Comandos	Descrição
git --version	mostra a versão do git
git --v	mostra a versão do git
git config --global user.name "Dário da Silva Melo"	seta o usuario no git
git config --global user.e-mail "dario.melo@ufu.br"	seta o email no git
git config --global user.name	mostra o usuario do git
git config --global user.e-mail	mostra o e-mail que está setado
git config --global core.editor "code --wait"	configura o git para usar o editor de texto VsCode
git config --global init.defaultBranch main	define que ao criar uma Branch ela iniciara como main
git config --list	mostra a configuração do git
git status	lista os arquivos não rastriados (untracked)
git add .	adiciona todos os arquivos no rastreamento
git add -A	adiciona todos os arquivos no rastreamento
git add --all	adiciona todos os arquivos no rastreamento
git add (nome_arquivo.extenção)	adicona um arquivo especifico no rastreamento
git commit -m "Primeira Vesão"	salva as alterações no repostioria e cria a primeira vesão
git commit -a -m "Primeira Vesão"	salva as alterações no repostioria com a mesnsagem sem precisar de digitar "git add ."
git commit --amend -m "mensagem alterada"	faz o comite novamente (acrescenta um arquivo esquecio no mesmo commit e alterara a mensagem
git commit --amend --no-edit	faz o comite novamente (acrescenta um arquivo esquecio no mesmo commit e não altera a mensagem
git diff	mostra a diferança da versão Modified e Unmodified
git diff --cached	mostra a diferença da versão Staged e Unmodified
git log	mostra a lista dos commit's realizados com informações
git log --oneline	mostra de forma resumida a lista de commit's
git log -1	lista o ultimo commit
git log -2	lista o penultimo commit e assim vai git log -3...-n
git log --oneline -3	lista os três ultimo commit de forma resumida
git log --patch	mostra um historio de todas as modificações ocorridas
git log -p	mostra um historio de todas as modificações ocorridas
git log --stat	mostra um historio dos arquivos modificados e commitados
git log --shortstat	mostra quantos aquivos forão alterados e que linha
code .	abri o VsCode no diretorio corrente
git checkout numero_do_comiite	mostra a versão dos arquivos referente ao commit que deseja
git checkout nome_do_arquivo.extenção	todas as alterações feitas no arquivo expecifico são perdidas e volta as informações do ultimo commit (não podem estar na area de preparação)
git checkout .	todas as alterações feitas nos arquivos são perdidas e volta as informações do ultimo commit (não podem estar na area de preparação - Staged)
git restore --staged nome_do_arquivo.extenção	tira o arqquivo da área de preparação (Staged - arquivo rastreado). Se já existe um commit
git restore --staged .	tira todos arqquivo da área de preparação (Staged - arquivo rastreado). Se já existe um commit
git rm --cached (nome_aquivo.extenção)	tira o arqquivo da área de preparação (Staged - arquivo rastreado). Se não existir um Commit

Comandos	Descrição
git rm --cached -r .	tira todos os arquivos da area de preparação (Staged- arquivo rastreado). Se não exiistir um Commit
git reset --hard	perde tudo que estiver na area de modificação (modified) e preparação (Staged). Volta para o Ultimo Commit
git clean -f	apaga todos os arquivos não rastreados. no estado: Unmodified
git rm -fr ./git	apaga o repositorio (não faça isso - não é uma opção valida)
touch .gitignore	cria o orquivo .gitignore (para não versionar arquivos). Exemplo *.exe, conf.doc, etc. A # é o comentário
git update-index --skip-worktree main.txt	faz com que o arquivo não seja mais ratreado
git update-index --no-skip-worktree main.txt	faz com que o arquivo volte a ser rastreado
git clone pasta_origem pasta_clone	faz o clone da pasta de origem
git clone https://....	cria um clone do repositorio no github
git remote -v	mostra o servidor remoto associado para burcar e enviar informações
git remote add <alias> https://....	faz a ligação entre o repositorio e o servidor remoto. Um Repositorio pode apontar para varios servidores
git remote remove <alias>	remove o alias (caminho do servidor)
git remote set-url <alias> https://...	atualiza o caminho do servidor remoto para aquele alias.
git push	envia as informações do repositorio para o servidor remoto
git pull	traz a ultima versão no servidor remoto
ssh-keygen cd ~/.ssh/ start . muda o nome dos arquivos eval $(ssh-agent) ssh-add ~/.ssh/nome_do_arquivo (não tem extenção)	
git branch	lista as branch
git branch --list	lista as branch
git branch nome_da_branch	cria uma nova branch
git checkout nome_da_branch	entra na branch desejada
git checkout -b nome_da_branch	cria uma nova branch e entra nela
git switch nome_da_branch	entra na branch desejada
git switch -	volta para ultima branch que eu tinha selecionado
git switch -c	cria uma nova branch e entra nela
git checkout -f nome_da_branch	entra na branch desejada apagando todas as modificações feitas nos arquivos rastrados na branch que eu estava
git push --set-upstream <alias_remoto> <nome_branch>	seta a nova branch para o servidor remoto
git push -u <alias_remoto> <nome_branch>	seta a nova branch para o servidor remoto
git branch -d <nome_branch>	apaga a branch
git branch -D <nome_branch>	força a delecção da branch (caso o git pergute se deseja realmente apagar)
git push --delete <alias_remoto> <nome_branch>	apaga a branch remota
git branch -m <novo_nome_branch>	renomeando a branch

Comandos	Descrição
git branch -m <nome_branch> <novo_nome_branch>	renomeando a branch (caso não esteja nela)
git log <nome_branch> --oneline -5	Variações do log: pode listar os commites e outras funcionalidades do log aprenas passando o nome da branch como parametro.
git merge <nome_branch_que_vai_ser_margiada>	vair margiar (mesclar) a branch desejada para a branch que está logado.
git branch --no-merged	lista as branch que não foram margiadas.
git branch --merged	lista as branch que foram margiadas.
git merge --abort	no caso de conflito este comoando serve para cancelar o margiamento.
git reset --hard	no caso de conflito este comoando serve para cancelar o margiamento.
git tag <nome_da_tag>	cria uma tag simples (lightweight)
git tag -a -m "mensage" <nome_da_tag>	cria uma tag complexa (annotated)
git tag <nome_da_tag> <numero_do_comite>	cria uma tag simples no commit desejado
git tag -a -m "mensage" <nome_da_tag> <numero_do_comit	cria uma tag complexa no commit desejado
git tag	lista tags
git tag -l	lista tags
git tag --list	lista tags
git tag -n	lista a tags com as mensagens
git push <alias_remoto> <nome_da_tag>	envia a tag para o servidor
git push --tags	envia todas as tags para servidor
git checkout <nome_da_tag>	vai para o commit que a tag esta ligada (a tag é um ponteiro que aponta para o commit)
git diff <nome_da_tag_01> <nome_da_tag_02>	faz a compração dos commits lidado as tags (podemos fazer comparação de versões)
git tag -d <nome_da_tag>	apago a tag localmente
git push --delete <alias_remoto> <nome_tag>	apaga a tag no servidor
git stash	salva as alterações em arquivos RASTREADOS na memoria, caso precise sair da branch e não queira dar o commit
git stash list	lista as stash que estão salvas na memoria
git stash apply	aplica as mudanças nos arquivos ratreados (na branch que estiver ativa)
git stash apply stash@{0.....n}	aplica a mudança desejada (salfa em forma de pilha. O valor zero é a ultima mudança)
git stash drop stash@{0.....n}	apaga a stash na memoria desejada (caso não informe a stash apaga com indice 0) - USAR ESTA
git stash pop stash@{0.....n}	apaga a stash na memoria desejada (caso não informe a stash apaga com indice 0)
git stash branch stash@{0.....n}	cria uma branch com as alterações salve em memoria. (caso não informe a stash desejada o git pega a com indice 0)
git revert HEAD	cria um novo commit desfanzendo todas as alterações feitas no commit referente ao ponteiro HEAD
git revert HEAD --no-edit	reverte o que já tinha sido revertido (cria um novo commit)
git reset --hard HEAD~1	apaga o ultimo commit
git reset --hard HEAD~5	apaga os ultimos 5 commits (git reset --hard HEAD~1.....n)
git reset --mixed HEAD~1	apaga o ultimo commit e não apaga as alterações feitas (fica na area modified)
git reset --soft HEAD~1	apaga o ultimo commit e não apaga as alterações feitas (fica na area starged)
git push --force	atualiza o servidor sobescrevendo o que exite no remoto (o historico do repositorio remoto será perdio e substituido pelo local)
git push --force-with-lease	atualiza o servidor, vair subescrever a linha do tempo, mas desde que nenhuma alteração ser perdida durante o processo.

Comandos	Descrição
git rebase main	mescla a branch com main (no caso da main ter sido alterada e ter commits que a branch não tenha)
git rebase --interactive HEAD~2	Unir os dois ultimos commits em um (o editor vair abrir a palavra squash na frente dos commits que vão ser fundidos menos no 1°)
git rebase --interactive HEAD~3	Unir os três ultimos commits em um
git rebase --abort	cancelar o rebase devido aos conflitos
git rebase --continue	após resolvido os conflitos emitir este comando para executar o rebase
git pull --rebase	mescla o servidor remoto com main (no caso o servidor remoto ter sido alterada e ter commits que a branch não tenha)
git fetch <alias_remoto> <branch_do_servidor>	traz uma branch especifica para o servidor especifica do servidor.
git branch -a	lista todas as branch inclusive as que foi baixada e que ainda não estaligada a sua main (não foi feito chechout na branch)
git cherry-pick <n°_do_commit_branch_desejada>	faz um merge de um commite anterior da branch desejada (você não quer mesclar com o commit autal da branch)
git bisect start git bisect good <n°_do_commit> git bisect bad <n°_do_commit> git bisect reset	Faz uma busca binaria rasreando a modificação não desejada
git fetch git branch -a git log <alias_remoto>/main --oneline git diff <alias_remoto>/main git checkout <alias_remoto>/main git switch - git merge	git pull = git fetch + fet merge (faço o git fetch para saber o que tem de diferente antes de mesclar) mostras as branch que eu busquei no servidor (ainda não estão ligadas ao meu repositorio local) mostra as diferenças
git config --global alias.<nome_do_alias> <comando>	Exemplo: git config --global alias.s status (cria um apelido para o comando)
git config --global alias.<nome_do_alias> '<comando>'	colocar aspas simple se o comando tiver paramentros Ex.: git config --global alias.line 'log --oneline'
git config --global unset alias.<nome_do_alias>	apaga o apelido
git branch | grep "entre com nome parcial da branch"	Exemplo: git branch | grep ac -> vai lista todas as branch que tem 'ac' em seu nome
git log --oneline | grep "entre com nome parcial da msn	Exemplo: git log --oneline | grep ac -> vai lista todos os commits de tem uma mensagem tem 'ac' em sua composicao
sourcetree	ferramenta grafica para trabalhar com o git (windows e mac)
gitkraken	ferramenta grafica para trabalhar com o git (windows, mac e linux)
git GUI	gerramenta grafica que veem com o git