



Introdução ao Git

Vinícius Lima

vinicius@infomach.com.br

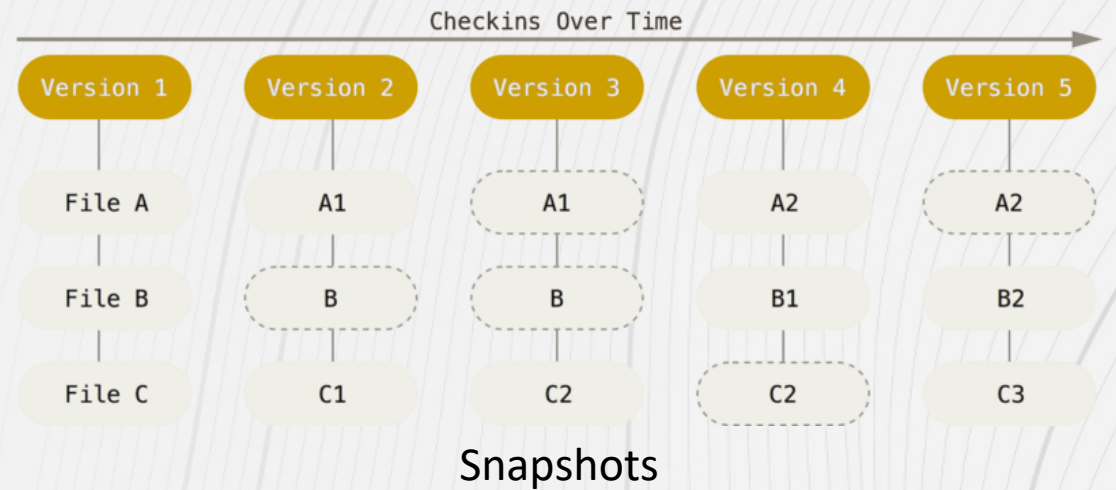
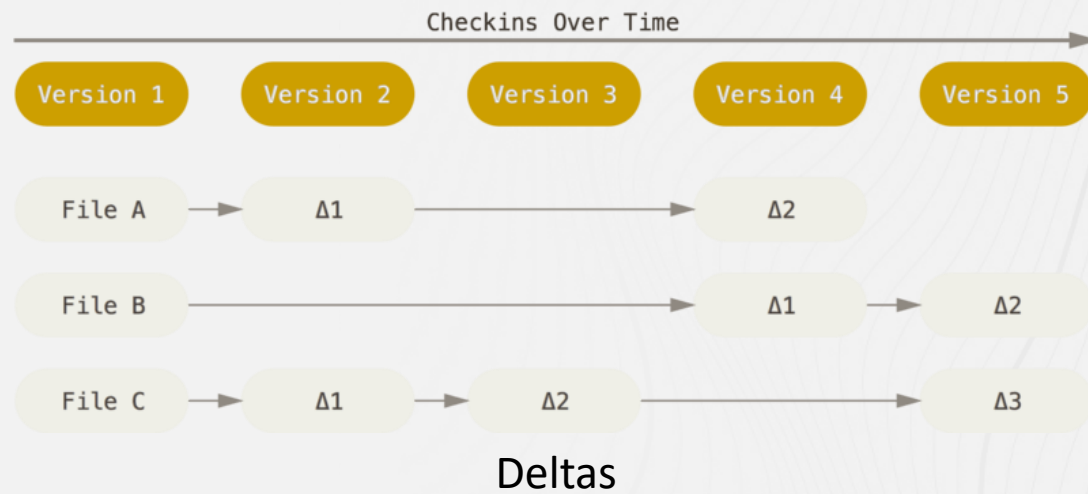
O que é “controle de versão”?

- Controle de versão é um sistema que armazena as mudanças de um arquivo ou conjunto de arquivos em uma linha do tempo.
- Permite retornar um arquivo ou todo um projeto a um estado anterior.
- Comparar mudanças que ocorreram durante a linha do tempo.
- Um histórico de quem modificou o que no projeto.

O que é Git?

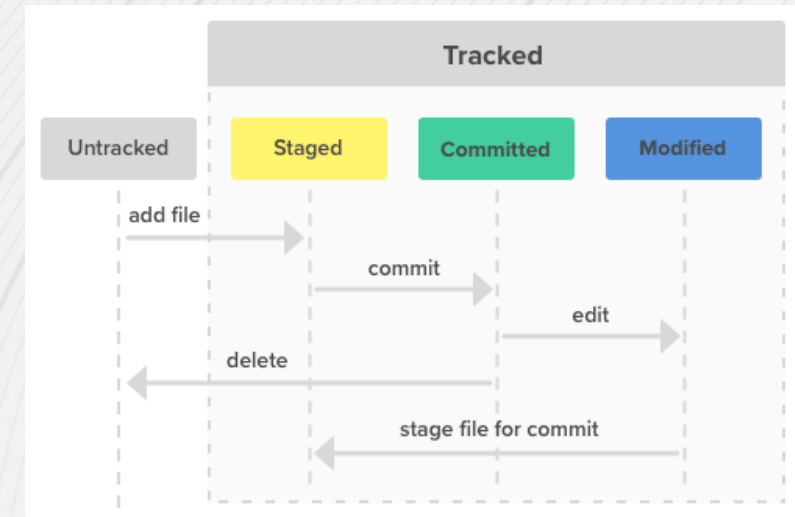
- Git é um sistema de controle de versões distribuído, usado para registrar o histórico de edições de qualquer tipo de arquivo.
- Cada diretório de trabalho do Git é um repositório com um histórico completo e habilidade total de acompanhamento das revisões.
- Não depende de acesso a uma rede ou a um servidor central.

Snapshots, não diffs!!



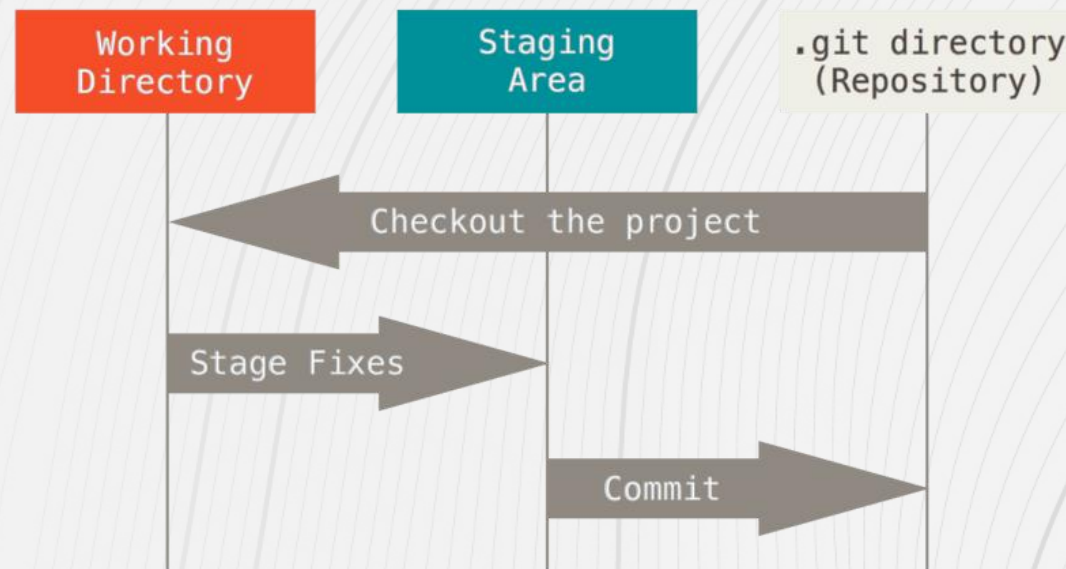
“Os Três Estágios”

1. Modificado: arquivo modificado mas não enviado (commit) a base de dados local.
2. Staged: o arquivo modificado foi marcado (add) para fazer parte do próximo snapshot.
3. Enviado (committed): o dado está salvo na base de dados local.



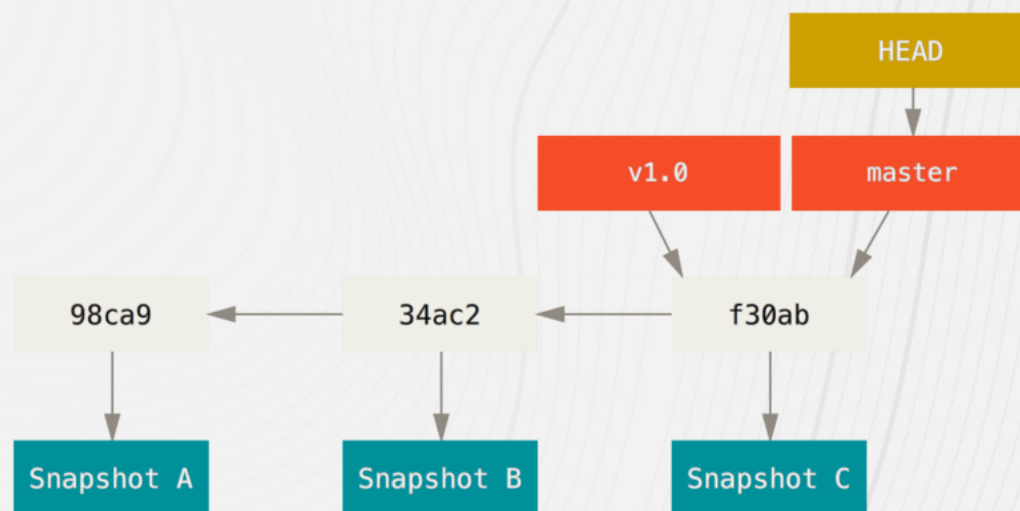
As três principais seções de um projeto Git

- Working tree: um ponto único na versão do projeto.
- Staging area: um arquivo no diretório Git com informações sobre o que irá no próximo commit. Também chamado de *index*.
- Diretório Git: onde ficam armazenados os metadados e objetos do projeto.

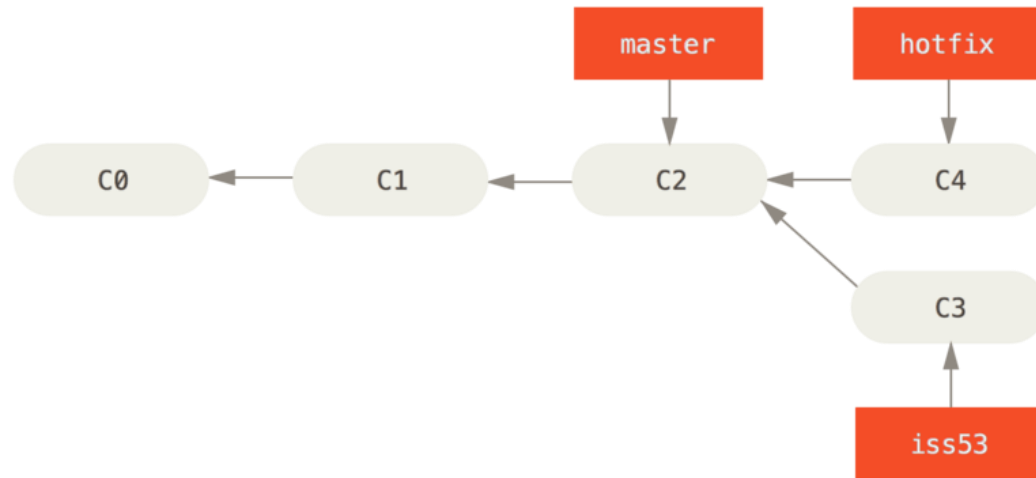


Git branching

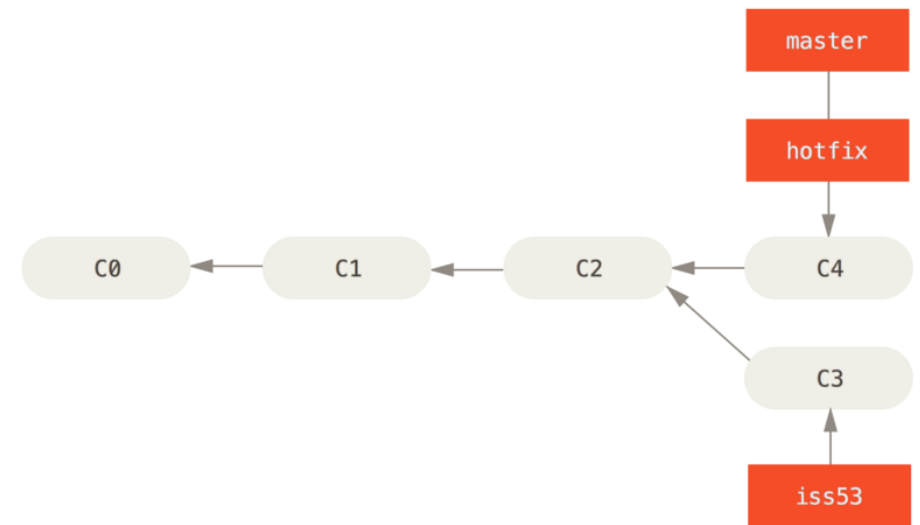
- Uma branch em Git é um simples ponteiro móvel para um dos commits.
- A branch padrão geralmente é chamada “master”.
- Toda vez que você faz um commit na branch, o ponteiro se move para frente.



Branching e Merging

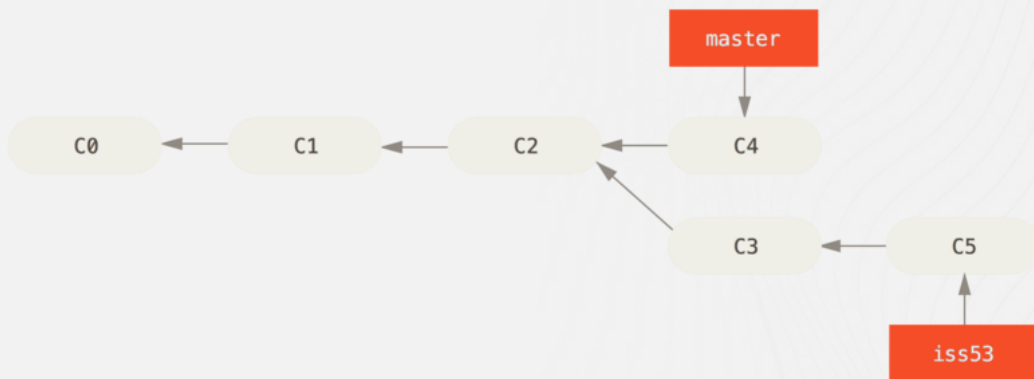


Novas branches

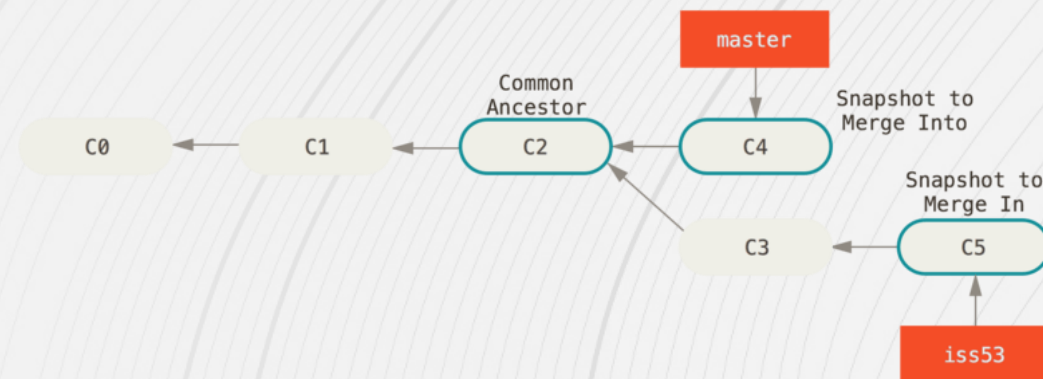


Fast-forward

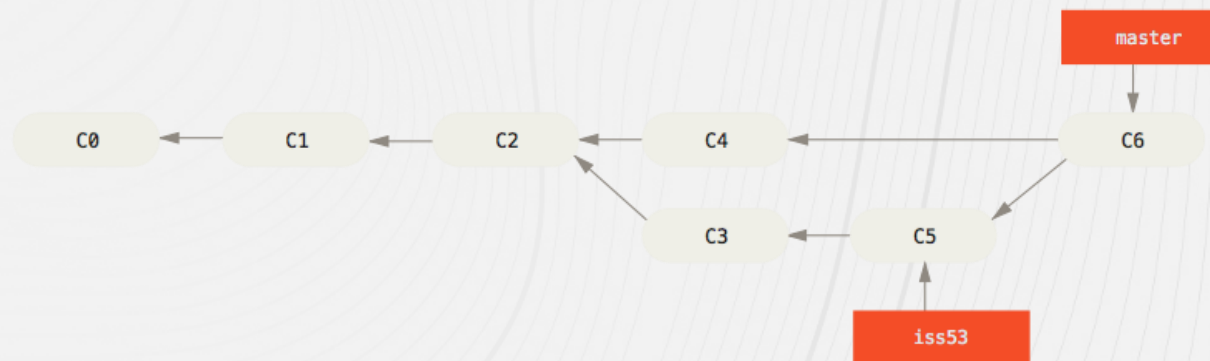
Branching e Merging



Continuando trabalho na branch



Histórico de desenvolvimento divergente



Estratégia "recursiva"

Principais comandos

git config -global user.name "[name]"

git config -global user.email "[email]"

git init [repo]

git clone [url]

git add [file]

git add *

git add .

git commit -m "[Commit message]"

git reset [file]

git reset [commit]

git status

git rm [file]

git log

git branch

git branch [branch name]

git branch -d [branch name]

git checkout

git checkout -b [branch name]

git remote add [local name] [remote server link]

git push [local name] [branch]

git push -all [local name]

git fetch

git merge [source branch]

git pull [repository]

Pro Git Book

2º edition – Scott Chacon e Ben Straub



Vinícius Lima
vinicius@infomach.com.br