



escola
britânica de
artes criativas
& tecnologia

Profissão: Analista de dados



CONTROLE DE VERSÃO II



GUIA DA AULA 1



Adicione e salve alterações



Acompanhe aqui
os temas que
serão tratados
na videoaula

● Setup

● Adicionar
e salvar



Setup

- Autenticação

```

In [ ]: import os

username = "andre-marcos-perez"
os.environ["GITHUB_USER"] = username

!git config --global user.name "${GITHUB_USER}"
  
```

```

In [ ]: import os
        from getpass import getpass

usermail = getpass()
os.environ["GITHUB_MAIL"] = usermail

!git config --global user.email "${GITHUB_MAIL}"
  
```



```

In [ ]: import os
        from getpass import getpass

        usertoken = getpass()
        os.environ["GITHUB_TOKEN"] = usertoken
  
```

- Projeto

```

In [ ]: !git clone \
        https://${GITHUB_USER}:${GITHUB_TOKEN}@github.com/andre-marcos-perez/da-ebac.
  
```

```

In [ ]: %cd /content/da-ebac/
  
```



Adicionar e salvar

- Fluxo:

O `git` define um fluxo de trabalho para manter o rastreamento das ações (criar, modificar, salvar etc.) realizadas nos arquivos de um repositório. Neste fluxo, arquivos são movidos entre áreas ou zonas dependendo da ação que é realizada. São elas:

1. **working**: trabalho;
2. **staging**: preparação;
3. **repository**: alterações salvas localmente;
4. **remote**: alterações salvas remotamente.



- Adicionar:

O comando `git add` move arquivos da *working* para *staging* área. Se um arquivo for alterado/removido após ter sido adicionado, este deve ser adicionado novamente.

A documentação desse comando pode ser encontrada no *link*

<https://git-scm.com/docs/git-add>

Os usos mais comuns do comando são:

```
git add <nome-do-arquivo-1> <nome-do-arquivo-2> ...
```

```
git add <nome-do-dir>
```



Exemplo:

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

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

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

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

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

O arquivo `.gitignore` é utilizado para indicar ao `git` quais arquivos devem ser ignorados pelo comando `git add`

```
In [ ]: !head -n 10 .gitignore
```



- salvar:

O comando `git commit` move arquivos da *staging* para *repository* área. A todo `commit` é atribuída uma chave identificadora única para rastreamento (*hash*). Ações nos arquivos "comitados" são salvas no repositório local dentro do diretório `.git`.

A documentação desse comando pode ser encontrada no *link*

<https://git-scm.com/docs/git-commit>

O uso mais comum do comando é:

```
git commit -m "<mensagem-descrevendo-as-alterações>"
```



Exemplo:

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

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

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

