ANOTAÇÕES CURSO DE GIT/GITHUB

Todos os comandos do git começam com a palavra “git” seguido do que queremos. Com “git config” conseguimos indicar diversas configurações do nosso git, como nome de usuário, e-mail, IDE, dentre outros.

Criar um repositório

Pelo cmd, acessar a pasta do projeto, e digitar “git init”. Isso irá criar o repositório com a pasta oculta .git .

Branch: são as versões do sistema. A versão principal se chama Master, que criada quando fazemos o primeiro commit.

Commit: é o envio para o git as modificações feitas no projeto, que normalmente vem acompanhado de um comentário, que é uma breve explicação do que foi feito. Para fazer um commit: git commit -m “comentário”.

Readme: é um arquivo .txt ou .md que explicam o projeto, contendo instruções de uso, versão e mais informações relevantes.

Comando git status: Na pasta do projeto, esse comando mostra a inclusão de novos arquivos.

Git log: mostra a lista de todos os commits feitos no atual branch.

Git branch: mostra todos os branchs do repositório e indica o branch atual com um \*

Quando modificamos, criamos ou deletamos um arquivo no projeto, o git apenas identifica essa mudança. Para que isso seja salvo no git, é preciso adicionar essas mudanças ao git com “git add -A” e depois fazer o commit (ou usar o comando “git commit -am ‘’ ”). Sem o commit, essas mudanças ficam apenas no computador.

Git reset

Para voltar a um commit do passado, basta usar o comando git reset --“reset-type” “commit id”.

Reset-type:

-soft: volta para o commit desejado, mas mantém as alterações feitas (apenas retira o commit), é como se ele voltasse no tempo de “git add -A”.

-mixed: ele volta no tempo antes de “git add -A”.

-hard: apaga tudo que foi feito desde a última modificação

Criar um branch

Ao criar um novo branch, este irá assumir o estado do branch atual. Para criar, basta usar “git branch nome-do-branch”

Para mudar de branch, devemos usar o comando “git checkout nome-do-branch”.

Git checkout serve para voltar para um determinado branch, commit ou arquivo específico.

Git diff

Esse comando lista detalhadamente tudo o que foi modificado desde a última versão.

O comando git diff –name-only mostra apenas os arquivos que foram modificados. Se você quer saber as modificações de um arquivo específico, basta “git diff nome-do-arquivo”.

Voltando apenas um arquivo

O comando deve ser “git checkout HEAD -- nome-do-arquivo”. “HEAD” é o branch atual.

Mandando o repositório local para o github

Todo o processo pode ser conferido aqui: <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

O primeiro passo é criar um repositório no github. Depois, no diretório do projeto e usando o git Bash (cmd do git), colocar o comando:

ssh-keygen -t ed25519 -C "ferreirasantos.leonard@gmail.com"

Ao teclar enter, a bash mostra onde será guardado os arquivos de configuração. Basta teclar enter novamente. Depois, ele ira pedir uma senha, pode esquecer e ir botanto enter até o fim. Então será gerado dois arquivos, um para a chave privada, e outro para a pública.

Agora devemos ir até o seu perfil do github – settings – SSH and GPG Keys – new SSH key. Dá um nome para chave (ex: pc home) e colar a chave pública gerada anteriormente.

Agora basta voltar ao bash e fazer a conexão entre o repositório local e o remoto, usando esse código:

git remote add origin https://github.com/Santos-Leonardo/learning-git.git

Agora que a conexão está feita, devemos apenas enviar os arquivos locais para o repositório online, com:

git push -u origin master

git push envia os arquivos locais para origin usando a branch master