

Git e GitHub na Prática para Desenvolvedores Modernos





**OSANAM
GIORDANE**



Head of Innovation
Cloud/DevOps Engineer



O que aprenderemos hoje



Fundamentos do Git

Conceitos básicos e comandos essenciais



GitHub na prática

Repositórios, Pull Requests e Colaboração



Fluxos de trabalho

Git Flow e GitHub Flow



Hands-on

Prática com exercícios reais



História



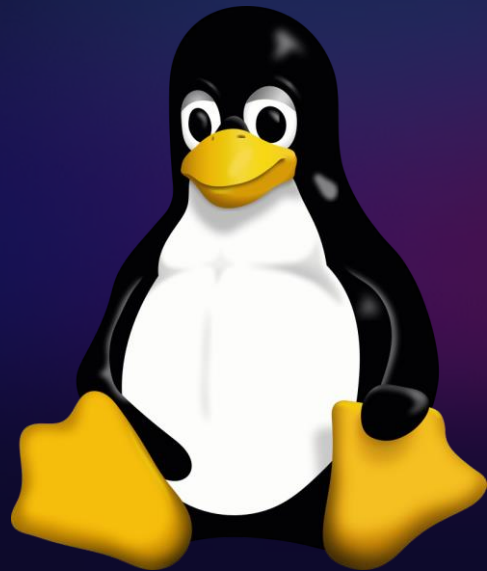
1

É um SCV centralizado baseado na ferramenta de controle de versão CVS

2

Além da grande quantidade de erros encontrados, obteve sucesso antes da chegada do Subversion.

LINUS TORVALDS



LINUX



O que é GIT

BitKeeper

Ferramenta antes utilizada para controle de versão no Linux, deixando de ser Open Source.

Melhor que as anteriores

Linus queria uma ferramenta que não atuasse como o CVS e erros do Subversion.

Principal Ferramenta

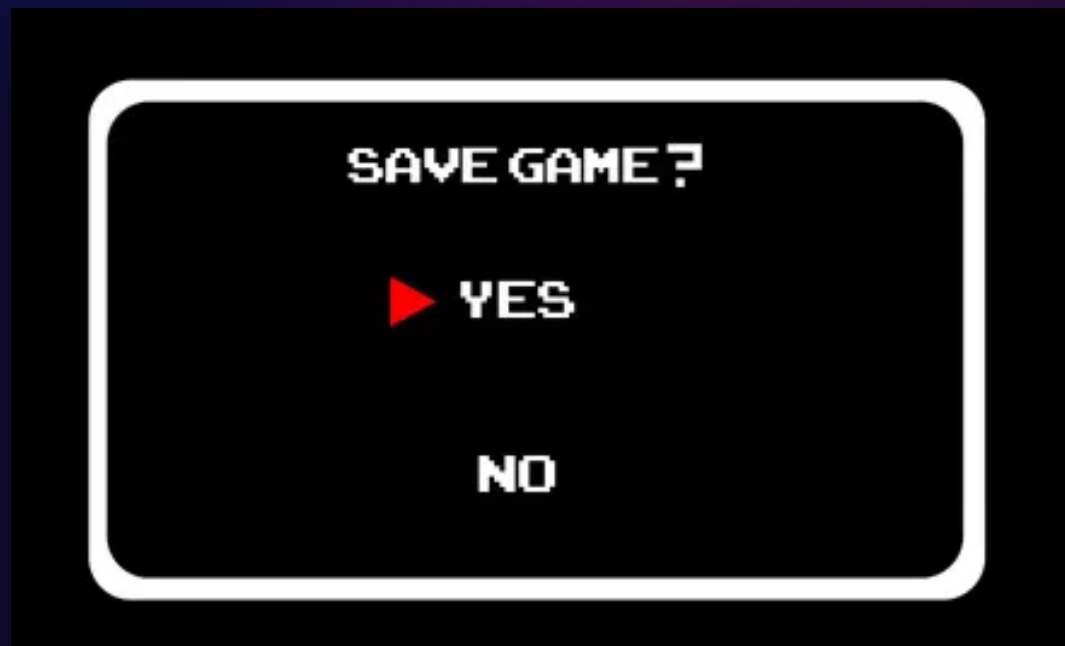
GIT é a ferramenta de controle de versão mais utilizada no mercado mundial.



Por que o GIT é tão importante?

Fazendo uma analogia de **games**, o software que desenvolvemos é o nosso '**jogo**' e o GIT o nosso '**save**', ele nos permitirá criar checkpoints e voltar em qualquer um deles sempre que quisermos.

Em resumo, o **GIT** te livra de fazer algumas besteiras.



Conceitos Fundamentais do Git

Repositório

Armazena todo o histórico do projeto e seus arquivos.

Commit

Snapshot do seu código em um momento específico.



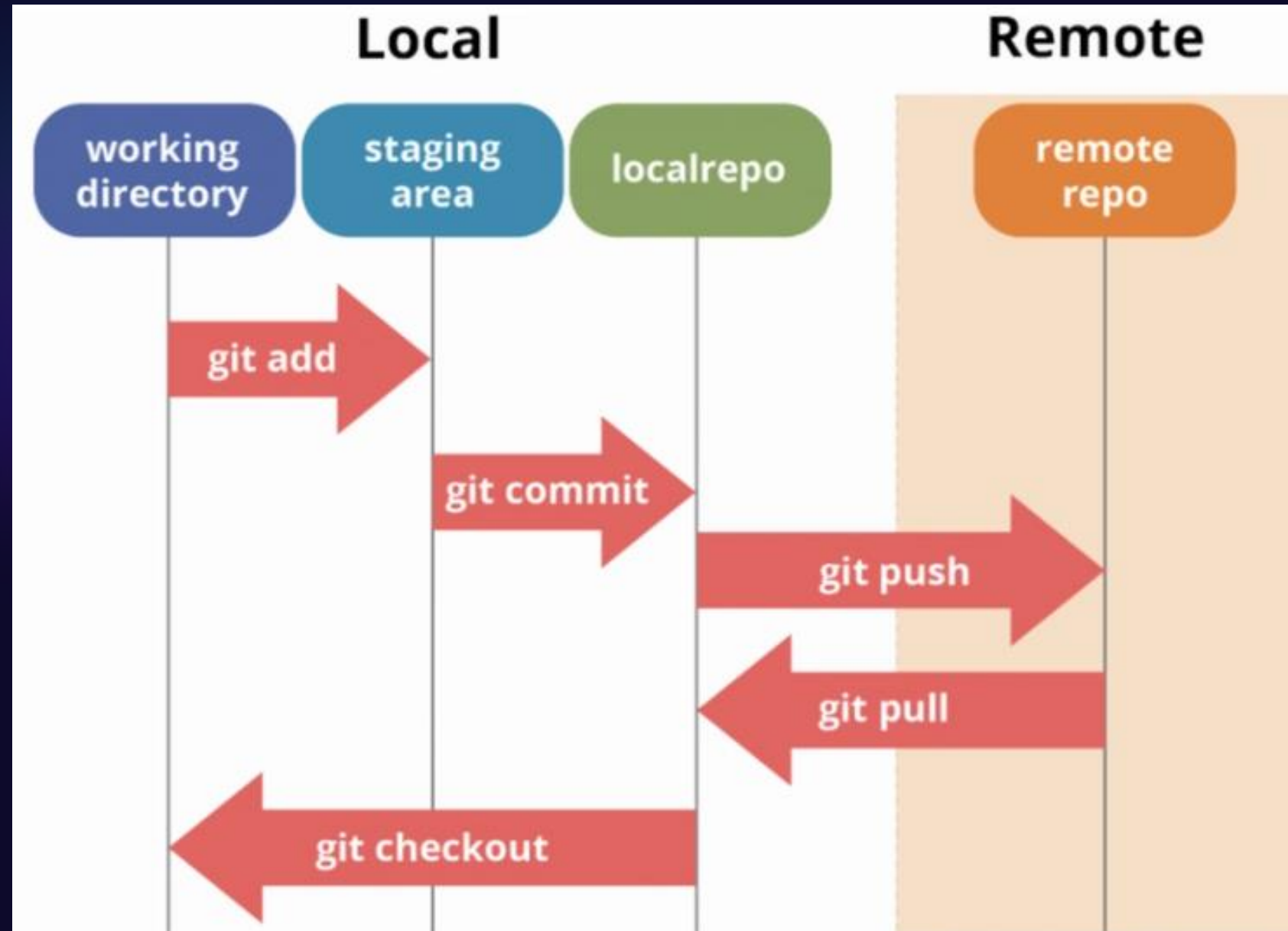
Working Directory

Onde você modifica os arquivos do seu projeto.

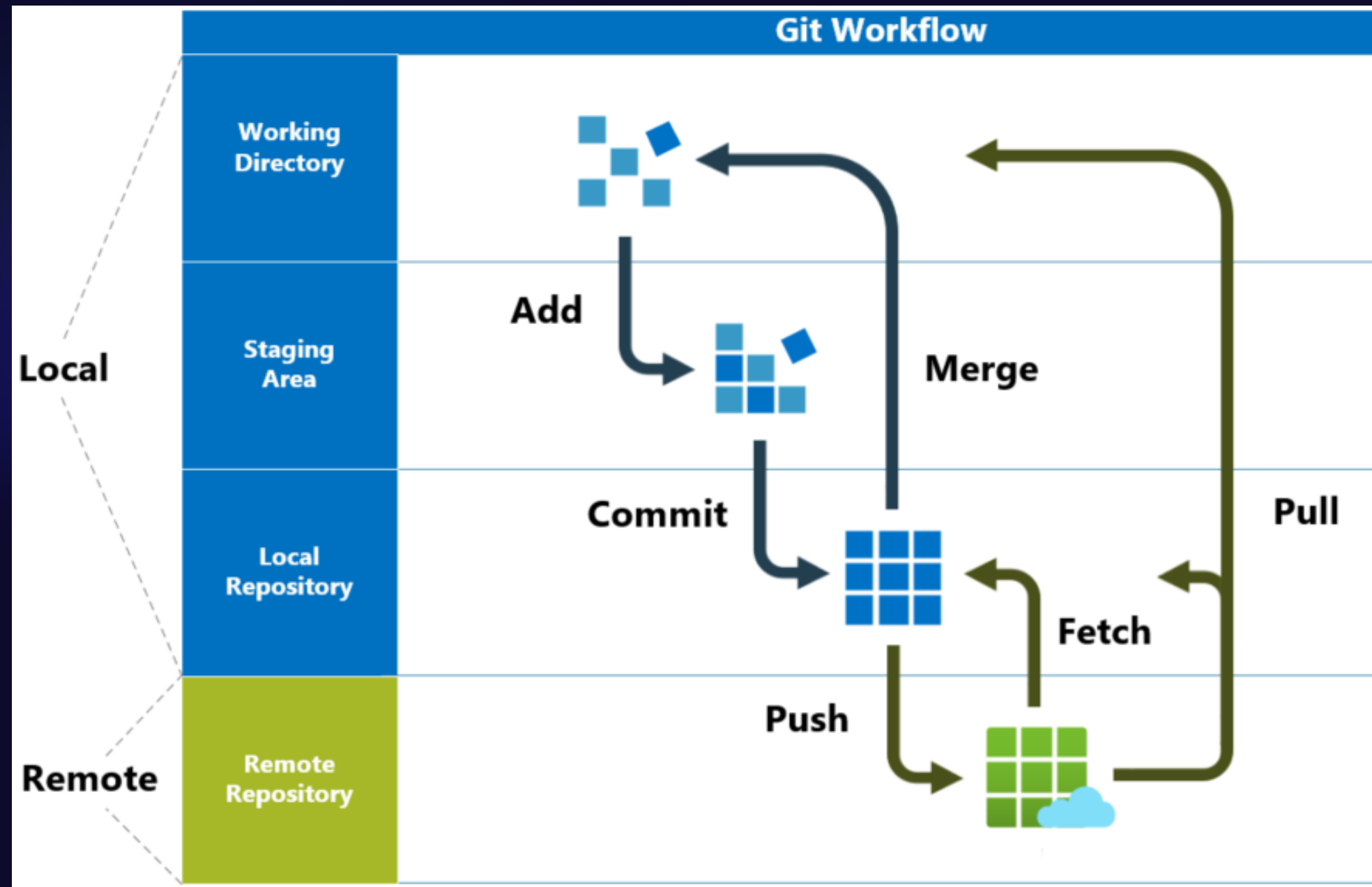
Staging Area

Área intermediária para preparar mudanças para commit.

Fluxo de trabalho com GIT



Fluxo de trabalho com GIT



Comandos Essenciais do Git

Configuração e Início

- git config
- git init
- git clone

Ciclo Básico

- git add
- git commit
- git status
- git log

Trabalho Remoto

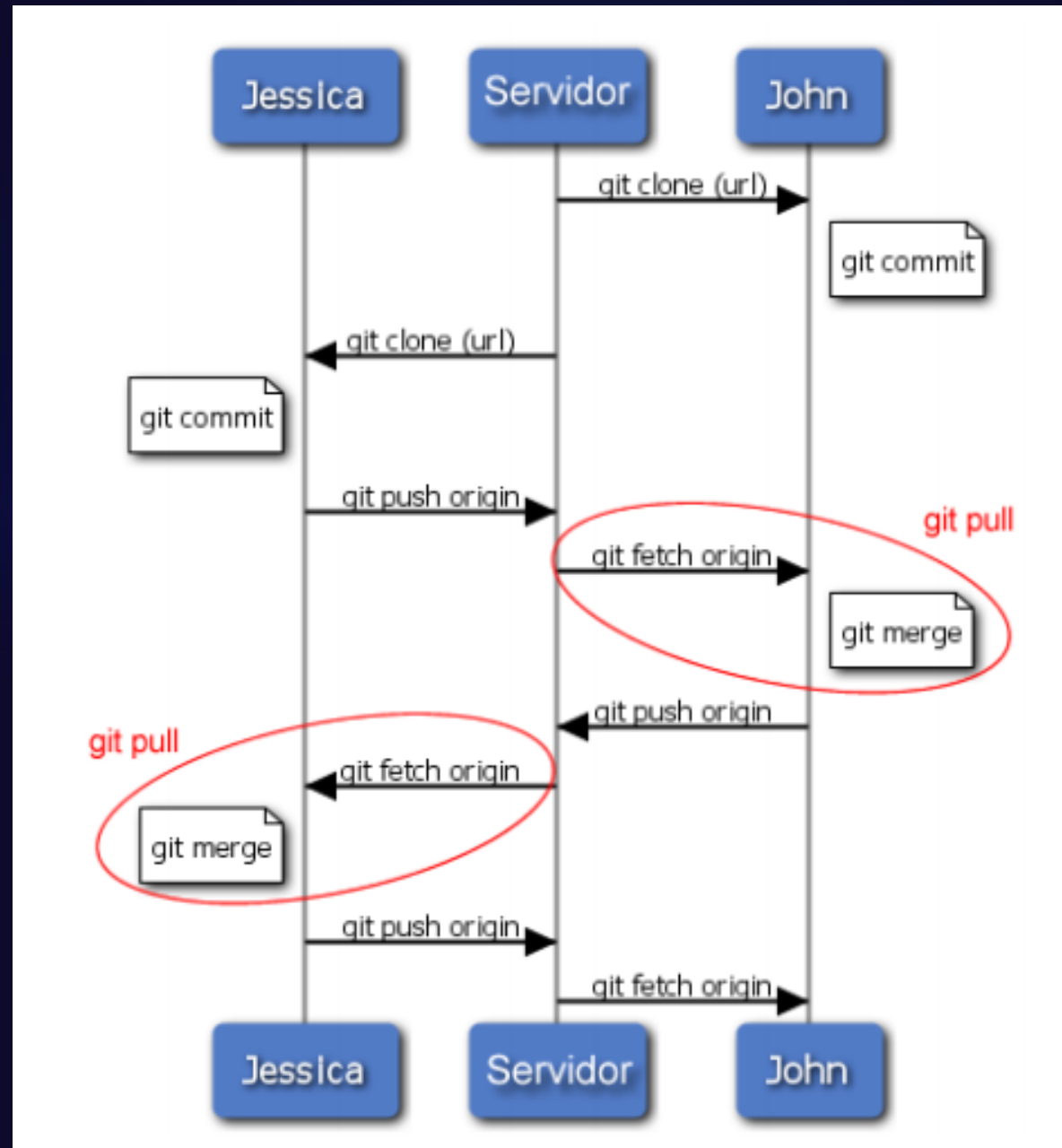
- git push
- git pull
- git fetch
- git remote

Branches

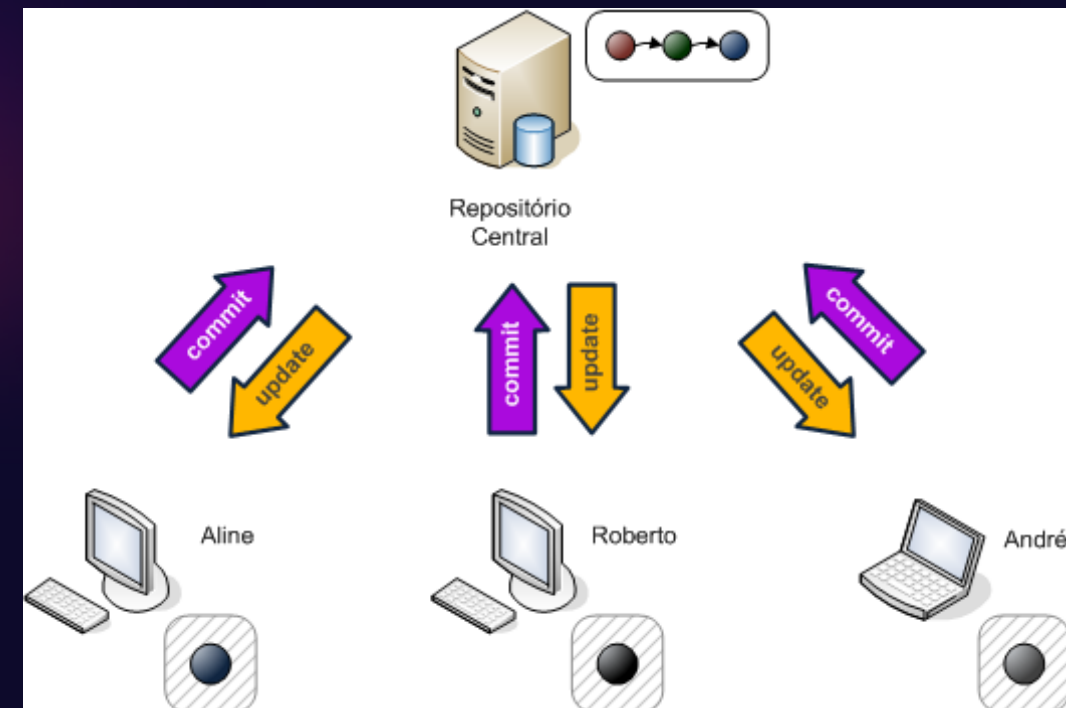
- git branch
- git checkout
- git merge
- git rebase

Trabalhando em Colaboração

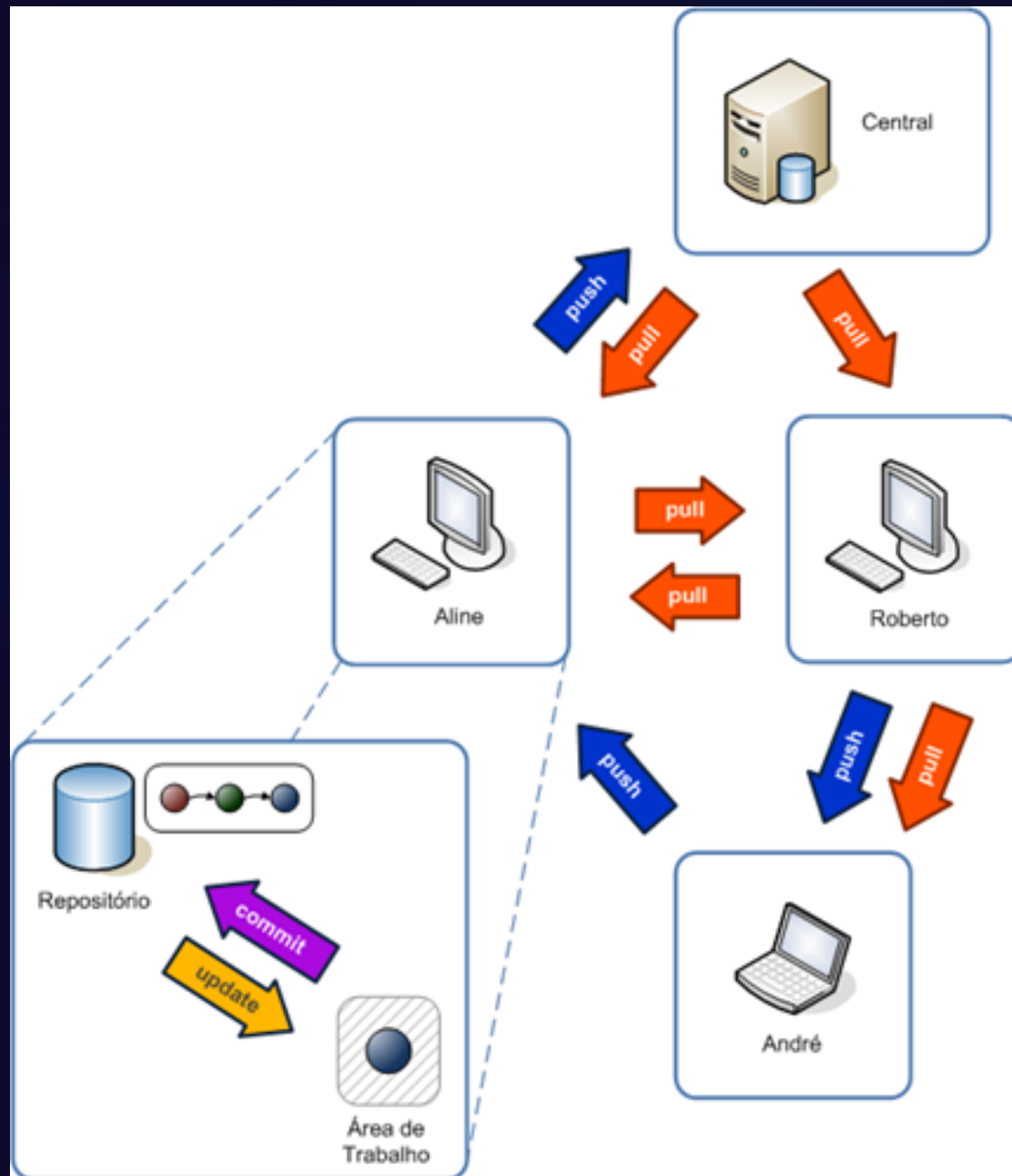
FLUXO DE TRABALHO



REPOSITÓRIO CENTRALIZADO (REMOTO)



Trabalhando em Colaboração



REPOSITÓRIO DESCENTRALIZADO

O QUE É

Branch

Branch, ou ramo em português, é uma cópia das linhas de código de um software gerenciada por um sistema de controle de versão (VCS). A ramificação serve para ajudar as equipes de desenvolvimento a consertar bugs e inserir funções, separando o trabalho em andamento do código testado e estável.



Sobre Banches (Ramificações)

Liberade para criar

Permitem que você desenvolva recursos, corrija erros ou experimente com segurança novas ideias em uma área contida do seu repositório.

Branch a partir de outra

Você sempre cria um branch a partir de um branch existente. Normalmente, você pode criar um novo branch a partir do branch-padrão do seu repositório.

Trabalho isolado

Você então poderá trabalhar nesse novo branch isolado das mudanças que outras pessoas estão fazendo no repositório.



GitHub: Além do Git

Repositórios

Armazenamento centralizado para projetos com interface visual amigável. Inclui gerenciamento de acesso e recursos extras como GitHub Pages.

Pull Requests

Mecanismo para propor mudanças e facilitar revisão de código. Permite discussões detalhadas sobre alterações antes da integração.

Issues e Projects

Rastreamento organizado de tarefas e bugs com labels e milestones. Quadros kanban para visualizar e gerenciar o fluxo de trabalho.

Fluxos de Trabalho com Git



Git Flow

Ideal para releases planejadas e versões estáveis

2

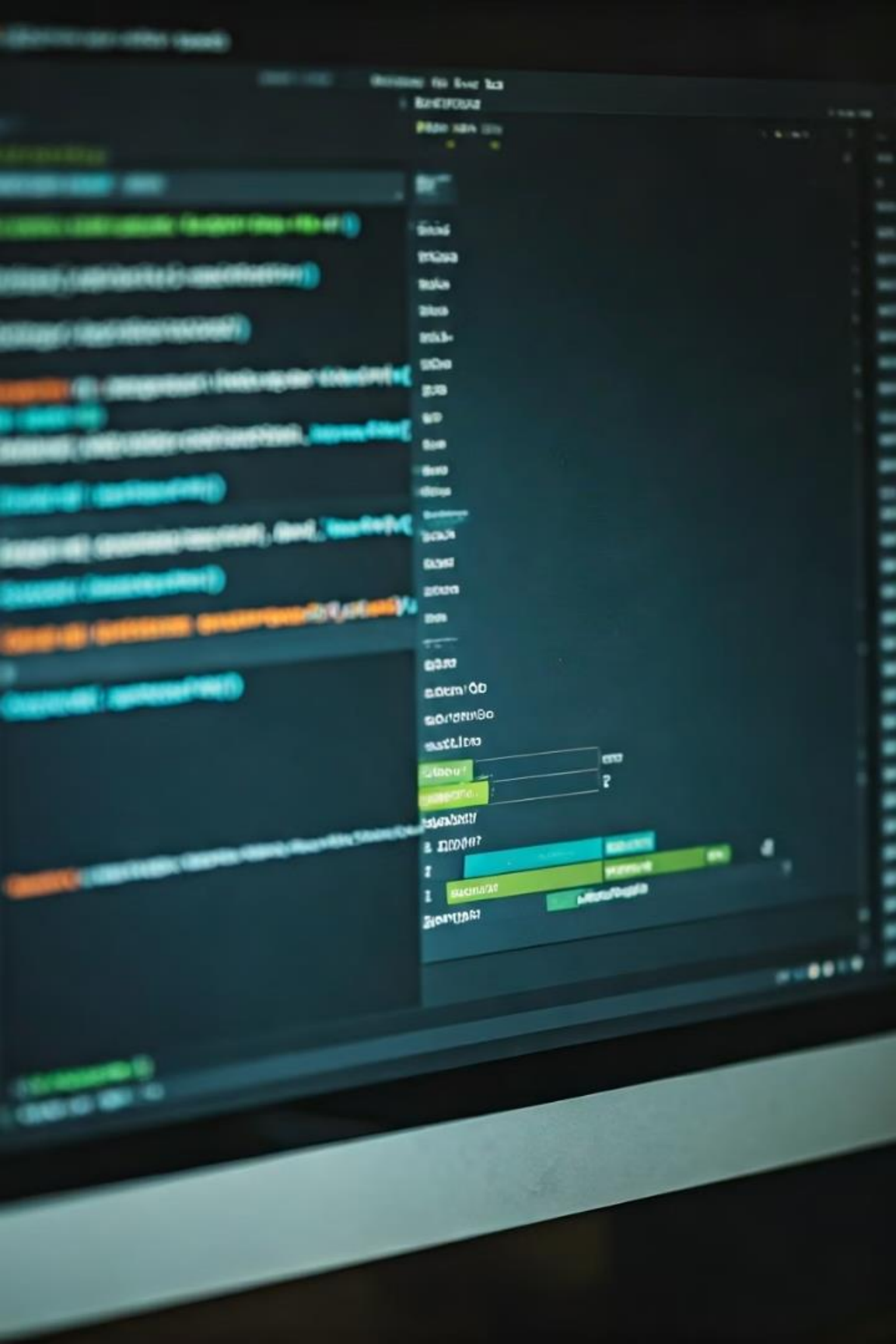
GitHub Flow

Simplicidade para deploy contínuo



Trunk-Based

Minimiza branches para integração contínua



Resolvendo Conflitos

Identifique o conflito

Git marca os arquivos conflitantes com marcadores especiais (<<<<<<, =====, >>>>>>).

Edite os arquivos

Decida quais mudanças manter, removendo os marcadores de conflito.

Marque como resolvido

Use git add para marcar arquivos como resolvidos.

Complete o merge

Finalize com git commit para completar o processo de merge.

Dúvidas



Exercício Prático



Configurar Git

Configure seu nome e email no Git. Clone o repositório de exemplo.



Criar branch e modificar

Crie uma branch para sua feature. Modifique arquivos e faça commits.



Enviar para GitHub

Push da branch para o repositório remoto. Crie um Pull Request.



Revisar e colaborar

Faça revisão do código de um colega. Resolva conflitos se necessário.



OBRIGADO

codefc.io

youtube.com/@codefc



CODE FC

