



Minicurso Git & GitHub

Controle de Versão na
Prática

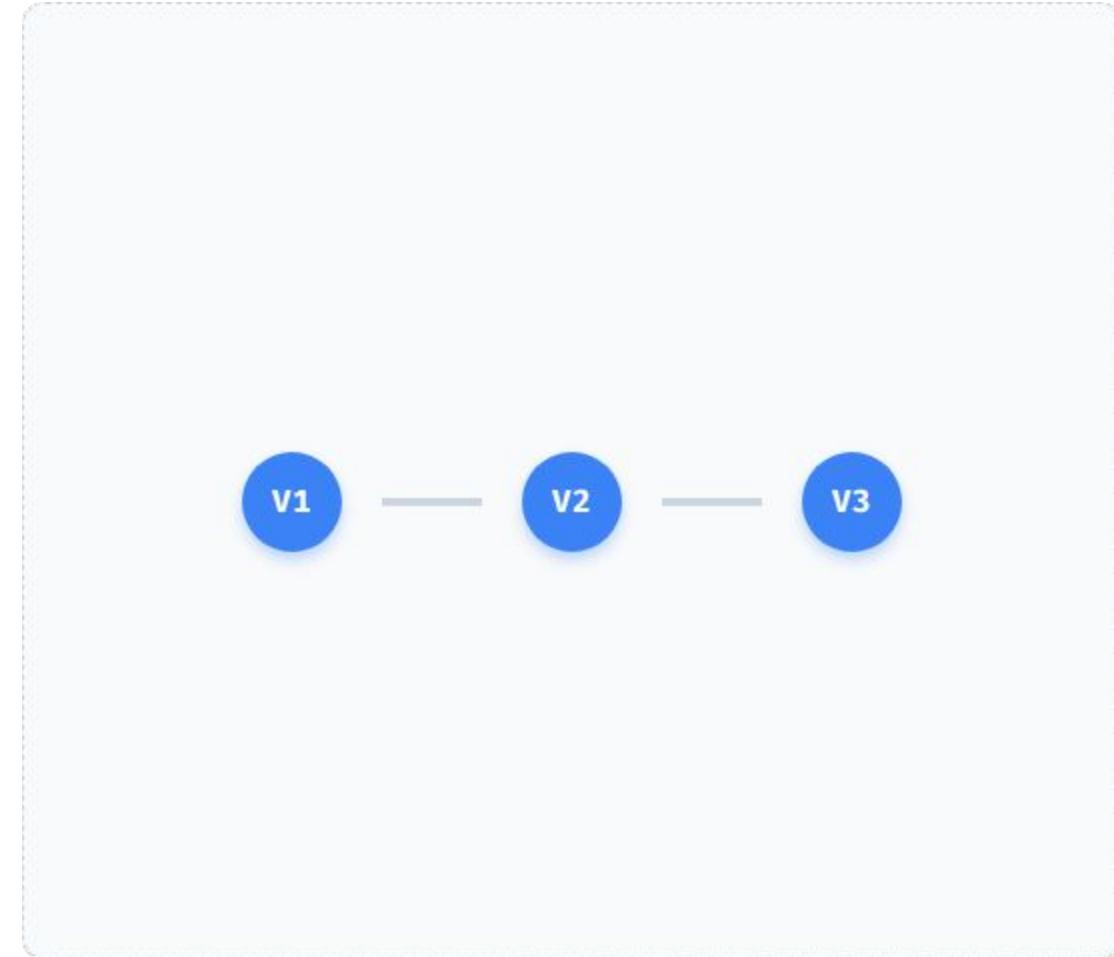
Instrutora: Lavínia

Baseado no material do Prof. André Restivo

Aula 1: Introdução

O que é Controle de Versão (VCS)?

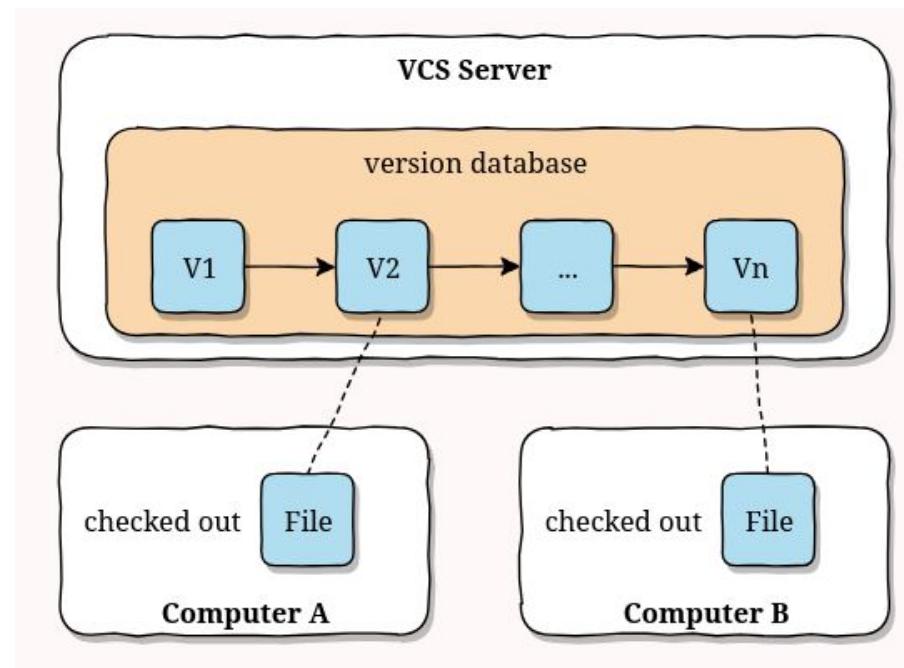
- ✓ Um sistema que registra mudanças em arquivos ao longo do tempo.
- ✓ Permite **reverter** arquivos ou o projeto inteiro para um estado anterior.
- ✓ Permite **comparar** mudanças ao longo do tempo.
- ✓ Identifica **quem** modificou algo e quando.
- ✓ Também conhecido como SCM (Source Control Management).



Centralizado vs. Distribuído

Centralizado (ex: SVN)

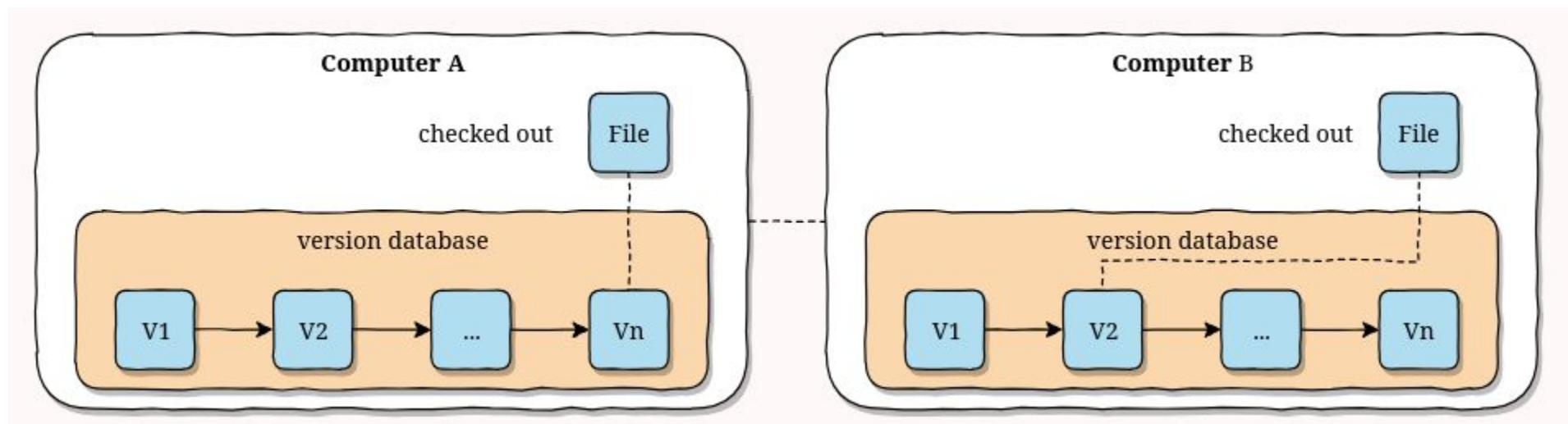
- Um único servidor contém todos os arquivos.
- Falha única compromete tudo.



Centralizado vs. Distribuído

Distribuído (ex: Git, Mercurial, Bazaar, Darcs)

- Clientes espelham totalmente o repositório.
- Não há diferença entre servidor e cliente.



A Grande Diferença: Git ≠ GitHub

Git (O Motor)

Software local, gratuito e open-source.

Roda no terminal (linha de comando) do seu próprio computador.

Não precisa de internet. Pode ser usado sozinho e offline para sempre.

GitHub (O Hospedeiro)

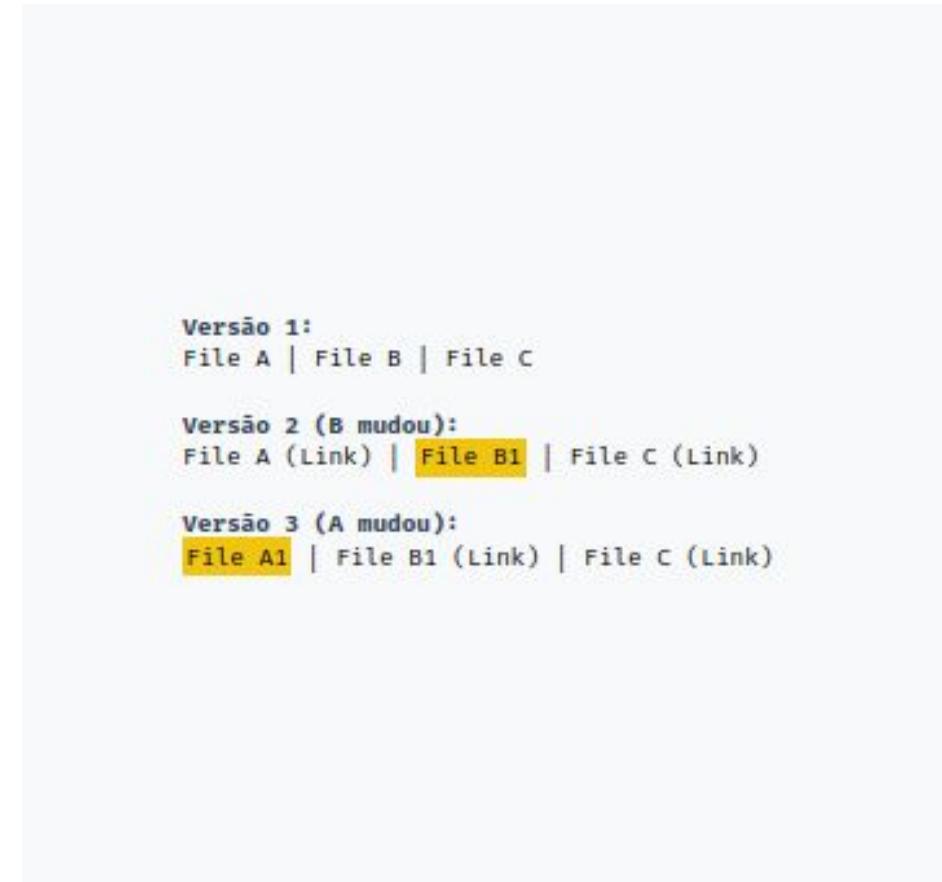
Serviço na nuvem gerido pela Microsoft.

Hospeda repositórios Git online e oferece ferramentas visuais.

Facilita muito o trabalho em equipe (Pull Requests, Revisão, Issues).

Conceitos Básicos: Snapshots

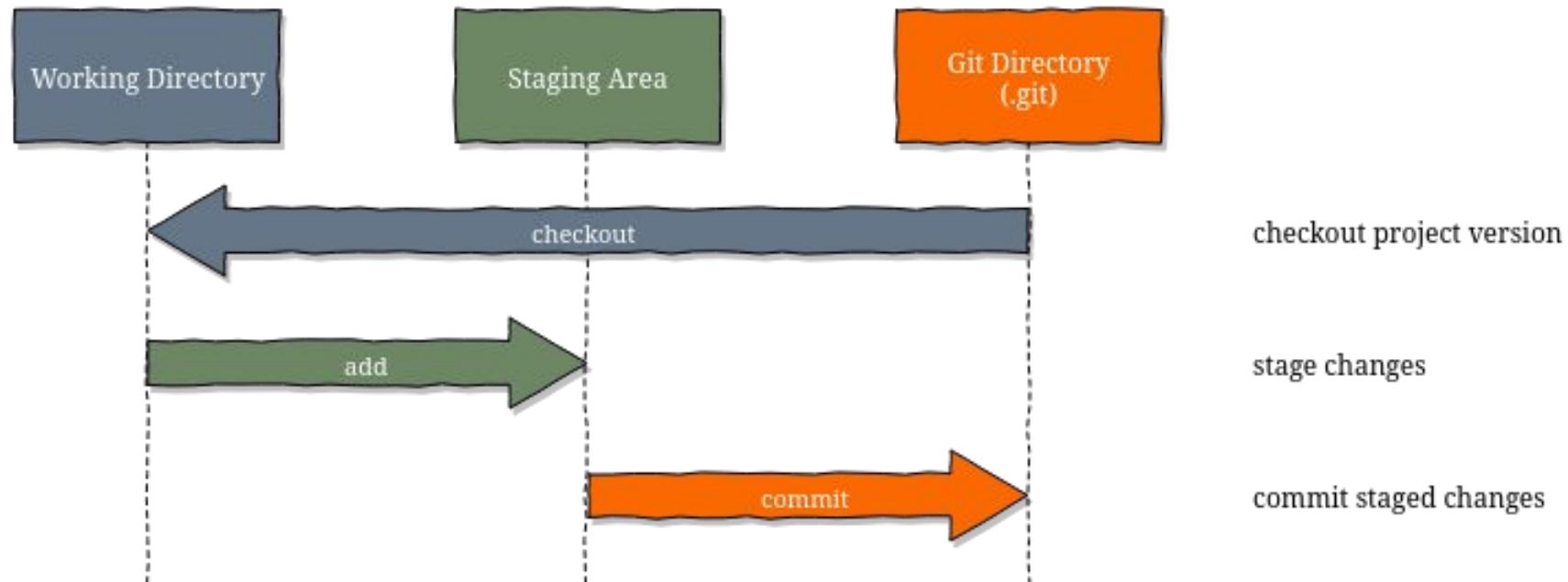
- ✓ O Git pensa nos dados como um conjunto de instantâneos (snapshots) de um mini sistema de arquivos.
- ✓ Cada **commit** é uma "foto" de **todos os seus arquivos naquele momento**.
- ✓ Se um arquivo não mudou, ele apenas cria um link para a versão anterior (eficiência).
- ✓ **Integridade:** Tudo é identificado por um hash SHA-1 (ex: 7e16b5...).



As 3 Áreas do Git



As 3 Áreas do Git



Ciclo de Vida dos Arquivos



Untracked

Novo arquivo, o Git ainda
não conhece.



Unmodified

Arquivo rastreado sem
alterações recentes.



Modified

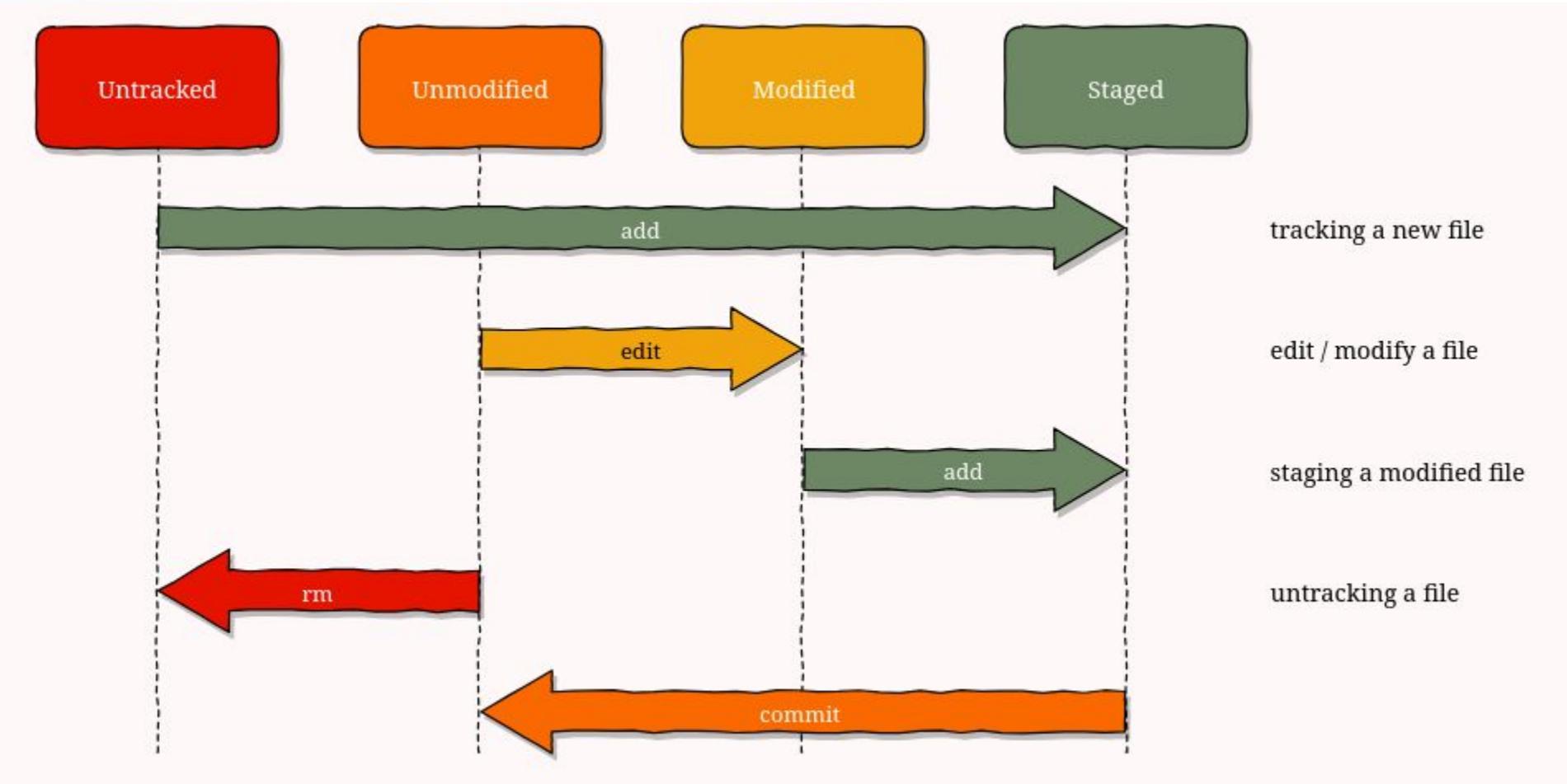
Arquivo alterado, mas não
preparado.



Staged

Preparado para o próximo
commit.

Ciclo de Vida dos Arquivos



Fluxo Básico de Comandos

- > **git init:** Transforma uma pasta comum em um repositório Git local, criando o diretório oculto `.`.
- + **git add :** Move as mudanças dos seus arquivos do Working Directory para a Staging Area. Use `git add .` para adicionar todos.
-  **git commit -m "mensagem":** Tira a "foto" (snapshot). Move o que está no Staging para o Re却itório de forma permanente com uma nota explicativa.
-  **git status:** Comando de verificação. Mostra exatamente em que estado os seus arquivos se encontram (Untracked, Modified ou Staged).
-  **git log:** Exibe a linha do tempo do projeto. Mostra o autor, a data e a mensagem de todos os commits realizados.