Configurar e-mail e nome

git config --global user.name "João Silva"

git config --global user.email [joao@treinaweb.com.br](mailto:joao@treinaweb.com.br)

listar configurações existentes

git config --list

git config –l

listar somente as configurações globais

git config --global -l

criando a chave ssh

ssh-keygen -t rsa -b 4096 -C joao@treinaweb.com.br ssh-add ~/.ssh/id\_rsa

cria o repositório no diretório local

git init

cria um repositório bare

git init –bare

sincronizando com o git hub

git remote add origin git@github.com:treinaweb-git/primeiro-repo.git

git push -u origin máster

clonando repositório remoto

git clone [git@github.com:treinaweb-git/primeiro-repo.git](mailto:git@github.com:treinaweb-git/primeiro-repo.git)

verifica o estado do repositório e a área de staging

git status

status resumido

git status –s

git status –short

adiciona arquivos no staging

git add file.txt

adiciona arquivos modificados ao staging de uma vez

git add .

git add –A

cria um commit diretamente com todos os arquivos modificados

git commit –a

incluir a mensagem de commit inline

git commit -m "First Commit"

adiciona alterações no commit anterior (use com cuidado)

git commit –amend

envia as alterações para o repositório, onde origin é o remote e máster o branch

git push origin máster

baixa commit do repositório remoto

git pull origin

atualiza as referencias com um repositório remoto

git fetch origin

ignorando arquivos

.gitignore

* Ignora todos os arquivos em html
  + \*.html
* Exceto do arquivo index.html
  + \*.html
* Ignora o README no diretório atual, mas permite subdir/README
  + /README
* Ignora todos os arquivos no diretório de build/
  + Build/
* Ignora os arquivos .txt do diretório doc/, mas permite subdiretórios (doc/server/arch.txt)
  + doc/\*.txt
* ignora todos os arquivos .pdf na arvore doc/
  + doc/\*\*/\*.pdf

cria uma nova branch

git checkout -b nova-branch

alterar entre as branches

git checkout máster

voltar para branch anterior

git checkout –

remover uma branch local

git branch -d branch

remover a branch remota

git push --delete origin branch

utilizando merge

exemplo: entra na branch develop

git checkout develop

combina a branch feature/a com develop

git merge feature/a

alterando cores no log

git log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit

ou essa opção

git config --global alias.lg "log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit"

criar uma tag

git tag v1.0.0

publicar tags no remoto

git push origin –tags

\*\*estudar mais sobre tags no git

Margeando branches

git merge develop

escolhendo nosso lado do conflito de alterações

git checkout --our file.txt

escolhendo as alterações remoto

git checkout --theirs file.txt

abortando o merge

git merge –abort

git reset -> não se deve utilizar em branches publicas

três modos: --mixed | --soft | --hard

mixed -> modo padrão altera as áreas staging e repositório, porem o working fica intacto

soft -> mantem o staging sem alteração

hard -> altera tudo

git revert -> para ser utilizado em branches publicas