

Dominando Git e GitHub: Guia Prático para Iniciantes

W por Wellington Oliveira



Sumário

1. Introdução
2. Conceitos Básicos do Git
3. Trabalhando com Repositórios
4. GitHub e Repositórios Remotos
5. Trabalhando com Branches e Merge
6. Resolvendo Conflitos
7. Conclusão
8. Recursos Adicionais



Introdução

O Git e o GitHub são ferramentas essenciais para qualquer desenvolvedor moderno. Este eBook ensinará os conceitos fundamentais do Git e como utilizá-lo com o GitHub para versionamento e colaboração eficiente em projetos.

Capítulo 1: Conceitos Básicos do Git

O que é o Git?

O Git é um sistema de controle de versão distribuído que permite acompanhar alterações no código ao longo do tempo.

Instalando o Git

- **Windows:** Baixe o instalador em git-scm.com.
- **Linux:** sudo apt install git (Debian/Ubuntu) ou sudo yum install git (CentOS/RHEL).
- **MacOS:** brew install git.

Configurando o Git

```
# Configurar nome e e-mail globalmente  
git config --global user.name "Seu Nome"  
git config --global user.email "seuemail@example.com"
```

Capítulo 2: Trabalhando com Repositórios

Criando um Repositório Local

```
mkdir meu-projeto  
cd meu-projeto  
git init
```

Adicionando e Confirmando Alterações

```
echo "Hello, Git!" > arquivo.txt git add arquivo.txt  
git commit -m "Adicionando primeiro arquivo"
```

Visualizando Histórico de Commits

```
git log --oneline --graph --all
```



1 Criar repositório

Inicialize um novo repositório Git local

2 Adicionar arquivos

Adicione arquivos à área de staging

3 Commit

Confirme as alterações no repositório

4 Visualizar histórico

Veja o histórico de alterações

Capítulo 3: GitHub e Repositórios Remotos

Criando um Repositório no GitHub

1. Acesse github.com e crie um novo repositório.
2. Copie a URL do repositório.

Conectando o Re却itório Local ao GitHub

```
git remote add origin https://github.com/seuusuario/meu-projeto.git  
git branch -M main  
git push -u origin main
```

Repositorio Local

Seu código e histórico de versões no seu computador

Repositorio Remoto

Cópia do seu código hospedada no GitHub para compartilhamento e backup

Capítulo 4: Trabalhando com Branches e Merge

Criando e Mudando de Branch

```
git branch nova-feature  
git checkout nova-feature
```

Fazendo Merge de Branch

```
git checkout main  
git merge nova-feature  
git push origin main
```



Capítulo 5: Resolvendo Conflitos

Quando dois desenvolvedores modificam a mesma linha de um arquivo, um conflito pode ocorrer. Para resolver:

1. Edite o arquivo conflitante.
2. Remova as marcações de conflito (<<<<<, =====, >>>>>).
3. Faça um novo commit.

```
git add arquivo.txt  
git commit -m "Resolvendo conflito"  
git push origin main
```

Identificar Conflito

O Git marca os conflitos nos arquivos afetados

Editar Arquivo

Escolha qual versão manter ou combine as alterações

Resolver e Commit

Adicione o arquivo resolvido e faça um novo commit

Capítulo 6: Conceitos Avançados e Exemplos

Por que usar o Git?

- O Git permite acompanhar todas as modificações feitas no código, facilitando a colaboração e a recuperação de versões anteriores.
- Exemplo:

```
git diff HEAD~1 arquivo.txt # Mostra diferenças entre a versão atual e a anterior
```

Commits e Versionamento

- Commits representam pontos no tempo do desenvolvimento do projeto.
- Exemplo prático de commit com mensagem detalhada:

```
git commit -m "Corrigindo bug na função de login e melhorando logs"
```

Trabalho em equipe com Branches

- Branches permitem que desenvolvedores trabalhem em funcionalidades separadas sem interferir na versão principal do código.
- Exemplo de criação e troca de branch:

```
git checkout -b nova-feature # Cria e muda para uma nova branch
```

Resolução de conflitos em detalhes

- Quando dois desenvolvedores alteram a mesma linha de um arquivo, o Git indica um conflito.
- Passos para resolução:
 1. Editar o arquivo manualmente.
 2. Remover as linhas de conflito (<<<<<, =====, >>>>>).
 3. Testar o código após a correção.
 4. Criar um novo commit para registrar a solução.

Conclusão

Agora você tem um conhecimento sólido sobre Git e GitHub! Continue praticando e explorando novas funcionalidades para melhorar seu fluxo de trabalho e colaboração em equipe.