



# OFICINA

INTRODUÇÃO AO GIT/GITHUB - DIA 1





# TÓPICOS ABORDADOS

03

## O QUE É E COMO FUNCIONA O GIT

Contextualizando a criação do Git e seu funcionamento;

04

## PREPARANDO O AMBIENTE

Configurações iniciais;

05

## INIT

Inicializando um repositório local e sincronizando com um repositório remoto;

07

## ADD, COMMIT, PUSH

Realizando o primeiro commit e push para o repositório remoto;

09

## FORK, CLONE

Buscando projetos de repositórios remotos

11

## GITHUB DESKTOP

Simplificando o dia-a-dia;

12

## GUIA DE COMANDOS

Lista de principais comandos

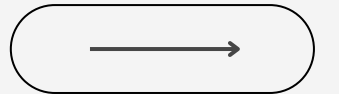
13

## EXERCÍCIOS PRÁTICOS

Proposta de atividade prática.



# O QUE É E COMO FUNCIONA O GIT



- É uma ferramenta de código aberto com o objetivo de realizar controle de versões de arquivos;
- 
- Introduz estágios para as modificações: Working Tree (estamos trabalhando e realizando modificações nos arquivos), Stage ("preparando" o terreno para armazenar modificações) e HEAD (ponteiro para o último "commit" realizado);



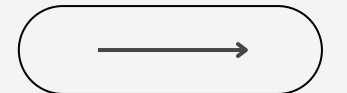
O Git foi desenvolvido pelo Linus Torvalds, após divergências e a ferramenta utilizada para codificação do kernel Linux se tornar proprietária





# PREPARANDO O AMBIENTE

LINUX (DEBIAN/UBUNTU E DERIVADOS)



```
sudo apt update  
sudo apt install git
```

WINDOWS

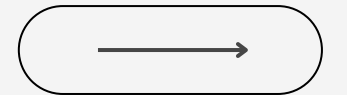
Instalador: <https://git-scm.com/install/windows>

```
git config --global user.name "Seu nome"  
git config --global user.email seuemail@email.com
```

```
git config --list
```



# INIT



- Comando para inicializar um repositório Git na sua máquina;

git init

- Para ver o status do seu repositório:

git status

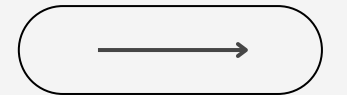
Assegure-se de utilizar o comando dentro do diretório que você deseja versionar;





# INIT

06

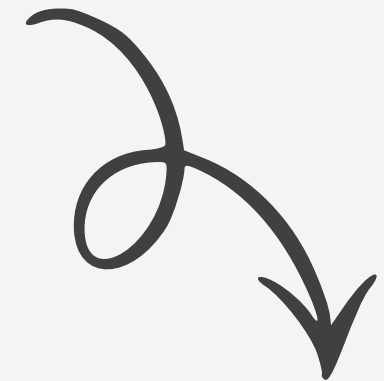


- Sincronizando com um repositório remoto:

```
git remote origin <url do repositório>
```

- Mudando o nome da branch para ficar igual ao remoto:

```
git branch -m main
```



É necessário ter uma conta no Github para vincular a um repositório remoto



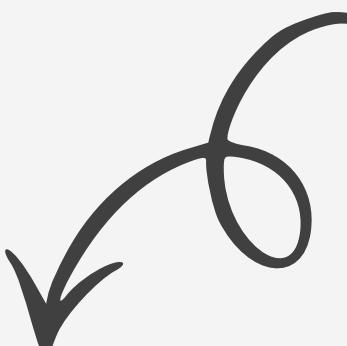
# ADD, COMMIT, PUSH

- Após criar e modificar arquivos, o comando add adiciona essas alterações para serem armazenadas com o commit;

git add <nome do arquivo>  
git add \*

- Para fazer o commit e armazenar as mudanças:

git commit -m "texto explicando a sua alteração"



Se houver vários arquivos alterados e você quer armazenar todos no mesmo commit, usar git add \*



# ADD, COMMIT, PUSH

- Para publicar seu commit no repositório remoto pela primeira vez, usar o comando:

```
git push -u origin main
```

- Depois do primeiro push, basta usar:

```
git push
```

Esse vídeo demonstra a criação de um token de acesso para realizar o push para seu repositório remoto

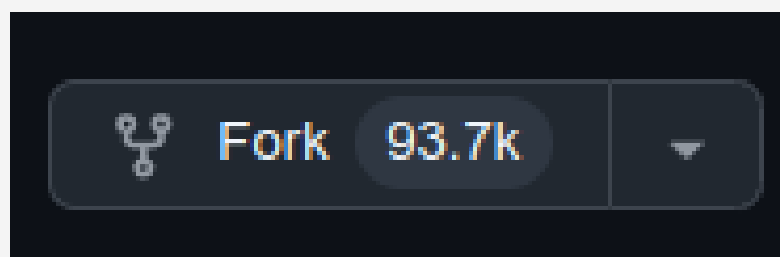
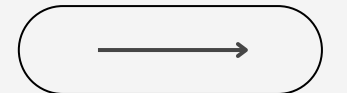
O `-u` vincula o repositório remoto como referência para o repositório local. Se não usar, precisa incluir "origin main" sempre que fizer push





# FORK E CLONE

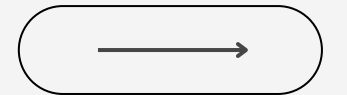
- Fazer um fork é criar uma cópia de um projeto de terceiro no seu repositório remoto;
- Permite usar outros projetos como base, alterando o que quiser sem afetar o original;
- Quem acessar o projeto original pode ver quantos forks foram feitos a partir dele;
- Quem acessar um projeto “forkado” pode ver o projeto original.



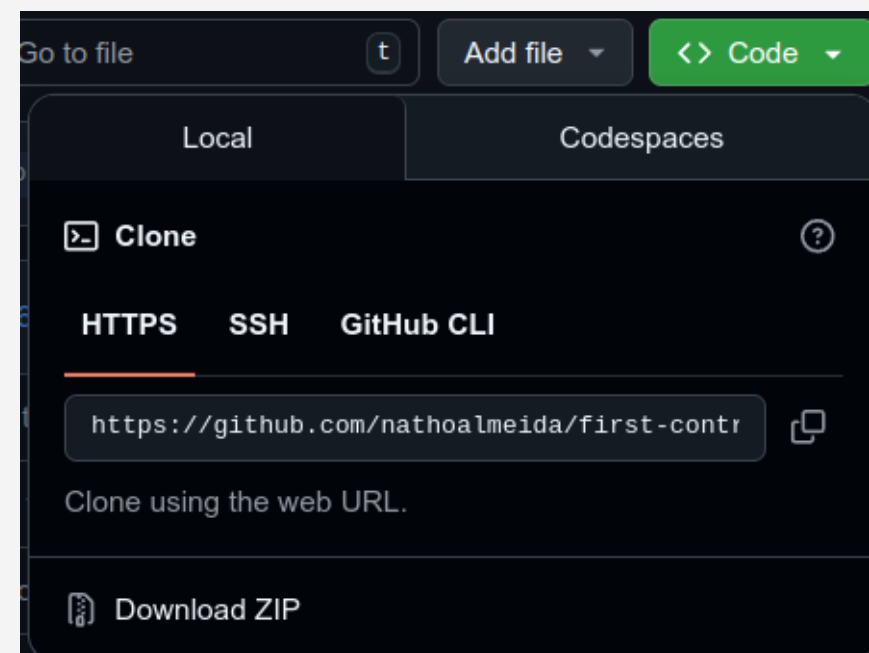


# FORK E CLONE

- Já clonar um repositório significa fazer uma cópia de um repositório remoto, criando um repositório local;
- Você pode, por exemplo, realizar o “fork” de um projeto e depois clonar o repositório para sua máquina;



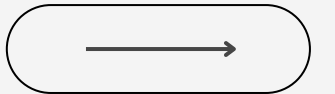
```
git clone <url do repositório>
```





# GITHUB DESKTOP

- Alternativa mais “amigável”, pois não é necessário utilizar linhas de comando para realizar as ações;
- É uma opção para ganho de produtividade, mas compreender o mecanismo por trás dos comandos é fundamental para o melhor uso do Git.

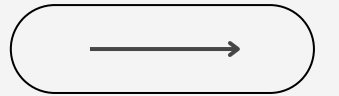




# GUIA DE COMANDOS

12

```
git config --list  
git config --global user.name "Seu nome"  
git config --global user.email seuemail@email.com
```



```
git init  
git status  
git log ← mostra histórico de commits
```

```
git add <nome_do_arquivo>  
git add *
```

```
git remote origin <url_repositorio_remoto>
```

```
git commit -m "mensagem"
```

```
git branch -m novo_nome_da_branch
```

```
git push -u origin main  
git push
```





# EXERCÍCIOS PRÁTICOS

## PROPOSTA 1: CRIAR UMA CONTA NO GITHUB

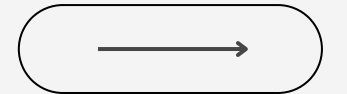
- Crie sua conta no GitHub;
- Gere seu primeiro token de acesso;

## PROPOSTA 2: UTILIZAR O FORK

- Faça um fork do repositório-exemplo utilizado hoje;
- Faça download desses slides ou utilize o git clone;

## PROPOSTA 3: CRIAR O PRÓPRIO REPOSITÓRIO LOCAL

- Crie um repositório local;
- Vincule a um repositório remoto;
- Faça seu primeiro commit!





Obrigada pela participação!

Não perca as atividades promovidas pelo  
PET-SI! Acompanhe as redes sociais:



@petsi\_ufsm



[Link para o Discord](#)