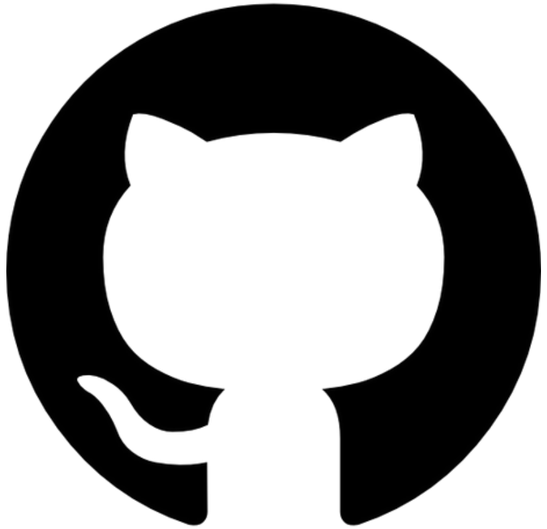


Oficina de Git e Github

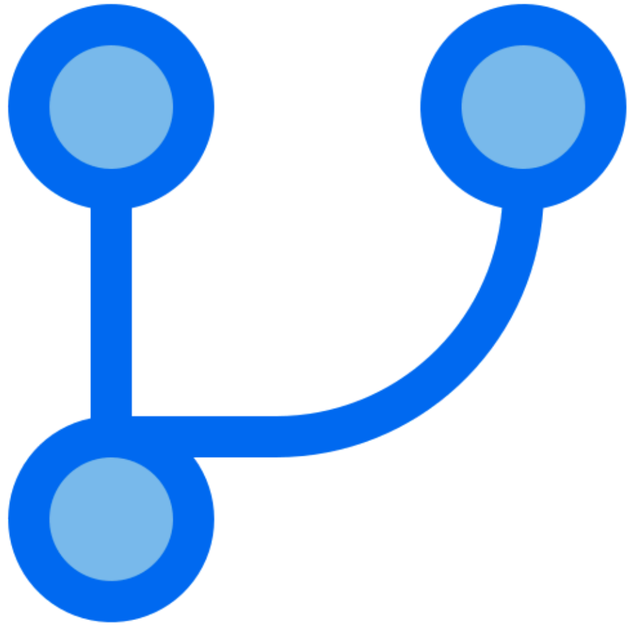
Módulo 3: Branches

Introdução

Neste módulo, vamos aprender a trabalhar com branches (ramificações) no Git.
Branches permitem desenvolver novas funcionalidades sem interferir no código principal e facilitam a colaboração em equipe.

O que é um Branch?

- É uma **linha paralela de desenvolvimento**.
- O branch principal geralmente se chama **main** (ou master em projetos antigos).
- Usado para testar novas ideias, corrigir bugs e criar funcionalidades sem quebrar o projeto principal.



Criando e Listando Branches

- Comando:

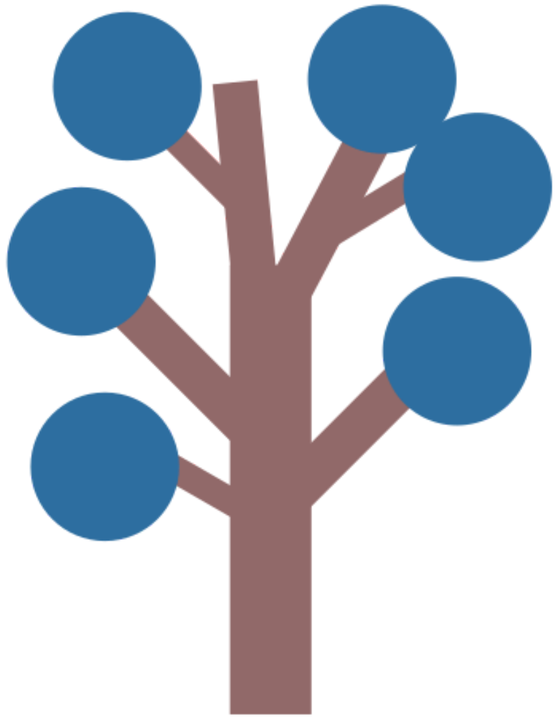
  ```
  git branch
  ```

- Lista todos os branches existentes.
- O branch atual aparece com um *.

- Comando:

  ```
  git branch nome-da-branch
  ```

- Cria uma nova branch, mas **não troca para ela**.



Mudando de Branch

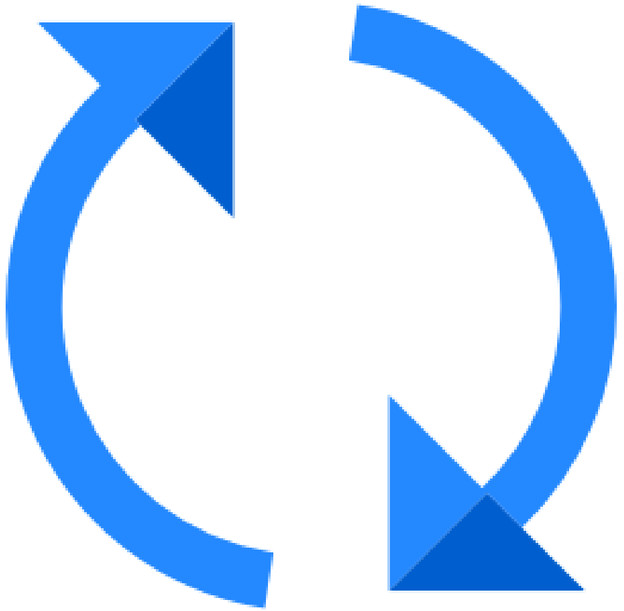
- Comando:

  ```
  git checkout nome-da-branch
  ```

- Alterna para outra branch.
- Comando:

  ```
  git checkout -b nome-da-branch
  ```

- Cria e já muda para a nova branch.



Mesclando Branches (Merge)

- Comando:

```
git merge nome-da-branch
```

- Mescla as alterações de outra branch na branch atual.
- Muito usado para juntar desenvolvimento paralelo ao branch principal.

Resolvendo Conflitos

- Acontecem quando duas pessoas alteram a **mesma parte do código** em branches diferentes.
- O git pede que você **edite manualmente o arquivo** para escolher qual versão manter.
- Depois de resolver:

```
git add arquivo
git commit -m “Conflito resolvido”
```

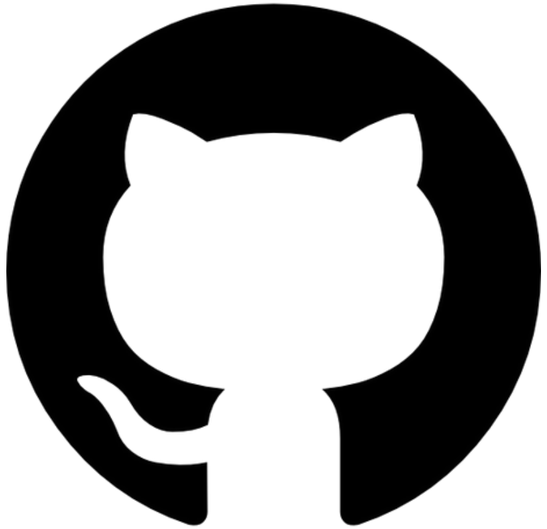
Fluxo de Colaboração

1. Criar uma branch para cada nova tarefa ou correção.
2. Fazer commits nela.
3. Mesclar (merge) com a branch principal após testes.
4. Em projetos no GitHub, usa-se **Pull Request** para revisão antes da mesclagem.

Conclusão do Módulo

Agora você já sabe:

- Criar e alternar entre branches.
- Mesclar alterações e resolver conflitos.
- Organizar o trabalho em equipe usando ramificações.



No próximo módulo:

Fork e Pull Request