**LISTA DE COMANDOS GIT**

1. **git init:** Cria um novo repositório Git vazio na pasta atual.
   1. **Exemplo:** Para inicializar um novo repositório Git em um diretório chamado "meu\_projeto", você pode executar o seguinte comando no terminal: **git init**
2. **git config --global :** Define configuração para todos os repositórios Git.
   1. **Exemplo 1:** Para definir o nome do usuário como "João Silva", você pode executar o seguinte comando no terminal: **git config --global user.name** "João Silva"
   2. **Exemplo 2**: Para definir o email do usuário como "joao.silva@example.com", você pode executar o seguinte comando no terminal: **git config --global user.email** "[joao.silva@example.com](mailto:joao.silva@example.com)"
3. **git config --list:** Mostra todas as configurações do Git em todos os repositórios.
   1. **Exemplo:** Para mostrar todas as configurações do Git em todos os repositórios, você pode executar o seguinte comando no terminal: **git config --list**
4. **git add:** Adiciona arquivos ao índice Git (também conhecido como "área de preparação").
   1. **Exemplo:** Para adicionar um arquivo chamado "index.html" ao índice Git, você pode executar o seguinte comando no terminal:
   2. **git add index.html** (adiciona um arquivo por vez)ou
   3. **git add .** (adiciona todos os arquivos de uma vez)
5. **git commit:** Registra as alterações feitas no repositório Git.
   1. **Exemplo:** Para registrar as alterações feitas no arquivo "index.html" com uma mensagem de confirmação, você pode executar o seguinte comando no terminal: **git commit -m** "Adicionado conteúdo ao arquivo index.html"
6. **git status:** Mostra o estado atual do repositório Git, incluindo quais arquivos foram modificados, adicionados ou excluídos.
   1. **Exemplo:** Para verificar o estado atual do repositório Git, você pode executar o seguinte comando no terminal: **git status**
7. **git log:** Mostra o histórico de commits do branch atual.
   1. **Exemplo:** Para mostrar o histórico de commits do branch atual, você pode executar o seguinte comando no terminal: **git log**
8. **git log --oneline:** Mostra o histórico de commits resumido em uma linha.
   1. **Exemplo:** Para mostrar o histórico de commits resumido em uma linha, você pode executar o seguinte comando no terminal: **git log** --oneline
9. **git branch:** Cria, lista ou exclui branches no repositório Git.
   1. **Exemplo:** Para criar um novo branch chamado "novo-recurso", você pode executar o seguinte comando no terminal: **git branch** novo-recurso
10. **git checkout (criar branch):** Altera o branch atual para outro existente.
    1. **Exemplo:** Para mudar para o branch "novo-recurso", você pode executar o seguinte comando no terminal: **git checkout** novo-recurso
11. **git checkout (alternar entre commits):** Altera o HEAD para um commit específico.
    1. **Exemplo:** Para alterar o HEAD para um commit específico usando seu hash, você pode executar o seguinte comando no terminal: **git checkout** <hash\_do\_commit>
12. **git push:** Envia as alterações locais para um repositório remoto.
    1. **Exemplo:** Para enviar as alterações do branch atual para o branch remoto "master", você pode executar o seguinte comando no terminal: **git push** origin master
13. **git remote add:** Adiciona um novo repositório remoto ao seu repositório local
    1. **Exemplo:** Para adicionar um novo repositório remoto chamado "**origin**" com a URL "**https://github.com/usuario/meu\_projeto.git**", você pode executar o seguinte comando no terminal:
    2. **git remote add** origin [https://github.com/usuario/meu\_projeto.g](https://github.com/usuario/meu_projeto.git)it
14. **git remote -v:** Mostra uma lista de todos os repositórios remotos vinculados ao seu repositório local, juntamente com suas URLs.
    1. **Exemplo:** Para verificar a lista de todos os repositórios remotos vinculados ao seu repositório local, você pode executar o seguinte comando no terminal:
    2. **git remote** -v