



Introdução a versionamento com Git

O controle de versão é uma necessidade fundamental na colaboração e gerenciamento de projetos de software. Essa apresentação explorará como o Git e as plataformas de hospedagem podem impulsionar a eficiência e a produtividade de equipes de desenvolvimento.



by **Leonardo Arthur Scaravonatto Zanardi**

O que é controle de versão?

Um controle de versão é um sistema que acompanha e registra as alterações feitas em arquivos ao longo do tempo. Ele permite a recuperação de versões anteriores dos arquivos, facilita o trabalho colaborativo e ajuda a evitar conflitos em projetos. Exemplos incluem Git, Subversion e Mercurial.

Importância do Controle de Versão na Colaboração de Projetos

Rastreamento de Mudanças

O controle de versão permite acompanhar todas as alterações feitas em um projeto, facilitando a identificação de erros e a reversão de modificações indesejadas.

Colaboração em Equipe

Múltiplos desenvolvedores podem trabalhar simultaneamente no mesmo projeto, sincronizando suas contribuições de maneira organizada.

Backup e Recuperação

O controle de versão atua como um sistema de backup, possibilitando a restauração de versões anteriores do projeto em caso de acidentes ou problemas.

Importância do Controle de Versão na Colaboração de Projetos

Rastreamento de Mudanças

O controle de versão permite acompanhar todas as alterações feitas em um projeto, facilitando a identificação de erros e a reversão de modificações indesejadas.

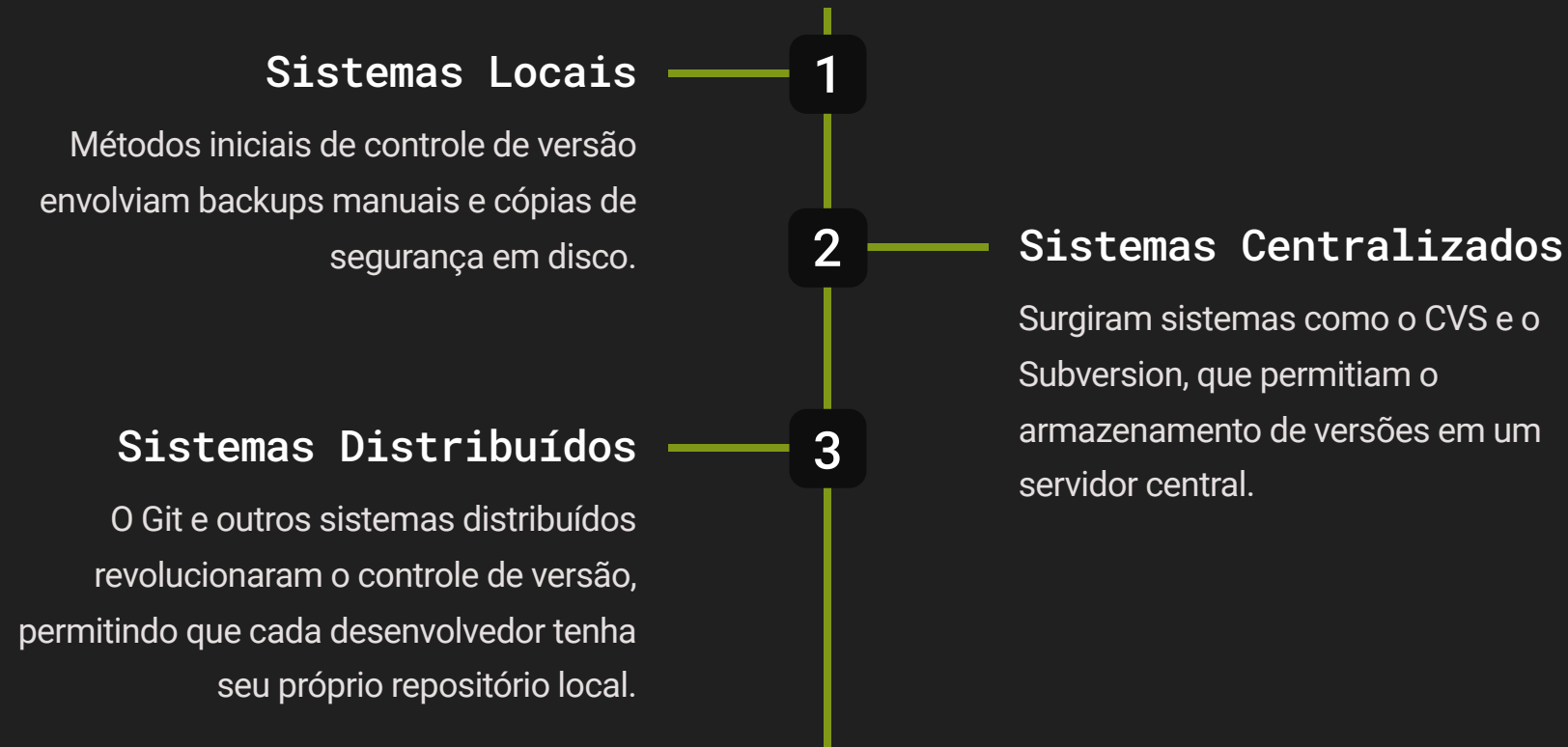
Colaboração em Equipe

Múltiplos desenvolvedores podem trabalhar simultaneamente no mesmo projeto, sincronizando suas contribuições de maneira organizada.

Backup e Recuperação

O controle de versão atua como um sistema de backup, possibilitando a restauração de versões anteriores do projeto em caso de acidentes ou problemas.

Evolução do Controle de Versão



Conceitos Básicos do Git

1 Repositório

O Git organiza as versões do projeto em um repositório, que pode estar no computador local ou em um servidor remoto.

2 Branches

Branches permitem que múltiplas linhas de desenvolvimento aconteçam simultaneamente, facilitando a experimentação e a resolução de conflitos.

3 Commits

Cada alteração feita no código é registrada como um commit, com uma mensagem descritiva e um hash único.

4 Merges

O Git permite a fusão de branches, combinando as modificações feitas em paralelo em uma única versão coerente.

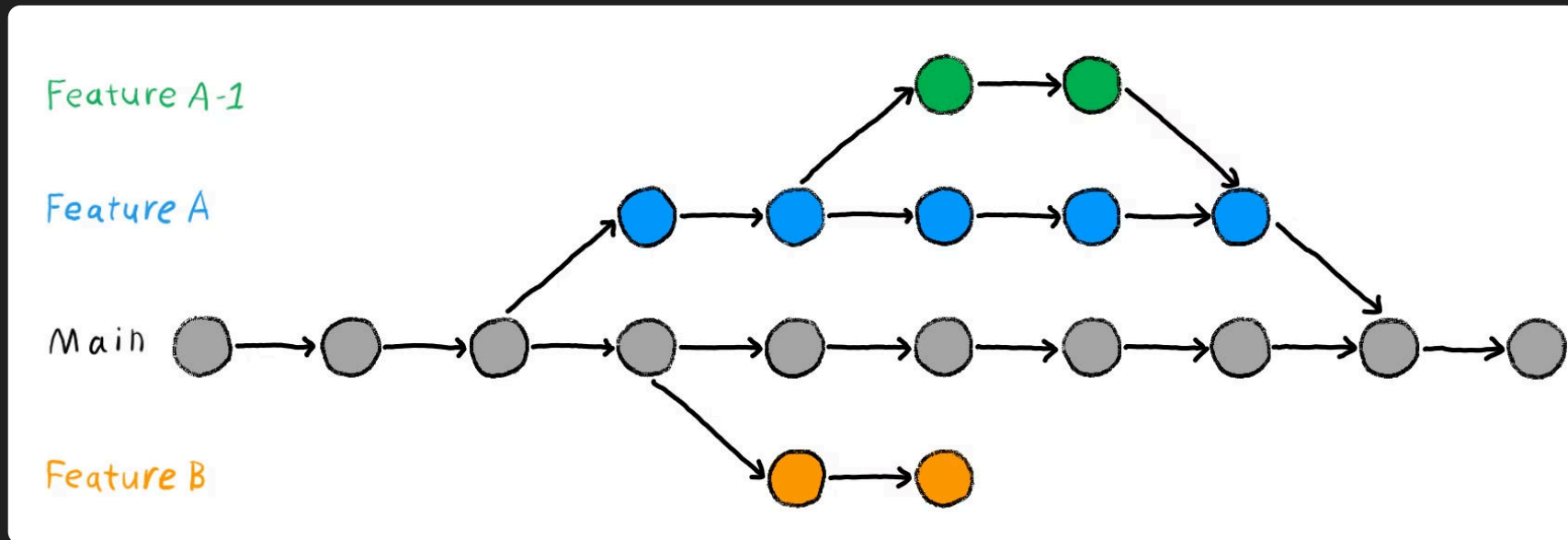
5 Main e Master

Branch principal, onde fica o código que está pronto. Normalmente é bloqueada, não sendo possível fazer um commit direto. Nesse caso, são abertos Merge Requests

6 Merge Requests (MRs)

Commits que finalizam um desenvolvimento realizado em uma branch, mergeando o código com a branch principal (main/master)

Imagem explicativa de conceitos



Principais Comandos do Git



Clone

Baixa uma cópia do repositório remoto para seu computador local.



Pull

Atualiza seu repositório local com o que está na branch branch



Add

Adiciona as alterações no estado de pre commit



Commit

Registra as alterações feitas no repositório local.



Push

Envia as alterações do repositório local para o remoto.

Exemplo de um fluxo de Trabalho com Git

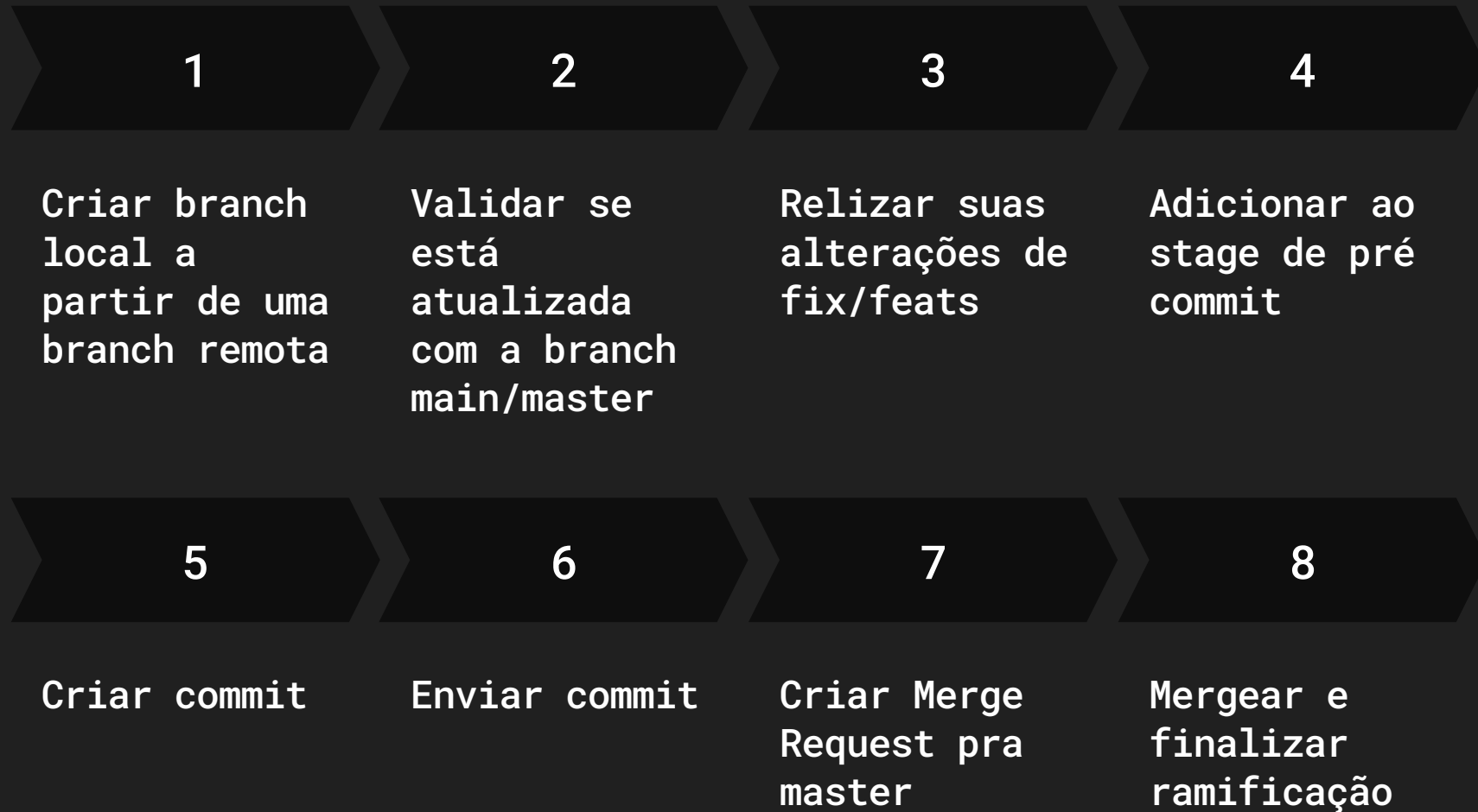
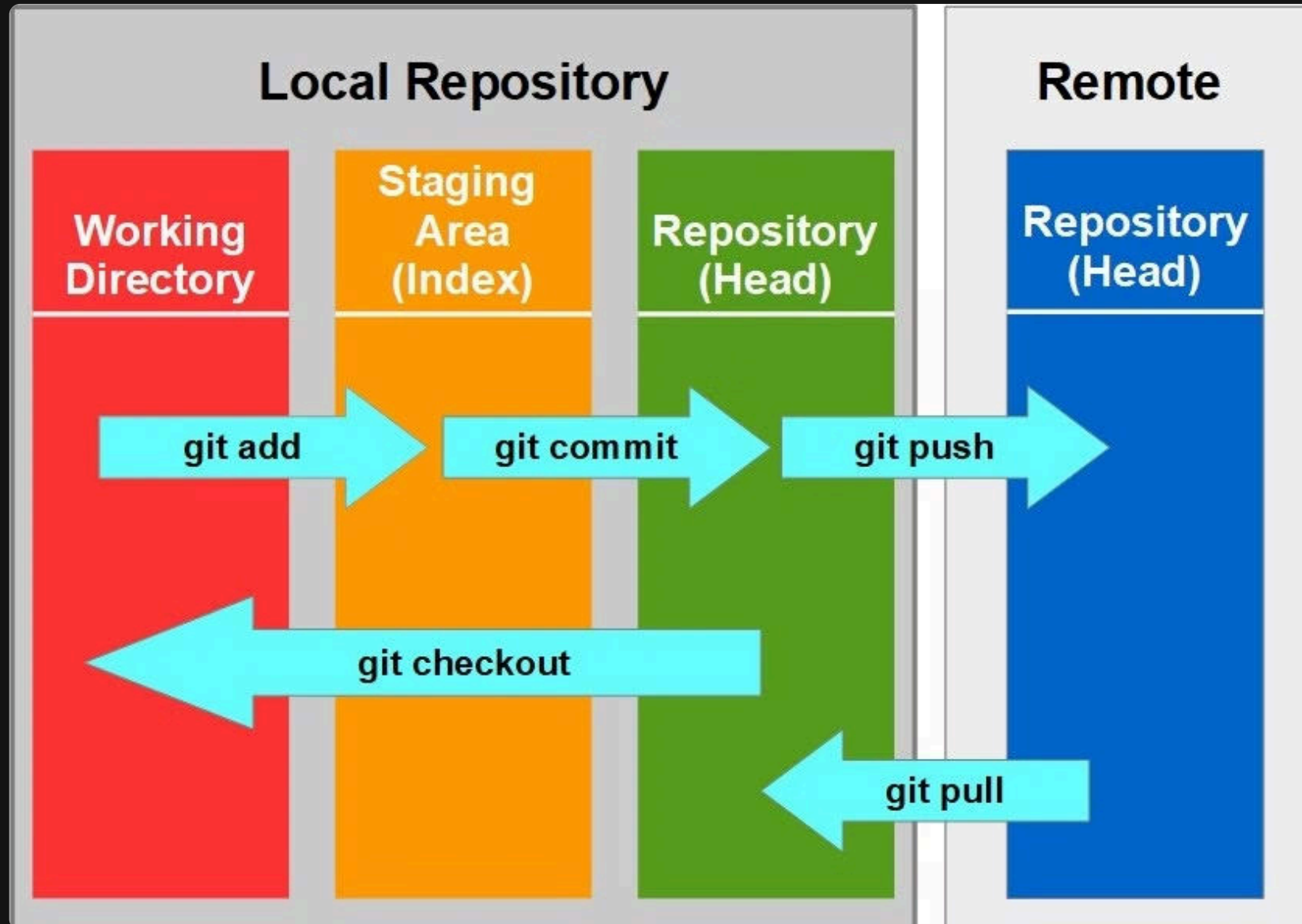


Diagrama de sequência de um fluxo



Boas Práticas e Dicas para Uso Efetivo do Git

Mensagens de Commit (Convencional commits)

Escreva mensagens claras e concisas que descrevam as alterações realizadas.

Branches Nomeadas

Use nomes significativos para suas branches, como "feature/nova-funcionalidade" ou "fix/correcao-bug".

Revisão de Código

Implemente revisões de código para garantir a qualidade e a consistência do projeto.

Repositórios Remotos e Plataformas de Hospedagem

GitHub

A maior plataforma de hospedagem de repositórios Git, com recursos avançados de colaboração e integração.

GitLab

Outra plataforma popular que oferece ferramentas de CI/CD, gerenciamento de projetos e muito mais.

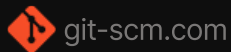
Bitbucket

Plataforma da Atlassian com ênfase em integração com outras ferramentas da empresa.

Azure DevOps

Serviço da Microsoft que combina controle de versão, planejamento de projetos e CI/CD.

Documentação extra:



Git - Downloading Package

The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

DanielHe4rt/ git4noobs

Aulão de Git para iniciantes

15 Contributors 1 Issue 1k Stars 104 Forks



GitHub - DanielHe4rt/git4noobs: Aulão de Git para iniciant...

Aulão de Git para iniciantes. Contribute to DanielHe4rt/git4noobs development by creating an account on GitHub.