**Para o git para de monitorar um arquivo**

git rm -f {arquivo}

**Restaurar um arquivo excluído**

O comando verificar o reposito online se oq esta na minha maquina esta

git checkout -b {nome\_da\_branch}

**para renomear um arquivo**

git mv arquivo\_origem arquivo\_destino

e funciona. De fato, se você fizer algo desse tipo e consultar o status, você verá que o Git considera que o arquivo foi renomeado.

No entanto, isso é equivalente a rodar algo como:

mv README.txt README   
git rm README.txt   
git add README

**git stash**

Muitas vezes, quando você está trabalhando em uma parte do seu projeto, as coisas estão em um estado confuso e você quer mudar de branch por um tempo para trabalhar em outra coisa. O problema é, você não quer fazer o commit de um trabalho incompleto somente para voltar a ele mais tarde. A resposta para esse problema é o comando **git stash**.   
Você quer mudar de branch, mas não quer fazer o commit do que você ainda está trabalhando; você irá fazer o stash das modificações. Para fazer um novo stash na sua pilha, execute:

git stash

Seu diretório de trabalho estará limpo.   
Neste momento, você pode facilmente mudar de branch e trabalhar em outra coisa; suas alterações estão armazenadas na sua pilha. Para ver as stashes que você guardou, você pode usar

git stash list

Você pode aplicar aquele que acabou de fazer o stash com o comando mostrado na saída de ajuda do comando stash original: **git stash apply**. Se você quer aplicar um dos stashes mais antigos, você pode especificá-lo, assim: **git stash apply stash@{2}**. Se você não especificar um stash, Git assume que é o stash mais recente e tenta aplicá-lo.

**git tag**

Git tem a habilidade de criar tags em pontos específicos na história do código como pontos importantes. Geralmente as pessoas usam esta funcionalidade para marcar pontos de release (v1.0, e por aí vai). Nesta seção, você aprenderá como listar as tags disponíveis, como criar novas tags, e quais são os tipos diferentes de tags.   
Para listar as **tags** execute:

git tag

Para criar uma **tag** basta executar o seguinte comando, caso não queira criar a **tag** anotada, somente retire os parâmetros **-a** e **-m**.

git tag -a v1.4 -m 'my version 1.4'