COMANDOS GIT

8. Vamos começar com dois conceitos básicos e muito importantes: o “pull” e o “push”.

* **Pull:**significa “puxar”, isso remete a fazer o download dos arquivos do repositório, onde você “puxa os arquivos para seu computador”.
* **Push:** é traduzido para “empurrar”, pois faz menção ao envio (upload) dos arquivos para nosso repositório. Seria “empurrar arquivos do seu computador para o repositório”.

9. Os comandos são utilizados no console do seu editor de texto (IDE) durante a programação do código. Outro ponto importante, o comando de push pode ser utilizado apenas o commit.

* **Git clone**

O comando Git clone realiza a cópia de um repositório para sua IDE, e lhe dá a possibilidade de ver e trabalhar no código e arquivos do repositório.

**Clonar branch:***git clone <https://github.com/exemplolink>*

* **Git branch**

O comando git branch é utilizado para manipular as branches (ramificações que foram explicadas anteriormente).

**Criar branch:** *git branch <nome-da-branch>*

**Criar branch já com push:***git push -u <local-remoto> <nome-da-branch>*

**Ver lista de branches:***git branch –list (dois sinais de subtração antes do termo “list”)*

**Excluir uma branch:** *git branch -d <nome-da-branch>*

* **Git branch**

O comando git checkout é utilizado para “entrar” na branch em que vamos trabalhar. Ele também é válido para trocar de branch ou mesmo verificar commits e arquivos.

**Entrar/trocar e conferir arquivos e commits:** *git checkout <nome-da-branch>*

* **Git status**

O comando git status é utilizado para consultar informações, se há necessidade de realizar um pull ou push e se a branch está atualizada.

**Verificar informações da branch:** *git status*

* **Git add**

O comando git add serve para adicionar uma alteração no próximo commit. Ele também pode ser usado para adicionar todas as alterações de uma vez.

**Adicionar alteração no commit:** *git add <arquivo>*

**Adicionar todas alterações no commit:***git add -A*

* **Git commit**

O comando git commit é um dos mais usados pelos desenvolvedores. Ele define pontos de verificação ao longo do desenvolvimento do código.

**Observação:** o git commit não envia os arquivos para o repositório, apenas salva-os no dispositivo de trabalho local.

g*it commit -m “Observação do commit”*

* **Git push**

O comando git push é executado após a realização do commit. Ele é usado para fazer o upload do commit já realizado.

**Observação:**o git push realiza o upload apenas dos arquivos e códigos que já estão em um commit.

**Realizar um push no código para um repositório:***git push <repositório-remoto> <nome-da-branch>*

**Realizar um push no código para um diretório com uma branch recém criada:***git push -u origin <nome-da-branch>*

* **Git pull**

O comando git pull é utilizado para obter os dados de um repositório remoto, ou seja, ele baixa as alterações do código e aplica imediatamente. É importante frisar que alguns conflitos podem ocorrer durante o processo e que você deverá resolvê-los manualmente.

**Realizar o download das alterações que estão presentes no repositório:***git pull <repositório-remoto>*

* **Git revert**

O comando git revert serve para voltar a uma versão anterior do projeto, caso algum erro ocorra no arquivo você está trabalhando.

Primeiro, utilizamos o comando git log–oneline para que possamos ver as versões e os códigos respectivos de cada uma. Após ter feito isso, podemos realizar o git revert.

**Realizar a consulta de versões do projeto:** *git log–oneline*

**Realizar a reversão com o git reverse:***git reverse 0000000*(Exemplo de número de versão)

* **Git merge**

O comando git merge é empregado para fazer uma mescla das alterações com os arquivos originais. Ele é usado após o desenvolvedor concluir uma branch com uma correção, ou nova implementação, que deverá ser atrelada a branch original.

Para fazer isso, você precisa estar na branch principal, usar o comando de atualização local, git fetch, e finalizar com o merge.

**Realizar a consulta se está na branch principal:** *git checkout <nome-da-branch-principal>*

**Realizar a atualização local:***git fetch*

**Realizar o merge:***git merge <nome-da-branch-com-o-recurso>*