

# Git: O Sistema de Controle de Versão Distribuído

Git é uma ferramenta essencial para desenvolvedores de software, facilitando o trabalho em equipe e o gerenciamento de versões de projetos. Ele foi criado por Linus Torvalds em 2005 e se tornou um dos sistemas de controle de versão mais populares do mundo.



by **Victor Santos Rohod**







# O que é o Git?

## Controle de Versões

O Git rastreia as mudanças feitas nos arquivos de um projeto ao longo do tempo, permitindo que você volte para versões anteriores.

## Trabalho em Equipe

O Git facilita a colaboração em projetos, permitindo que várias pessoas trabalhem simultaneamente no mesmo código.

## Gerenciamento de Ramos

O Git permite que você crie diferentes ramos do código, permitindo que você experimente novas funcionalidades sem afetar o código principal.



# Por que o Git é importante?

1

## Histórico de Mudanças

O Git registra todas as alterações feitas nos arquivos de um projeto, proporcionando um histórico completo do desenvolvimento.

2

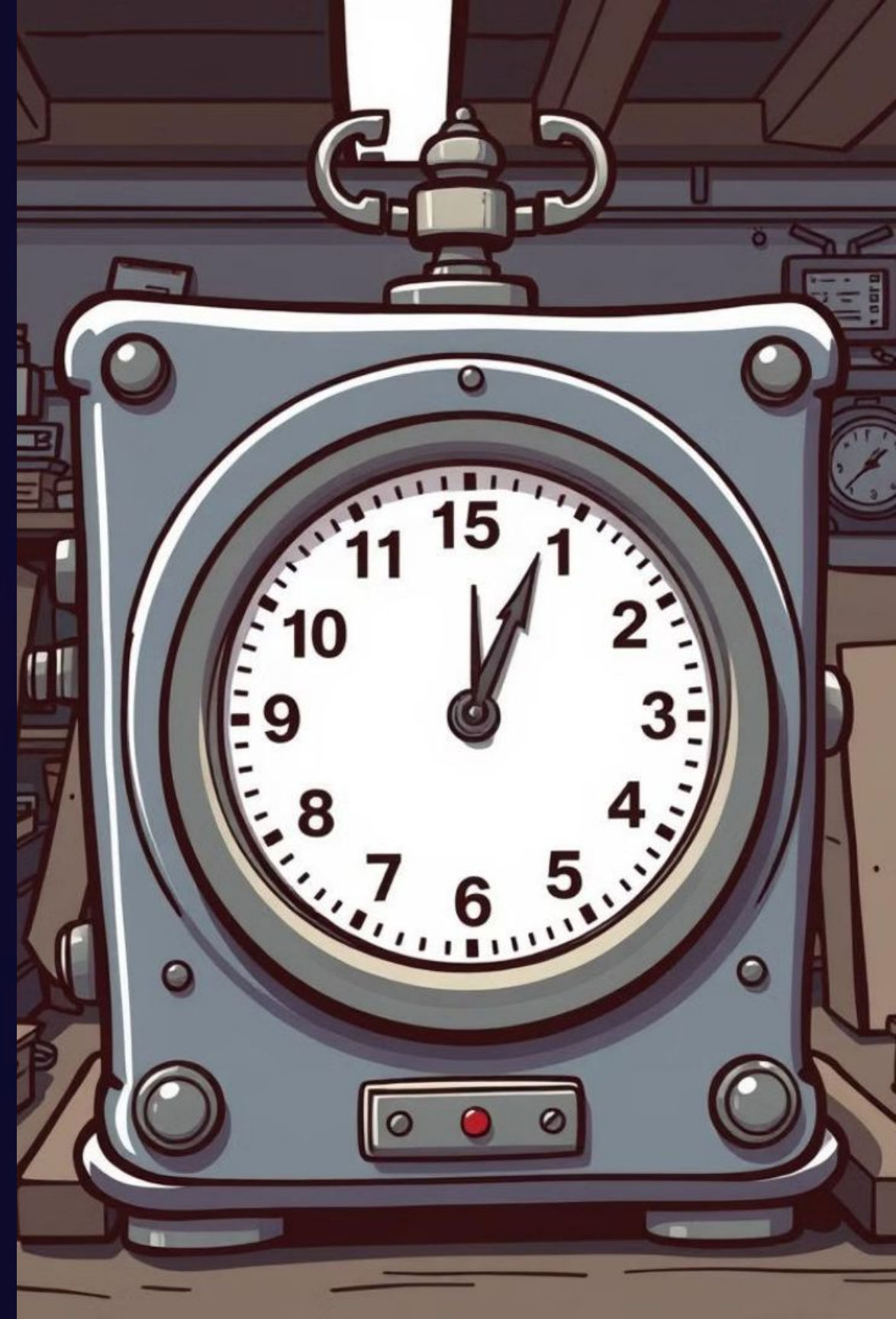
## Prevenção de Perdas

Com o Git, é possível recuperar versões anteriores do projeto, evitando perdas de código ou erros graves.

3

## Integração de Código

O Git facilita a integração de código de diferentes desenvolvedores, garantindo que as mudanças sejam combinadas de forma eficiente.



# Principais funcionalidades do Git



## Commit

Salva as alterações feitas nos arquivos no repositório local.



## Pull

Baixa as alterações do repositório remoto para o repositório local.



## Merge

Combina as alterações de um ramo com outro.



## Stash

Salva as alterações não commitadas temporariamente.



## Log

Mostra o histórico de commits do repositório.



## Push

Envia as alterações do repositório local para o repositório remoto.



## Branch

Cria um novo ramo de trabalho, permitindo desenvolver novas funcionalidades sem afetar o código principal.



## Checkout

Troca entre diferentes ramos de trabalho.



## Revert

Desfaz alterações específicas de um commit.



## Status

Exibe o estado atual do repositório.

# Repositórios e branches



1

## Repositório Local

Armazena o código do projeto no computador do desenvolvedor.

2

## Repositório Remoto

Armazena o código do projeto em um servidor, acessível a todos os membros da equipe.

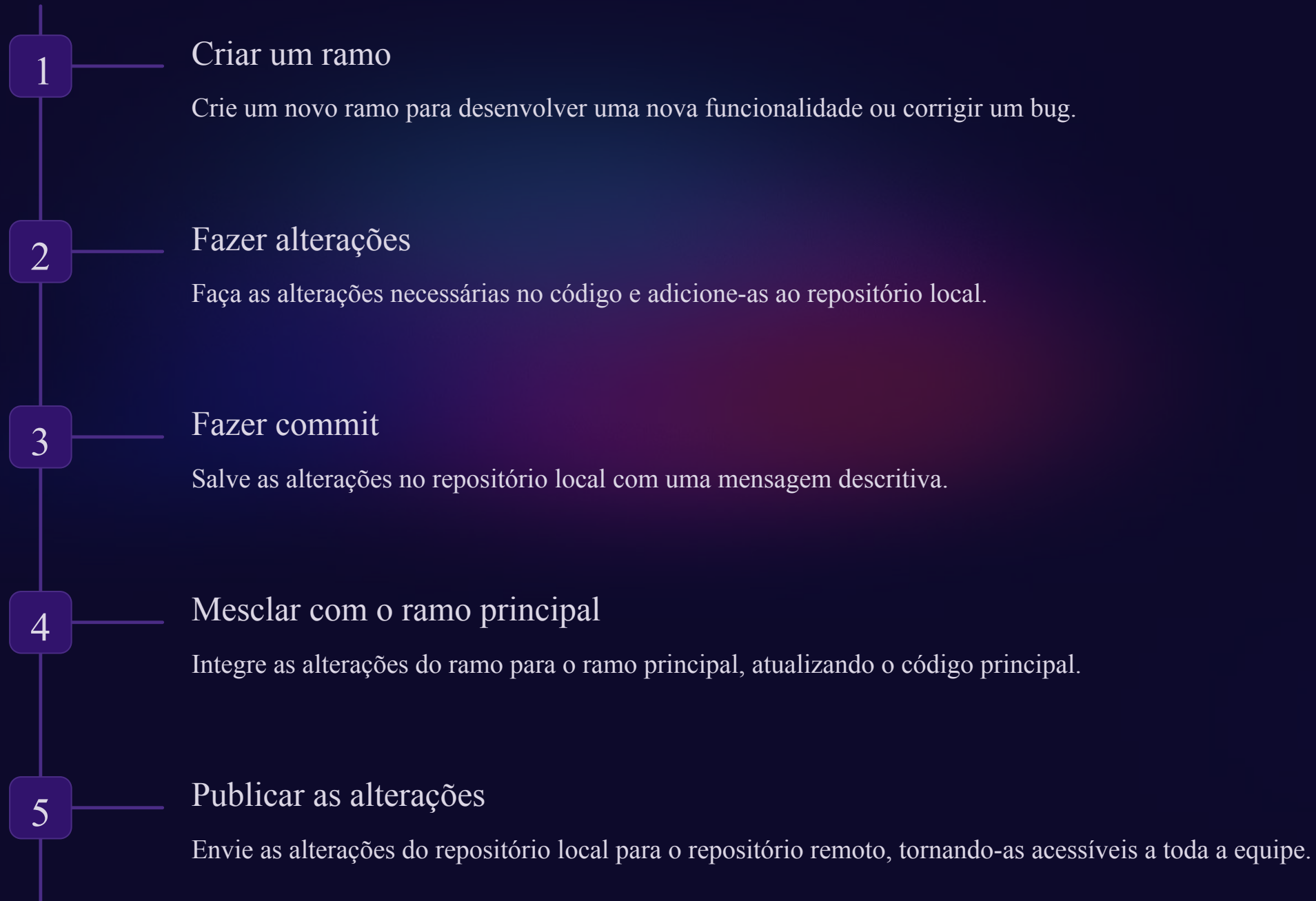
3

## Branches

Permitem que você trabalhe em diferentes versões do projeto, sem afetar o código principal.



# Fluxos de trabalho com o Git



# Comandos básicos do Git



## git clone

Cria uma cópia local do repositório remoto, permitindo que você faça o download do código-fonte e comece a trabalhar em um projeto.



## git add

Adiciona arquivos modificados ao repositório local, preparando-os para serem commitados. É importante adicionar os arquivos que você deseja registrar para que as alterações sejam rastreadas.



## git commit

Salva as alterações no repositório local com uma mensagem descritiva, registrando o que você fez e fornecendo um histórico do projeto.



## git push

Envia as alterações do repositório local para o repositório remoto, compartilhando suas alterações com outros membros da equipe ou com o mundo.



## git pull

Baixa as alterações do repositório remoto para o repositório local, garantindo que você tenha a versão mais recente do código e evite conflitos.

# Trabalhando com conflitos de merge



## Conflitos

Ocorrem quando duas pessoas fazem alterações no mesmo trecho de código.



## Resolução

É necessário analisar as alterações conflitantes e escolher qual versão manter.



## Ferramentas

O Git oferece ferramentas para facilitar a resolução de conflitos, como o "git mergetool".

# erge Confl





# Dicas e boas práticas de uso do Git



1

## Mensagens Claras

Escreva mensagens de commit claras e concisas, descrevendo o propósito das alterações.

2

## Commits Pequenos

Faça commits pequenos e específicos, facilitando a revisão e o rollback de alterações.

3

## Revisão de Código

Peça a um colega para revisar seu código antes de mesclar com o ramo principal.

4

## Documentação

Mantenha a documentação do projeto atualizada, refletindo as alterações feitas com o Git.

# O poder do Git para a colaboração

O Git é uma ferramenta poderosa que revolucionou o desenvolvimento de software, facilitando a colaboração, o gerenciamento de versões e o controle de mudanças.

Dominar o Git é fundamental para qualquer desenvolvedor que busca trabalhar em projetos de forma eficiente e profissional.

