



**git**

uma ferramenta de  
versionamento

# Revisão



# Problemas (versionamento)

1

Compartilhamento dos projetos

2

Controle de alterações

3

Trabalho distribuídos em grupos e offline

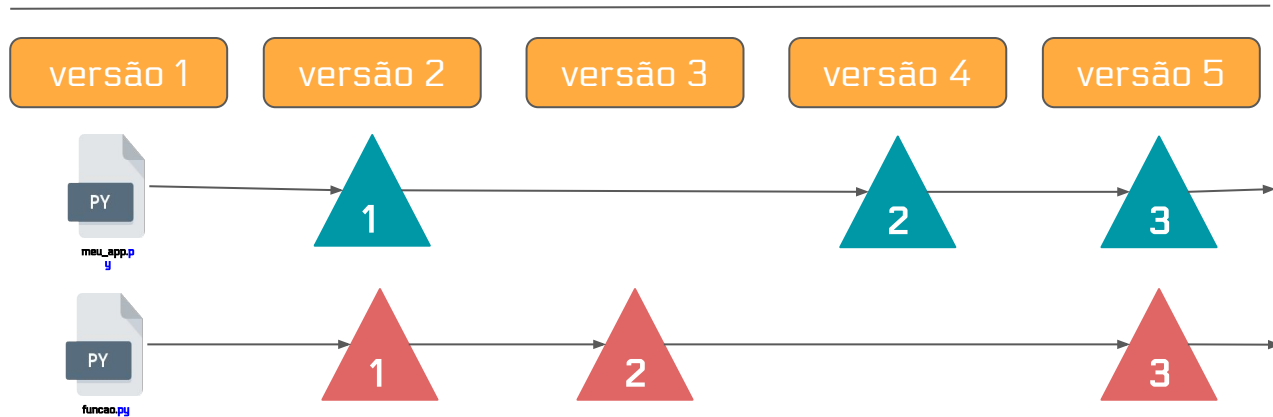
# Tipos de Versionamento



**delta differences**

salva apenas as diferenças  
dos arquivos de uma versão  
para outra.

versões ao longo do tempo



# Tipos de Versionamento



**delta differences**

salva apenas as diferenças  
dos arquivos de uma versão  
para outra.

1



fortemente dependente do arquivo  
original

2



baseado nas diferenças dos  
arquivos entre as versões

3



versionamento horizontal

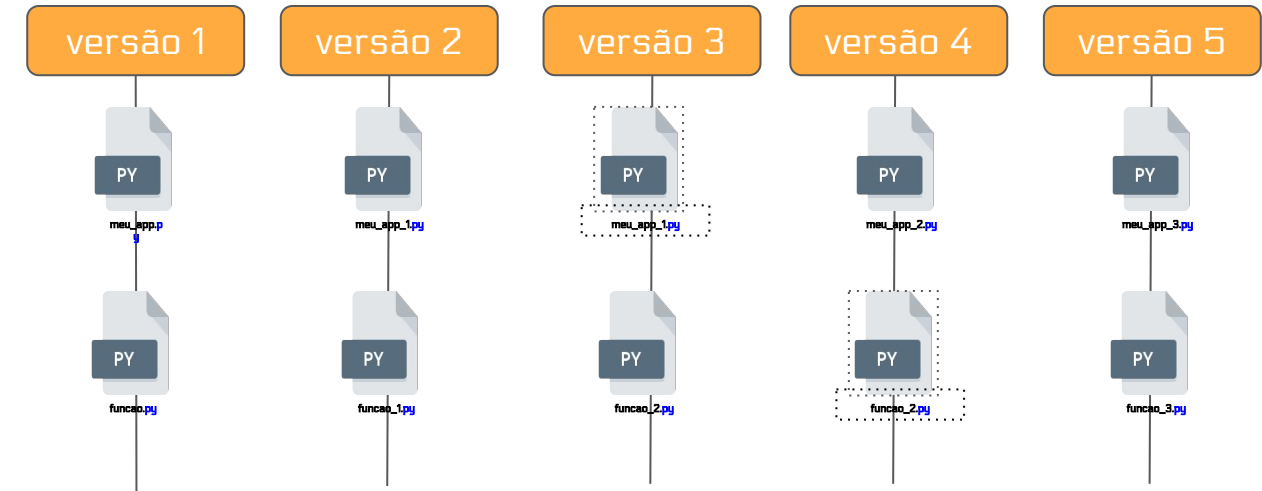
# Tipos de Versionamento



**snapshots**

cópias inteiras dos arquivos  
versionados

versões ao longo do tempo



# Tipos de Versionamento



## snapshots

cópias inteiras dos arquivos  
versionados

1



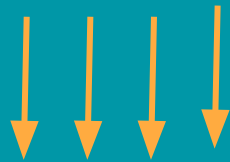
arquivos independentes por versão

2



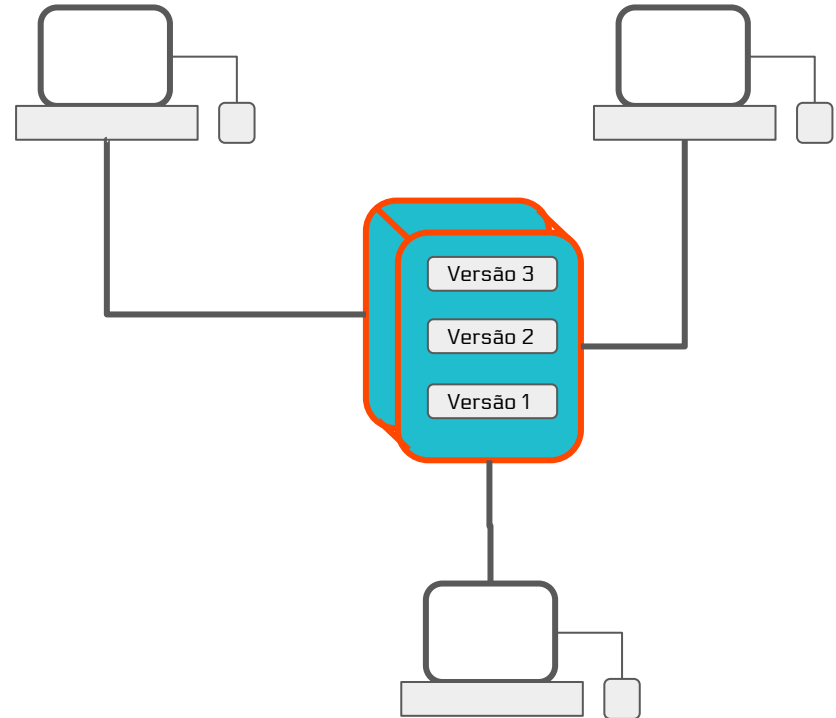
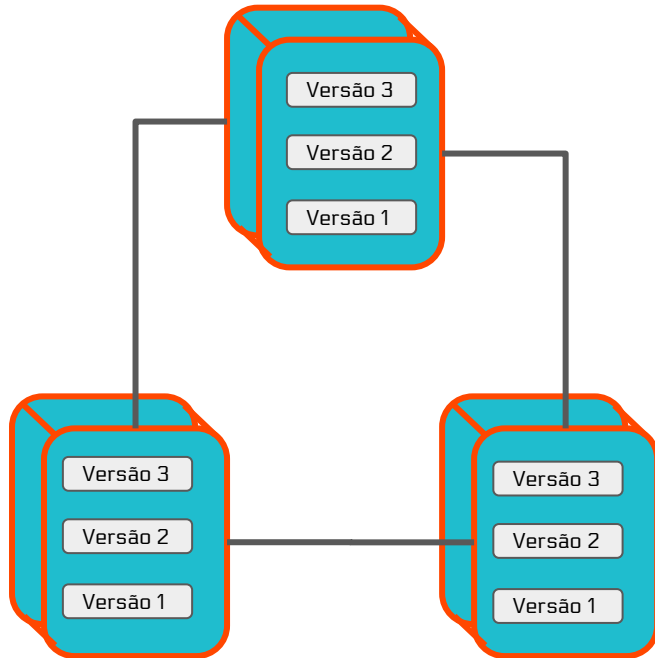
copia o arquivo inteiro da  
versão.

3



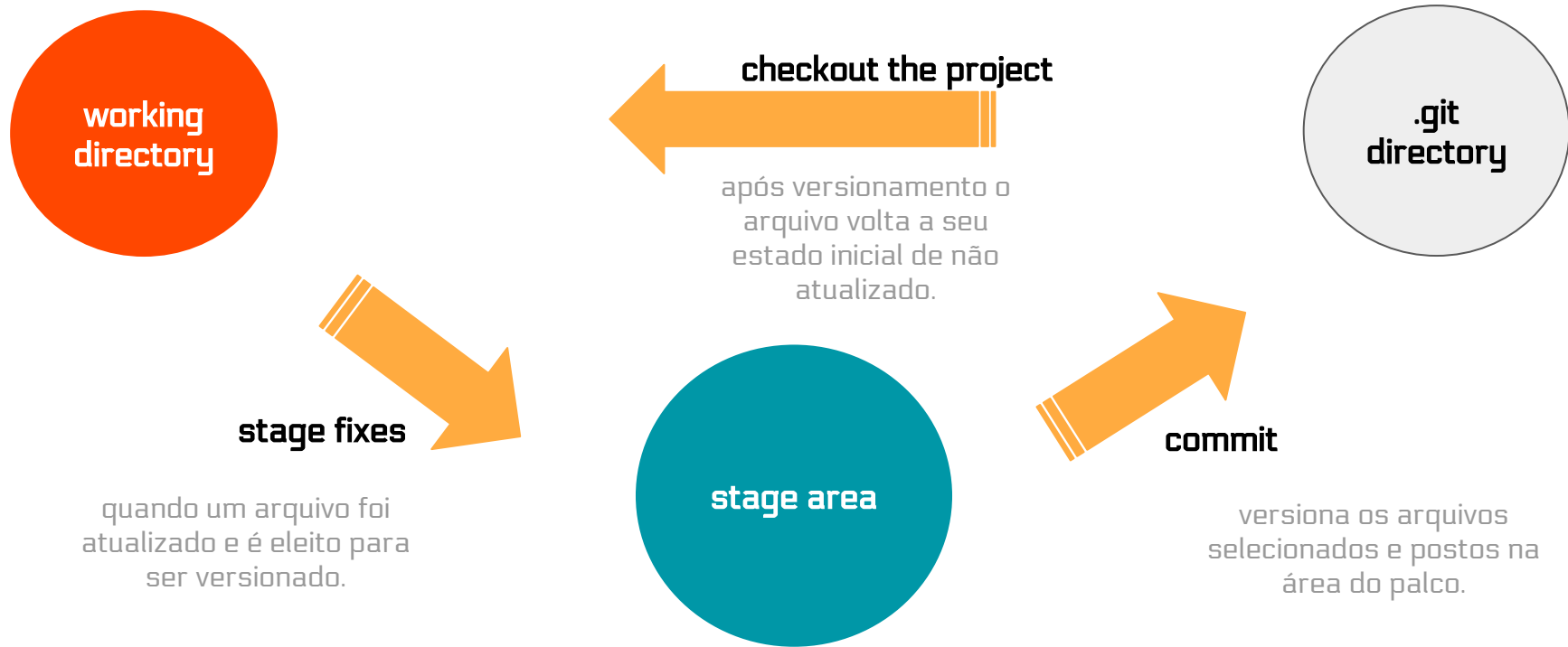
versionamento vertical

# Sistemas de Controle de Versão

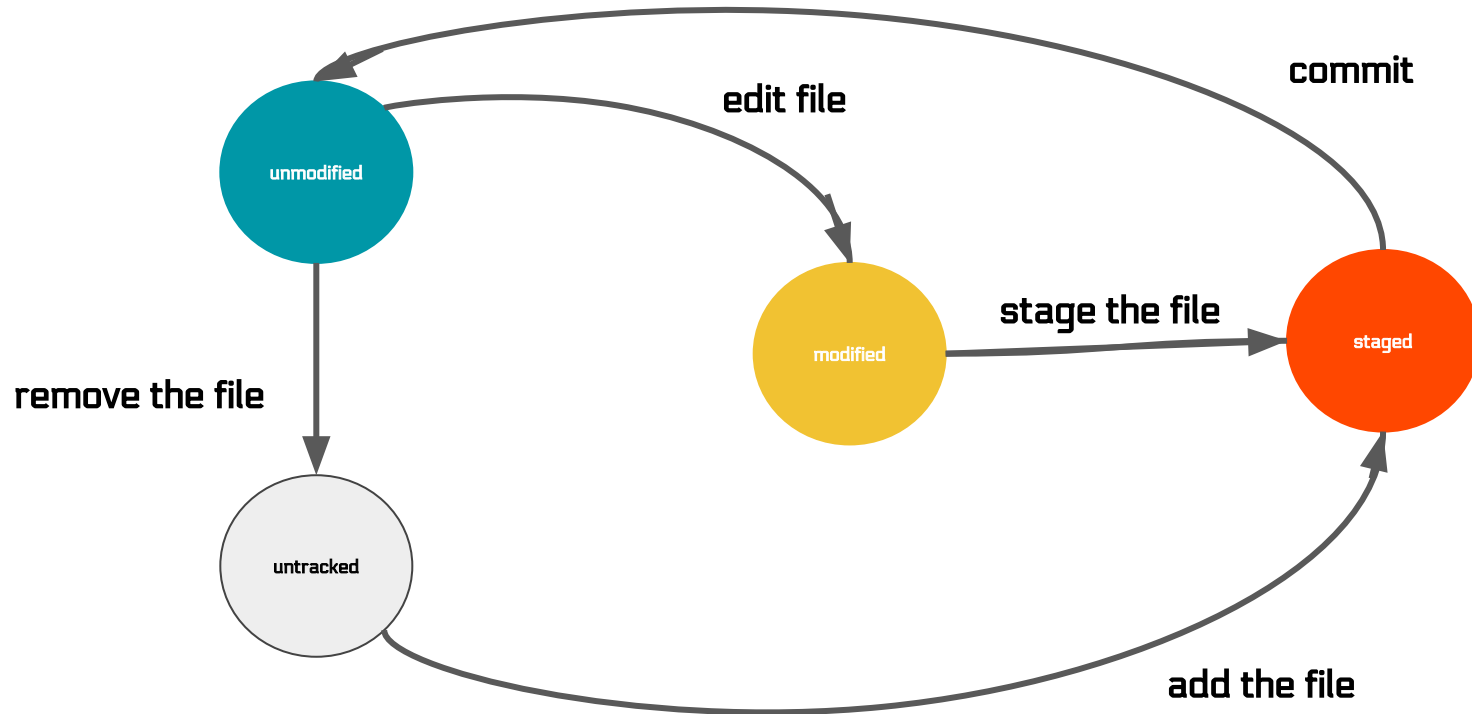




# Git Área



# Estados dos Arquivos



# Comandos Git



# Comandos Git

git init



transforma um  
diretório em diretório  
de versionamento git

# Git Área



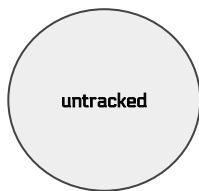
# Comandos Git

Crie um arquivo de texto:  
meu\_primeiro\_arquivo.txt

# Como faz? – Git Área



# Como faz? – Estados dos Arquivos





# Comandos Git

## git status



lista todos os arquivos no  
working directory e os seus  
respectivos estados.

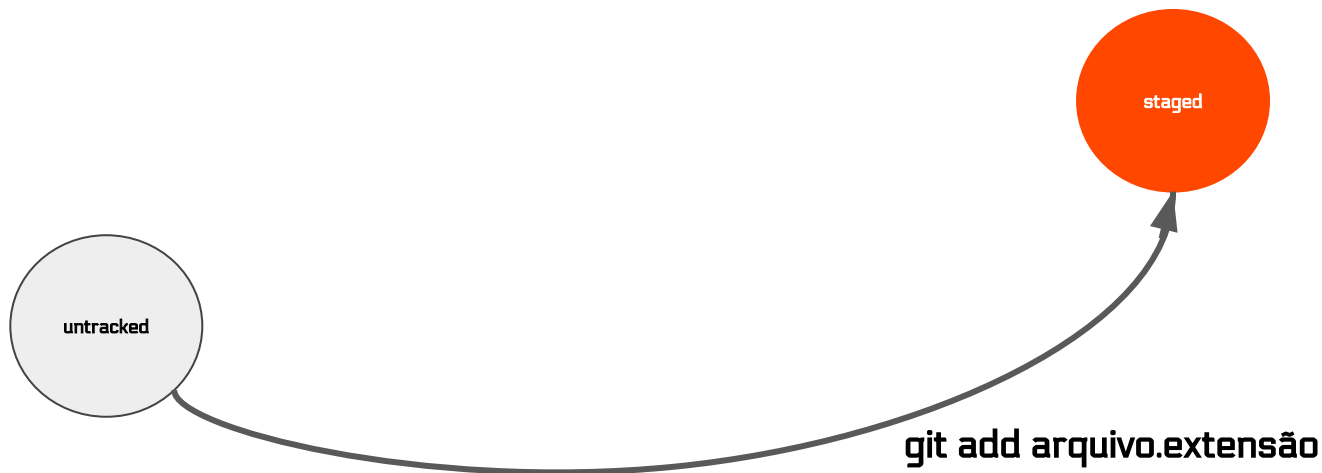
# Comandos Git

`git add nome_arquivo.extensão`

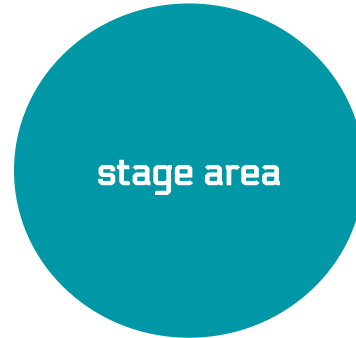


adiciona um arquivo no  
palco para ser  
versionado.

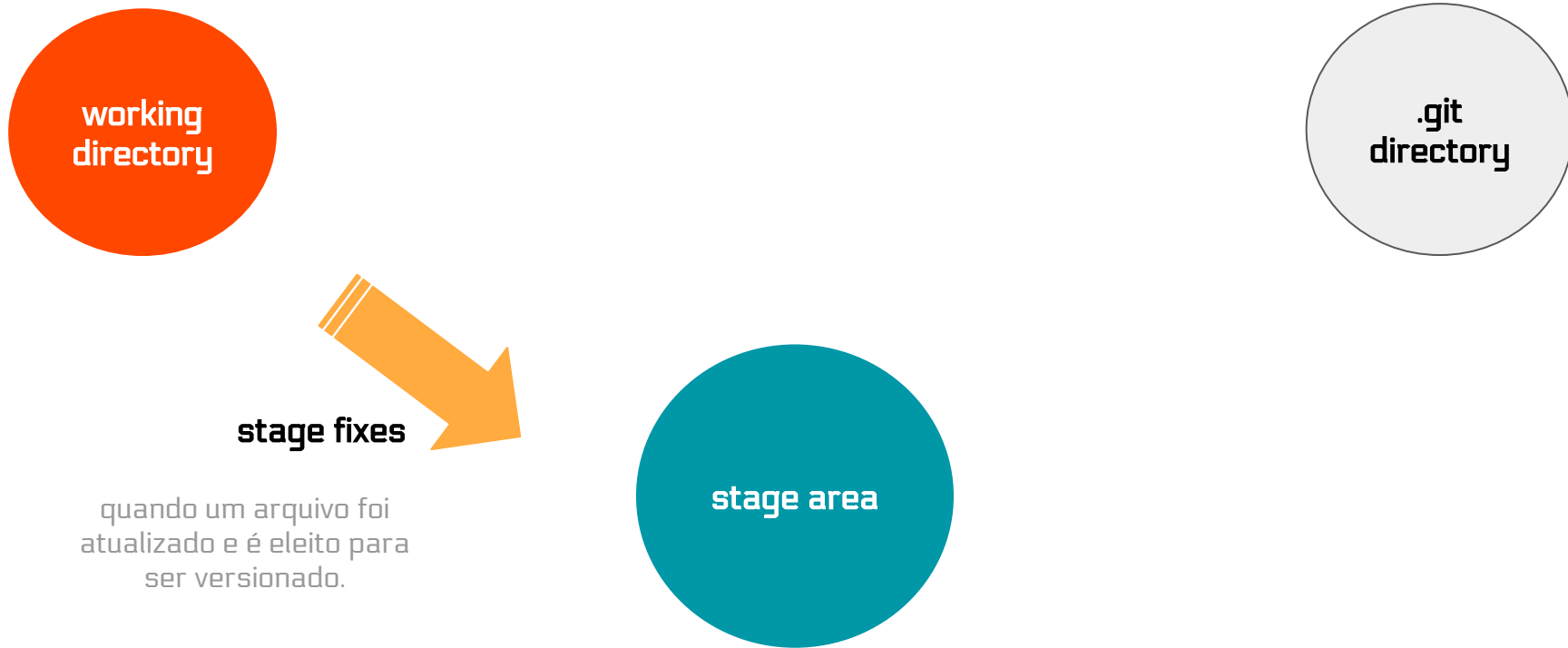
# Como faz? – Estados dos Arquivos



# Como faz? – Git Área



# Como faz? – Git Área



## git status --short



lista todos os arquivos no  
working directory e os seus  
respectivos estados.

# Comandos Git

## git status -s



lista todos os arquivos no  
working directory e os seus  
respectivos estados.

# Não é Comando Git

.gitignore



não é um comando, mas ajuda  
a organizar os tipos de  
arquivos que queremos  
ignorar



# Comandos Git

## git commit



versiona todos os arquivos que  
estão no palco (staged area)

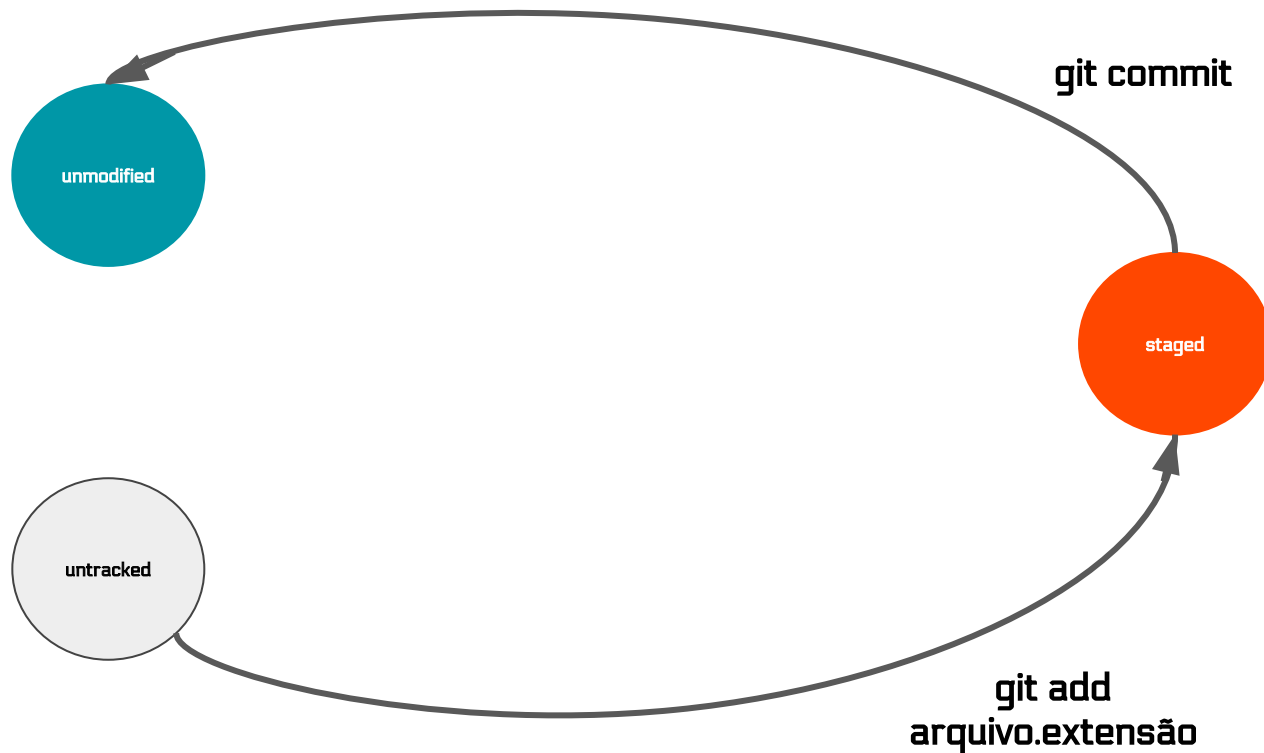
# Comandos Git

```
git commit -m 'Meu primeiro commit'
```

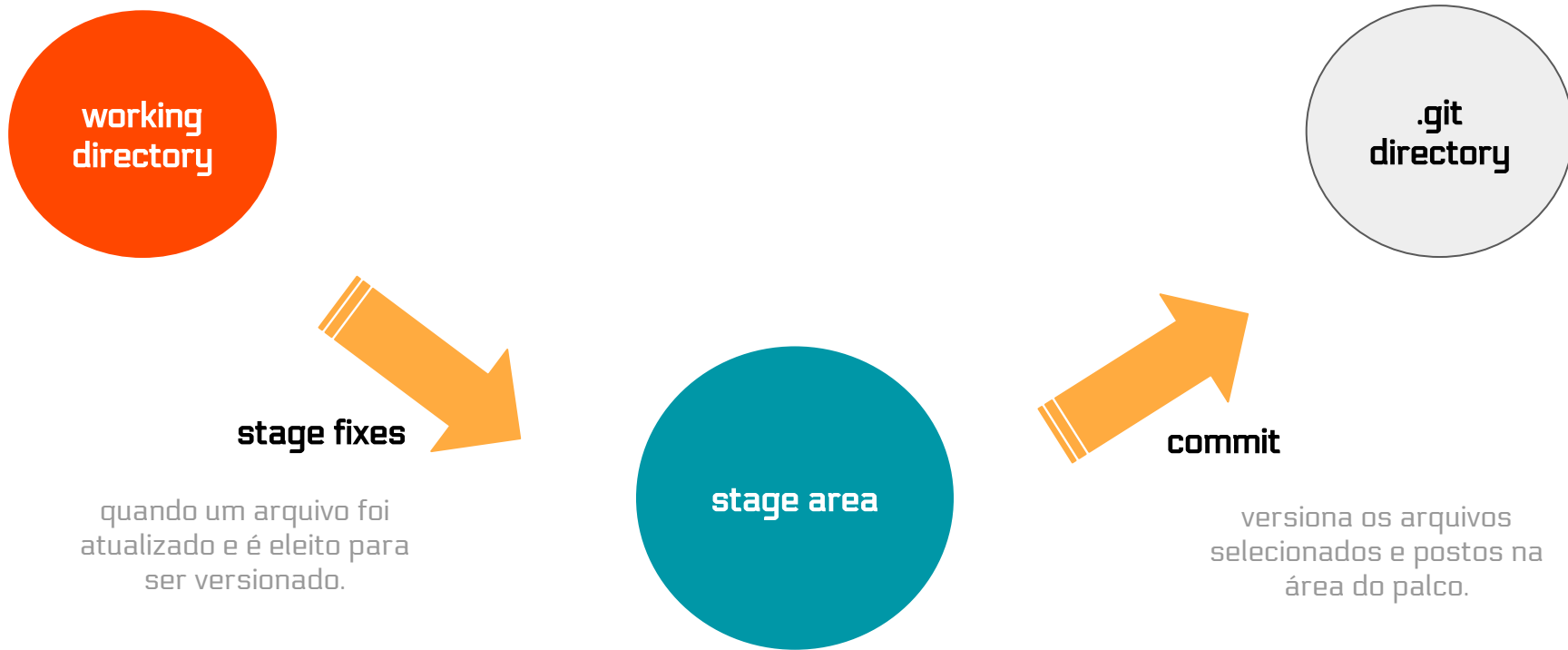


versiona todos os arquivos que  
estão no palco (staged area)

# Como faz? – Estados dos Arquivos



# Git Área



# Comandos Git

git diff



apresenta quais arquivos  
foram alterados em relação a  
última versão

# Comandos Git

git diff



apresenta quais arquivos  
foram alterados em relação a  
última versão e suas  
diferenças

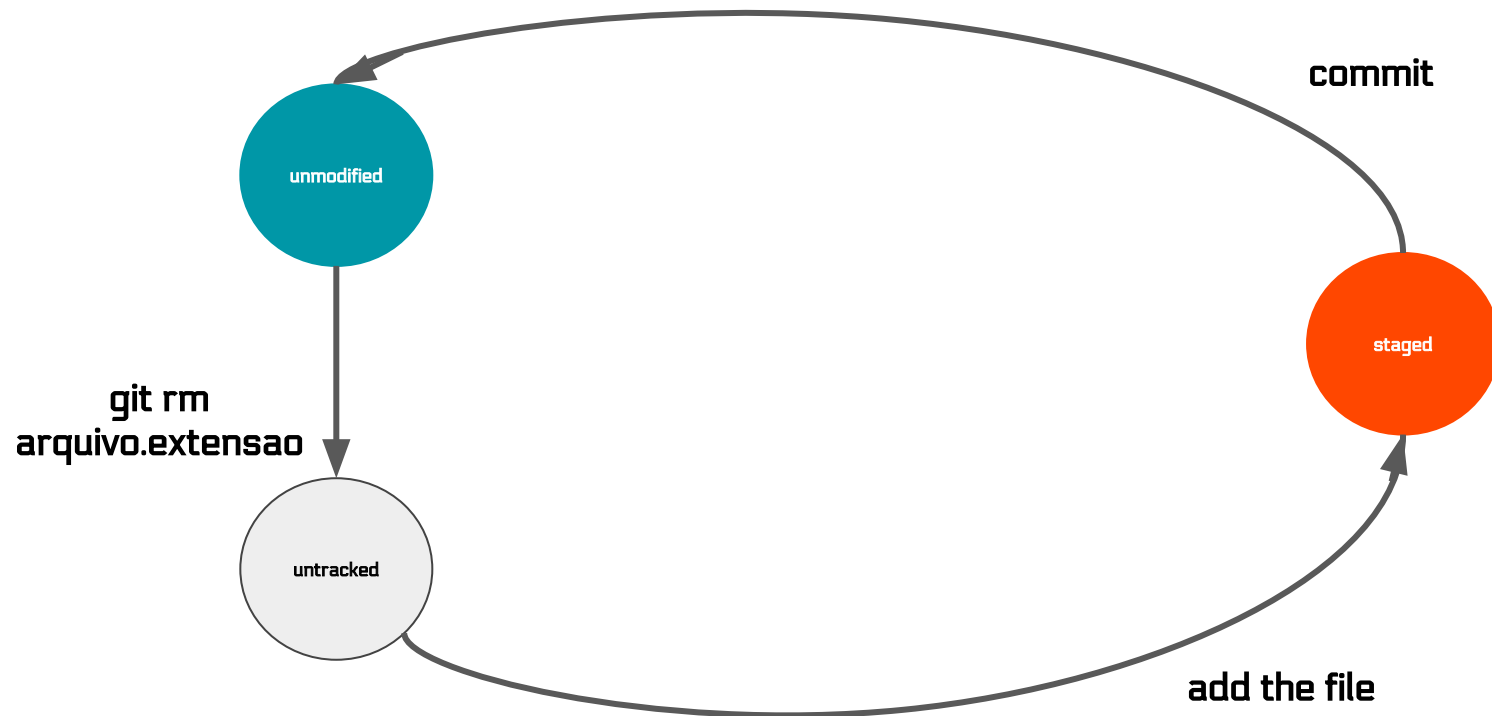
# Comandos Git

`git rm nome_arquivo.extensão`



remove o arquivo do  
versionamento e do diretório.

# Como faz? – Estados dos Arquivos





# Comandos Git

```
git mv arquivo_origem arquivo_destino
```



renomeia o arquivo e mantém  
o histórico de versionamento  
sobre o novo nome

# Comandos Git

git log



apresenta o histórico de  
versionamento

# Comandos Git

```
git log -p
```



apresenta o histórico de versionamento: com as diferenças de uma versão para outra.

# Comandos Git

```
git log -p -2
```



apresenta o histórico de versionamento: com as diferenças de uma versão para outra. [limite]

# Comandos Git

`git log --stat`



apresenta o histórico de  
versionamento: com o estado  
de cada versão

# Comandos Git

```
git log --pretty=oneline
```



apresenta o histórico de  
versionamento: de modo  
resumido a uma linha

# Comandos Git

```
git log --pretty=format
```



apresenta o histórico de  
versionamento: de modo  
resumido a uma linha

# Comandos Git

## git log --pretty=format

%h - hash curta

%an - nome do autor

%ae - e-mail do autor

%ad - data da autoria

%ar - data da autoria relativa

%cn - nome do autor commit

%ce - e-mail do autor commit

%cd - data da autoria commit

%s - conteúdo



# Comandos Git

`git log --since (ou --after)`

`--until (ou --before)`

`--author (autor)`

`--committer`

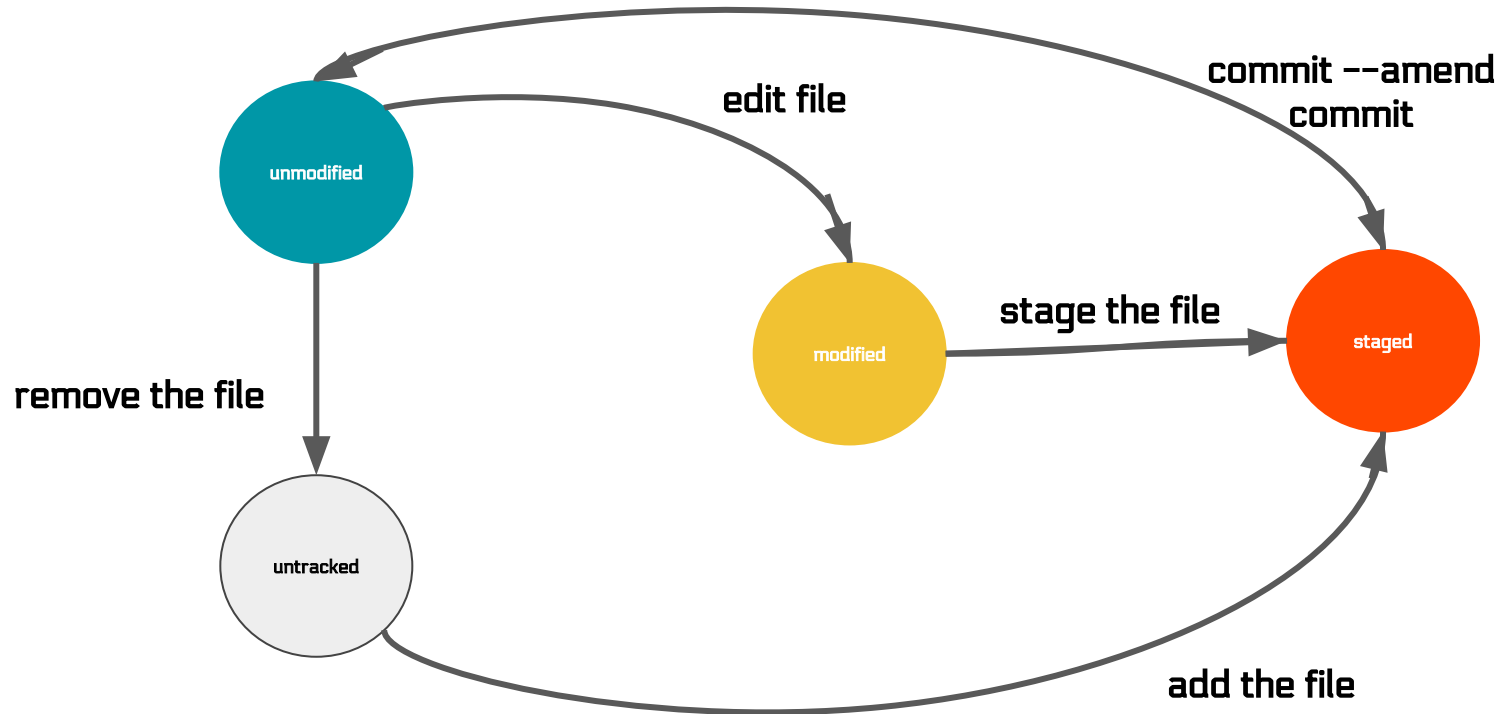
`--grep`

## git commit --amend



you forgot to add  
a file before versioning.  
it returns all files from  
commit to the stage area

# Estados dos Arquivos

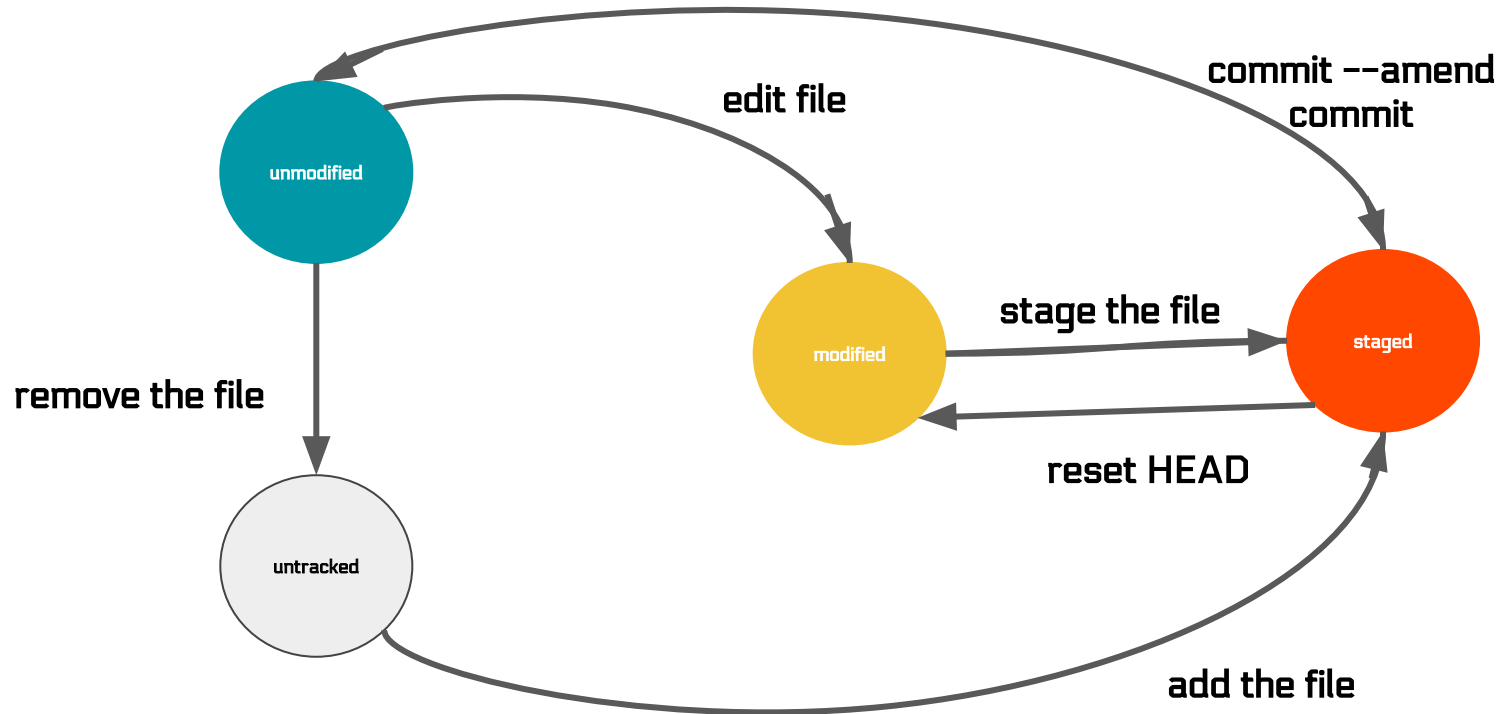


## git reset HEAD arquivo



retira o arquivo da área do palco e leva para modified

# Estados dos Arquivos

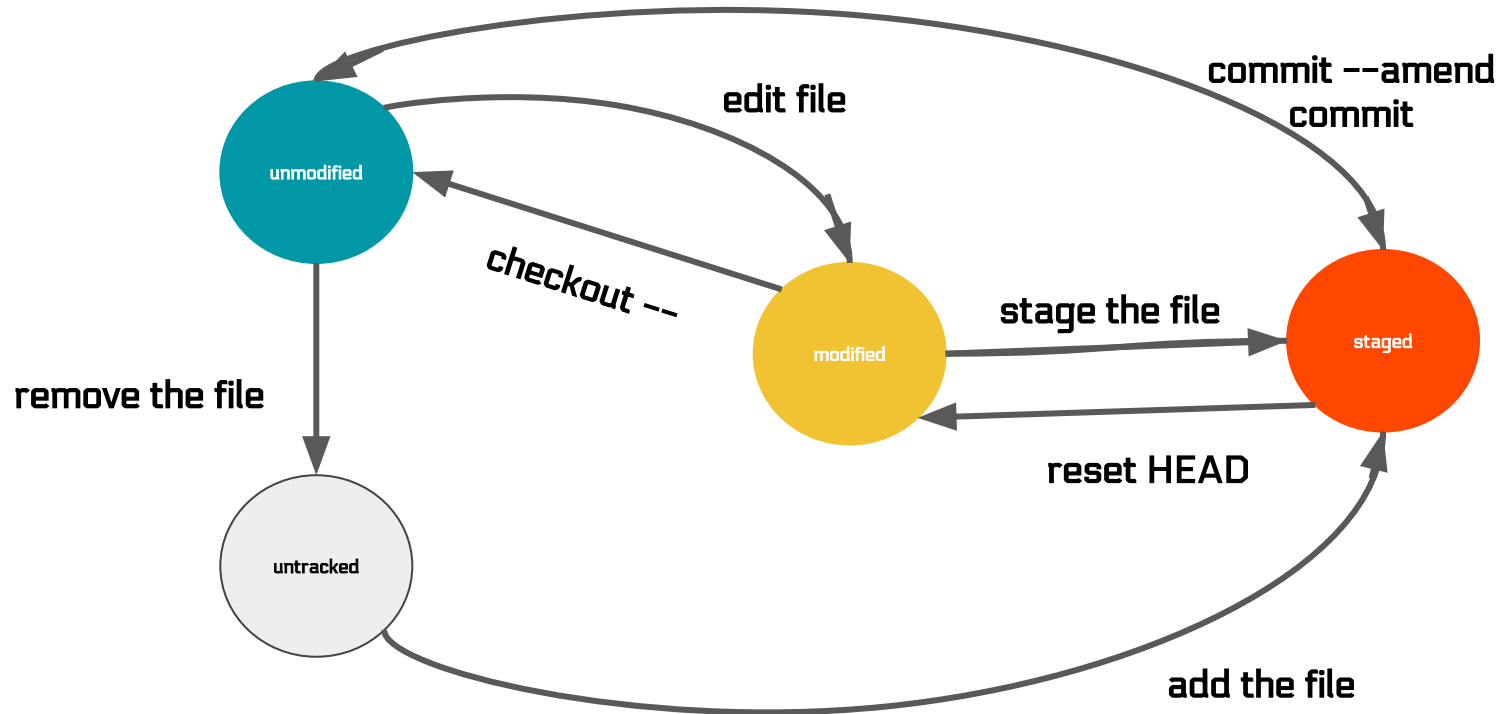


## git checkout -- arquivo



descarta todas as  
atualizações feitas no arquivo  
modified – voltando para  
unmodified.

# Estados dos Arquivos



## git checkout -- arquivo



descarta todas as  
atualizações feitas no arquivo  
modified – voltando para  
unmodified.



# Trabalhando repositórios



# Comandos Git

`git remote -v`



lista todos os repositórios  
remotos do versionamento  
com uma url curta

# Comandos Git

```
git remote add <apelido> <url>
```



adiciona um repositório  
remoto com um apelido para a  
url do repositório.

# Comandos Git

```
git remote show <remote>
```



mostra as novidades de um repositório remoto específico.

# Comandos Git

```
git fetch <remote>
```



descarrega todas as  
informações do repositório

# Comandos Git

```
git push <remote> <ramo>
```



carrega todas as informações  
de um ramo para o repositório