Para instalar o GIT na sua máquina:

<https://git-scm.com/download>

Uma vez instalado no terminal

**RELAÇÃO DE COMANDOS QUE VOCÊ PRECISA SABER**

git init // Inicia a linha do tempo

git add // adiciona ou atualiza mudanças para irem para a linha do tempo

git commit // adiciona um ponto na linha do tempo

git log // visualiza os pontos na linha do tempo - commit

git status // que informa o estado das alterações do nosso projeto

git show // apresenta determinado ponto na história

git branch // gerenciar novas linhas do tempo

git checkout // manipula as linas do tempo

git merge // vai unir as linhas do tempo

git push // envia alterações locais para o repositório remoto

git clone // clonar um projeto / repositorio

git pull // puxa do repositório remoto

**PARA OPERAR O GIT – USE OS COMANDOS**

Comandos inicialização do GIT

git init -->> Inicia um repositório

Comando para adicionar os arquivos no repositório

git add arquivoinicial.bat

ou

git add \*

ou

git add .

Para comentar uma alteração onde se cria um ponto na história na linha do tempo.

git commit -m " Mensagem sobre  a alteração que foi feita"

Para ver um ponto na história no log

git log

Para ver o estado do desenvolvimento. Para ver isto modifique o um arquivo do repositório e veja que ele deixa o nome do arquivo em vermelho na lista que resultou da consulta.

git status

Copie o numero do commit para ver o ponto na estória e para isto use o comando

git show numero-do-commit (basta os quatros primeiro números do commit para ele localizar )

ou

git show  ( Sem número nenhum mostra o último ponto na estória

Começa uma nova funcionalidade no seu projeto sem estragar o que já foi feito

Se você quer começar uma nova versão por um novo caminho branch

git branch                 (Branch = significa ramificação)

git branch novonome criou uma nova funcionalidade para o projeto

Para mudar sua ramificação, branch,  você  usa o comando checkout

git checkout novonome  (novonome para a nova branch )

Verifique digitando o comando

git status

Como agora temos duas linhas do tempo uma chamada Master e outra chamada novonome, podemos navegar entre elas através do comando checkout.

git checkout master

git checkout novonome

E olhar todas as nossas linhas do tempo usando somente o comando

git branch

Geralmente a master é onde está o arquivo em produção.

Para listar o conteúdo do diretório use sempre o comando

ls -al

Ele sempre vai mostrar toda a estrutura do container e a pilha.

Use o comando Clear para limpar a tela.

Agora vamos adicionar novas funcionalidades em projeto em produção.

git merge novonome

Com este comando você vai unir as Branch´s, os caminhos da linha do tempo.

git log

git show

Sempre para ver o que está acontecendo

Agora você quer deletar a branch da nova funcionalidade, depois de aplicar em seu projeto.

git branch -D novonome ( A flag –D é um comando para o git apagar a Branch )

git branch

**Voce quer colocar seu projeto na nuvem agora**

**GIT HUB**

O git hub é um repositório remoto

Crie a sua conta no https://github.com

Crie um repositório remoto

Como o repositorio no desktop já esta esta criado eu vou fazer o link, a junção do repositório remoto com o repositório no desktop

Para isto vou usar o comando

git remote add origin url  ( Copie a url que o git hub produziu quando você criou o repositório.

git remote add origin [**https://github.com/swilliamsilva**](https://github.com/swilliamsilva/nnnnnnnnnn)

Para ver os meus repositórios remotos e o estado deles eu uso o comando

git remote -v

Para subir a aplicação para o repositório remoto você pode usar o comando

git push

Para criar a branch no repositório on line use o comando

git push -u origin master

Ele vai pedir seu nome ou email e sua senha de usuário do git hub

Para evitar ter que colocar o usuário e senha sempre que for fazer um push use o comando

git config.credential.helper store

Então ele vai gravar suas credenciais na configuração ( git config )

**Para você pegar um projeto já iniciado, para trabalhar com ele on line**

**Você precisa resolver um conflito?**

Copie o link do projeto e no git hub e copie no repositório do git desktop

usando o comando

git clone **url**

git clone https;//github.com/simuleagora/ppppppp

ls -al

git status

git checkout -b teste

Ele vai criar uma branch chamado teste

Antes de enviara a resolução, precisamos atualizar o projeto local.

git pull

O git pull puxa as atualizações que foram feitas nas nuvens para o seu projeto.

Você  precisa voltar um determinado arquivo para um determinado instante do tempo

Tem que descobrir o momento com o git log

git checkout numero-da-alteracao

Recuperar um projeto deletado

git checkout -webpack.nomedoarquivo