

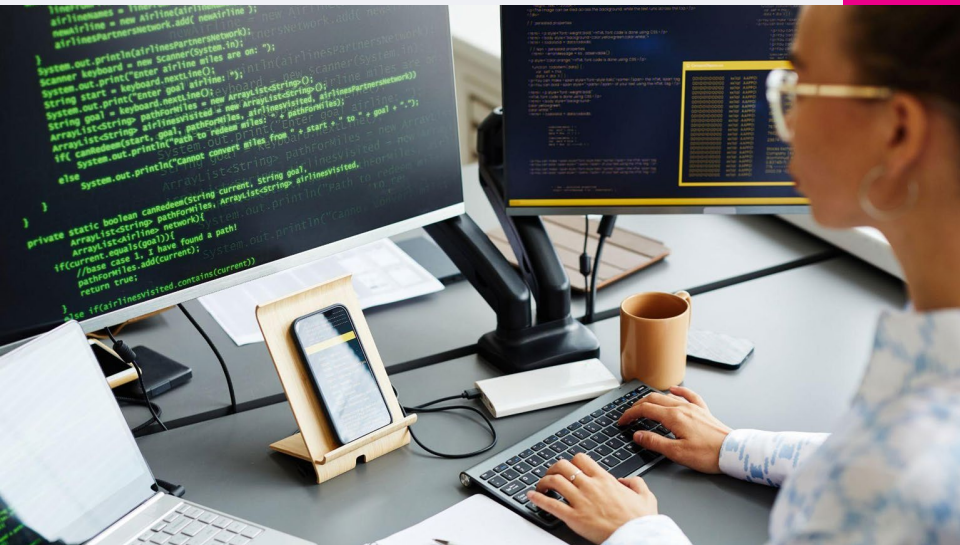


BOOTCAMP I

GIT & GITHUB

Prof. Marcelo Carboni Gomes

GIT & GITHUB



Git é um sistema de controle de versão distribuído que permite gerenciar e acompanhar mudanças no código de software ao longo do tempo.

O GitHub é uma plataforma baseada na web que oferece serviços para hospedagem de repositórios Git.



Controle de Versão: Conceito Básico

- Guardar versões históricas de um projeto.
- Colaborar em projetos com múltiplas pessoas simultaneamente.
- Reverter para versões anteriores caso algo dê errado.



Principais Funcionalidades

- Rastreamento de histórico: O Git registra todas as mudanças no código, permitindo visualizar todo o histórico de modificações;
- Branching e merging: O Git permite criar branches (ramificações) para trabalhar em funcionalidades ou correções sem interferir no código principal. Depois, as mudanças podem ser mescladas (merge) de volta à versão principal;
- Trabalho colaborativo: Em equipes, o Git permite que várias pessoas contribuam para o mesmo projeto simultaneamente, sem sobrescrever o trabalho umas das outras.



Na Prática

- Repositório (Repo): É onde o histórico de todas as alterações é armazenado. Pode ser local (no seu computador) ou remoto (em plataformas como GitHub);
- Commit: É uma "foto" do estado atual do projeto em um dado momento. Cada commit contém um conjunto de mudanças e uma mensagem descritiva;
- Branch: Uma ramificação do projeto principal. Você pode criar branches para desenvolver novas funcionalidades sem impactar a versão principal (geralmente chamada de main ou master).



Na Prática

- Merge: Processo de unir mudanças de uma branch para outra;
- Pull e Push: O pull baixa atualizações de um repositório remoto, enquanto o push envia suas alterações locais para o repositório remoto.



Principais Funcionalidades

- Colaboração simplificada: GitHub permite que várias pessoas trabalhem em um mesmo projeto simultaneamente, revisando e integrando mudanças de maneira organizada;
- Gerenciamento de Projetos: Ferramentas como Issues e Projects ajudam a organizar e priorizar tarefas e bugs dentro de um projeto;
- Integração com ferramentas externas: GitHub permite integração com serviços de CI/CD (integração e entrega contínua), automatizando testes e implantações.



Principais Funcionalidades

- Repositório (Repo): Criar repositórios públicos (acessíveis por qualquer pessoa) ou privados (somente para membros do projeto). Compartilhar repositórios para colaboração com outros desenvolvedores;
- Fork e Pull Request: Fazer um fork de um repositório cria uma cópia completa do projeto em sua conta GitHub. Isso permite que você trabalhe em uma versão separada do projeto sem afetar o repositório original. Após fazer modificações em seu fork, você pode propor que essas mudanças sejam integradas no repositório original, submetendo um pull request. O responsável pelo repositório original pode revisar, discutir e eventualmente aprovar as mudanças;



Principais Funcionalidades

- **Issues:** São ferramentas para reportar bugs, discutir novas funcionalidades ou fazer anotações sobre o projeto. Cada issue pode ser atribuído a uma pessoa ou equipe, facilitando o gerenciamento de tarefas.
- **Projects** Uma maneira visual de organizar o progresso de um projeto usando quadros (parecidos com ferramentas como o Trello), permitindo acompanhar o progresso das issues e tarefas.



Integração entre Git e GitHub

1. Pré-requisitos

Git instalado no seu sistema: ``git --version``

Conta no GitHub: github.com

Opcional: Configuração de chave SSH ou token de acesso pessoal (PAT)

2. Inicializando um projeto Git

No terminal, dentro da pasta do seu projeto:

```
git init
```

Configura seu nome e e-mail:

```
git config --global user.name "Seu Nome"
```

```
git config --global user.email "seu@email.com"
```



Integração entre Git e GitHub

3. Adicionando arquivos e fazendo commit

```
git add .
```

```
git commit -m "Primeiro commit"
```

4. Criando um repositório no GitHub

Vá em github.com → botão 'New repository'

Defina nome, visibilidade e crie o repositório sem README se já tem um local

Copie a URL do repositório. Exemplo (HTTPS):

<https://github.com/seu-usuario/nome-repositorio.git>



Integração entre Git e GitHub

5. Conectando o repositório remoto

Com HTTPS:

```
git remote add origin https://github.com/seu-usuario/nome-repositorio.git
```

Com SSH:

```
git remote add origin git@github.com:seu-usuario/nome-repositorio.git
```

6. Enviando seu código para o GitHub

```
git push -u origin master
```

Ou, se o branch principal for 'main':

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



Integração entre Git e GitHub

7. Verificando

Acesse o GitHub e confira seu código no repositório.

Resumo de comandos comuns

Comando

`git status`

`git add arquivo.txt`

`git commit -m "msg"`

`git push origin branch`

`git pull origin branch`

`git clone URL`

Descrição

Verifica status dos arquivos

Prepara arquivo para commit

Salva alterações localmente

Envia alterações ao GitHub

Baixa alterações do GitHub

Clona um repositório do GitHub