**Iniciar projeto**

**Configurações iniciais:**

1. Remover as credenciais
2. Adicionar o nome de usuário: git config --global user.name “<nome do usuário>”
3. Adicionar o nome de usuário: git config --global user.email “<e-mail do usuário>”
4. Verificar se não existem erros no cadastro: git config --global --list

► Caso o usuário ou e-mail sejam inseridos erroneamente, basta executarmos novamente os comandos acima que os dados serão subscriscritos.

**Criar um projeto remoto & local:**

1. Criar um novo repositório remoto no GitHub
2. Criar um repositório um diretório local.
3. Adicionar git no repositório local: git init

► Caso o usuário deseje remover o controle de versão com git no projeto, basta excluir o arquivo oculto que estará no diretório (pasta) do projeto, chamado .git

**Puxar projeto remoto:**

1. Aponta para o projeto do GitHub: git remote add origin “<URL GitHub>”
2. Verificar se foi cadastrado a URL: git remote -v
3. Puxar o projeto do GitHub para o diretório local: git pull origin main
4. Mudar branch: git checkout -b main
5. Se quisermos verificar a branch atual: git branch

► A partir daqui tudo que for alterado no diretório local pode ser enviado para o remoto (GitHub)

► Caso seja necessário alterar o endereço HTTP do projeto remoto, basta executar o comando git remote remove origin

**Criar um projeto remoto & local:**

1. Adicione algo no repositório local.
2. Verificar alterações realizadas: git status
3. Envia para a área de stage: git add **.**
4. Verificar alterações realizadas: git status
5. Salva o registro de commit: git commit -m “<mensagem do commit>”

► Add com commit ao mesmo tempo: git commit -a -m “<Mensagem>” ou git commit -am “<Mensagem>”

► Commit com duas mensagens: git commit -m “<Mensagem>” -m “<Mensagem>”

► Alterar mensagem do commit: git commit --amend “<Nova Mensagem>” caso já esteja no repositório remoto será necessário executar git push --force

**Enviar para o repositório remoto:**

1. Enviar todos os commits: git push -u origin main

**Resumo da página acima:**

1. Ter um repositório do projeto no GitHub (Nuvem)

2. Remover credenciais (PC)

3. Adicionar nome e e-mail de usuário com: (Terminal)

git config --global user.name “<nome de usuário>”

git config --global user.email “<e-mail de usuário>”

4. Crie um diretório local, caminhe até ele e dê o git init (Terminal)

5. Adicione a URL: git remote add origin “<URL>” (Terminal)

6. Depois: (Terminal)

git pull origin main

git checkout -b main ou git banch -M main

\* Se o repositório do GitHub estiver sem nenhum arquivo o pull não precisará ser realizado.

7. Depois das alterações no diretório:

git add **.**

git commit -m “<mensagem>”

git push -u origin main

**Stage Area**

**Enviar arquivos ou diretórios para a Stage Area**

Comando: git add <diretório ou arquivo alvo>

Exemplos:

Envio variado: git add arquivo\_1.txt arquivo\_2.txt diretorio\_1

Envio de todos I: git add \* (o asteristico significa todos)

Envio de todos III: git add --a (o “a” significa “all” ou todos)

Envio de todos III: git add -A (o “A” significa “All” ou todos)

Envio de todos do diretório atual: git add **.** (o ponto final significa aqui ou diretório atual)

**Remover arquivos ou diretórios da Stage Area**

Comando: git restore --staged <diretório ou arquivo alvo>

Exemplos:

Remoção variada: git restore --staged arquivo\_1.txt arquivo\_2.txt diretorio\_1

Remoção de todos: git restore --staged \* (o asteristico significa todos)

Remoção do um diretório atual: git restore --staged **.** (o ponto final significa aqui ou diretório atual)

**Commits**

**Criar commits:**

Comando: git commit -m “<Mensagem>”

Variações:

Commit com uma mensagem: git commit -m “<Mensagem>” -m “<Mensagem>”

Commit e add I: git commit -a -m “<Mensagem>”

Commit e add II: git commit -am “<Mensagem>”

**Mudar mensagem dos commits:**

Comando: git commit –amend ”<Nova Mensagem>”

**Reverter commits mantendo os históricos:**

Comando: git revert <HASH ou HEAD>

Variações:

Reverter o último commit sem abrir o vim: git revert --no-edit <hash>

Reverter o último commit abrindo o vim: git revert <hash>

Reverter o último commit: git revert head (se quisermos voltar o revert basta fazer novamente o git revert head)

Reverter commit específico: git revert <hash> (teremos conflito caso o commit revertido tenha sido alterado posteriormente, evitar seu uso)

Reverter vários commits: git revert <hash 1> < hash 2> < hash 3> (teremos conflito caso o commit revertido tenha sido alterado posteriormente, evitar seu uso)

Reverter um intervalo de commits: git revert head~3**..**head (teremos conflito caso o commit revertido tenha sido alterado posteriormente, evitar seu uso)

**Reverter commits apagando os históricos:**

Comando: git reset <HASH ou HEAD>”

Variações:

Reverter apagando o commit e mantendo as alterações: git reset --soft head~1

Reverter apagando o commit e mantendo as alterações: git reset --hard head~1

Reverter apagando o commit e mantendo as alterações: git reset --hard head~3

Reverter apagando o commit e mantendo as alterações: git reset --hard a154s25

► Se o commit já estiver na nuvem devemos digitar git push origin master --force ou git push origin main --force

**Reverter commits II:**

Comando: git checkout <HASH ou HEAD>”

Variações:

Reverter apagando o commit e mantendo as alterações: git reset --soft head~1

git restore --source HEAD --staged --worktree nome-do-arquivo

**Consulta**

**Comandos de Consulta:**

Comando: git commit -m “<Mensagem>”

Variações:

Consultar estado dos diretórios ou arquivos: git status

Consultar estados detalhados dos commits: git log

Consultar estados resumidos dos commits: git log –oneline

Consultar estados de commits com heads: git reflog

**Remoto**

**Inserir projeto remoto no computador:**

Comando: git remote add origin <URL>

Variações:

Verificar: git remote -v

Remover: git remote remove origin

**Puxar projeto remoto para o local**

**Puxar e mudar branch**

Comando 1: git pull origin main

Comando 2: git checkout -b main

**Branch**

**Ver branches**

Verificar branch atual e branches existentes: git branch

Variações:

Criar branch: git branch <nome da nova branch>

Mudar de branch: git checkout -b <nome da branch>

**Enviar para a nuvem**

**Comando de envio**

Enviar todos os commits e arquivos: git push -u origin main

Enviar todos os commits e arquivos: git push -u origin main –force

Enviar todos os commits e arquivos: git push -u origin main -f

**Secure SHell (SSH)**

**Envio sem credenciais**

Verificar chaves SSH: ls -al ~/.ssh

Criar chave SSH: ssh-keygen -t ed25519 -C "<seu-email@example.com>"

Adicionar SSH ao agente 1/2: eval "$(ssh-agent -s)"

Adicionar SSH ao agente 2/2: ssh-add ~/.ssh/id\_ed25519

Copiar a chave pública que aparecer: cat ~/.ssh/id\_ed25519.pub

Adicione em: GitHub > Configurações > SSH and GPG Keys > New SSH key

Testa a conexão: ssh -T git@github.com

Testa a conexão git config --global user.name "<nome>" e git config --global user.email "<e-mail>"

Adicionar o remote: git remote add origin <SSH>