



escola
britânica de
artes criativas
& tecnologia

Profissão: Analista de dados



CONTROLE DE VERSÃO III



GUIA DA AULA 2



Trabalhando com branches

 Listar

 Criar



Acompanhe aqui
os temas que
serão tratados
na videoaula



Listar

O comando `git branch` lista as *branches* do repositório. A documentação do comando pode ser encontrada no link <https://git-scm.com/docs/git-Branch>. Os usos mais comuns do comando são:

```
git branch
```

```
git branch -a
```

Exemplo

:

```
In [ ]: !git branch
```

```
In [ ]: !git branch -a
```



Já o comando `git fetch` atualiza os metadados das *branches* no repositório local. A documentação do comando pode ser encontrada no *link* <https://git-scm.com/docs/git-fetch>. O uso mais comum do comando é:

```
git fetch --prune
```

Exemplo

:

```
In [ ]: !git fetch --prune
```



Criar

O comando `git checkout` cria novas *branches*. Ele também é utilizado para mudar o contexto do desenvolvimento entre *branches*. *Branches* remotas são criadas através do comando `git push`, caso não existam. A documentação do comando pode ser encontrada no *link* <https://git-scm.com/docs/git-checkout>. Os usos mais comuns do comando são:

```
git checkout <nome-da-branch>
```

```
git checkout -b <nome-da-nova-branch> <nome-da-branch-referencia>
```

Exemplo

:

```
In [ ]: !git branch -a
```

```
In [ ]: !git checkout -b dev main
```



```
In [ ]: !git branch -a
```

```
In [ ]: # alterar o arquivo hello.py
```

```
In [ ]: !git status
```

```
In [ ]: !git add hello.py
```

```
In [ ]: !git commit -m "arquivo hello.py alterado em dev"
```



```
In [ ]: !git status
```

```
In [ ]: !git push origin dev
```

```
In [ ]: !git checkout main  
!git branch -a
```

```
In [ ]: !git checkout dev  
!git branch -a
```

