**ENTENDENDO O QUE É GIT**

* Versionamento de códigos

**COMANDOS BÁSICOS**

* + **Dir (directory) =** listar as pastas (diretórios)
  + **Cd (change directory) =** mudar os diretórios (navegar em um específico)
  + **Cd .. =** voltar pro diretório anterior
  + **Cls (clear screen) =** limpar a interface
  + **TAB =** auto completar
  + **Mkdir (make directory) =** criar pasta (diretório)
  + **Echo (printa algo na tela) =** usado para criar arquivos. EX.:(ECHO HELLO > HELLO.TXT)
  + **Del (delete)/Rm (remove) =** deleta somente os arquivos do diretório
  + **Rmdir /S /Q (remove directory) =** remove todo o diretório

**TÓPICOS FUNDAMENTAIS**

**SHA1**

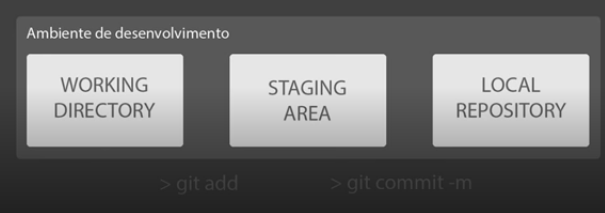
* Algoritmo de encriptação. Contém 40 dígitos. É uma forma curta de representar o estado do arquivo. Openssl sha1 + texto.txt.
* Maneira simples de observar se um arquivo teve ou não mudanças.

**OBJETOS INTERNOS DO GIT**

* **BLOBS =** ARQUIVOS (POSSUEM UM SHA1 PRÓPRIO);
* **TREES – ARMAZENAM BLOBS =** APONTA PARA OS BLOBS E ARMAZENA O NOME DESSES ARQUIVOS (TAMBÉM POSSUI SHA1 PRÓPRIO);
* **COMMITS – OBJETO QUE DÁ SENTIDO A ALTERAÇÃO FEITA =** APONTA PARA UMA TREE, APONTA PARA UM PARENTE (ULTIMO COMMIT), AUTOR E MENSAGEM. POSSUI UM TIMESTAMP. DÁ SIGNIFICÂNCIA PARA ALTERAÇÃO FEITA.

**PRIMEIROS COMANDOS COM O GIT**

* **GIT INIT – cria-se um repositório**
* **GIT ADD**
* **GIT COMMIT**

****

* **GIT PUSH –** Empurrar arquivo local para o GitHub
* **GIT PULL –** Puxar arquivo do GitHub para o Git

PASSO A PASSO PARA CRIAÇÃO DE REPOSITÓRIO

CRIAR NO GIT HUB 🡪 CLONAR NO NOSSO SERVIDOR LOCAL (GIT CLONE + URL) 🡪 ADICIONAR OS ARQUIVOS A ESSA PASTA 🡪 GIT ADD (ADICIONAR OS ARQUIVOS NO SERVIDOR LOCAL) 🡪 GIT COMMIT –M (PARA COMITAR AS MUDANÇAS E ESCREVER UMA MENSAGEM) 🡪 GIT PUSH (PARA EMPURRAR NOSSO ARQUIVO PARA A NUVEM)

**CADA MUDANÇA A SER ADICIONADA (GIT ADD. OU GIT ADD –A, PARA ADICIONAR TODA A PASTA**