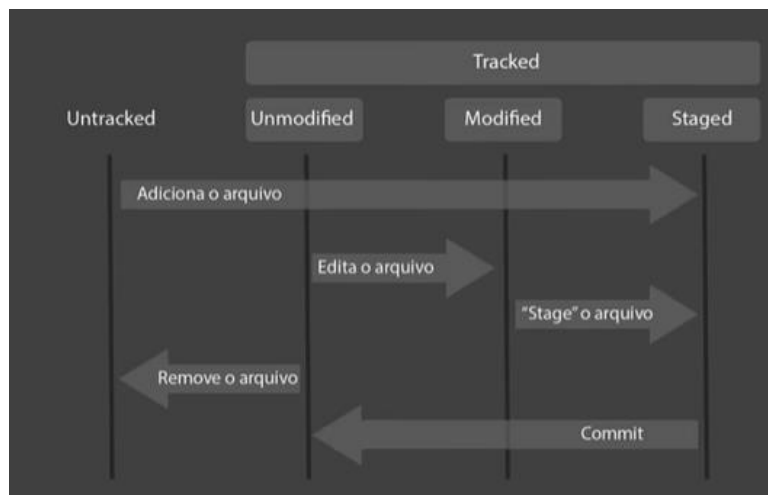


## Passo a passo no ciclo de vida

- Git Init: Usado para criar nossa pasta .git e além disso inicializa o conceito do Git chamado repositório. Quando usamos ele, de fato estamos criando um repositório no Git dentro daquela pasta.
- Tracker / Untracked: Dentro do Tracked, que são os arquivos de fato rastreados pelo Git, ele pode se subdividir em 3 fases diferentes, sendo elas – Unmodified, Modified e Staged. Temos também o outro grupamento que é o Untracked que são os arquivos que o Git ainda não tem ciência deles. Unmodified é o arquivo que ainda não foi modificado, Modified é um arquivo que sofreu alguma modificação e o Staged é um conceito chave para que entendamos que é onde ficam os arquivos que estão sendo preparados para que façam parte de outro tipo de agrupamento.



Quando utilizamos o **git add** nós tínhamos um arquivo que estava untracked pois havíamos acabado de criar ele, e usando esse comando, movemos esse arquivo direto para o Staged. Os arquivos Unmodified são arquivos dentro do repositório que ainda não sofreu nenhuma modificação e quando entramos no arquivo e modificamos algo, ele vai diretamente para o Modified pois o Git vai comparar o SHA1 dos arquivos e vai perceber que aquele arquivo sofreu alteração. Se rodarmos o **git add** novamente nesse arquivo que está no Modified, ele vai também para o Staged. Se temos o nosso arquivo Unmodified e removemos esse arquivo, ele retorna para o Untracked mesmo.

Quando nós enviamos o arquivo para o Staged, significa que esse arquivo está sendo preparado para receber um Commit e esse Commit retorna todos esses arquivos para o Unmodified para que se inicie novamente o ciclo.

- Repositórios: Nesse setor temos a separação de dois ambientes, sendo ele o ambiente de desenvolvimento que vai estar na máquina e teremos um servidor. O git vai ter a versão dele no servidor, que no caso é o Github, e ele tem a versão que está na nossa máquina. Então, as alterações que fazemos na máquina, não repercutem imediatamente na versão que está no repositório remoto.

No meu ambiente de desenvolvimento temos o Repositório de Trabalho, que é o que estamos manipulando no livro de receitas. Temos também a área de Staging que foi o que falamos no tópico anterior. Nosso arquivo vai se movendo entre esses dois ambientes a medida que vamos adicionando novos arquivos e modificações. Quando fazemos o Commit, ele passa a fazer parte do meu repositório local que, por sua vez, pode ser empurrado para o repositório remoto.



Vamos entender na prática como esses conceitos são aplicados.

Vamos ao nosso Git Bash e vamos colocar mais arquivos no nosso repositório de livro de receitas e vamos passar novamente por esse ciclo. De início vamos dar ls para mostrar os arquivos que tem lá dentro e vamos utilizar agora o **git status** que será muito importante nos ajudando a monitorar o status desses arquivos. Aplicando esse comando, nosso terminal irá nos retornar o seguinte:

```
Igor Carrilho@LAPTOP-J2KR9VMJ MINGW64 /c/workspace/livro-receitas (master)
$ git status
On branch master
nothing to commit, working tree clean

Igor Carrilho@LAPTOP-J2KR9VMJ MINGW64 /c/workspace/livro-receitas (master)
$ |
```

Isso significa que nosso repositório de trabalho não tem nenhuma alteração e está limpo. Vamos então adicionar um arquivo ali criando ali uma pasta chamada receitas utilizando o **mkdir receitas** e vamos mover o strogonoff para essa pasta utilizando o comando **mv strogonoff ./receitas/**

```
Igor Carrilho@LAPTOP-J2KR9VMJ MINGW64 /c/workspace/livro-receitas (master)
$ ls
receitas/ 'strogonoff de frango.md'

Igor Carrilho@LAPTOP-J2KR9VMJ MINGW64 /c/workspace/livro-receitas (master)
$ mv strogonoff\ de\ frango.md ./receitas/

Igor Carrilho@LAPTOP-J2KR9VMJ MINGW64 /c/workspace/livro-receitas (master)
$ |
```

Agora com o arquivo já dentro da pasta receitas, vamos novamente dar um git status e veremos que agora ele nos trará outras informações:

```

MINGW64:/c/workspace/livro-receitas
Igor Carrilho@LAPTOP-J2KR9VMJ MINGW64 /c/workspace/livro-receitas (master)
$ mv strogonoff\ de\ frango.md ./receitas/

Igor Carrilho@LAPTOP-J2KR9VMJ MINGW64 /c/workspace/livro-receitas (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    strogonoff de frango.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        receitas/

no changes added to commit (use "git add" and/or "git commit -a")
Igor Carrilho@LAPTOP-J2KR9VMJ MINGW64 /c/workspace/livro-receitas (master)
$

```

Ele está falando que deletou pois perante ao sistema operacional, parece que você deletou aquele arquivo, porem apenas movemos para outra pasta e o Git ainda não conhece essa pasta. Então, quando o Git não sabe de onde vem aquela pasta, ele dá aquele status de untracked files.

Vamos agora adicionar esses arquivos para a área Staged utilizando o **git add strogonoff receitas/** e após isso daremos git status novamente e o terminal nos retornará a seguinte mensagem:

```

Igor Carrilho@LAPTOP-J2KR9VMJ MINGW64 /c/workspace/livro-receitas (master)
$ git add strogonoff\ de\ frango.md receitas/
warning: LF will be replaced by CRLF in receitas/strogonoff de frango.md.
The file will have its original line endings in your working directory

Igor Carrilho@LAPTOP-J2KR9VMJ MINGW64 /c/workspace/livro-receitas (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    strogonoff de frango.md -> receitas/strogonoff de frango.md

Igor Carrilho@LAPTOP-J2KR9VMJ MINGW64 /c/workspace/livro-receitas (master)
$

```

Ele nos informa que os arquivos mudaram de estado para Staged e podem ser commitados. Caso a gente quiser voltar ele para Unstaged, basta utilizarmos o comando que o terminal nos indica que é o “git restore --staged <file>”. Mas o que queremos é fazer o commit, e para isso iremos utilizar **git commit -m “cria pasta receitas”**. O terminal nos trará as informações do commit como a versão resumida do SHA1 e nos mostrar que temos 1 arquivo modificado e o nome do arquivo. Se dermos git status agora, não teremos mais nenhum arquivo para commitar e a nossa arvore de trabalho estará limpa.

Se por exemplo formos ao arquivo que temos e mudar algo, ele voltara para o status de Modified e entramos novamente nesse ciclo para que seja feito todo o processo novamente. Vamos adicionar um outro arquivo que irá servir de Index par que as pessoas possam ver todos os arquivos que têm dentro da nossa pasta de livro de receitas. Para isso, vamos utilizar echo > README.md.