



Jonathan da Cruz

Controle de versão - GIT

Uma breve história de como tudo começou



Tópicos

- [Sobre o curso](#)
- [História](#)
- [Local Version Control System](#)
- [Centralized Version Control](#)
- [Distributed Version Control](#)



Jonathan da Cruz

Sobre o curso

- Git é um sistema de controle de versão de código aberto e amplamente utilizado.
- Permite que você mantenha o histórico de todas as alterações realizadas em seu projeto
- Com Git, você pode facilmente reverter alterações, resolver conflitos e colaborar com segurança em projetos

[Voltar ao slide de tópicos](#)



Por que controlar a versão do projeto?

[Voltar ao slide de tópicos](#)

História

1982

Sistema para controle local de arquivos. Famoso na época RCS manipulava apenas arquivos isolados.

2000

Lançamento do Subversion que possui um servidor centralizado.

2005

Lançamento do GIT. A partir de então equipes podem trabalhar facilmente de forma colaborativa

Atualmente

Empresas do mundo todo compartilham recursos humanos em ambientes cada vez mais colaborativos.



[Voltar ao slide de tópicos](#)

Tipos de Versionamento de código ao longo da história.

[Voltar ao slide de tópicos](#)

Local Version Control System

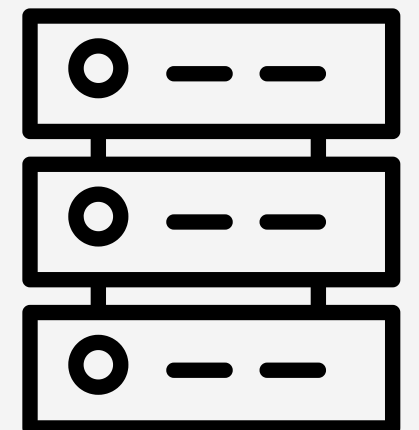
Alto risco/ Alta facilidade

Copiar arquivos para outro diretório mantendo nomes diferentes para releases. Pode apagar sem nenhum controle e dificilmente ocorre a reversão.

Utilização do RCS

Um utilitário lançado para manipular arquivos individuais, instalava um banco de dados local e foi uma evolução para reverter alterações.

Computador local



[Voltar ao slide de tópicos](#)

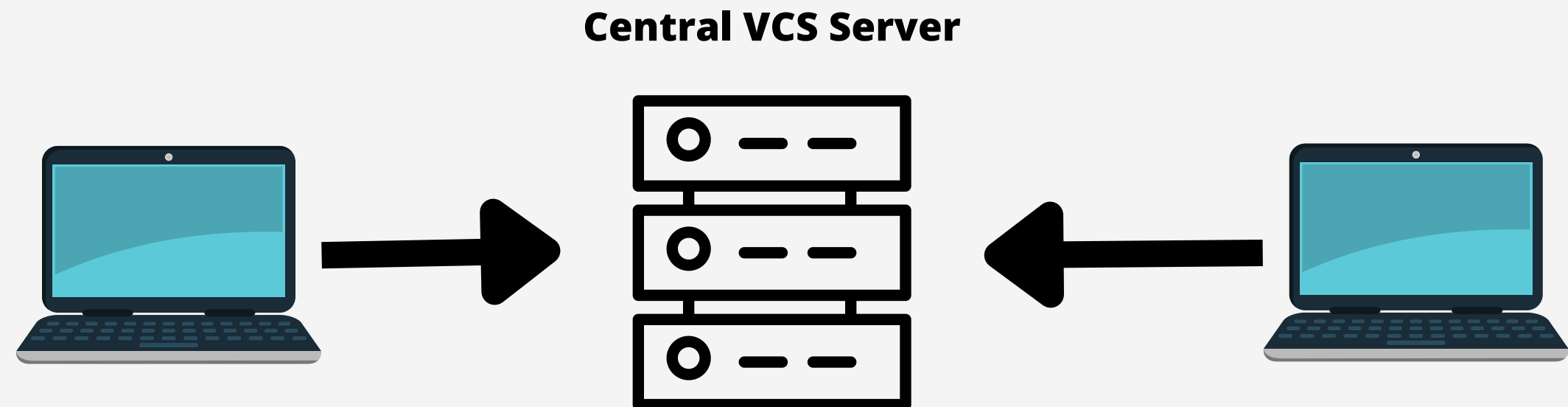
Centralized Version Control System

Colaborativo

Por muitos anos essa foi a solução padrão. Com o passar do tempo os problemas de queda do servidor central eram um problema pois ninguém conseguia publicar o código. Um exemplo Apache Subversion

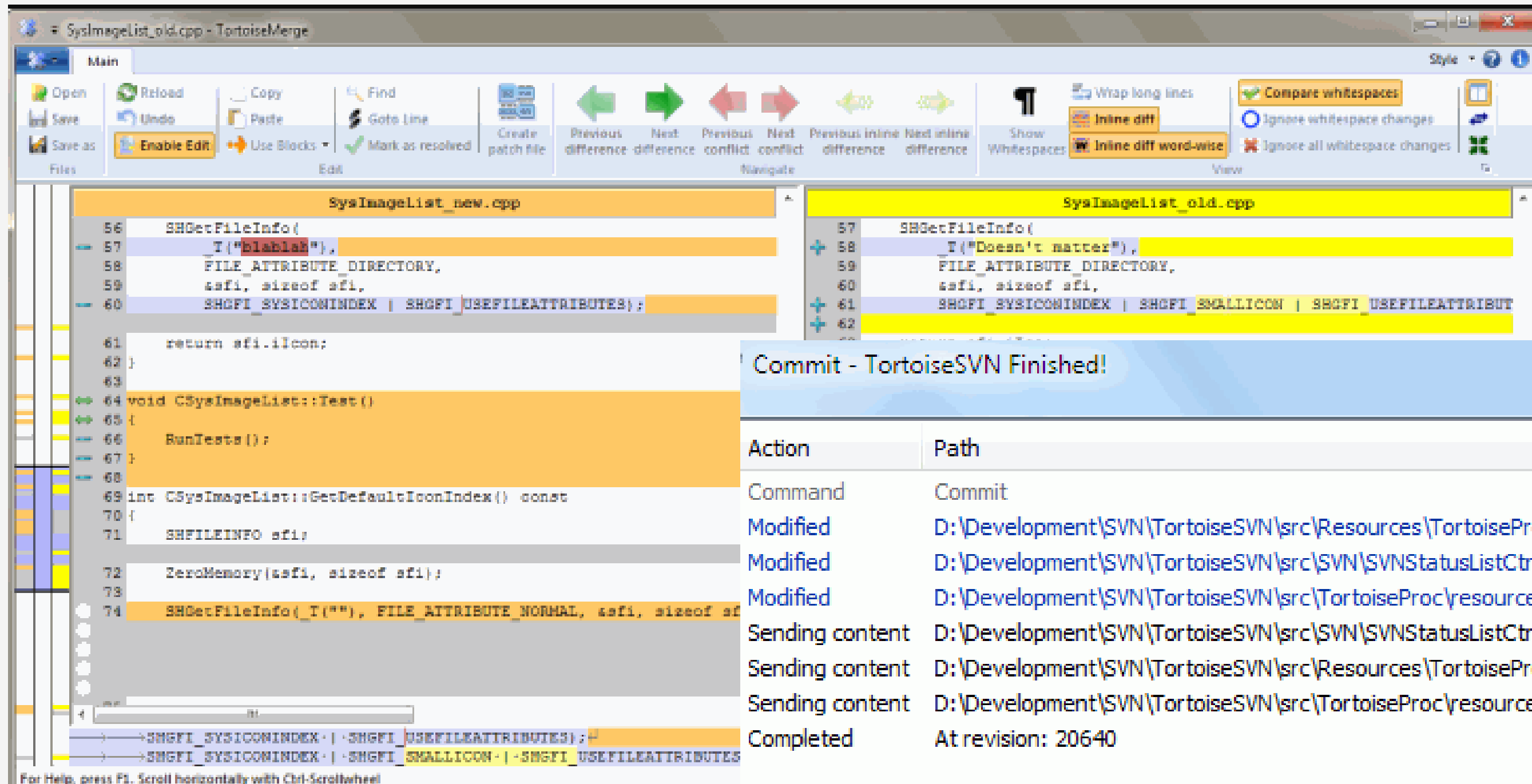
Servidor Centralizado

Ele possui um único servidor central que contem todos os arquivos e seu versionamento e assim quem precisa trabalhar com esses arquivos pode apenas fazer um check e buscar a versão atualizada desse servidor central



[Voltar ao slide de tópicos](#)

TortoiseSVN - Client Windows



Commit - TortoiseSVN Finished!

Action	Path	Mime type
Command	Commit	
Modified	D:\Development\SVN\TortoiseSVN\src\Resources\TortoiseProcENG.rc	
Modified	D:\Development\SVN\TortoiseSVN\src\SVN\SVNStatusListCtrl.cpp	
Modified	D:\Development\SVN\TortoiseSVN\src\TortoiseProc\resource.h	
Sending content	D:\Development\SVN\TortoiseSVN\src\SVN\SVNStatusListCtrl.cpp	
Sending content	D:\Development\SVN\TortoiseSVN\src\Resources\TortoiseProcENG.rc	
Sending content	D:\Development\SVN\TortoiseSVN\src\TortoiseProc\resource.h	
Completed	At revision: 20640	

19 kBytes transferred in 0 minute(s) and 9 second(s)

Modified: 3

OK

Cancel

Distributed Version Control

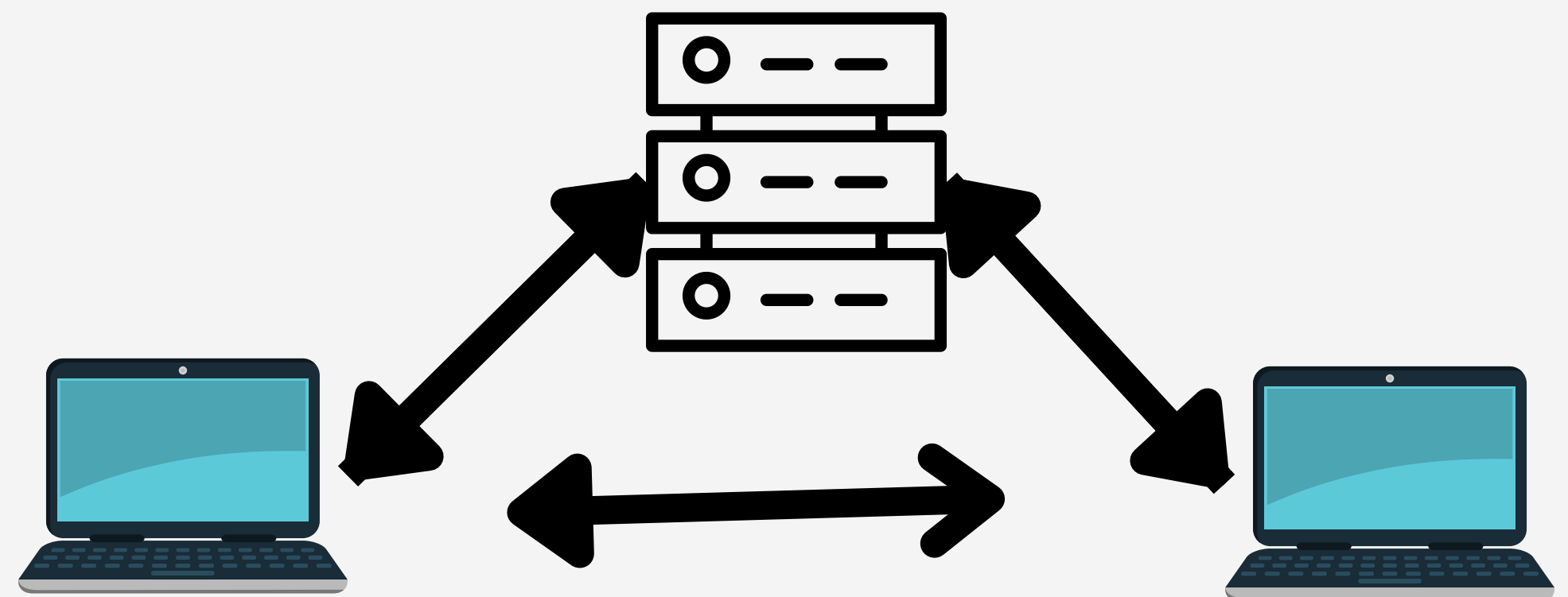
Equipes globalizadas

O método faz uma cópia local do repositório completo para o computador de cada membro da equipe.

O servidor não precisa armazenar um arquivo físico para cada branch — ele só precisa das diferenças entre cada commit.

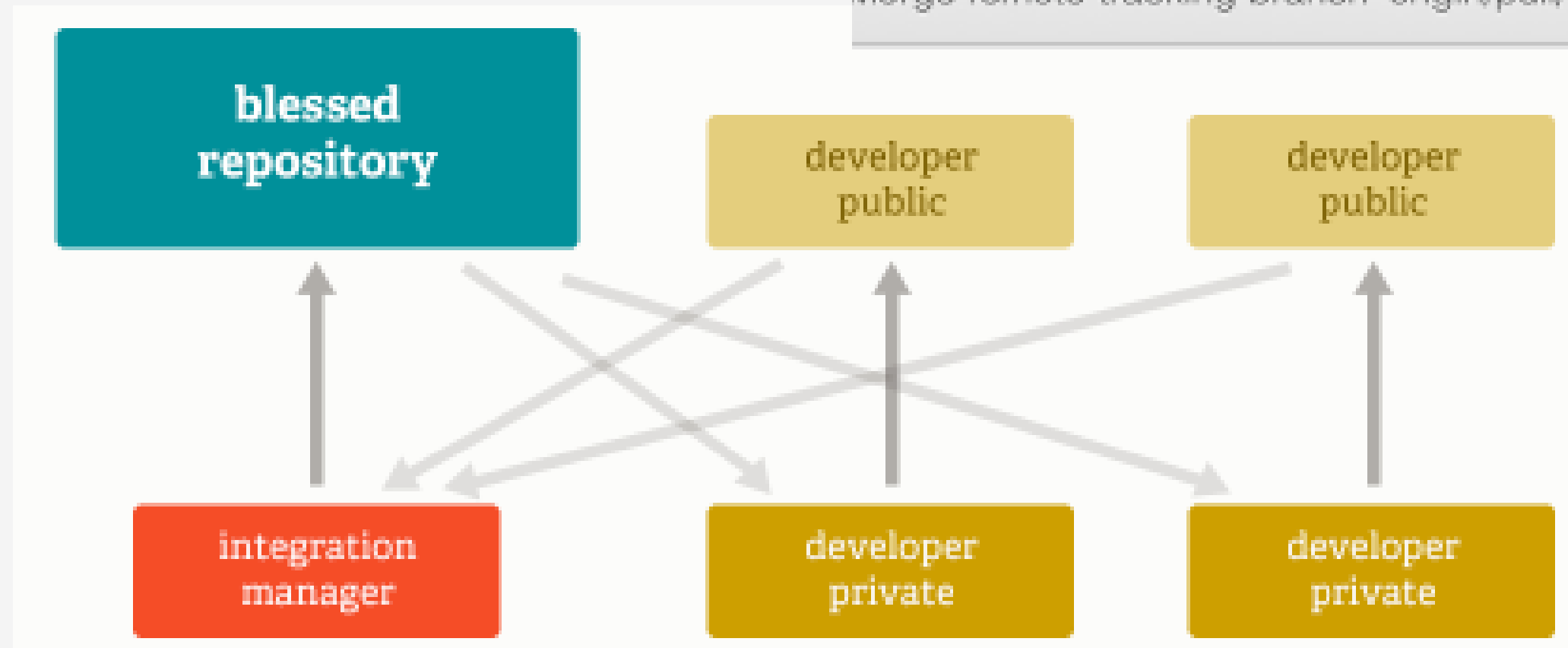
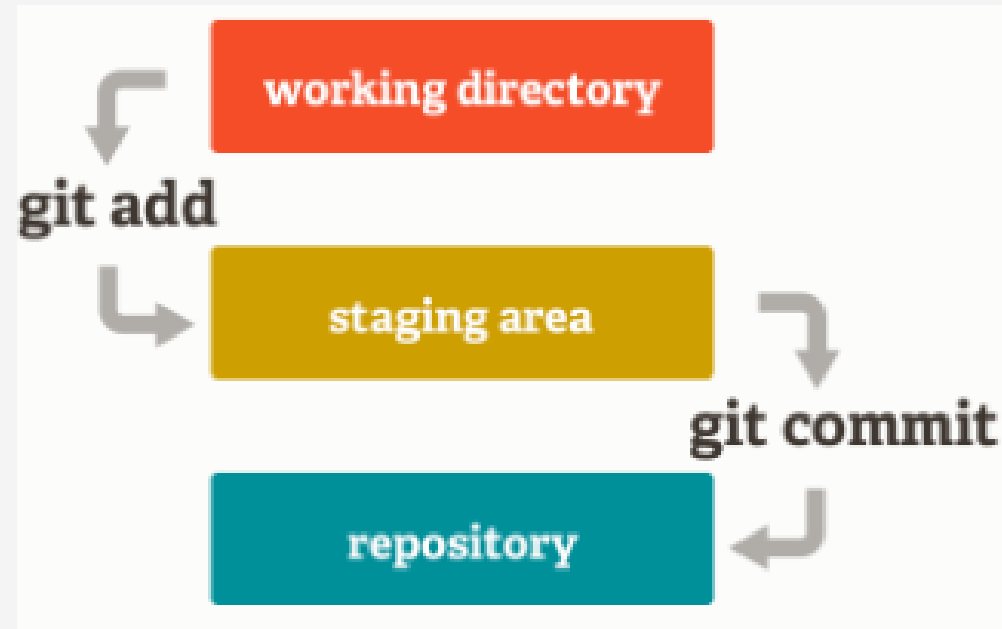
Vantagens e mais vantagens

Quando um membro da equipe clona um repositório, ele basicamente cria um backup externo, facilmente podendo reverter o código base. Permite desenvolver offline, fazendo o processo local, apenas necessita conexão para empurrar o código para o servidor.



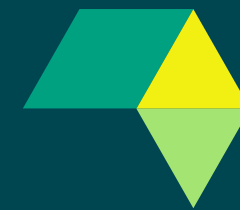
[Voltar ao slide de tópicos](#)

DVC - Git



12	
Scott Chacon Revert "refactor helpers to use content_tag/image_tag/link_to"	Jun 12 2575264
Scott Chacon Merge pull request #148 from linquize/git-clean	Jun 12 7f760bb
Scott Chacon Merge remote-tracking branch 'origin/pull/111/head'	Jun 12 a4fad60
Scott Chacon Merge remote-tracking branch 'origin/pull/109/head'	Jun 12 64b93fa

Até logo



Jonathan da Cruz



[Voltar ao slide de tópicos](#)