

GIT: Sistema de controle de versão de código-fonte

Disciplina: Programação aplicada à engenharia cartográfica

Maurício C. M. de Paulo - D.Sc.

4 de fevereiro de 2026

GitHub é uma plataforma online para:

- Hoster repositorios Git
- Compartilhar código
- Colaborar em projetos

Em resumo: GitHub = Git + colaboração + nuvem

GitHub e Git: qual a diferença?

- **Git**

- Ferramenta local
- Controla versões do código
- Funciona sem internet

- **GitHub**

- Serviço online
- Hospeda repositórios Git
- Facilita colaboração

- **Repositórios Git+GitHub:**

- **Remotos:** repositórios conectados de controle de versão
- **Locais:** cópias locais do repositório

Repositórios no GitHub

Um **repositório** é:

- Um projeto versionado com Git
- Armazenado online no GitHub

Pode conter:

- Código-fonte
- Documentação
- Histórico de alterações

Cada repositório possui:

- Um endereço (URL)
- Permissões de acesso

Ações comuns no GitHub

- **Clone:** baixar o repositório para o computador
- **Push:** enviar commits para o GitHub
- **Pull:** baixar atualizações
- **Fork:** criar uma cópia do repositório
- **Pull Request:** propor alterações

Essas ações permitem o trabalho colaborativo.

Por que usar o GitHub?

- Trabalhar em equipe
- Manter histórico do projeto
 - Lembrar o que você fez
 - Achar onde foi que você quebrou o código
- Facilitar revisão de código
- Compartilhar projetos

Benefícios:

- Transparência
- Organização
- Reprodutibilidade

Cada aluno deve criar seu próprio usuário

- Acessar <https://github.com>
- Criar uma conta

Essas credenciais serão usadas:

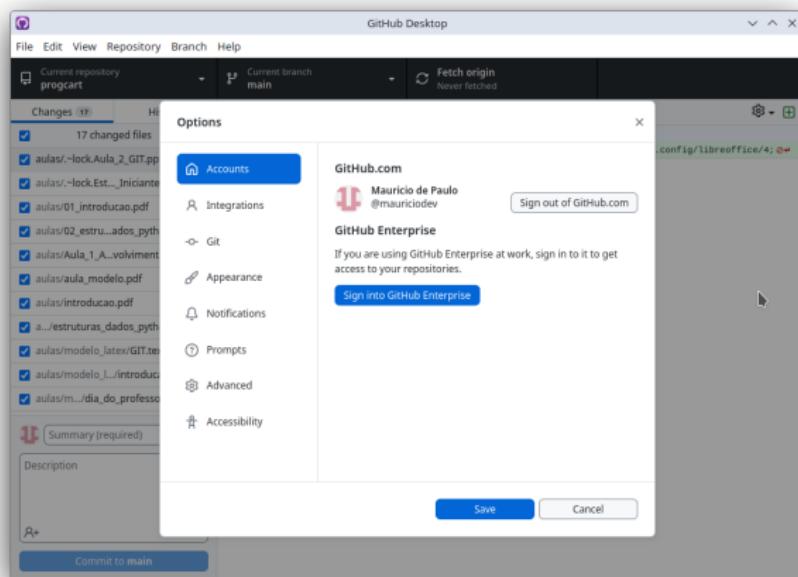
- No VSCodium ou GitHub Desktop
- Para baixar código
- Para submeter alterações

GitHub Desktop — Login do usuário

Passo 1 — Login

- Abrir o GitHub Desktop
- File → Options → Accounts
- Fazer login com a conta GitHub

Necessário para clonar e sincronizar repositórios.



GitHub Desktop — Clonar repositório

Passo 2 — Clonar

- File → Clone repository
- Escolher o repositório:
`https://github.com/mauriciodev/progcart`
- Definir pasta local

O repositório é baixado apenas uma vez.

figs/github_clone.png

Passo 3 — Alterações

- Editar arquivos no editor
- Salvar alterações
- Ver aba *Changes*

As diferenças são mostradas
linha a linha.

Passo 4 — Commit

- Selecionar arquivos
- Escrever mensagem
- Commit to main

Commit registra mudanças
localmente.

GitHub Desktop — Push e Pull

Passo 5 — Push

- Envia commits ao GitHub

Passo 6 — Pull

- Baixa atualizações remotas

Push = local → remoto
Pull = remoto → local

Git no terminal - Instalação

Windows (PowerShell):

```
winget install Git.Git
```

Linux (shell):

```
sudo apt install git
```

Git no terminal - Configuração

Após a instalação, configure suas credenciais:

```
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"
```

Essas informações identificam o autor dos commits.

Separação de arquivos:

- Arquivos **tracked**: já registrados no Git
- Arquivos **untracked**: ainda não versionados

Objetivo:

- Evitar versionar arquivos que não fazem parte do código-fonte

Atalho:

- Ctrl + Shift + P

Abrindo um repositório Git

Endereço do repositório:

```
git@github.com:mauriciodev/progcart.git
```

Esse endereço é usado para clonar o repositório.

GitHub — Autorização da máquina

Para acessar o GitHub a partir de um computador:

- É necessário cadastrar a chave SSH da máquina

Linux/Windows: criar chave SSH

- No terminal: `ssh-keygen -t ed25519`
- Abrir o arquivo `.pub` gerado na pasta `.ssh` do seu usuário.
- No GitHub acessar: `Settings → SSH and GPG Keys → New SSH Key.`
- Copiar o conteúdo do arquivo para o GitHub.

Exercícios

- ① Fazer um **fork** do repositório da disciplina:
 - <https://github.com/mauriciodev/progcart>
- ② Criar um **clone** do repositório na sua máquina
- ③ Verificar a timeline das versões
- ④ Verificar o histórico de um arquivo
- ⑤ Alterar um arquivo
- ⑥ Fazer um **commit**
- ⑦ Fazer um **push**
- ⑧ Acessar o GitHub e verificar a ferramenta de **Pull Request**