

# Módulo 15 - Git e GitHub

O Git e o GitHub são duas ferramentas fundamentais para o desenvolvimento de software colaborativo e o controle de versionamento de código. Vou explicar brevemente o que são e como funcionam:

## Git:

1. **O que é o Git:** O Git é um sistema de controle de versionamento de código distribuído. Ele permite que os desenvolvedores acompanhem as mudanças em seu código ao longo do tempo, facilitando a colaboração em projetos de software.
2. **Como funciona:** O Git funciona mantendo um registro de todas as alterações feitas em um projeto. Cada vez que você faz uma alteração, o Git cria um "commit" que contém uma captura do estado do código naquele momento. Isso permite que você reverta para versões anteriores do código, rastreie quem fez quais alterações e resolva conflitos em colaborações.

## 3. Comandos básicos do Git:

- `git init` : Inicia um novo repositório Git em um diretório.
- `git clone` : Clona um repositório Git existente para o seu computador.
- `git add` : Adiciona arquivos ao "staging area" (área de preparação) para serem incluídos no próximo commit.
- `git commit` : Cria um novo commit com as mudanças na área de preparação.
- `git pull` : Obtém as alterações do repositório remoto para o seu repositório local.
- `git push` : Envia seus commits para um repositório remoto.

## GitHub:

1. **O que é o GitHub:** O GitHub é uma plataforma baseada na web que utiliza o Git para hospedar repositórios de código. Ele é amplamente utilizado para colaboração em projetos de código aberto e privados. O GitHub oferece recursos adicionais, como rastreamento de problemas, integração contínua, gerenciamento de equipes e muito mais.
2. **Como funciona:** No GitHub, você pode criar repositórios para seus projetos e colaborar com outras pessoas. Os repositórios são armazenados na nuvem, o

que facilita o compartilhamento e a colaboração em equipes distribuídas. Os desenvolvedores podem fazer "forks" (cópias) de repositórios, fazer alterações em suas cópias e, em seguida, criar "pull requests" para sugerir que essas alterações sejam mescladas no repositório original.

### 3. Recursos do GitHub:

- **Pull Requests:** Usados para sugerir e revisar mudanças em um repositório.
- **Issues:** Usados para rastrear problemas, tarefas e melhorias no código.
- **Actions:** Permite automatizar tarefas de integração contínua e implantação contínua.
- **GitHub Pages:** Permite hospedar sites estáticos diretamente do repositório.

O Git e o GitHub são ferramentas poderosas para desenvolvedores, permitindo um controle de versionamento robusto e facilitando a colaboração em equipe em projetos de software. Eles são amplamente utilizados na indústria de desenvolvimento de software e são essenciais para qualquer pessoa que trabalhe com programação.

- `git merge` → une duas branches
- `git checkout outra-branch` → altera a branch para "nova-branch"
- `git checkout .` → volta o estado no projeto
- `git branch nova-branch` → Cria uma branch chamada "nova-branch"
- `git branch` → mostra todas as branches existentes
- `git rm -f` → remove um arquivo do repositório
- `git reset --hard hash_do_commit` → O HEAD aponta para o commit indicado através do hash. Todos os arquivos serão modificados para aquela versão do commit
- `git reset HEAD~1` → retorna 1 commit
- `git reset` → remove modificações feitas com o `git add`
- `git push` → envia seu código para o repositório remoto
  - `git push -u origin main` → utilize esse comando na primeira vez que enviar um código a um novo repositório remoto
- `git remote add origin <url>` → adiciona um novo repositório remoto
- `git log` → Mostra o histórico de commits

- `git commit -m "nome-do-commit"` → Realiza o commit do repositório
- `git add .` → Adiciona um novo arquivo ao repositório

## Comandos gerais

- `git push -u origin main`
- `git remote add origin <url-repositorio-remoto>`
- `git commit -m <sua-descrição-do-commit>`
- `git add .`
- `git init`

## Ordem para criar um repositório remoto