão e Recuperação

- **git reset**: Desfaz alterações no índice ou no histórico.

- **git revert**: Reverte um commit específico.

- **git stash**: Salva temporariamente alterações não confirmadas.

Outros Comandos Úteis

- **git tag**: Marca pontos específicos no histórico.

- **git diff**: Mostra diferenças entre arquivos.

- **git blame**: Mostra o autor de cada linha de um arquivo.