**Git/GitHub**

Cada arquivo é salvo com um código SHA1, se for alterado e salvo o Git gerará outra chave SHA1, se for salvo exatamente igual ao primeiro arquivo a chave gerada será igual à do primeiro arquivo;

***Comandos básicos de GIT CLI (Comand Line Interface):***

**cd** – altera o diretório atual

**cd ..** – sobe um nível no diretório

**clear** – limpa a tela

**ls** – mostras o que existe de visível no diretório atual

**ls -a** – mostra pastas ocultas no diretório atual

**mkdir** -criar uma nova pasta no diretório atual

**mv nome do arquivo ./pasta para a qual será movido**/ - move o arquivo para o caminho descrito

**git init** – O comando git init cria um novo repositório do Git. Ele pode ser usado para converter um projeto existente e não versionado em um repositório do Git ou inicializar um novo repositório vazio.

**git add \***- O comando git add adiciona uma alteração no diretório ativo à área de staging. Ele diz ao Git que você quer incluir atualizações a um arquivo específico no próximo commit. No entanto, git add não tem efeito real e significativo no repositório — as alterações não são gravadas mesmo até você executar git commit.

**git commit** **-m** - “digita aqui o que foi alterado”- Usado para subir uma captura de tela do diretório de staging para o histórico de commit de repositórios, junto com uma mensagem inserida.

**git remote add origin (digite o link do repositório no git hub)** – adiciona o repositório indicado do git hub

**git push origin master** – faz o upload dos arquivos do repositório local no branch master para o github indicado como “origin”

**git pull origin master** – utilizado quando se tentar fazer o push para o GitHub mas ocorre erro pq as versões estão diferentes.

**git status** – mostra o status (untracked, unmodified, modified, staged) dos arquivos

**git clone (digita a URL do repositório que quer copiar)**- usado para clonar um repositório para o computador junto com todos os dados de versionamento e não apenas como uma pasta comum.