



# Git For Dummies

Arthur Ladislau Pereira

# Objetivo

- Fazer uma introdução ao Git.
- Ensinar os principais Comandos.
- Ensinar boas Práticas.



# Sumário

- O que é Git?
  - O que é versionamento?
  - O que é a ferramenta?
- Por que usar Git?
  - Outras Ferramentas disponíveis no mercado.
  - Benefícios de Git
- Prática
  - Criando repositórios
  - Git Fetch
  - Git Pull
  - Git Add
  - Git Commit
  - Git Push
  - Git Revert
  - Git Reset
  - Git Branch
  - Git Checkout
  - Git Rebase
  - Resolvendo Conflitos

# O que é Git? - Versionamento

- Uso ou criação de versões diferentes de arquivo.
- Por que versionar arquivos?
  - Segurança
  - Controle de Histórico
  - Trabalho em equipes
  - Resgate de Versões estáveis
  - Ramificação de Projetos

# O que é Git? - Ferramenta

- Sistema de Controle de Versões Distribuído.
- Open Source.
- Desenvolvido por Linus Torvalds.
- Multiplataforma.
- Desenvolvido para trabalhar diretamente com ferramentas de controle de acesso.

# Por que usar Git? - Outras Ferramentas

- Modelo Client-Server:
  - CVS.
  - Subversion (SVN).
  - Vesta.
- Modelo Distribuído:
  - BitKeeper.
  - Mercurial.
  - GNU Arch.
  - GNU Bazaar.

# Por que usar Git? - Benefícios de Git

- Suporte para desenvolvimento não linear.
- Desenvolvimento Distribuído.
- Compatível com HTTP, FTP e SSH.
- Eficiente para grandes projetos.
- Toolkit-based.
- Autenticação Criptográfica de História.
- Algoritmos de Complementação de Merge.
- Garbage Collection.

# Prática

- Criando um Repositório



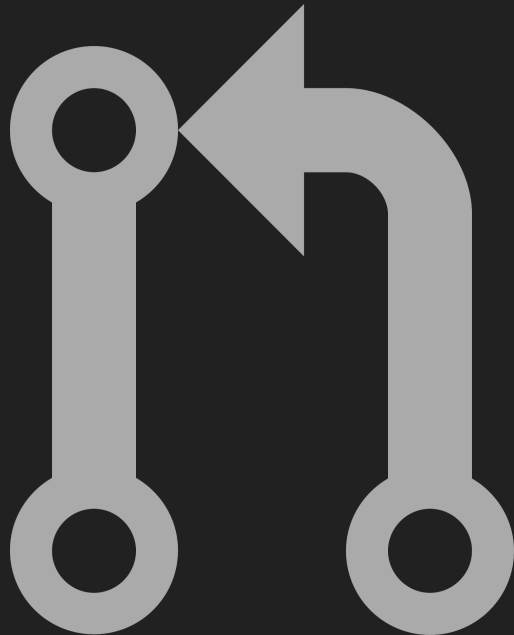


# Prática - Git Clone

- Cria uma cópia Local do repositório no diretório escolhido.
- `git clone [<url do repositório>]`
- Argumento `-r`: clone recursivo.
-

# Prática - Git Fetch

- Acha branches e tags junto a seus complementos.
- `git fetch [<-opções>] [<repositório>]`



# Prática - git pull

- Atualiza o repositório local de acordo com um repositório remoto.
- Basicamente faz um fetch e um merge com o repositório local.
- `git pull [<-opções>] [<repositório> <referência>]`
- Um pull sem argumentos faz a atualização de acordo com a branch atual.

# Prática - Git Add

- Adiciona mudanças ao próximo commit.
- `git add [<arquivos a serem adicionados>]`
- `git add .` adiciona todas as mudanças feitas no repositório.

# Práticas - Git Commit

- Grava mudanças do repositório local.
- `git commit`
- Todos os commits tem uma mensagem de commit.
- `git commit -m "[<mensagem>]"` | para já adicionar uma mensagem ao commit.
- `git commit -a` | para adicionar mudanças de arquivos já monitorados ao commit sem usar o `add`.

In case of fire



1. `git commit`



2. `git push`



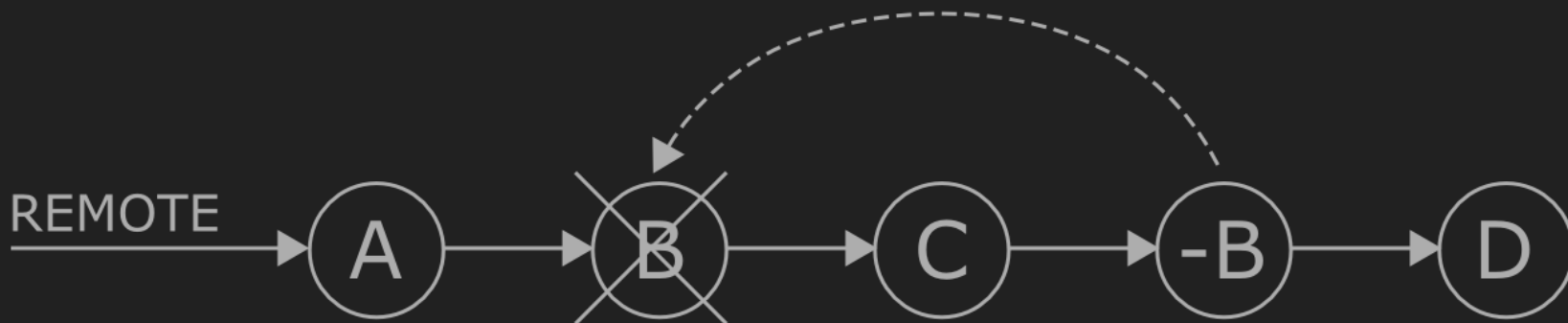
3. `leave building`

# Prática - Git Push

- Persiste remotamente as mudanças commitadas, faz um merge da versão atual do repositório local (só as mudanças commitadas serão consideradas) com o repositório remoto.
- `git push [<repositório/branch>]`
- Um push sem argumentos faz um merge com o repositório em que está trabalhando atualmente.

# Prática - Git Revert

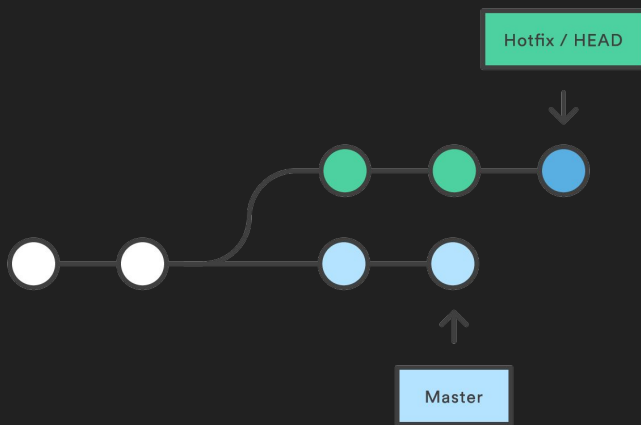
- Um revert desfaz commits específicos identificando-os pela Hash dos commits
- `git revert [<hash>]`
- Um revert gera automaticamente um commit indicando o que foi desfeito.



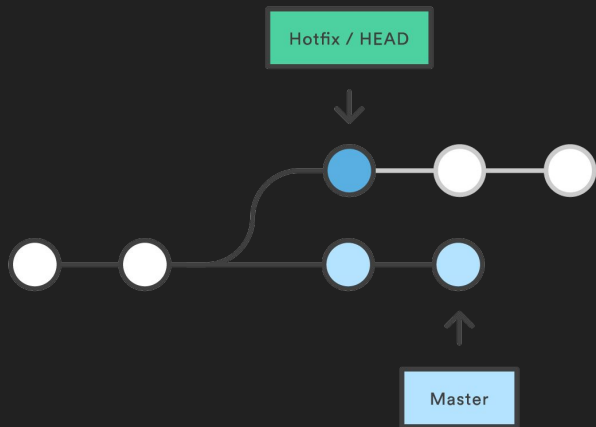
# Prática - Git Reset

- Retorna o repositório local para uma versão anterior usando a hash do commit desejado.
- `git reset [<hash>]`
- O reset não interage diretamente com o repositório remoto e pode ser desfeito com um `git pull`.

Before Resetting



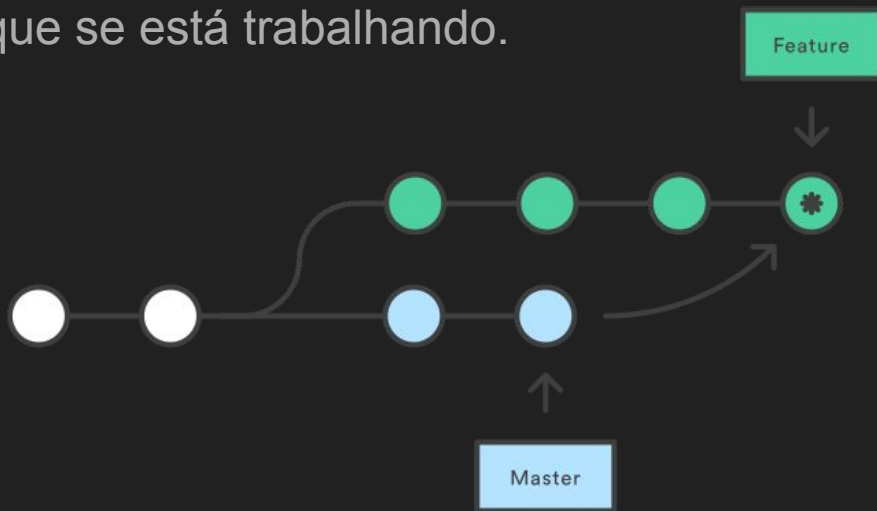
After Resetting





# Prática - Git Branch

- Cria uma nova Branch do repositório, uma nova cópia para desenvolvimento paralelo sem alterar o histórico atual
- `git branch [<nome da nova branch>]`
- Um branch sem argumentos lista todas as branches locais criadas do repositório em que se está trabalhando.

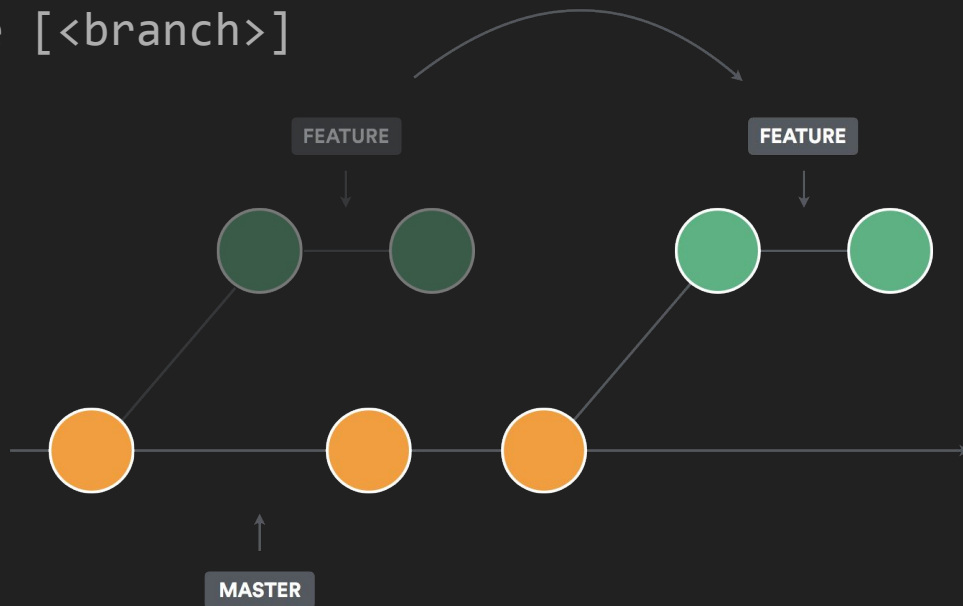


# Prática - Git Checkout

- Recupera a versão de uma dada branch ou recupera as versões remotas de arquivos específicos.
- `git checkout [<nome da branch>]`
- `git checkout [<caminho dos arquivos>]`
- `git checkout .` | recupera a versão remota de todos os arquivos do repositório.

# Prática - Git Rebase

- Avança a branch atual para o estado de uma branch desejada, basicamente trazendo os commits de uma branch para o repositório local para continuar o trabalho.
- `git rebase [<branch>]`



# Prática - Resolvendo Conflitos

- Quando se faz um pull alguns conflitos podem aparecer, especialmente se estamos trabalhando em equipe.
- O git nos ajuda a resolver estes conflitos colocando ambas as possibilidades em um único arquivo com alguns marcadores.
- Tudo o que temos que fazer é ver os marcadores de conflitos e editar os arquivos num editor de texto qualquer de acordo com a nossa necessidade.

**YOU EXPECTED A THANK YOU**

**BUT IT WAS ME, DIO!**

To Be Continued