

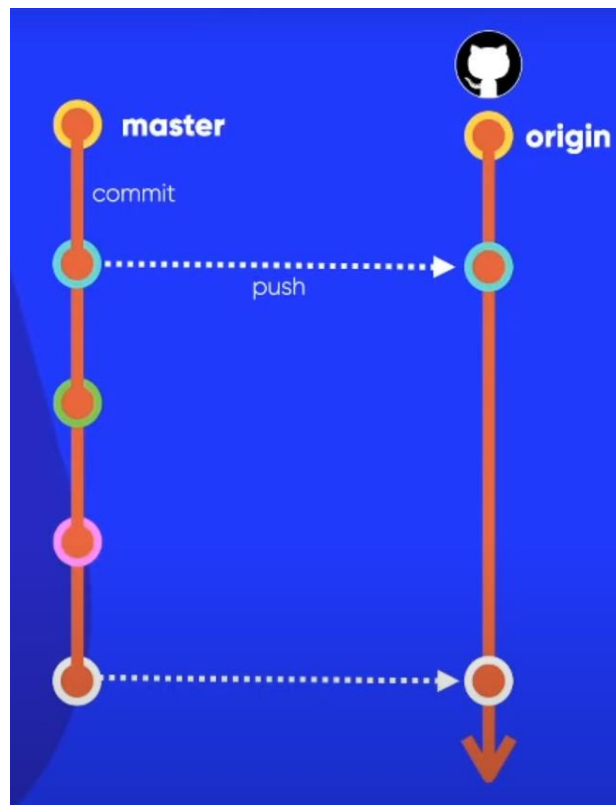
Estudo Autodidata Git e GitHub



Aula: 9

Tema: Branch ou ramificações

Fluxo de trabalho (Master, Origin, Commit e Push)



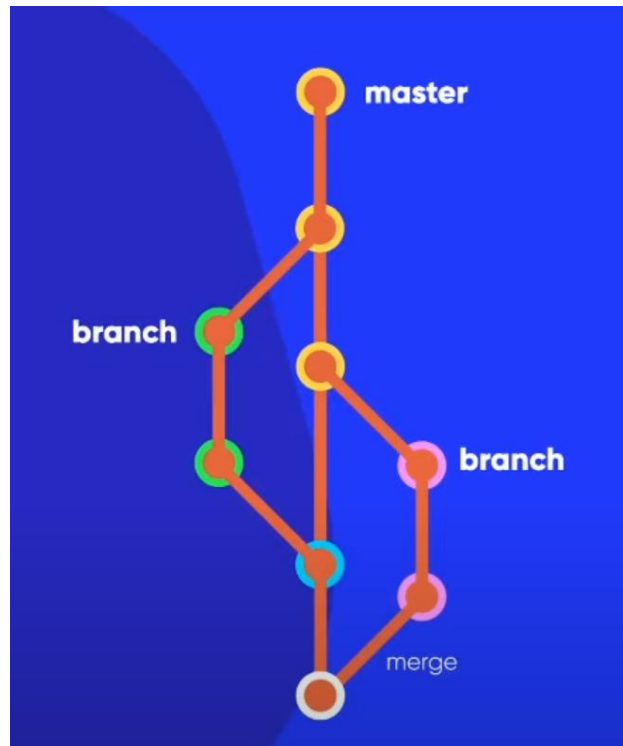
A imagem mostra uma **representação visual do fluxo de trabalho com commit e push entre um repositório local (master) e o repositório remoto (origin, geralmente no GitHub).**

Partes da imagem:

1. **master (lado esquerdo)**
 - Representa o seu **repositório local**, ou seja, o código que está no seu computador.
 - Cada bolinha colorida é um **commit** (uma modificação registrada no histórico do Git).
2. **origin (lado direito)**
 - Representa o **repositório remoto**, normalmente hospedado no **GitHub**.
 - Quando você faz push, os commits do master local são enviados para o origin.
3. **commit**
 - Você faz modificações no seu projeto e salva com `git commit -m "mensagem"`.
 - Isso **cria um novo ponto na linha do tempo do seu branch local**.
4. **push**
 - Você realiza as alterações no repositório remoto (origin) realizando os commits (no github desktop sendo na opção "COMMIT TO MAIN").
 - Esse processo copia os commits que estão no seu master local para o origin.

Termo	Significado
Commit	Salva uma mudança localmente
Push	Envia commits para o GitHub (origin)
Master	Nome padrão do branch principal local
Origin	Repositório remoto (normalmente no GitHub)

Branches e merge



Essa imagem é uma ótima representação gráfica de **como o Git lida com múltiplas ramificações** (branches) e depois une elas com o **merge**.

🔍 Partes principais da imagem:

📁 master

- É o **branch principal** (a linha de código principal).
- Tudo o que está pronto para produção ou uso geralmente está nesse branch.

📁 branch (1º)

- Um **novo branch** é criado a partir do master.
- Você pode imaginar isso como um **trabalho paralelo**, onde alguém está desenvolvendo uma nova funcionalidade sem mexer no código principal.
- Ele avança com seus próprios commits (as bolinhas coloridas).

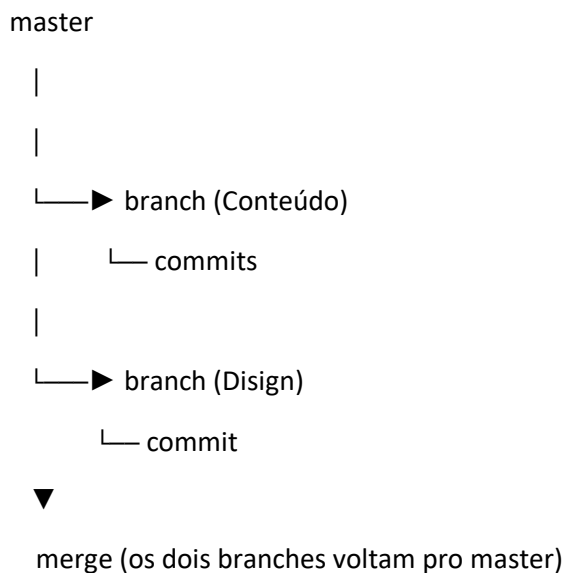
▮ branch (2º)

- Outro branch foi criado **em paralelo**.
- Isso mostra que **várias pessoas ou funcionalidades podem ser desenvolvidas ao mesmo tempo**.

🌀 merge

- Os dois branches (o verde e o rosa) são **fundidos (merge)** de volta no master.
- Isso junta o histórico e as mudanças feitas nos dois branches, consolidando tudo no principal.

Representação do fluxo de trabalho



ETAPA	SIGNIFICADO
Criação de branch	Começar um trabalho paralelo sem mexer no código principal (master)
Commits no branch	Registros de progresso e mudanças dentro da ramificação criada
Merge	Junção das mudanças de um branch de volta ao principal (master)

O melhor conselho sobre o uso de branch é: Quando for fazer alterações em um projeto, seja um site ou um aplicativo, ou dividir as funções a exercer como conteúdo e design, não faça isso na master, crie branches que vão auxiliar na organização e no desenvolvimento do projeto.

Sem falar que se um erro muito grande for feito ele não vai afetar a master, mas sim uma branch que pode ser descartada.