


CONFLITOS EM MERGE



Aprendendo com o GIT.

DANIELA VELTER



Capítulo 1:

Introdução ao Git

Introdução do GIT

O Git é uma ferramenta de controle de versão amplamente utilizada pelos desenvolvedores para gerenciar o histórico de seus projetos de software.

Com o Git, você pode acompanhar as alterações feitas no código ao longo do tempo e colaborar de forma eficiente com outros membros da equipe.

Vamos mergulhar nos principais comandos do Git para começar a sua jornada!





Capítulo 2: Configuração Inicial

Configuração Inicial

Antes de começar a usar o Git, é importante configurar suas informações de usuário.

Isso inclui seu nome de usuário e endereço de e-mail.

Utilize os seguintes comandos para configurar sua identidade no Git:

```
bash
```

```
git config --global user.name "Seu Nome"  
git config --global user.email "seuemail@example.com"
```




Capítulo 3: Inicializando um Repositório

Inicializando um Repositório


Para começar a rastrear as alterações em um projeto, você precisa inicializar um repositório Git.

Navegue até o diretório do seu projeto e execute o seguinte comando:



```
bash
```

```
git init
```



Capítulo 4: Adicionando Arquivos ao Stage

Adicionando Arquivos ao Stage

Antes de confirmar as alterações em um arquivo, você precisa adicioná-lo ao "stage" traduzido "Palco".

Isso é feito com o comando `git add`.

Por exemplo, para adicionar todos os arquivos modificados no diretório atual ao stage, você pode usar:

```
bash
```

```
git add .
```



Capítulo 5: Realizando Commits

Realizando Commits

Depois de adicionar os arquivos desejados ao stage, é hora de confirmar as alterações com um commit.

Cada commit deve ter uma mensagem descritiva que explique as alterações feitas.

Aqui está um exemplo de como fazer um commit:

```
bash
```

```
git commit -m "Adicionando funcionalidade de login"
```



Capítulo 6: Visualizando o Histórico

Visualizando o Histórico

Para visualizar o histórico de commits em um repositório Git, você pode usar o comando `git log`.

Isso mostrará uma lista de todos os commits, começando pelo mais recente. Por exemplo:

A terminal window with a dark background. The prompt 'bash' is shown in a lighter grey bar at the top. Below it, the command 'git log' is entered in a monospaced font, with 'git' in white and 'log' in orange. To the left of the terminal is a large orange arrow pointing right.

```
bash
```

```
git log
```



Capítulo 7: **Criando e Alternando** **Entre Branches**

Criando e Alternando Entre Branches

Branches são ramificações do seu código que permitem trabalhar em novas funcionalidades sem interferir no código principal.

Para criar uma nova branch, utilize o comando `git branch`. Por exemplo:

```
bash
```

```
git branch nova-feature
```

Criando e Alternando Entre Branches

Para alternar para uma branch específica, você pode usar o comando `git checkout`.

Por exemplo:



```
bash
```

```
git checkout nova-feature
```



Capítulo 8: Mesclando Branches

Mesclando Branches

Depois de trabalhar em uma nova funcionalidade em uma branch separada, você pode mesclar suas alterações de volta para a branch principal usando o comando `git merge`.

Por exemplo, para mesclar a branch `nova-feature` na branch atual:

```
bash
```

```
git merge nova-feature
```



Capítulo 9: Resolvendo Conflitos de Merge

Resolvendo Conflitos de Merge

Às vezes, ao mesclar branches, podem ocorrer conflitos se duas alterações diferentes forem feitas na mesma parte de um arquivo.

O Git irá sinalizar esses conflitos e você precisará resolvê-los manualmente.

Abra os arquivos em conflito em um editor de texto e resolva as diferenças.

Depois de resolver os conflitos, adicione os arquivos ao stage e faça o commit das alterações.



Capítulo 10: Sincronizando com Repositórios Remotos

Sincronizando com Repositórios Remotos

Para colaborar com outros desenvolvedores e manter seu código seguro, é essencial sincronizar seu repositório local com um repositório remoto, como GitHub ou GitLab.

Use os comandos `git remote add` e `git push` para configurar e enviar suas alterações para um repositório remoto.



Conclusão

Conclusão

Com esses comandos essenciais do Git, você está pronto para começar a gerenciar seus projetos de forma eficiente e colaborativa!

Experimente-os em seu próprio projeto e explore as possibilidades que o Git oferece para o desenvolvimento de software.



Agradecimentos

Obrigada Por Ler até aqui!

Esse Ebook foi gerado por IA,
diagramado por humano.

O passo a passo se encontra no meu
GitHub.



[https://github.com/Daniela2319/-
Criando-um-Ebook-com-ChatGPT.git](https://github.com/Daniela2319/-Criando-um-Ebook-com-ChatGPT.git)