* **Clonar um repositorio do GitHub**

**git clone [url]**

**Exemplo: git clone https://github.com/luc-git-projects/comandos.git**

* **Ver situação dos arquivos**

**git status**

* **Rastrear arquivos**

**git add [nome-arquivo]**

**Exemplo: git add index.html**

* **Rastrear todos os arquivos**

**git add .**

* **Commitar arquivos**

**git commit –m “[aqui fica a mensagem do commit]”**

**Exemplo: git commit –m “Arquivo inicial de citacoes”**

* **Rastrear e commitar arquivos de uma vez**

**git commit –a –m “[aqui fica a mensagem do commit]”**

**Exemplo: git commit -a -m "Inserindo titulo e diminuindo tamanho da pagina"**

**git commit –am “[aqui fica a mensagem do commit]”**

**Exemplo: git commit -am "Inserindo titulo e diminuindo tamanho da pagina"**



* **Buscar alterações para o GitHub**

**git pull**

* **Enviar alterações para o repostório remoto**

**git push**

* **Ver historico de alterações gravadas no repositório**

**git log**

* **Mostrar quantidade de logs que quisermos (são mostrados os ultimos commits)**

**git log –n [quantidade de logs a serem mostrados]**

**Exemplo: git log –n 2 (mostra os dois ultimos commits)**

* **Mostrar commits bem consisos**

**git log - -oneline**

* **Mostrar resumo de arquivos alterados e quantidade de linhas adiconadas e removidas**

**git log - -stat**

* **Mostrar as diferenças entre os arquivos anteriores a modificação e os arquivos que foram alterados que se encontram no diretório de trabalho**

**git diff**

* **Mostrar as diferenças entre um arquivo especifico antes da modificação e o arquivo que foi alterado que se encontra no diretório de trabalho**

**git diff [nome do arquivo]**

**Exemplo: git diff index.html**

* **Mostrar as diferenças entre os arquivos que estão na area de stage**

**git diff - -staged**

* **Mostrar as diferenças em um arquivo especifico na area de stage**

**git diff - –staged [nome do arquivo]**

**Exemplo:** **git diff - -staged index.html**

* **Mostrar as diferenças entre um arquivo especifico que foi comitado anteriormente e o arquivo alterado**

**git diff [nome do arquivo]**

**Exemplo: git diff index.html**

* **Mostrar as diferenças nos arquivos no diretorio de trabalho e na area de stage**
* **Primeiro temos que conseguir o código do último commit**
* **git log –n 1 - -oneline**
* **Depois usar o comando diff com o código do comit**
* **Git diff [código do commit]**
* **Exemplo: git diff 039e627**
* **Mostrando as diferenças entre intervalos de commits (arquivos tem que estar commitados)**

**git diff [código do commit anterior]..[código do commit posterior]**

**Exemplo: git diff 039e627..877c170**

* **Mostrando as diferenças entre intervalos de commits (mostra qualquer alteração, mesmo os arquivos não sendo commitados e eles podem estar no diretorio de trabalho ou na área de stage)**

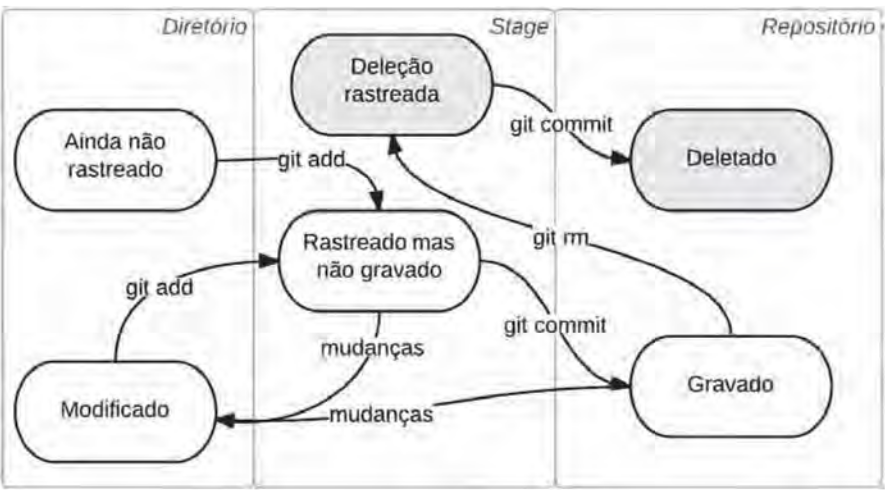
**git diff [codigo do ultimo commit que deseja ver]~[quantidade de commits anteriores]**

**Exemplo: git diff 877c170~2**

* Remoção do arquivo, mas arquivo ja tem que estar commitado

**git rm [nome do arquivo]**

**Exemplo: git rm produtos.html**



* **Mudar nome de arquivo**

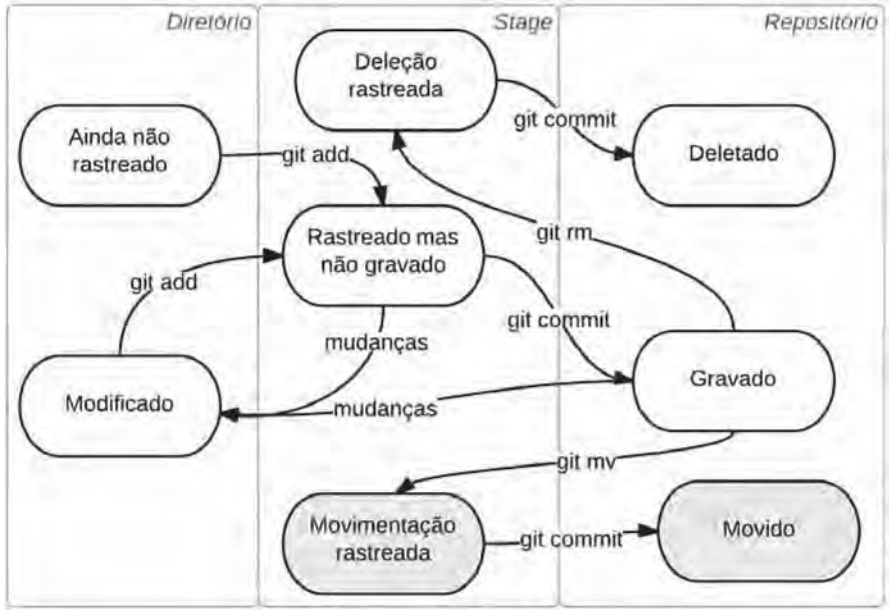
**git mv [nome do arquivo que vai ser mudado] [novo nome do arquivo]**

**Exemplo: git mv estilos.css principal.css**

* **Movendo arquivo para outro diretório (diretório já tem que existir)**

**git mv [nome do arquivo] [diretório/nome do arquivo]**

**Exemplo: git mv principal.js js/principal.js**



* **Desfazendo aterações não rastreadas no diretório de trabalho (arquivos dentro da área de stage não serão afetados)**

**git checkout - - [nome do arquivo]**

**Exemplo: git checkout -- index.html**

* **Recuperar arquivos que foram removidos acidentalmente**

**git checkout - - [nome do arquivo]**

**Exemplo: git checkout -- index.html**

* **Desfazendo alterações de arquivos na área de stage (retira o arquivo da área de stage, mas as mudanças efetuadas no arquivo seram preservadas)**

**git reset - - [nome do arquivo]**

**Exemplo: git reset -- index.html**

* **Removendo todas as aterações de todos os arquivos na área de stage (o repositório volta como estava no último commit)**

**‘ git reset - -hard**