

Ebook
Guideline

git BRANCH

A Máquina do Tempo de ramificações

Criado por Tiago Peniche





Imagine o poder de viajar no tempo e explorar diferentes linhas do passado e do futuro.

Com o Git Branch, você cria e assume o controle da sua própria "máquina do tempo", para os nerds de plantão o Git Branch é seu próprio Delorean guiando o desenvolvimento do seu projeto por múltiplas ramificações,



Cada branch é como um universo paralelo em constante evolução e que contribuirão para a Branch principal do projeto.

O QUE É UM BRANCH?



Um branch é como um portal que te leva a uma realidade alternativa do seu projeto. Imagine diversas ramificações (branches) que podem ser alteradas individualmente e mescladas na branch principal sem que haja conflito entre elas.

Cada branch representa uma linha de desenvolvimento independente, permitindo que você experimente novas ideias, faça correções de bugs ou trabalhe em paralelo em diferentes versões do seu código (ou em diferentes “linhas temporais”).

O Git Branch é um portanto um comando fundamental para você e sua equipe conseguirem gerenciar e organizar seu código em diferentes linhas de desenvolvimento paralelamente.



criando branches

Para criar um novo branch, basta utilizar o seguinte comando:



● ● ● Git Branch - Ebook

```
git branch <nome_do_branch>
```

Por exemplo, para viajar para uma "realidade" onde você implementa uma nova funcionalidade:



● ● ● Git Branch - Ebook

```
git branch nova-funcionalidade
```

LISTANDO BRANCHES

Para visualizar as branches disponíveis, utilize o comando:



```
● ● ● Git Branch - Ebook
```

```
git branch
```

Ele te mostrará todos as suas branches, incluindo a atual (indicada por um asterisco).

EXCLUINDO BRANCHES

Ao finalizar sua jornada em um branch, você pode fechá-lo com o comando:

```
● ● ● Git Branch - Ebook  
git branch -d <nome_do_branch>
```



É importante ter cuidado ao fechar as branches, pois a ação é irreversível.

MESCLANDO BRANCHES

Ao retornar ao presente, você pode integrar as mudanças do seu branch à linha do tempo principal (geralmente "master") com o comando:



● ● ● Git Branch - Ebook

```
git checkout master  
git merge <nome_do_branch>
```

Este comando irá combinar as alterações do branch especificado com a linha do tempo master. O Git merge só será citado por aqui mas ele merece um artigo próprio dada a complexidade e importância desse comando.

DICAS E TRUQUES PARA UTILIZAR MELHOR O GIT BRANCH

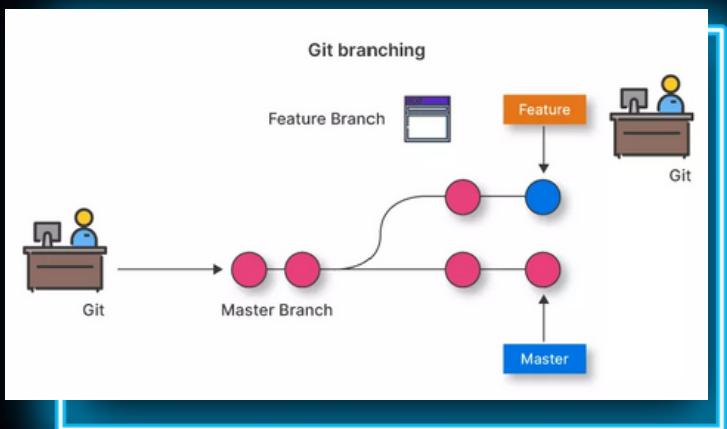
- Utilize nomes descritivos para seus branches, facilitando a identificação do destino de cada portal.
- Criar exemplo com código
- Crie branches para explorar diferentes funcionalidades, corrigir bugs ou realizar experimentos.



- exemplo de código
- Evite trabalhar diretamente na linha do tempo master, utilizando-a como referência principal.
- Faça backups frequentes do seu repositório para evitar perda de dados.

BENEFÍCIOS DO GIT BRANCH

- Desenvolvimento simultâneo e organizado das duas frentes do projeto.
- Integração eficiente das alterações sem conflitos.
- Testes e validação mais rápidos.
- Maior flexibilidade para experimentar novas ideias.
- Maior segurança contra erros e perda de dados.



OBRIGADO POR LER ATÉ AQUI



Este Ebook foi criado baseado em um [artigo homônimo](#) usando IA para coleta de informações e revisado e diagramado por mim

O passo a passo de como foi realizado esse projeto encontra-se em um repositório no meu Github abaixo



Penichezito/Ebook-Git-Branch

1 Contributor 0 Issues 0 Stars 0 Forks

Penichezito/Ebook-Git-Branch

Contribute to Penichezito/Ebook-Git-Branch development by creating an account on GitHub.

[GitHub](#)