# Introdução ao Git e ao GitHub

**Git:** Serve para que você possa criar uma linha do tempo com diferentes pontos das alterações feitas. Assim, você consegue manter o histórico das alterações de um mesmo arquivo ou arquivos ao longo do tempo. Ou seja, o Sistema de Controle de Versão Distribuído moderno mais usado no mundo, sendo projetado com desempenho, segurança e flexibilidade

**GitHub:** é o site mais conhecido de hospedagem de projetos git. ele serve para que possa manter as versões do projeto em que trabalha em um lugar centralizado, na internet, acessível a quem pertence ao projeto. Além de hospedar projetos git, o github dá visibilidade da linha do tempo de diferentes versões do projeto.

**Comandos básicos do Git:**

**Sistema Windows**

**dir** **=** (listar pastas)

**cd** = (mudar pastas) / cd .. = sair da pasta

**cls** **=** (limpar a tela)

**Tecla Tab =** (alto completa um nome na hora de digitar)

**mkdir =** (criar pastas/arquivos)

**echo =** printa de volta o que digitou

**echo “texto” > =** ele vai pegar o output, a saída dessa função echo e vai jogar em um arquivo.

**del “pasta” =** (deletar arquivos)

**rmdir “pasta” =** (deletar a pasta com todos seus arquivos)

**Tópicos fundamentais para entender o funcionamento do Git**

**SHA1:**

A encriptação gera um conjunto de caracteres identificador de 40 dígitos. É uma forma curta de representar um arquivo. Se você alterar uma vírgula que seja desse texto, já será gerado outro conjunto. Se você alterar a informação no arquivo para a que estava anteriormente e executar o comando SHA1 novamente,

ele retorna para a chave anterior

Comando que gera o SHA1 = openssl sha1 (nome do arquivo.txt)

**Objetos Fundamentais:**

**Blobs,** Git gera SHA1 e armazena Meta-Dados

**Trees,** armazenam blobs, responsável por montar a estrutura de um arquivo. Possuem os SHA1 dos metas-dados.

**Commit,** objeto mais importante, junta tudo. O SHA1 desse commit é o hash de toda essa informação.

**Iniciando o Git e Criando um commit**

**Comandos:**

**git init** =(iniciar repositorio no git)

**git add** =(mover arquivos e dar inicio ao versionamento)

**git commit** =(criar commites)

**Criar Repositório:**

**Acessar via terminal a pasta do seu projeto**

**git init** = Efetuar esse comando para criar uma pasta oculta .git

**ls -a** = (mostrar arquivos e pastas ocultas)

**Adicionar um Arquivo:**

**Adicionar um arquivo na pasta do seu projeto**

**git add \* =** Comando para adicionar as alterações feitas em todos os arquivos do projeto

**git commit -m "" =** Comando para fazer um commit do seu projeto (colocar uma mensagem dentro das strings "")

**Ciclo de vida dos arquivos no Git**

**Untracked:** são arquivos que o GIT ainda não tem ciência deles

**Tracked:** são os arquivos que o GIT já tem ciência deles:

**Unmodified** é o arquivo que não sofreu nenhuma alteração

**Modified** é o arquivo que sofreu alguma alteração

**Staged** são os arquivos que já foram adicionados no projetos e estão esperando o commit

**Configuração do seu Repositório no GIT**

Copiar o caminho html do seu repositório, que o GitHub vai dar e colocar no GIT

**git remote add origin (link do seu repositório GitHub)** = comando para adicionar seu repositório GitHub a sua máquina local. Origin, apelido do seu link repositório

**git remote -v** = Comando para listar seus repositórios cadastrados

**git push origin master** = Comando para empurrar seu repositório local para o remoto

**Resolvendo Conflitos**

**git pull origin master =** Comando para puxar um conteúdo do seu repositório remoto

Para resolver, assim que você puxar o conteúdo do seu repositório remoto, verifique as modificações que devem ser feitas, faças as devidas alterações e depois faça o passo de subir as suas alterações para o GitHub novamente.

**Como clonar um repositório público do GitHub na sua maquina:**

**Entre em algum repositório público e clique no botão CODE**

**Copie esse link e vá no seu terminal Git e faça - git clone (link)**

**Passo a Passo Git**

1. **Pwd** = mostra ao diretório/pasta que estou
2. **Mkdir**
3. **Ls**
4. **Cd**
5. **Code**. = abre o visual studio code na pasta atual
6. **Touch** = nome do arquivo que quero criar
7. **Git init**
8. **Git status=** mostra o estado atual dos arquivos na pasta. Untracked, não foi adicionado no git, Added, o arquivo foi adicionado no git e Modified, quando o arquivo já existe no git e foi modificado.
9. **Git add “nome do arquivo”**
10. **git commit -m ""**
11. **git remote add origin (link do seu repositório GitHub)**
12. **1°= git push origin master**

**2°= git push**

**Git diff= saber o que foi modificado no arquivo.**