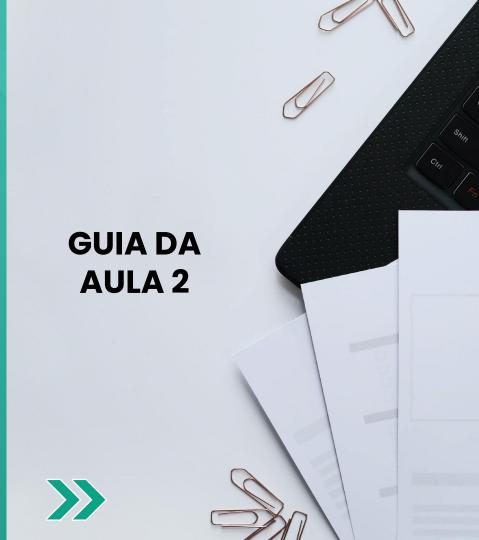


Profissão: Analista de dados





CONTROLE DE VERSÃO II







Visualize e desfaça alterações

- Visualizar
- Reverter



Acompanhe aqui os temas que serão tratados na videoaula







Visualizar

O comando git log lista os últimos *commits* (id, data, autor, mensagem etc.) em ordem cronológica. A documentação desse comando pode ser encontrada no *link* https://git-scm.com/docs/git-log

Os usos mais comuns do comando são:

```
git log

git log <nome-do-arquivo>

Exemplo:

In []: !git log

In []: !git log --oneline
```





Já o comando git diff mostra as diferenças entre um arquivo na working com a staging ou repository área, ou seja, entre a versão recentemente alterada com a última versão salva. A documentação desse comando pode ser encontrada no link https://git-scm.com/docs/git-diff

O uso mais comum do comando é:

git diff <nome-do-arquivo>

```
In []: !git status
In []: !git diff hello.py
```

((| |))



Reverter

O comando git reset move arquivos da staging de volta para a working área, essencialmente desfazendo o comando git add. A documentação desse comando pode ser encontrada no link https://git-scm.com/docs/git-reset
Os usos mais comuns do comando são:

```
git reset

git reset <nome-do-arquivo>

Exemplo:

In []: !git status

In []: !git reset

In []: !git status
```





Já o comando git checkout move arquivos da *repository* de volta para a *working* área, essencialmente desfazendo qualquer alteração feita nos arquivos. A documentação desse comando pode ser encontrada no *link* https://git-scm.com/docs/git-checkout
Os usos mais comuns do comando são:

```
git checkout

git checkout <nome-do-arquivo>

Exemplo:

In []: !git status

In []: !git checkout

In []: !git status
```

