Comandos Git:

git config --global user.name “Nome da Pessoa”

git config --global user.email “email.com”

git config --global color.ui true

git init = inicializa o seu repositório

git status = verifica o status dos arquivos

git add + [nome do arquivo] = prepara o arquivo para ser commitado

git add . = adiciona todos os arquivos que estão modificados e no untracked files

git reset HEAD [arquivos] = volta do estagio 2 para o 1

git commit -m [“mensagem”] = commita o arquivo, o -m é para poder colocar uma mensagem

git commit – a -m “mensagem” = commita direto sem passar pelo estágio 2

git log = visualiza as alterações

git log -p = mostra o detalhamento do log todo

git log -p -2 = mostra o detalhamento dos últimos dois commits

git log –stat = mostra os logs com staatisticas

git log –pretty=oneline = mostra log formatado em uma linha

git log --pretty=format:"%h - %an, %ar : %s" = idem

git log –since=2.days

git commit = abre editor padrão [vim] para commitar arquivos

git checkout + hash para voltar a versão

git checkout -b [nome do branch] = cria novo branch

git checkout [nome do branch] = alterna entre branches

git branch = lista os branches

git merge [nome do branche] = fazer merge

git rebase [nome do branche] = organiza os commites no branche atual

git reset HEAD~1 --soft = volta um commit, porém as alterações não foram alteradas

git reset HEAD~3 --hard = cuidado, esse c omando arranca fora o seus comitês (3 ultimos)

**git checkout – [exemplo.php]** = Imagine agora que você fez diversas alterações nesse arquivo, porém, por qualquer motivo, você se arrependeu de fazê-las (lembrando que você apenas fez as alterações, mas não as commitou), você vai fazer o conteúdo do arquivo voltar ao estado original

**git branch -D nome-do-branch = remove o branch**

vim .gitignore = criar arquivo para ignorar no versionamento, basta registrar neste arquivo

ssh-keygen = gerar chave

git remote add origin [caminho github]

git push [origin] [master] = subir os arquivos pro github

git clone <https://github.com/ArmandoJunior/git-code-education.git>

git branch -a = mostra branches locais e remotos

git branch -b [nome branch] /origin/branch01 = cria um branch baseado em um branch especifico

git pull = traz para o branch as alterações dispodiveis

Quando um branch local é removido, isso não significa que o branch remoto também será.

Nesse caso, há um comando específico para que possamos remover um branch remoto. Utilize o comando:

git push origin :nome-do-branch

Removendo um branch

Após uma funcionalidade ser desenvolvida e o merge realizado, você poderá optar por remover o branch. É isso mesmo =)

Remover um branch não significa que você removerá os commits que você realizou, pois o merge já foi feito.

Para remover um branch utilize o comando: **git branch -D nome-do-branch**

Teste isso agora mesmo em seu computador!

### Voltando ao estado original

Vamos imaginar que você possui um arquivo no controle de versão chamado exemplo.php.

Imagine agora que você fez diversas alterações nesse arquivo, porém, por qualquer motivo, você se arrependeu de fazê-las (lembrando que você apenas fez as alterações, mas não as commitou). Para você fazer o conteúdo do arquivo voltar ao estado original, digite:

git checkout -- exemplo.php

Rodando esse comando, todos as alterações realizadas serão perdidas e o arquivo voltará exatamente como estava antes.

Esse recurso é extremamente útil, porém deve ser usado com cuidado.