

Deletando arquivos do seu repositório Git

1. Faça um *clone* ou *pull* no GitHub do seu projeto criado na última aula prática. Para isso, utilize um dos comandos:

```
git pull origin master (para o caso de já ter o repositório na máquina, se não tiver, use o comando abaixo) ou git clone <endereço>
```

2. Faça um novo arquivo e coloque-o na pasta do projeto.

3. Apague da pasta os outros arquivos do projeto.

4. Faça um `git status` para ver a situação do repositório local. A mensagem de ser parecida com a mensagem abaixo:

```
~/Área de Trabalho/integrador(master) $ git status
No ramo master
Changes not staged for commit:
  (utilize "git add/rm <arquivo>..." para atualizar o que será submetido)
  (utilize "git checkout -- <arquivo>..." para descartar mudanças no
diretório de trabalho)
```

```
deleted:      triggers/TR_MONITOR_DELETE_ALUNO_CURSO.sql
deleted:      triggers/TR_MONITOR_DELETE_PESSOAS.sql
```

Arquivos não monitorados:

(utilize "git add <arquivo>..." para incluir o que será submetido)

```
triggers/NovoArquivo.sql
```

nenhuma modificação adicionada à submissão (utilize "git add" e/ou "git commit -a")

5. Faça um `git add` para adicionar o arquivo que você acabou de criar. A mensagem deve ser parecida com a mensagem abaixo:

```
~/Área de Trabalho/integrador(master) $ git add .
warning: You ran 'git add' with neither '-A (--all)' or '--ignore-removal',
whose behaviour will change in Git 2.0 with respect to paths you removed.
Paths like 'triggers/mssql/sapiens/TR_MONITOR_DELETE_ALUNO_CURSO.sql'
that are
removed from your working tree are ignored with this version of Git.
```

```
* 'git add --ignore-removal <pathspec>', which is the current default,
  ignores paths you removed from your working tree.
```

```
* 'git add --all <pathspec>' will let you also record the removals.
```

Run 'git status' to check the paths you removed from your working tree.

6. A mensagem acima indica que os arquivos deletados não foram adicionados. Porém, os arquivos que você apagou não foram removidos do repositório. Para “gravar” essa remoção a partir do próximo *commit*, execute o comando: `git add -all`

7. Rode o `git status` novamente e veja que o Git agora registrou que os arquivos serão deletados do repositório, a partir do próximo *commit*. A mensagem deve ser parecida com essa:

```
~/Área de Trabalho/integrador(master) $ git status
```

```
No ramo master
```

```
Mudanças a serem submetidas:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
deleted:    triggers/TR_MONITOR_DELETE_ALUNO_CURSO.sql
deleted:    triggers/TR_MONITOR_DELETE_PESSOAIS.sql
new file:   triggers/NovoArquivo.sql
```

8. Faça um *commit* para gravar as mudanças.

9. Faça um *push* para seu repositório no GitHub. Lembrando que é necessário ou configurar a chave SSH, para o caso de seu origin apontar para um link SSH, ou inserir seu usuário e senha do GitHub.

Fork

Como mostrado na última aula, o GitHub possui uma ferramenta para facilitar a cópia de um projeto público para o nosso próprio GitHub. Para fazer isso é só acessar a página do projeto no GitHub e clicar em "Fork" no canto superior direito da tela.

1. Faça o Fork desse projeto: <https://github.com/vinduarte/det476teste>
2. Baixe-o para a sua máquina, usando o comando *clone*.
3. Faça alguma alteração, registrando-a num *commit*.
4. Faça *push* dessa mudança para seu GitHub.
5. Mostre ao professor.

Dica:

Daqui para frente na disciplina, os materiais estarão disponíveis no seguinte projeto do GitHub: <https://github.com/vinduarte/det476-material>. Vocês poderão clonar esse projeto e fazer *push* para ver se tem atualização. Ou então poderão fazer Fork do projeto, mas aí perderão as atualizações (pensem porquê).