



git

uma ferramenta de
versionamento

Agenda

Versionar?

**Tipos de
Versionamento**

Como fazer?

Git





Versionar?

Meu primeiro programa

```
1 print( 'Meu primeiro app')
```

meu_app.py

Meu primeiro programa

```
1 print( 'Meu primeiro app')
```



meu_app.py

meu_app.py

Meu primeiro programa

```
1 print( 'Meu primeiro app')
```

meu_app.py



meu_app.py



Meu primeiro programa

```
1 print( 'Meu primeiro app')
```

meu_app.py



meu_app.py



Quantidade linhas:
1 linha

Tamanho do Arquivo:
2 Kbps

Conteúdo da linha:
print('Meu
primeiro app')

Meu primeiro programa

```
1 print( 'Meu primeiro app')
```

meu_app.py



meu_app.py

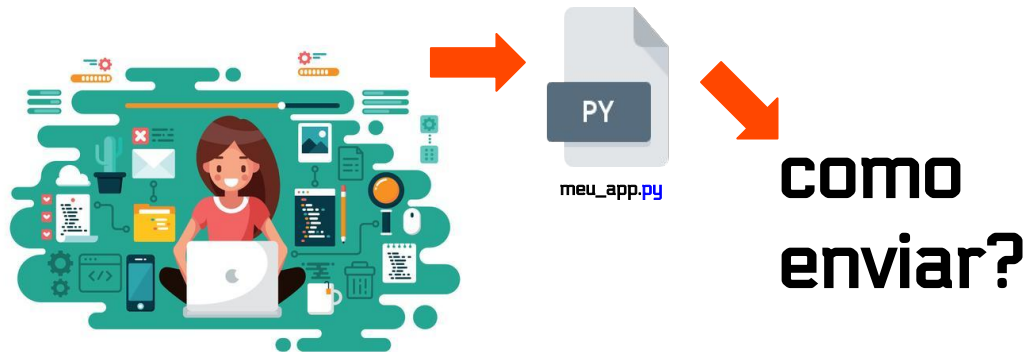


Quantidade linhas:
1 linha

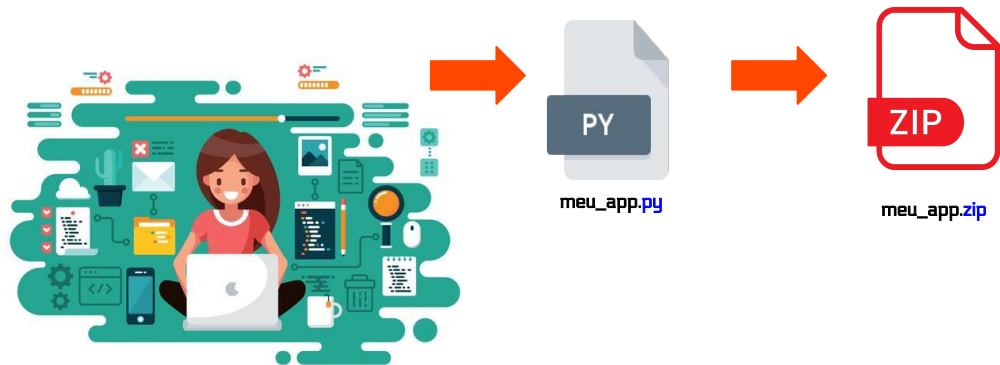
Tamanho do Arquivo:
2 Kbps

Conteúdo da linha:
print('Meu
primeiro app')

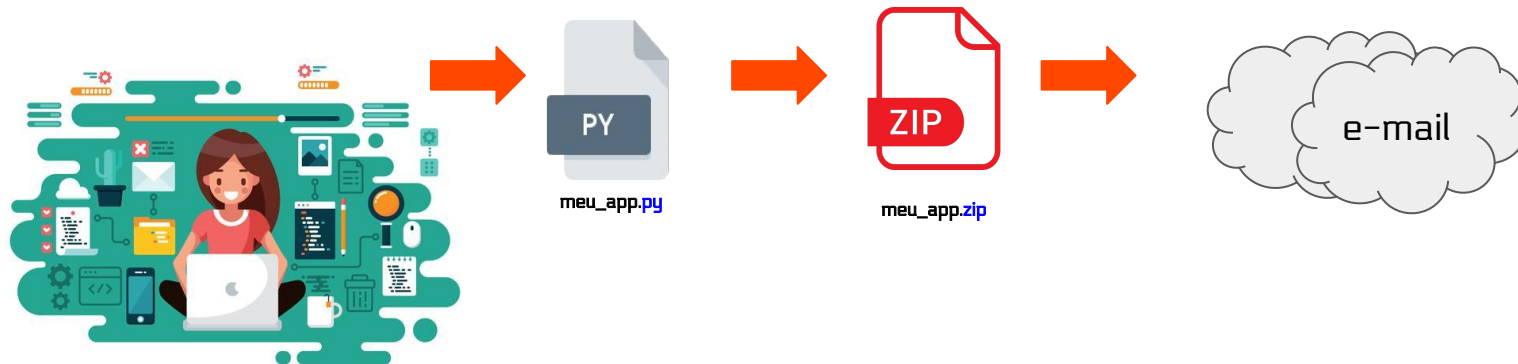
Meu primeiro programa



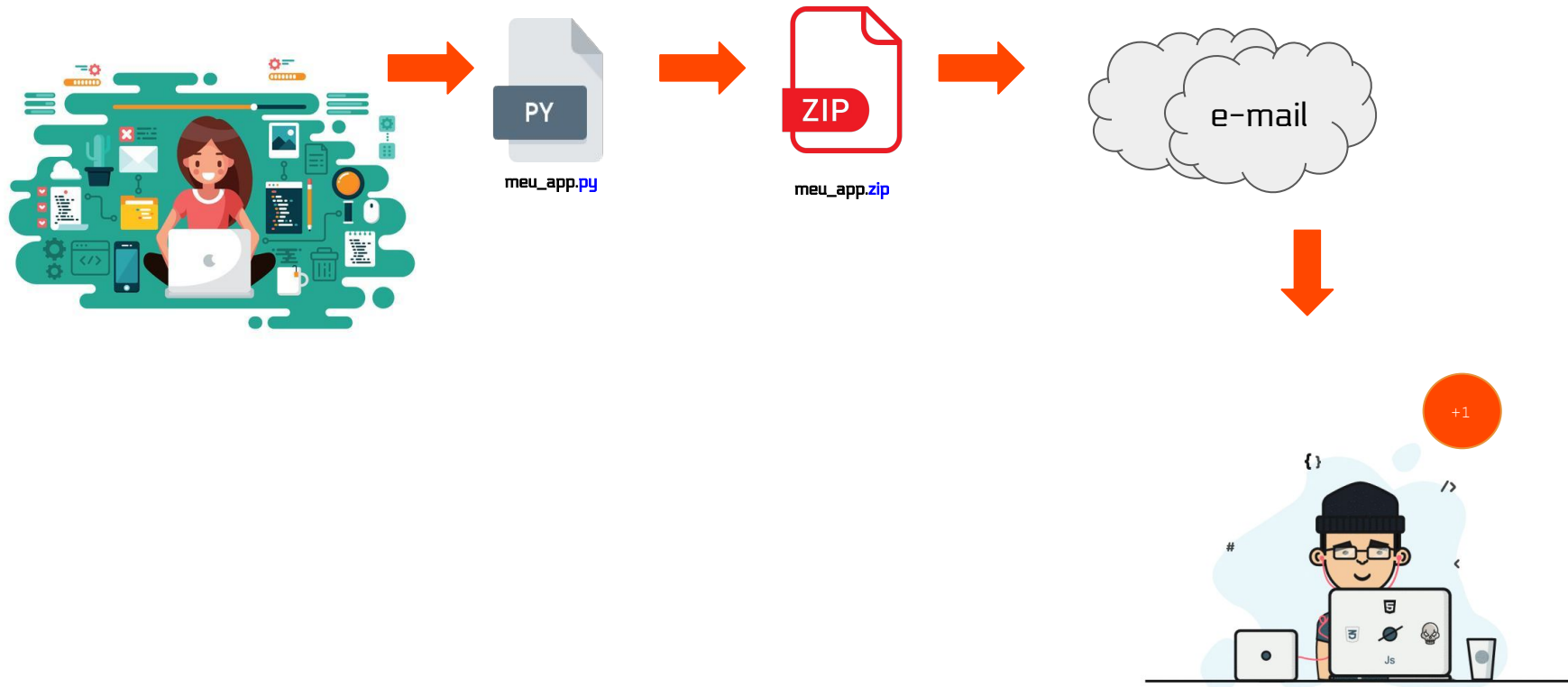
Meu primeiro programa



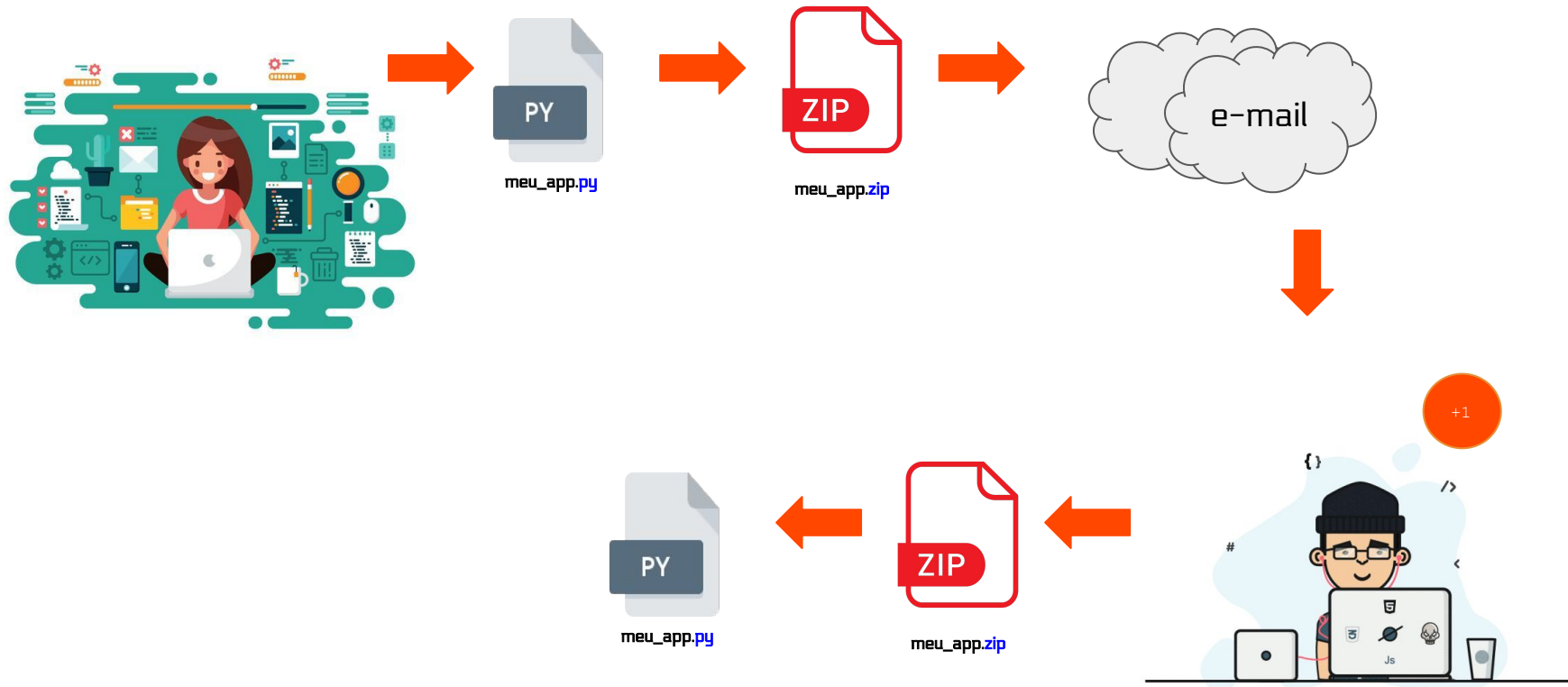
Meu primeiro programa



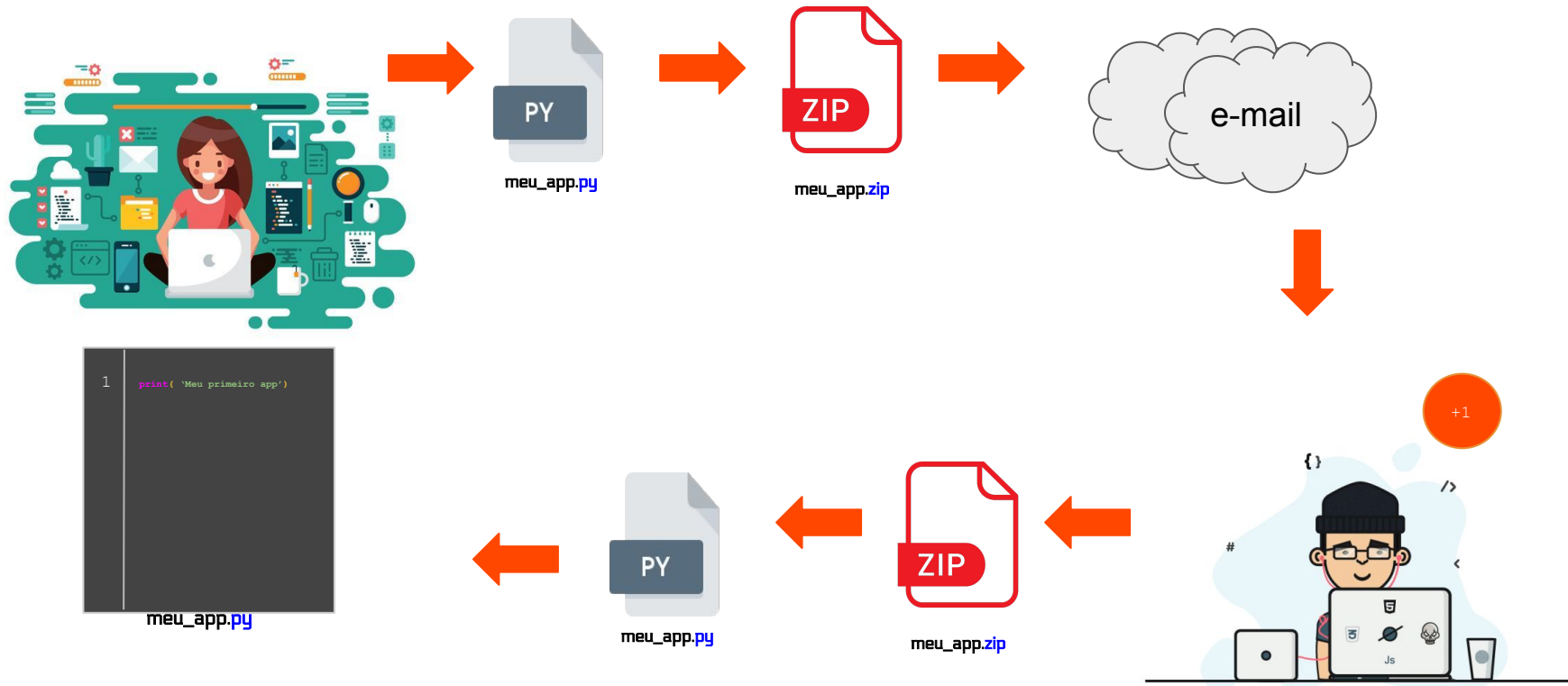
Meu primeiro programa



Meu primeiro programa



Meu primeiro programa



Meu primeiro programa

```
1 #nosso primeiro programa
2 print( 'Meu primeiro app')
3 v1 = input('Digite v1')
```

meu_app.py

Meu primeiro programa

```
1 #nosso primeiro programa
2 print( 'Meu primeiro app')
3 v1 = input('Digite v1')
```

meu_app.py



meu_app.py



Meu primeiro programa

```
1 #nosso primeiro programa
2 print( 'Meu primeiro app')
3 v1 = input('Digite v1')
```

meu_app.py



meu_app.py

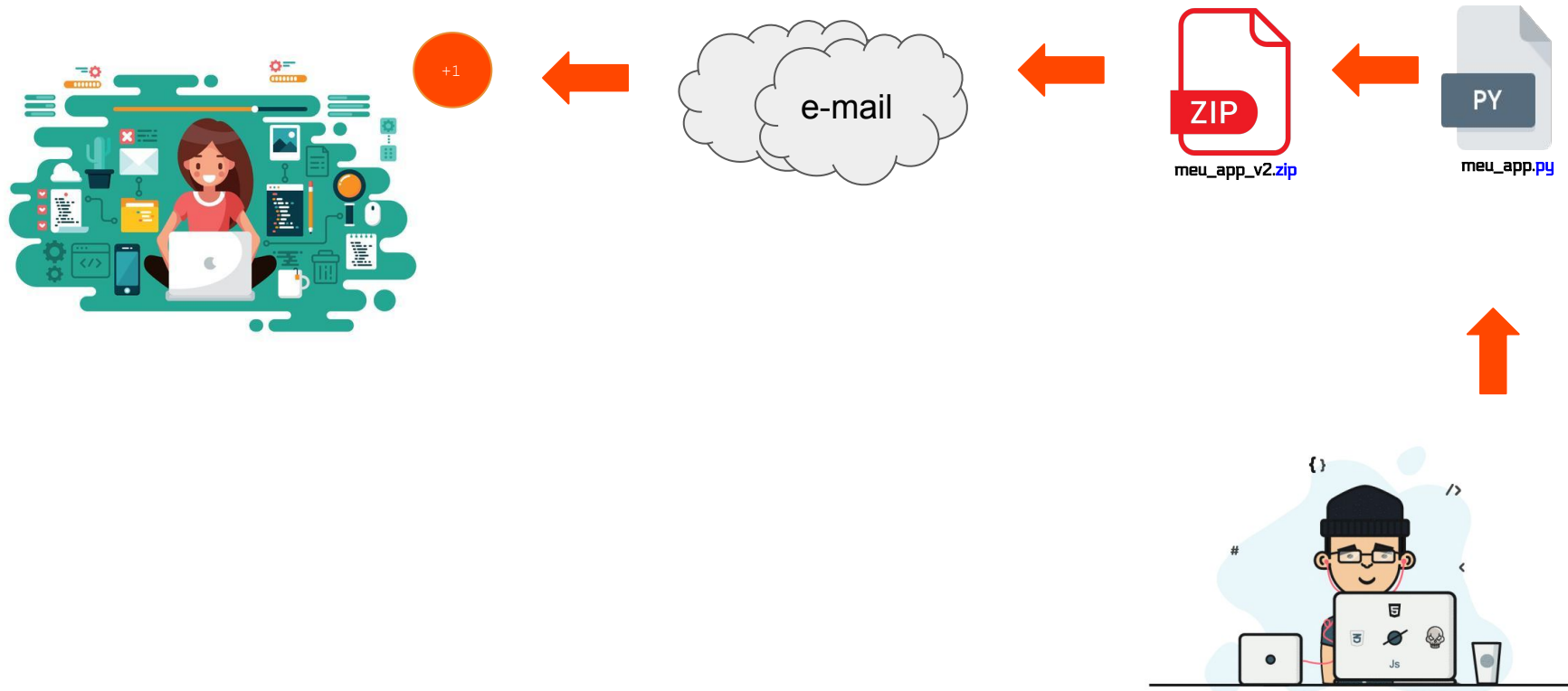


Quantidade linhas:
3 linha

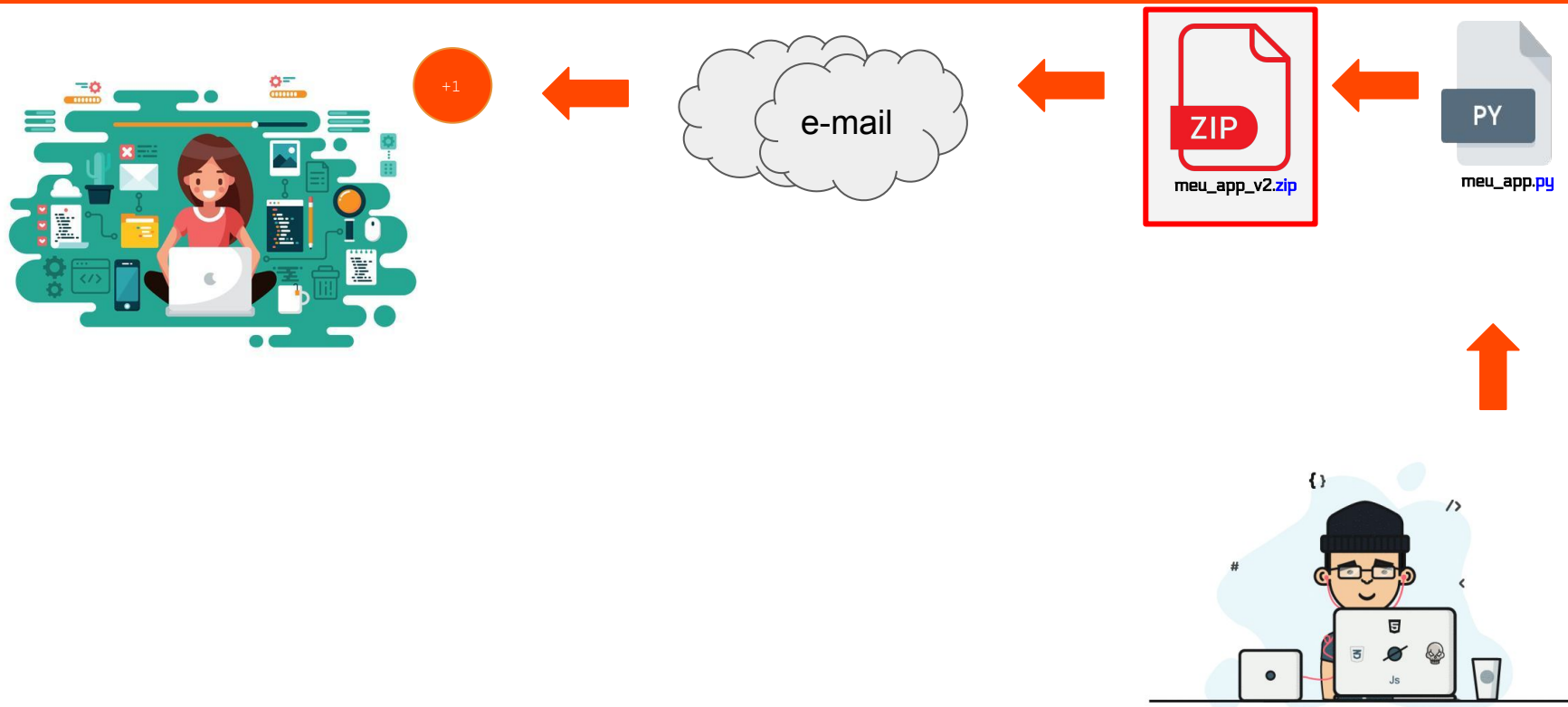
Tamanho do Arquivo:
2,8 Kbps

Conteúdo da linha:
#nosso primeiro programa
print('Meu primeiro app')
v1 = input('Digite v1')

Meu primeiro programa



Meu primeiro programa



Meu primeiro programa



meu_app_v2.zip



Meu primeiro programa



Meu primeiro programa



```
1 print( 'Meu primeiro app')
```

meu_app.py



```
1 #nosso primeiro programa  
2 print( 'Meu primeiro app')  
3 v1 = input('Digite v1')
```

meu_app.py

Meu primeiro programa



mas e se ela continuar
mudando o código
enquanto ele estiver
fazendo?

```
1 print( 'Meu primeiro app')
2 v2 = input( 'Digite o ano')
```

meu_app.py



```
1 #nosso primeiro programa
2 print( 'Meu primeiro app')
3 v1 = input('Digite v1')
```

meu_app.py

Meu primeiro programa



vai se adaptar

```
1 print( 'Meu primeiro app')
2 v2 = input( 'Digite o ano')
```

meu_app.py



```
1 #nosso primeiro programa
2 print( 'Meu primeiro app')
3 v1 = input('Digite v1')
4 v2 = input( 'Digite o ano')
```

meu_app.py

Meu primeiro programa



meu_app.zip



meu_app_v2.zip



meu_app_v3.zip

```
1 #nosso primeiro programa
2 print( 'Meu primeiro app')
3 v1 = input('Digite v1')
4 v2 = input( 'Digite o ano')
```

meu_app.py

Meu primeiro programa



```
1 #nosso primeiro programa
2 print( 'Meu primeiro app')
3 v1 = input('Digite v1')
4 v2 = input( 'Digite o ano')
```

meu_app.py

Meu primeiro programa



meu_app.zip



meu_app_v2.zip



meu_app_v3.zip



meu_app_v4.zip



meu_app_v4_1.zip

Meu primeiro programa



meu_app.zip



meu_app_v2.zip



meu_app_v3.zip



meu_app_v4.zip



meu_app_v4_1.zip

**e se eu tiver que voltar uma parte do código para
como estava escrito na versão 2, mas manter todo
o restante?**

Meu primeiro programa



meu_app.zip



meu_app_v2.zip



meu_app_v3.zip



meu_app_v4.zip



meu_app_v4_1.zip

você teria que fazer na mão.

Meu primeiro programa



meu_app.zip



meu_app_v2.zip



meu_app_v3.zip



meu_app_v4.zip



meu_app_v4_1.zip

```
1 print( 'Meu primeiro app')
2 v2 = input( 'Digite o ano')
3 v3 = v2 * 0.1
4 if v3 > 10:
5     print( 'O valor > 100' )
6 elif v3 > 5:
7     print( '50 < valor
8         < 100' )
9 else:
10    print( 'Não sabemos' )
```

meu_app_v4.py



```
1 #nosso primeiro programa
2 print( 'Meu primeiro app')
3 v1 = input('Digite v1')
4 v2 = input( 'Digite o ano')
```

meu_app_v2.py

Meu primeiro programa



meu_app.zip



meu_app_v2.zip



meu_app_v3.zip



meu_app_v4.zip



meu_app_v4_1.zip

```
1 #nosso primeiro programa
2 print( 'Meu primeiro app')
3 v1 = input('Digite v1')
4 v2 = input( 'Digite o ano')
5 v3 = v2 * 0.1
6 if v3 > 10:
7     print( 'O valor > 100' )
8 elif v3 > 5:
9     print( '50 < valor
10             < 100' )
11 else:
12     print( 'Não sabemos' )
```

meu_app_v4_1.py



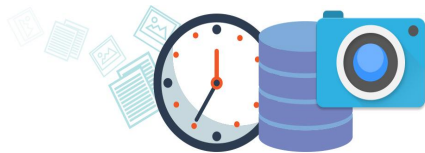
```
1 #nosso primeiro programa
2 print( 'Meu primeiro app')
3 v1 = input('Digite v1')
4 v2 = input( 'Digite o ano')
```

meu_app_v2.py

Tipos de Versionamento



Tipos de Versionamento



snapshots

cópias inteiras dos arquivos
versionados

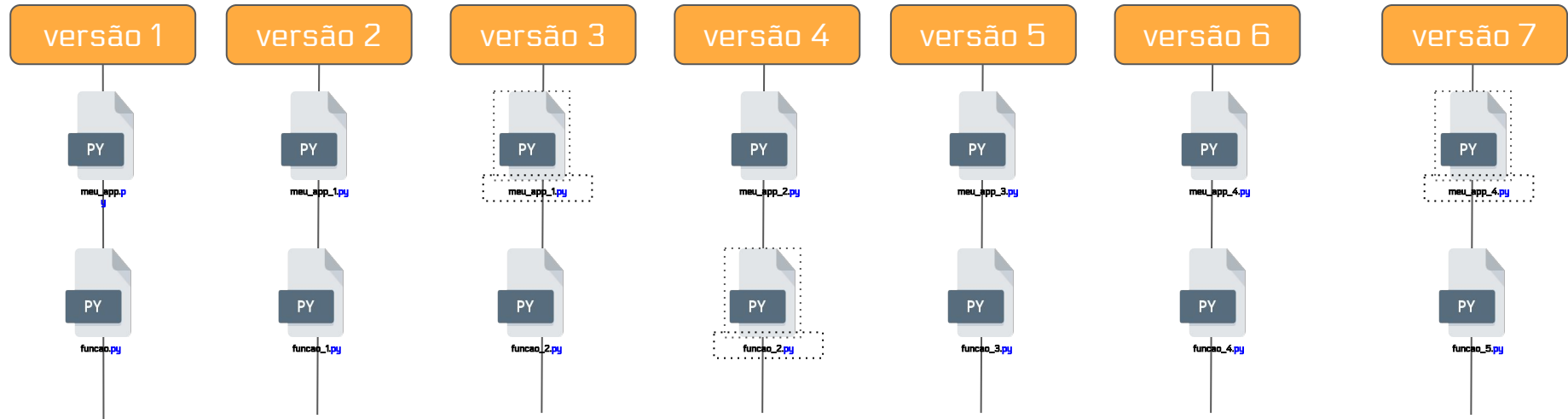


delta differences

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

Tipos de Versionamento

versões ao longo do tempo



Tipos de Versionamento



delta differences

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

```
1 print( 'Meu primeiro app')
```

`meu_app.py`



```
1 print( 'Meu primeiro app')  
2 v1 = input('Digite v1')
```

`meu_app.py`

Tipos de Versionamento

```
1 print( 'Meu primeiro app')
```



```
1 print( 'Meu primeiro app')  
2 v1 = input('Digite v1')
```



meu_app.py

meu_app_v2.py

Tipos de Versionamento

```
1 print( 'Meu primeiro app')
```



```
1 print( 'Meu primeiro app')  
2 v1 = input('Digite v1')
```

meu_app.py

meu_app_v2.py



- + linha 1:
- + `v1 = input('Digite v1')`

Tipos de Versionamento

```
1 print( 'Meu primeiro app')
```



```
1 print( 'Meu primeiro app')  
2 v1 = input('Digite v1')
```

meu_app.py

meu_app_v2.py



- + linha 1:
- + v1 = input('Digite v1')

Tipos de Versionamento

```
1 print( 'Meu primeiro app')
```

meu_app.py



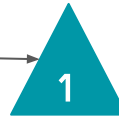
```
1 print( 'Meu primeiro app')
2 v1 = input('Digite v1')
```

meu_app_v2.py

versões ao longo do tempo

versão 1

versão 2



Tipos de Versionamento

```
1 print( 'Meu primeiro app')
```

meu_app.py



```
1 print( 'Meu primeiro app')
2 v1 = input('Digite v1')
```

meu_app_v2.py

versões ao longo do tempo

versão 1

versão 2



e se tivéssemos
outros arquivos?

Tipos de Versionamento

```
1 def imprimir(usuario):  
2     print( usuario )
```



```
1 def imprimir(usuario):  
2     print( usuario )  
3     print( usuario['id'] )
```

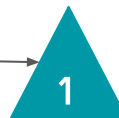
funcao.py

funcao_v2.py

versões ao longo do tempo

versão 1

versão 2



e se tivéssemos
outros arquivos?

Tipos de Versionamento

```
1 def imprimir(usuario):  
2     print( usuario )
```



```
1 def imprimir(usuario):  
2     print( usuario )  
3     print( usuario['id'] )
```

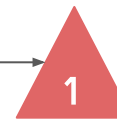
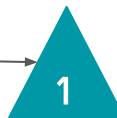
funcao.py

funcao_v2.py

versões ao longo do tempo

versão 1

versão 2



Tipos de Versionamento

```
1 def imprimir(usuario):  
2     print( usuario )
```



```
1 def imprimir(usuario):  
2     print( usuario )  
3     print( usuario['id'] )
```

funcao.py

funcao_v2.py

versões ao longo do tempo

versão 1

versão 2

versão 3



1



1

Tipos de Versionamento

```
1 def imprimir(usuario):  
2     print( usuario )  
3     print( usuario['id'] )
```

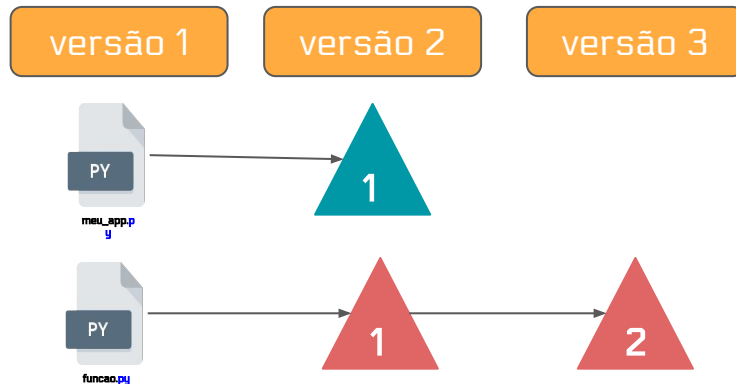


```
1 def imprimir(usuario):  
2     print( usuario )  
3     print( usuario['id'] )  
4     print( usuario['aNas'] )
```

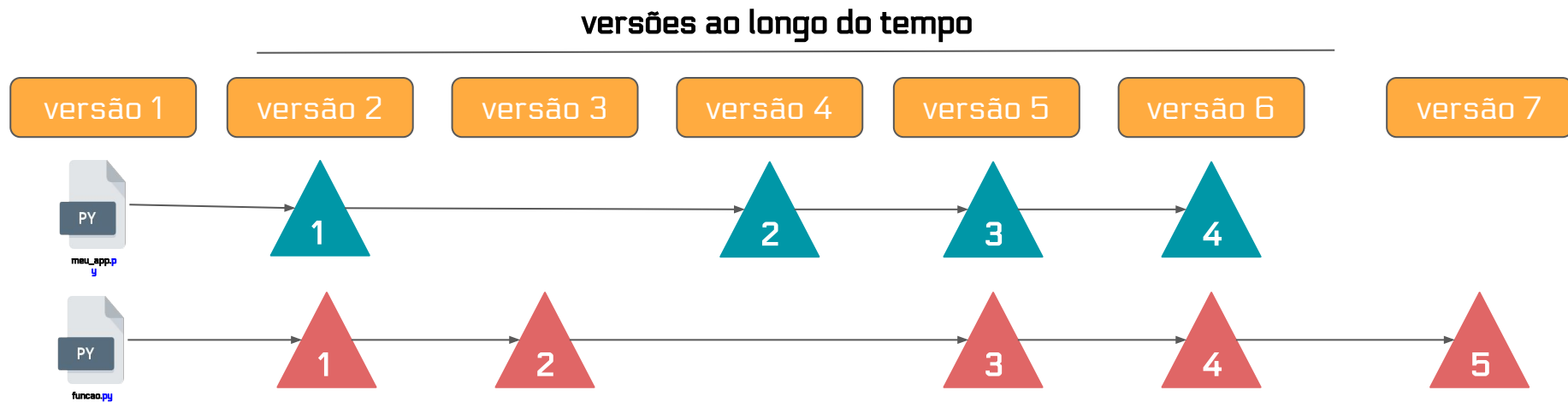
funcao_v2.py

funcao_v3.py

versões ao longo do tempo

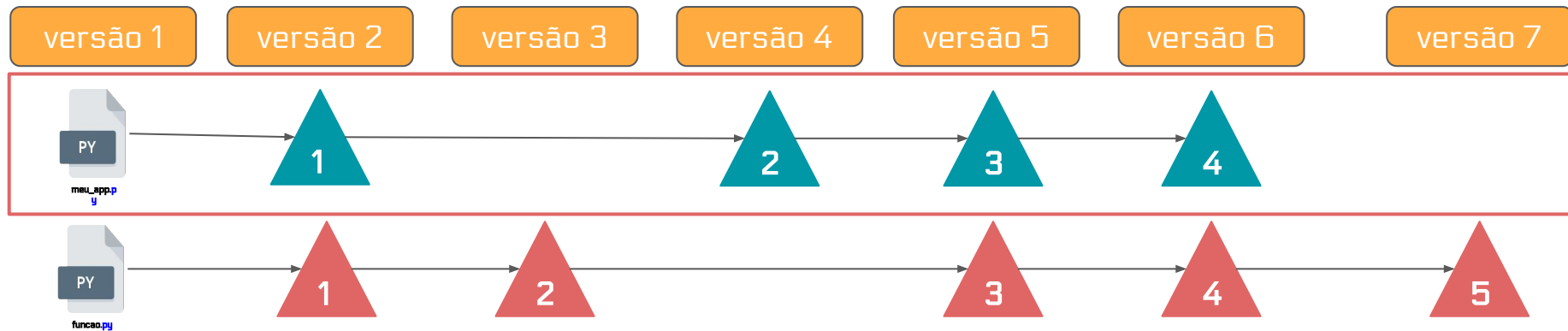


Tipos de Versionamento

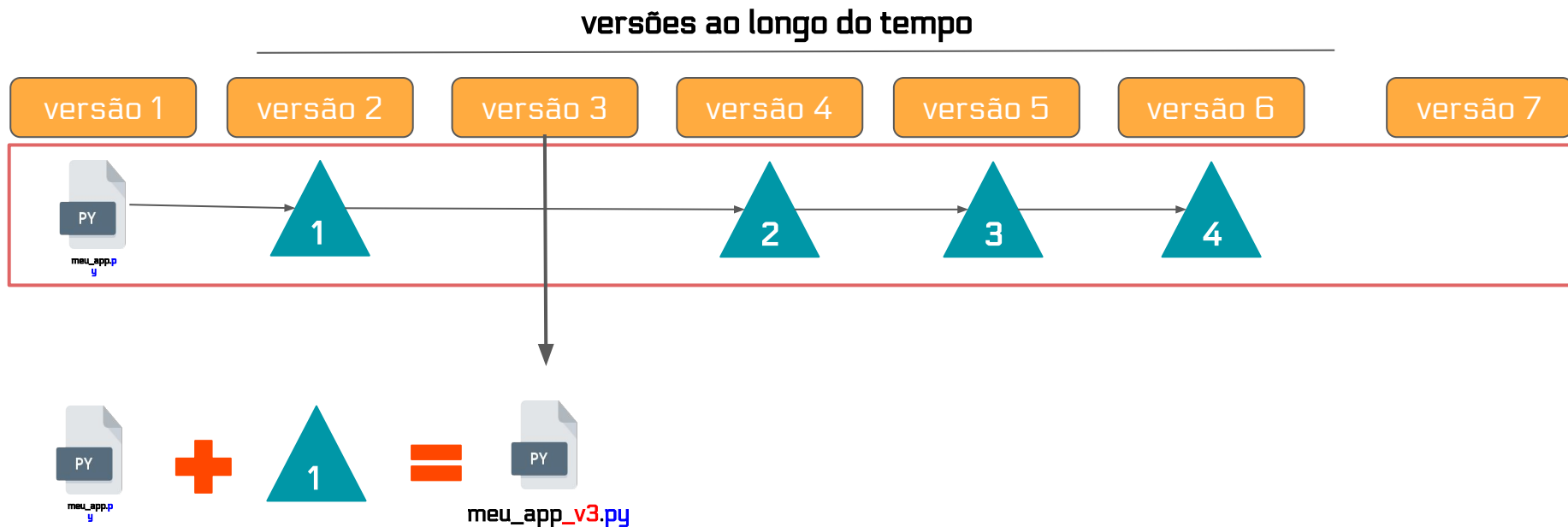


Tipos de Versionamento

versões ao longo do tempo

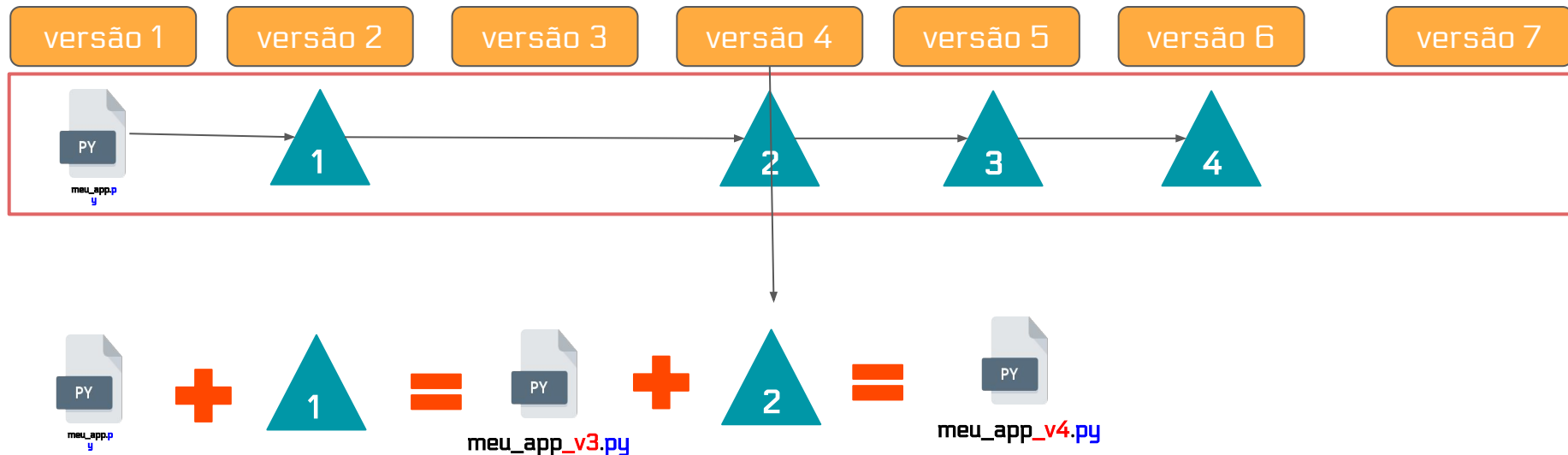


Tipos de Versionamento



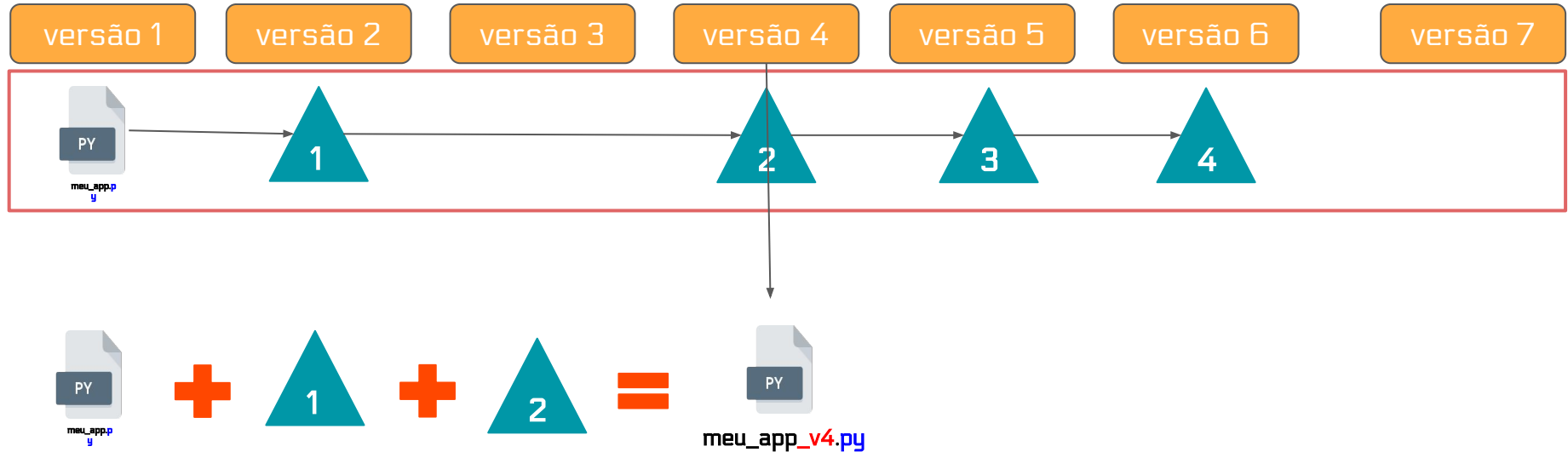
Tipos de Versionamento

versões ao longo do tempo



Tipos de Versionamento

versões ao longo do tempo



Tipos de Versionamento



snapshots

cópias inteiras dos arquivos
versionados



delta differences

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

Tipos de Versionamento



snapshots

cópias inteiras dos arquivos
versionados

```
1 print( 'Meu primeiro app')
```

meu_app.py



```
1 print( 'Meu primeiro app')  
2 v1 = input('Digite v1')
```

meu_app.py

Tipos de Versionamento

```
1 print( 'Meu primeiro app')
```



```
1 print( 'Meu primeiro app')  
2 v1 = input('Digite v1')
```

meu_app.py

meu_app_v2.py

versões ao longo do tempo

versão 1

versão 2



Tipos de Versionamento

```
1 def imprimir(usuario):  
2     print( usuario )
```



```
1 def imprimir(usuario):  
2     print( usuario )  
3     print( usuario['id'] )
```

funcao.py

funcao_v2.py

versões ao longo do tempo

versão 1



meu_app.py

PY

funcao.py

versão 2

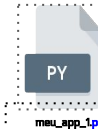


meu_app_1.py

PY

funcao_1.py

versão 3



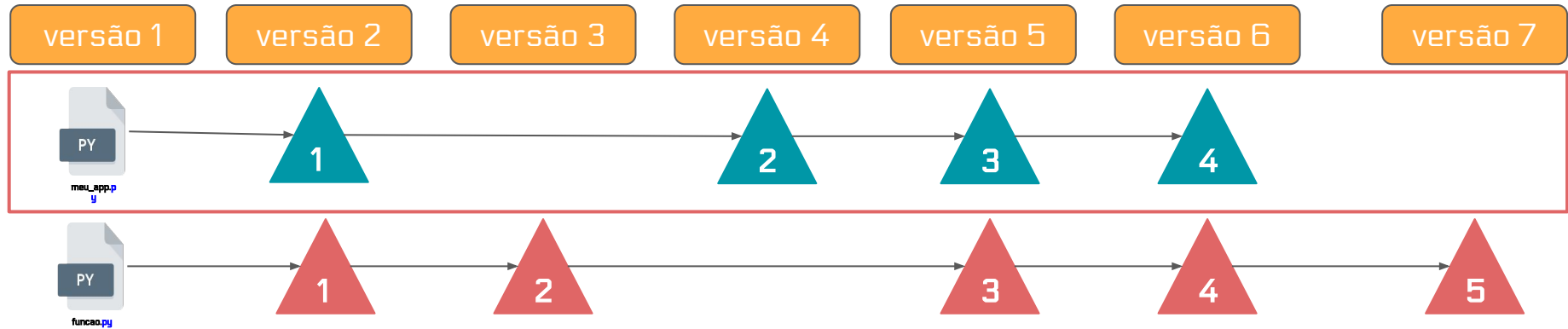
meu_app_1.py

PY

funcao_2.py

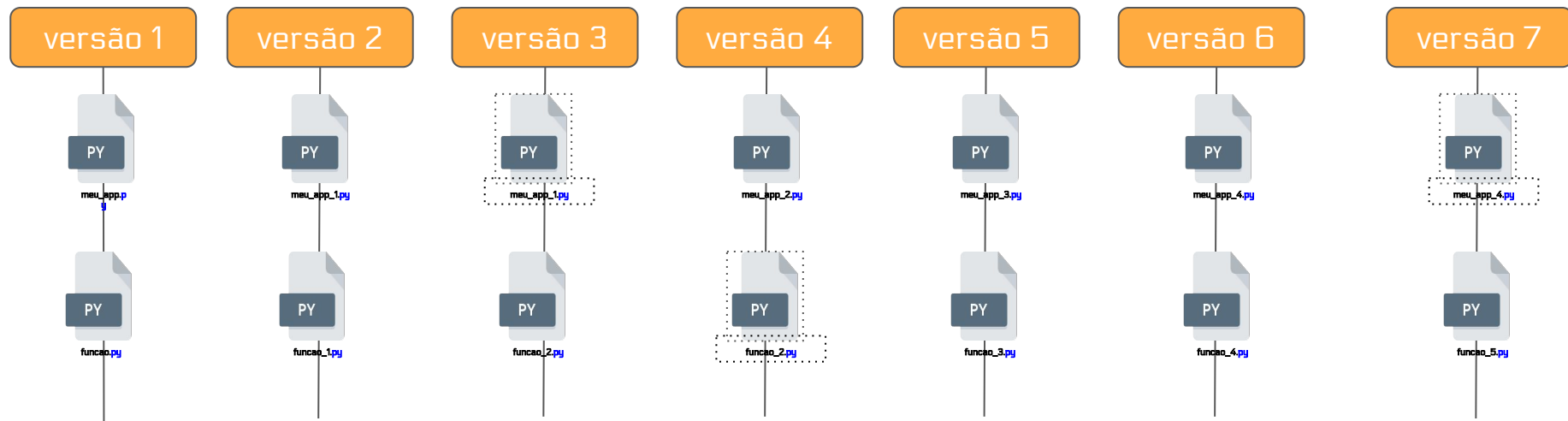
Tipos de Versionamento

versões ao longo do tempo



Tipos de Versionamento

versões ao longo do tempo





Como faz?

Como faz? – Git Área



Diretório onde estão os arquivos que estão sendo trabalhados.

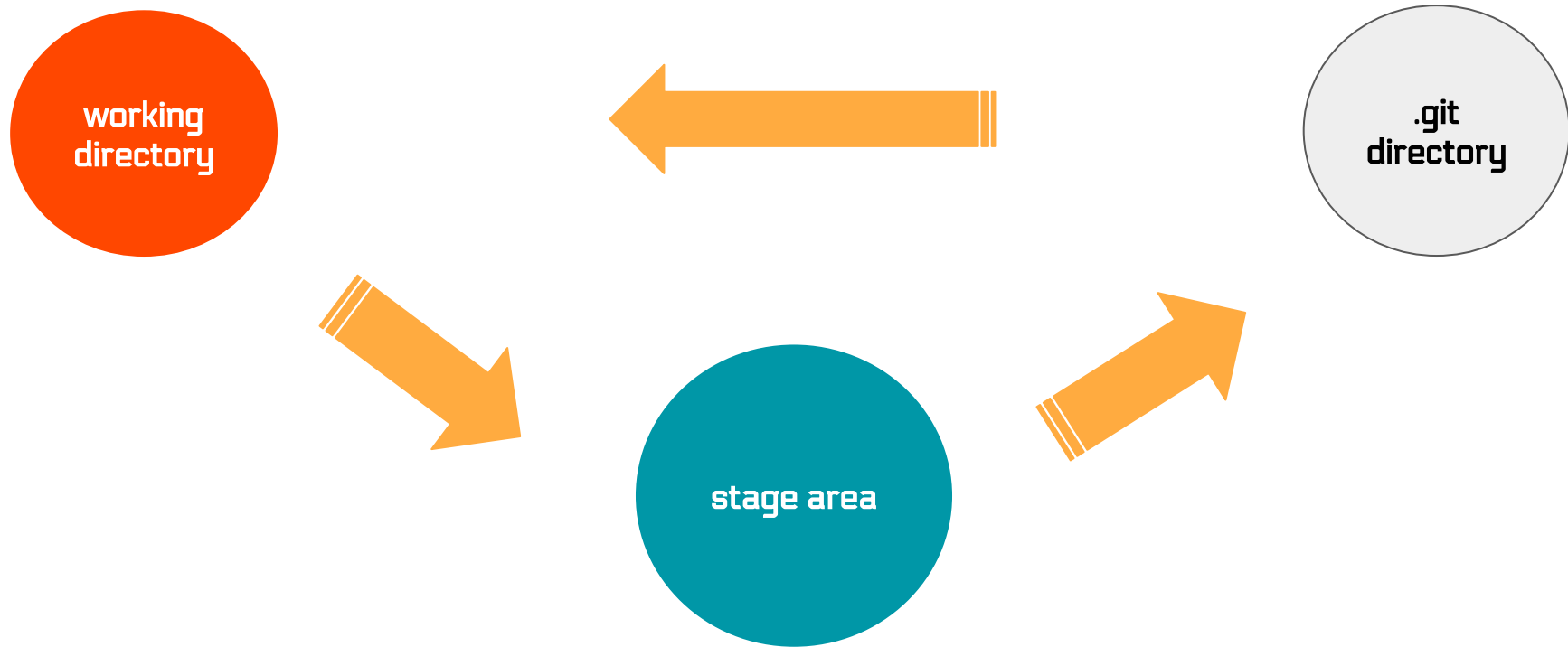


Área onde estão os arquivos que já foram trabalhados e se deseja versionar.

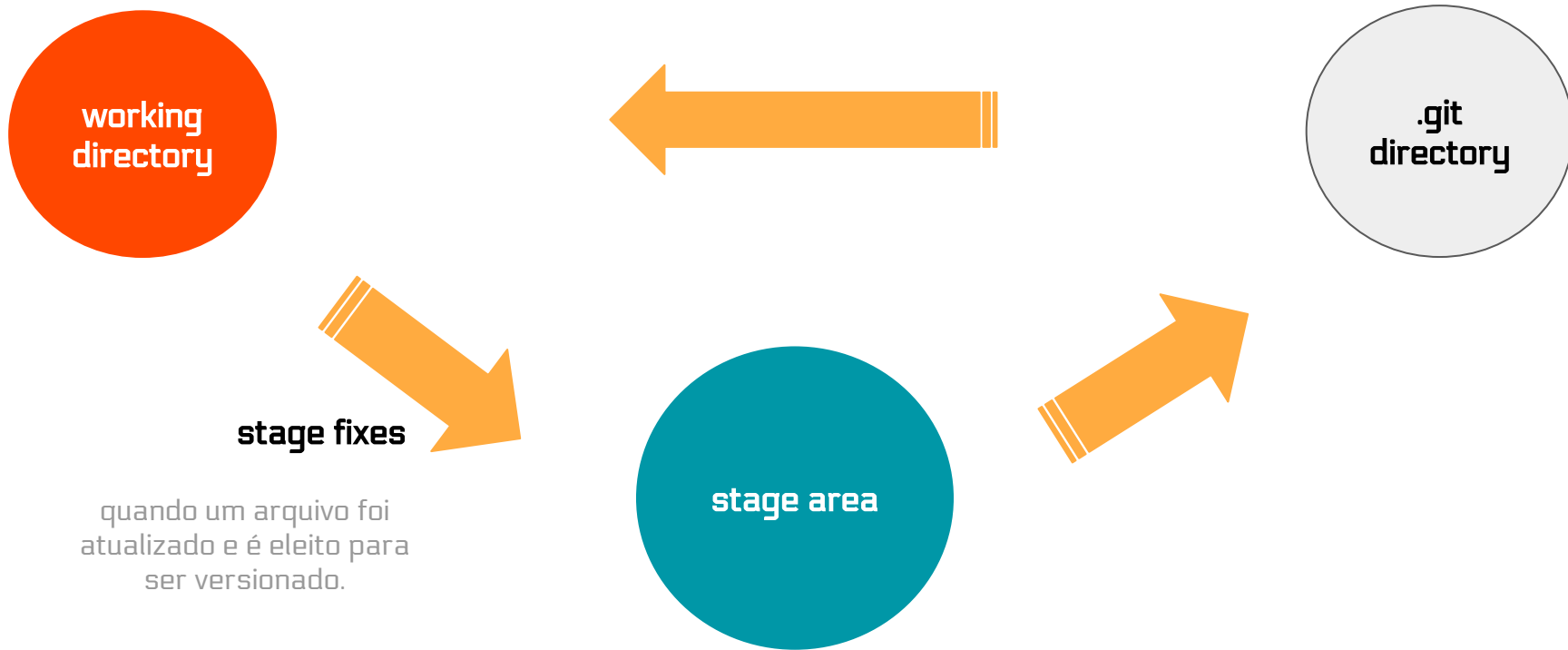


Diretório onde estão os arquivos versionados.

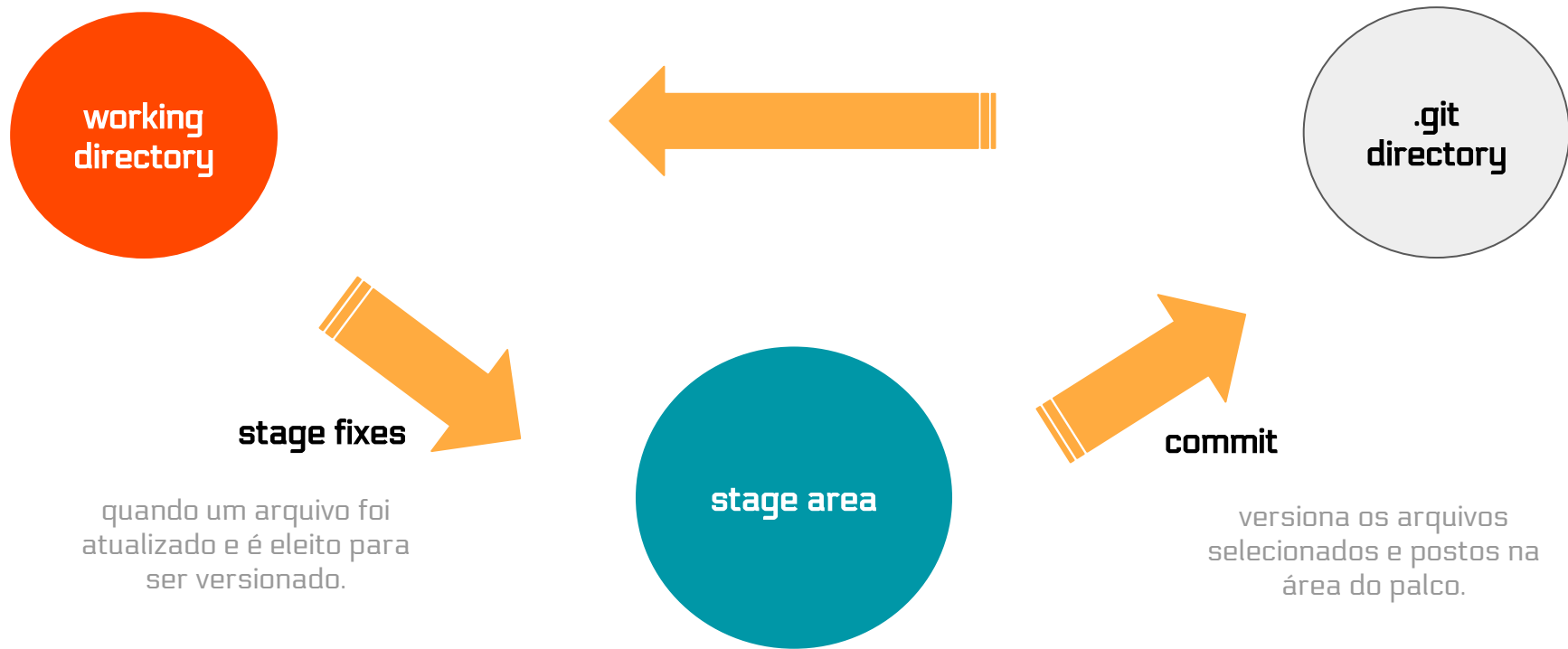
Como faz? – Git Área



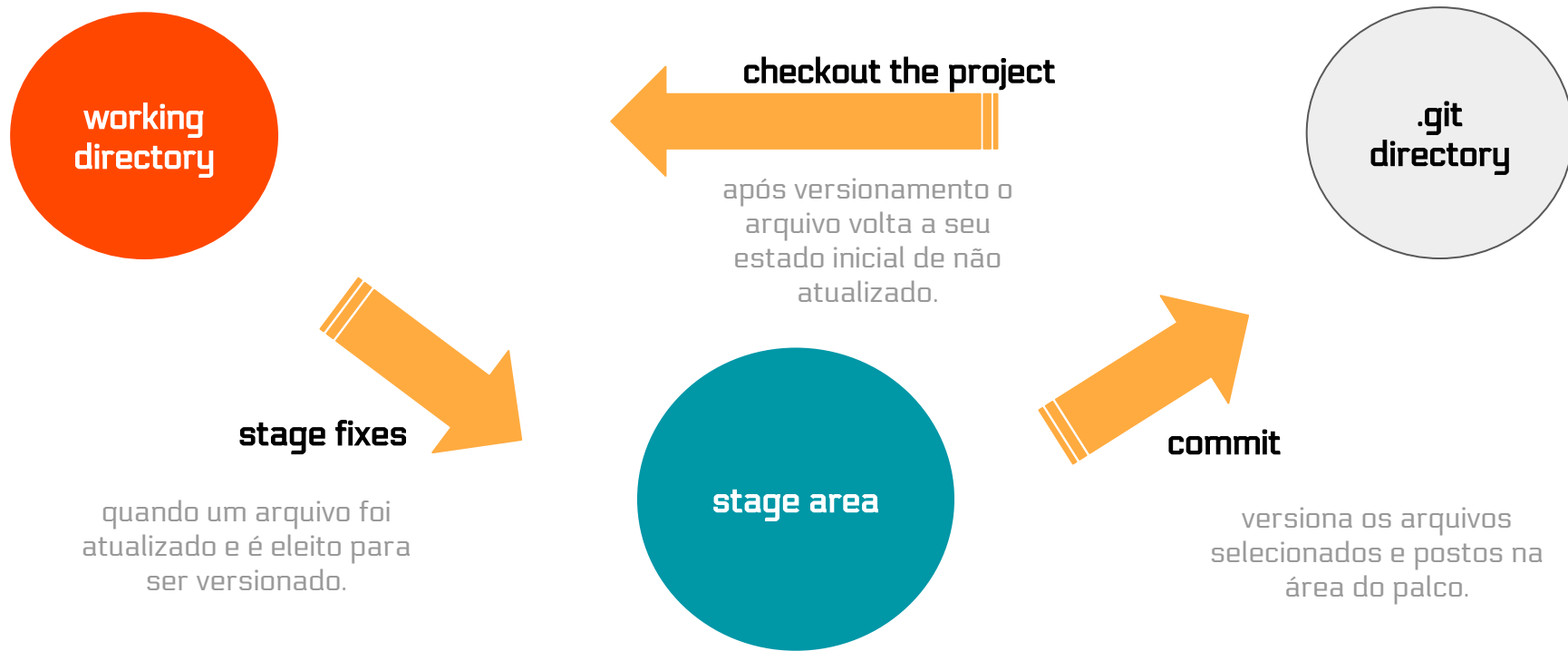
Como faz? – Git Área



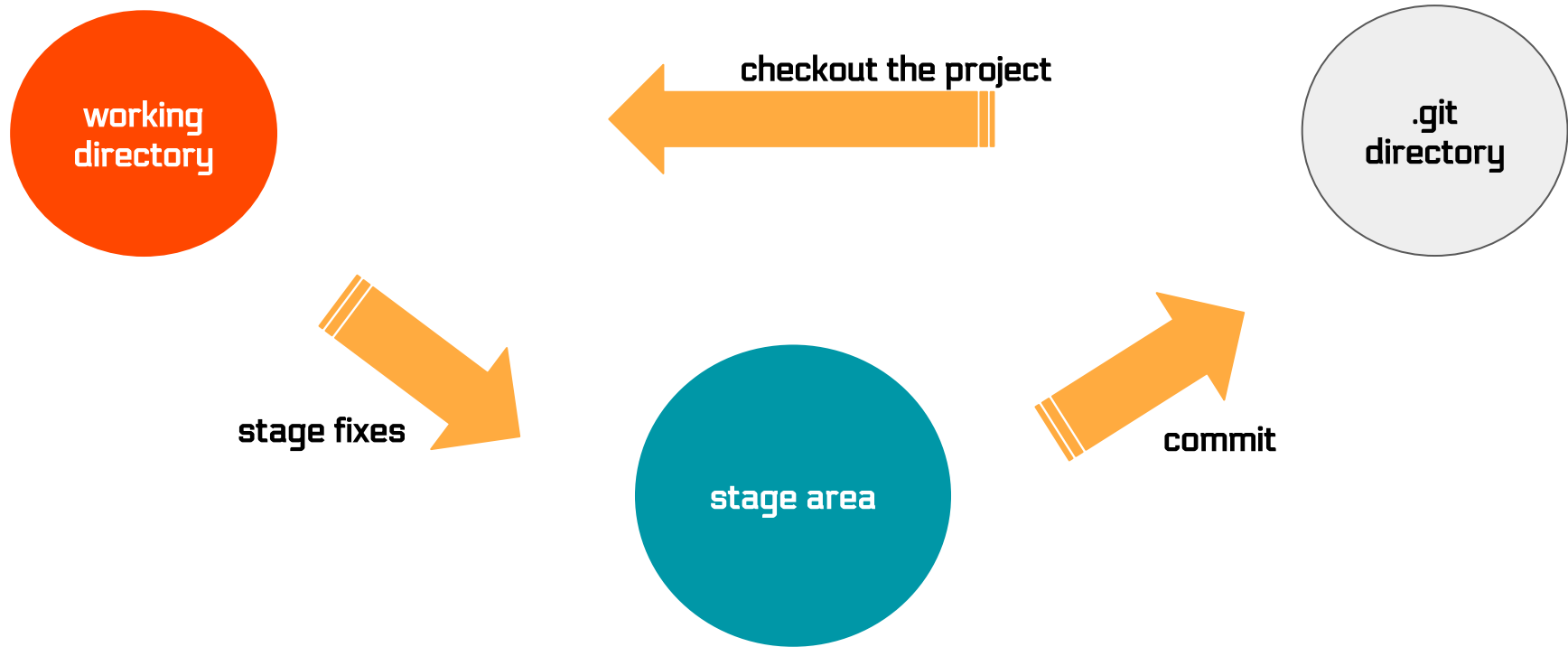
Como faz? – Git Área



Como faz? – Git Área



Como faz? – Git Área



Como faz? – Git Área



working
directory



+ Novo Arquivo

Como faz? – Git Área

working
directory

+ Novo Arquivo

1

novo_arquivo.py

Como faz? – Git Área

working
directory

+ Novo Arquivo

```
1 def imprimir(usuario):  
2     print( usuario )
```

novo_arquivo.py

Como faz? – Git Área

working
directory

+ Novo Arquivo

```
1 def imprimir(usuario):  
2     print( usuario )
```

novo_arquivo.py



Como faz? – Git Área

working
directory

+ Novo Arquivo

```
1 def imprimir(usuario):  
2     print( usuario )
```

novo_arquivo.py

add in stage
area



stage fixes

Como faz? – Git Área

working
directory

+ Novo Arquivo

```
1 def imprimir(usuario):  
2     print( usuario )
```

novo_arquivo.py

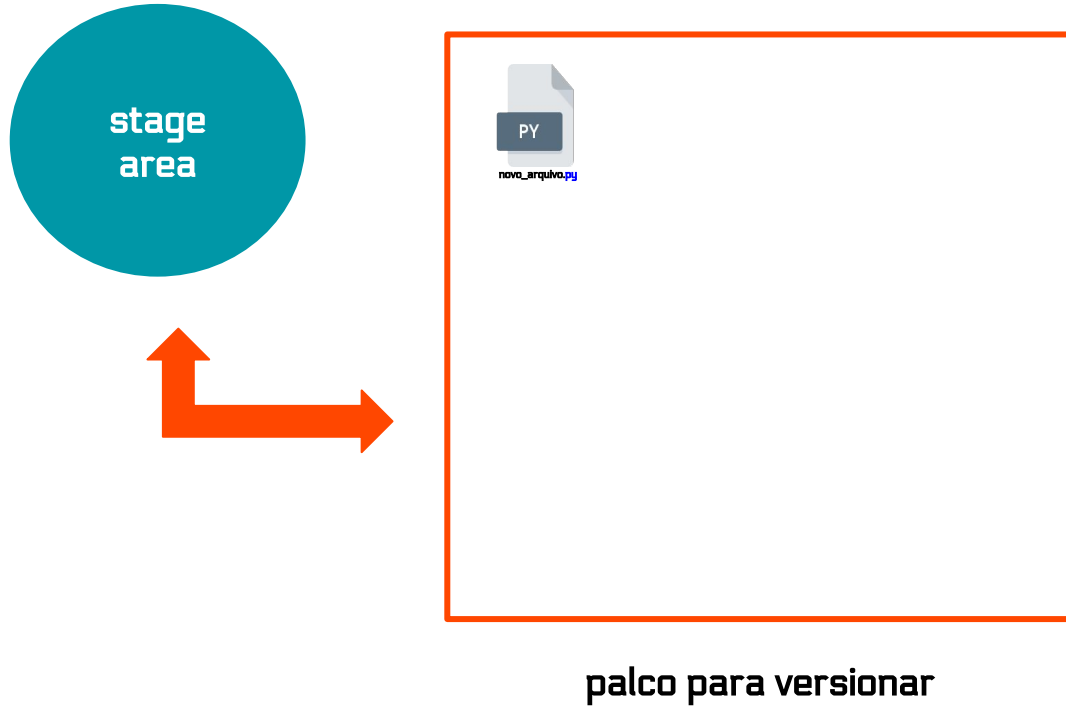
add in stage
area



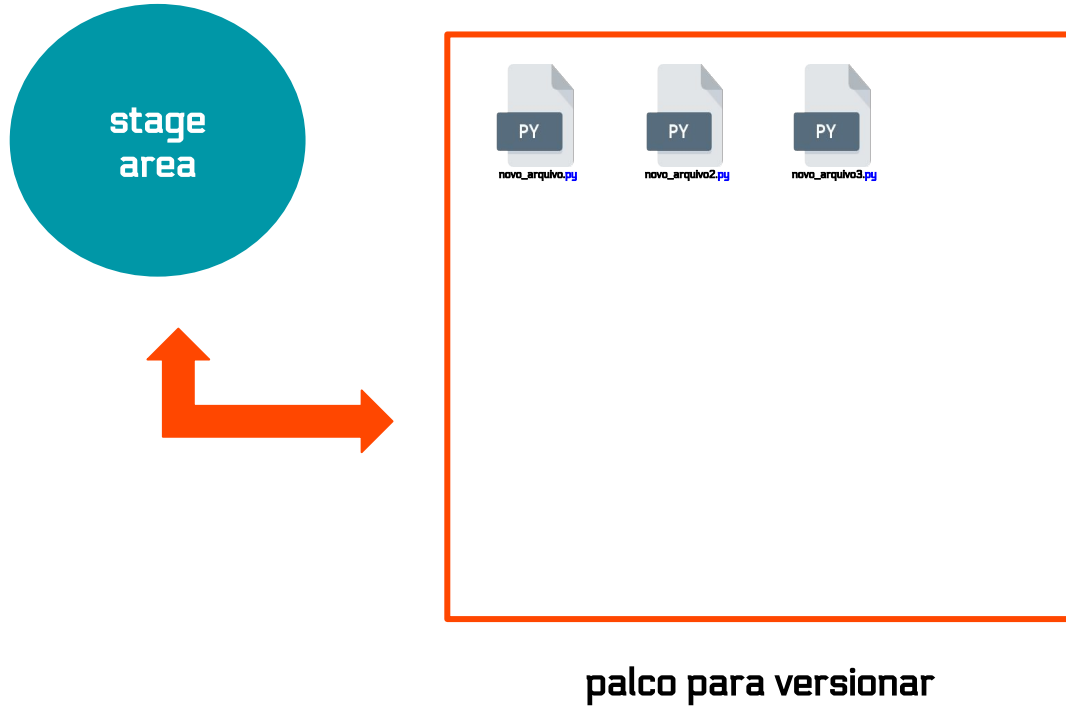
stage fixes

stage
area

Como faz? – Git Área



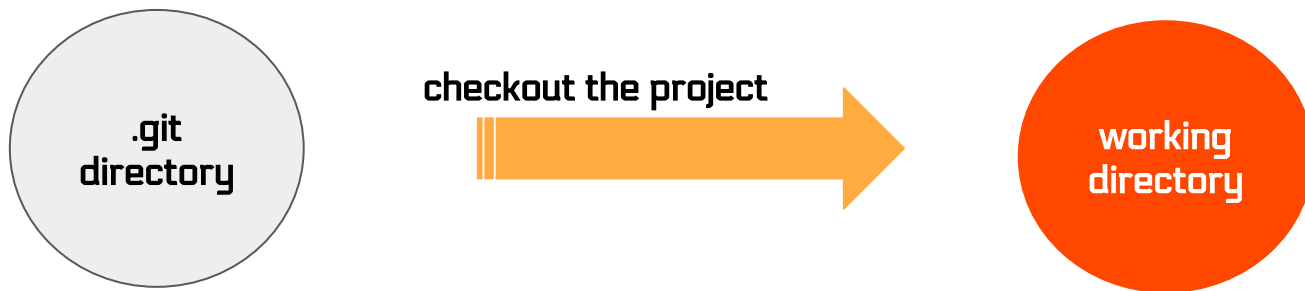
Como faz? – Git Área



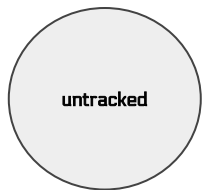
Como faz? – Git Área



Como faz? – Git Área



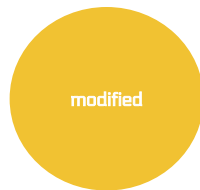
Como faz? – Estados dos Arquivos



arquivo não
versionado



arquivo versionado,
mas não modificado.

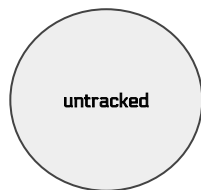
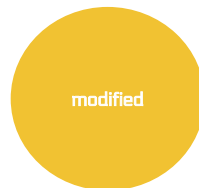


arquivo versionado
e modificado.

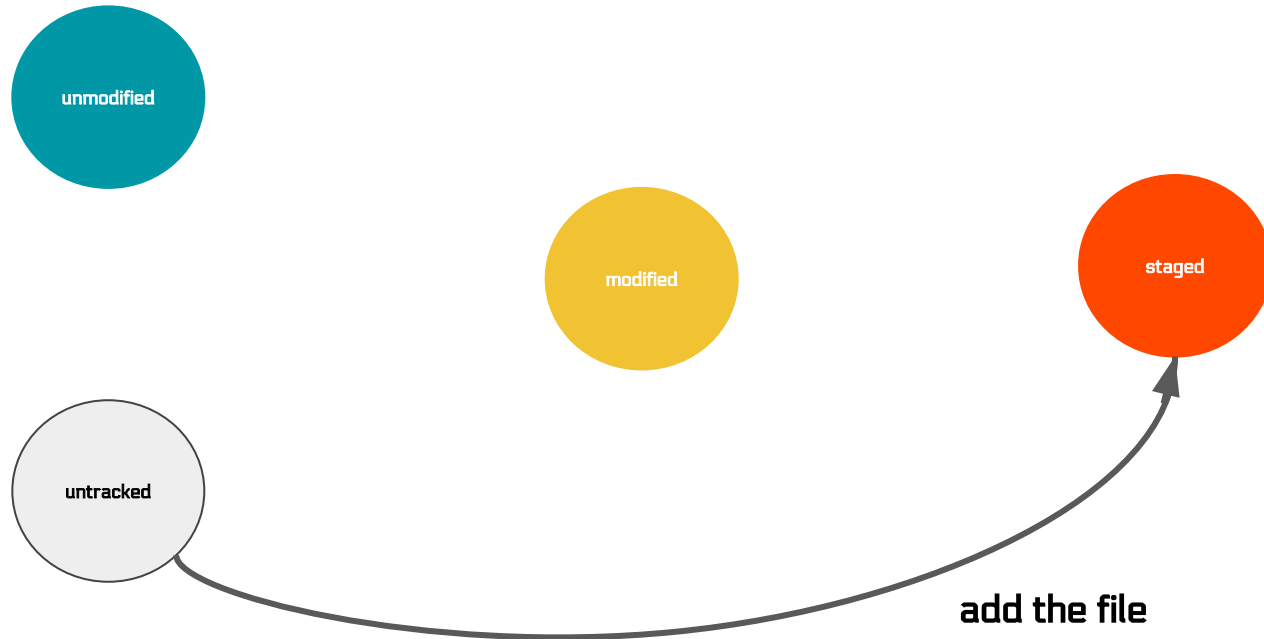


arquivos
versionados,
modificados e que
se deseja gerar uma
nova versão.

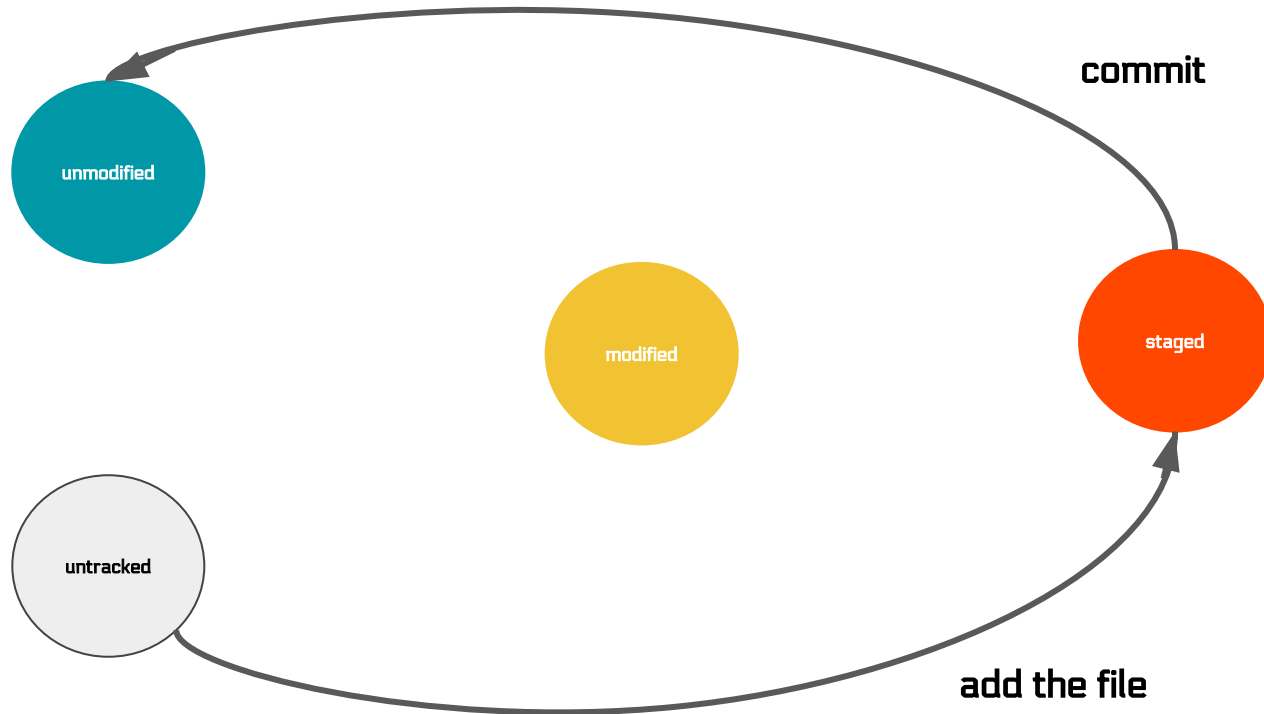
Como faz? – Estados dos Arquivos



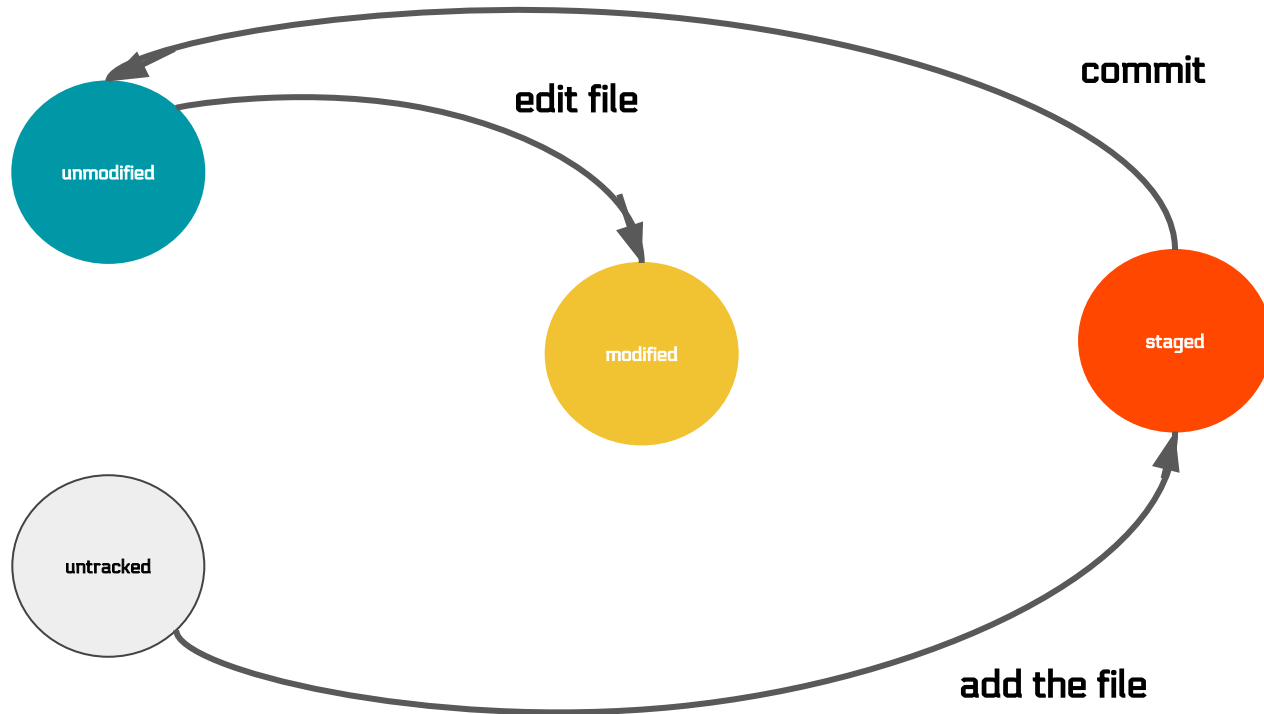
Como faz? – Estados dos Arquivos



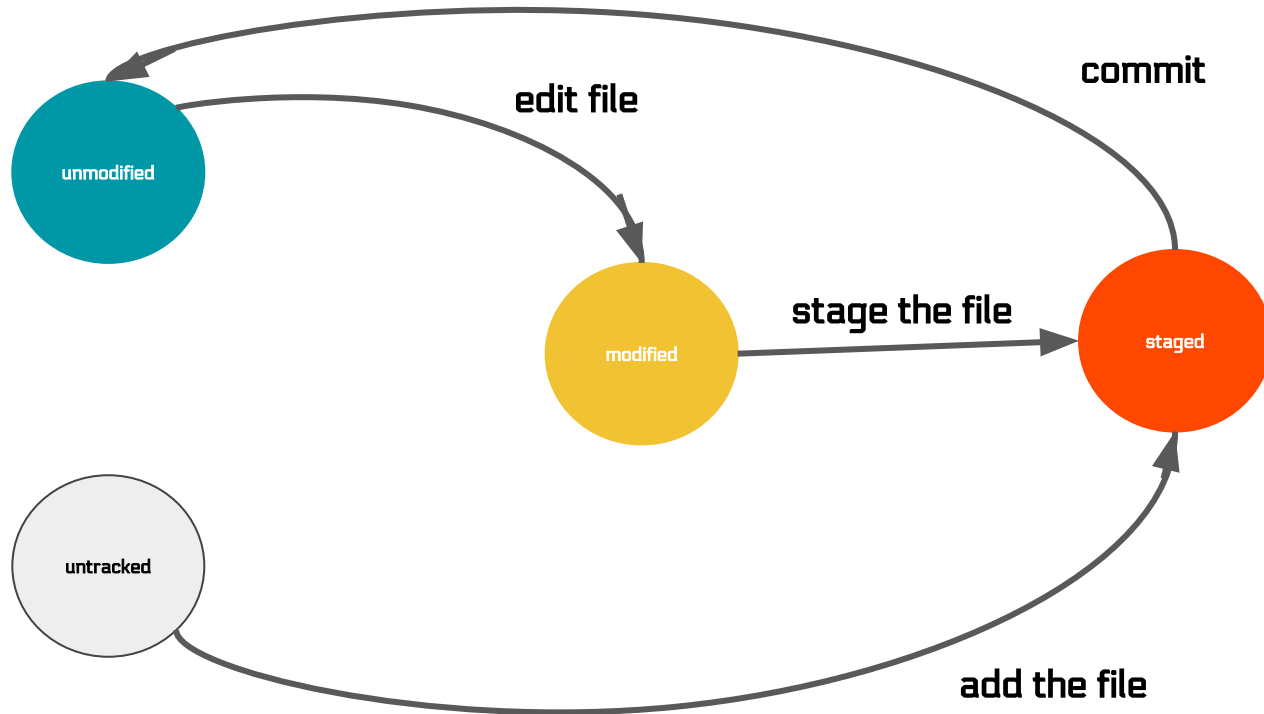
Como faz? – Estados dos Arquivos



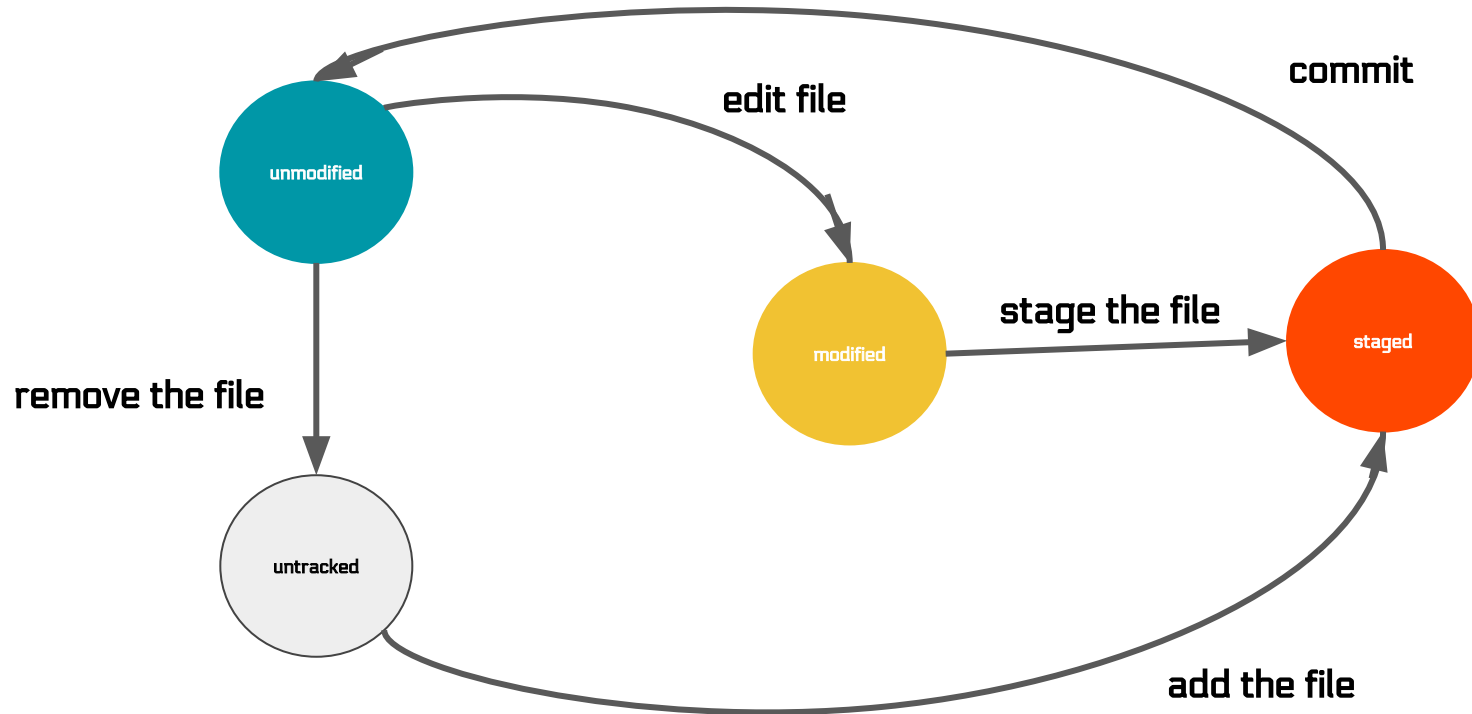
Como faz? – Estados dos Arquivos



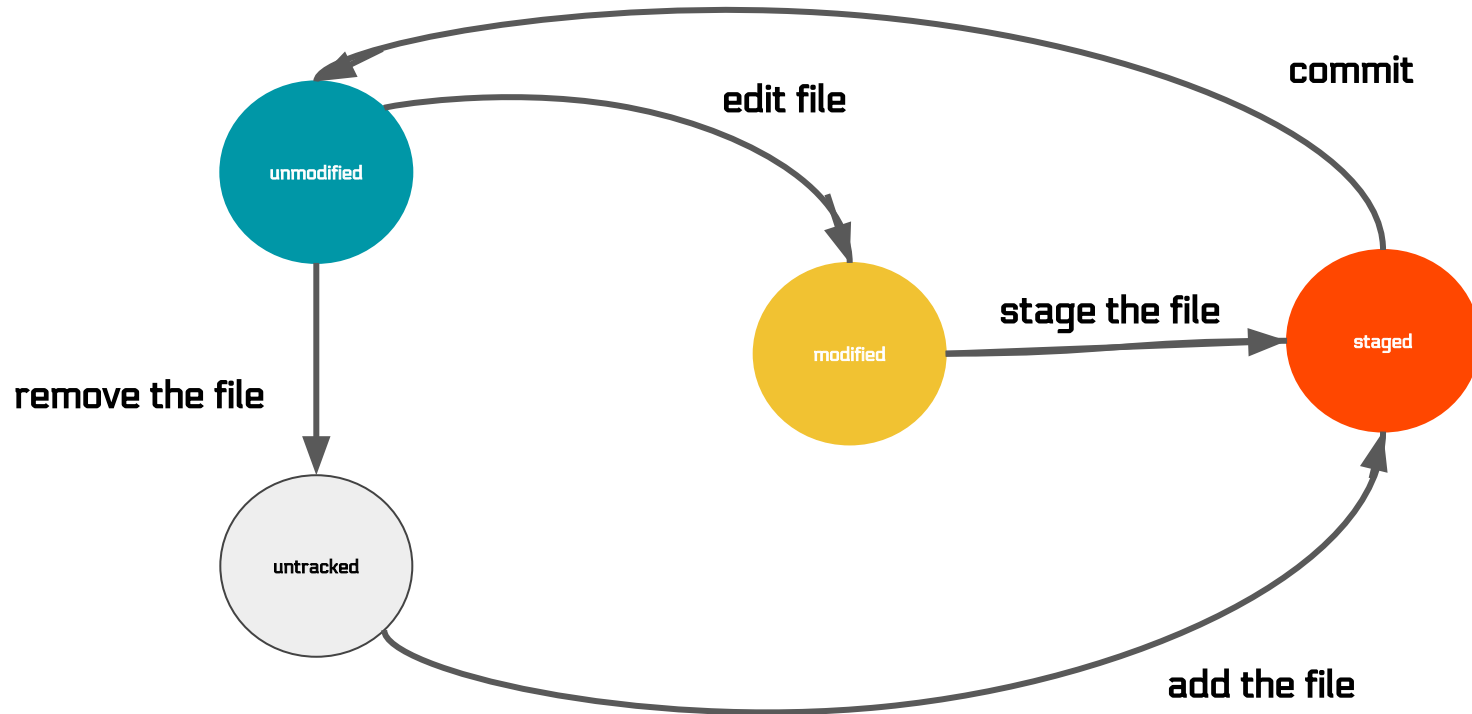
Como faz? – Estados dos Arquivos



Como faz? – Estados dos Arquivos



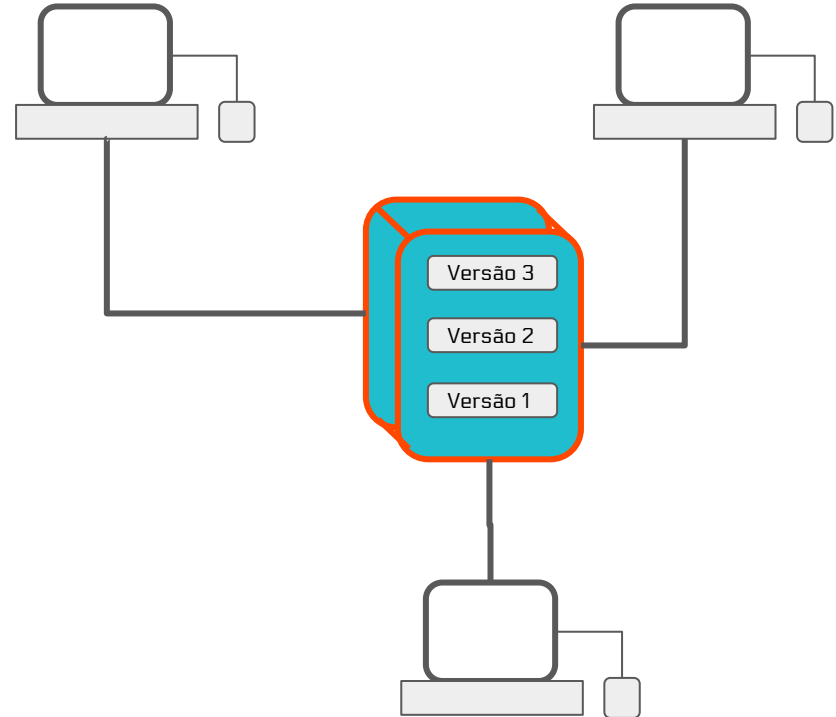
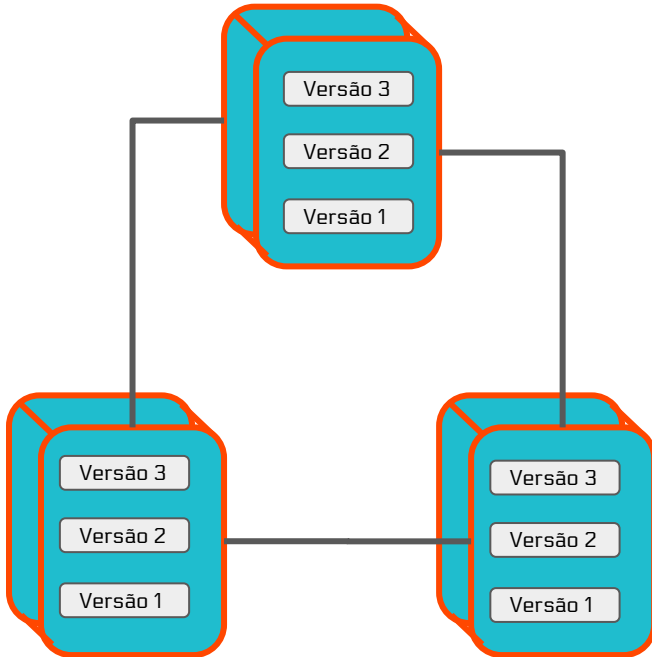
Como faz? – Estados dos Arquivos



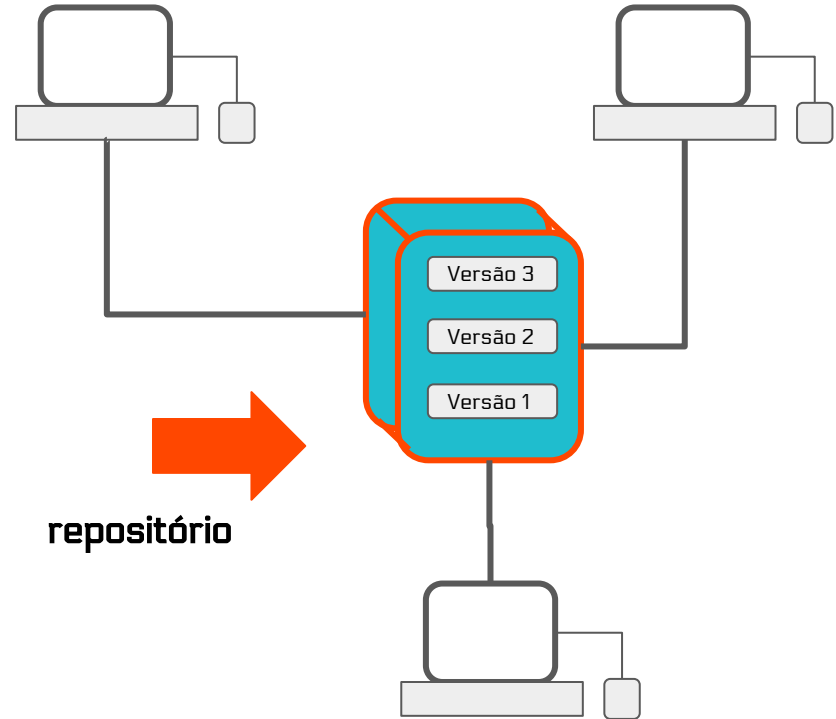
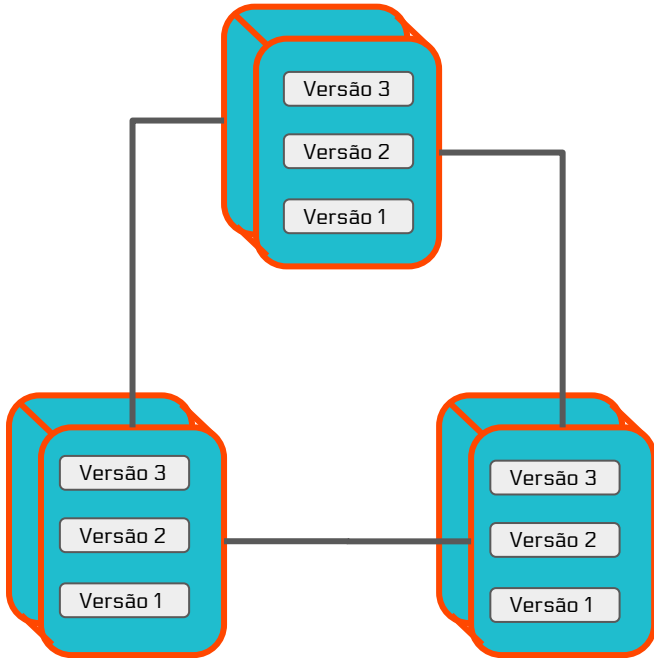


Git

Git - Sistemas de Controle de Versão



Git - Sistemas de Controle de Versão



Git – Sistemas de Controle de Versão



Git – Sistemas de Controle de Versão

