

GIT

Angela Mazzonetto
Márcio Nicolau

Universidade de Passo Fundo
Programa de Pós-Graduação em Computação Aplicada

25 de Agosto de 2016

Tópicos

- ▶ First steps;
- ▶ Git Essencial;
- ▶ Branching;
- ▶ Tools for GIT.

First steps

► O que é GIT?

Git considera que os dados são como um conjunto de snapshots (captura de algo em um determinado instante, como em uma foto) de um mini-sistema de arquivos.

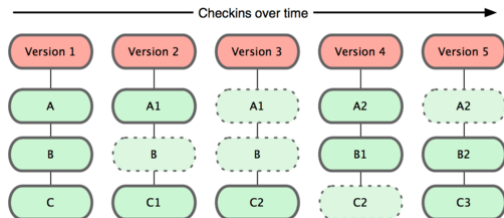


Figura 1. Git armazena dados como snapshots do projeto ao longo do tempo.

First steps

- ▶ Todo o histórico do projeto permanece no disco local;
- ▶ o Git pode buscar o arquivo de um mês atrás e calcular as diferenças localmente;
- ▶ há poucas coisas que não possam ser feitas caso esteja offline ou sem acesso a uma VPN;
- ▶ tudo no Git tem seu checksum (valor para verificação de integridade);
- ▶ o mecanismo que o Git usa para fazer o checksum é chamado de hash SHA-1:

24b9da6552252987aa493b52f8696cd6d3b00373

First steps

- ▶ Git faz com que seus arquivos sempre estejam em um dos três estados fundamentais: consolidado (committed), modificado (modified) e preparado (staged).
- ▶ Dados são ditos consolidados quando estão seguramente armazenados em sua base de dados local.
- ▶ Modificado trata de um arquivo que sofreu mudanças mas que ainda não foi consolidado na base de dados.
- ▶ Um arquivo é tido como preparado quando você marca um arquivo modificado em sua versão corrente para que ele faça parte do snapshot do próximo commit (consolidação).

First steps

- ▶ Isso nos traz para as três seções principais de um projeto do Git: o diretório do Git (git directory, repository), o diretório de trabalho (working directory), e a área de preparação (staging area).

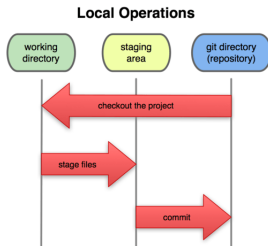
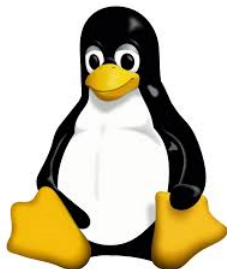


Figura 2. Diretório de trabalho, área de preparação, e o diretório do Git.

- ▶ O workflow básico do Git pode ser descrito assim:
 - Você modifica arquivos no seu diretório de trabalho.
 - Você seleciona os arquivos, adicionando snapshots deles para sua área de preparação.
 - Você faz um commit, que leva os arquivos como eles estão na sua área de preparação e os armazena permanentemente no seu diretório Git.

First steps

- Instalando e configurando....



- ▶ Inicializando um Repositório em um Diretório Existente
- ▶ Clonando um Repositório Existente
- ▶ Verificando o Status de Seus Arquivos
- ▶ Monitorando Novos Arquivos
- ▶ Seleccionando Arquivos Modificados
- ▶ Ignorando Arquivos
- ▶ Visualizando Suas Mudanças Seleccionadas e Não Seleccionadas
- ▶ Fazendo Commit de Suas Mudanças
- ▶ Pulando a Área de Seleção
- ▶ Removendo Arquivos
- ▶ Movendo Arquivos

- ▶ Visualizando o Histórico de Commits
- ▶ Limitando a Saída de Log
- ▶ Usando Interface Gráfica para Visualizar o Histórico
- ▶ Modificando Seu Último Commit
- ▶ Tirando um arquivo da área de seleção
- ▶ Desfazendo um Arquivo Modificado
- ▶ Trabalhando com Remotos
- ▶ Exibindo Seus Remotos
- ▶ Adicionando Repositórios Remotos
- ▶ Fazendo o Fetch e Pull de Seus Remotos
- ▶ Pushing Para Seus Remotos

- ▶ Inspeccionando um Remoto
- ▶ Removendo e Renomeando Remotos
- ▶ Tagging
- ▶ Listando Suas Tags
- ▶ Criando Tags
- ▶ Tags Anotadas
- ▶ Tags Assinadas
- ▶ Tags Leves
- ▶ Verificando Tags
- ▶ Taggeando Mais Tarde
- ▶ Compartilhando Tags
- ▶ Inspeccionando um Remoto

- ▶ Tagging
- ▶ Listando Suas Tags
- ▶ Criando Tags
- ▶ Tags Anotadas
- ▶ Tags Assinadas
- ▶ Tags Leves
- ▶ Verificando Tags
- ▶ Taggeando Mais Tarde
- ▶ Compartilhando Tags
- ▶ Inspeccionando um Remoto
- ▶ Removendo e Renomeando Remotos

- ▶ Tagging
- ▶ Listando Suas Tags
- ▶ Criando Tags
- ▶ Criando Tags
- ▶ Tags Anotadas
- ▶ Tags Assinadas
- ▶ Tags Leves
- ▶ Verificando Tags
- ▶ Taggeando Mais Tarde
- ▶ Compartilhando Tags

Ramificação (Branching) no Git

- ▶ Criar um branch significa dizer que você vai divergir da linha principal de desenvolvimento e continuar a trabalhar sem bagunçar essa linha principal.
- ▶ Básico de Branch e Merge
- ▶ Branch Básico
- ▶ Merge Básico

Git Distribuído - Fluxos de Trabalho Distribuídos

- ▶ Cada desenvolvedor pode contribuir com código para outros repositórios e ao mesmo tempo pode manter um repositório público em que outros podem basear seu trabalho e que eles podem contribuir.
- ▶ Exemplo Equipe.

Ferramentas GIT

- ▶ Git-flow
- ▶ Source Tree



The screenshot shows the Git website homepage. At the top left is the Git logo (a red diamond with a white 'G') followed by the text '--distributed-is-the-new-centralized'. To the right is a search bar with the placeholder text 'Search entire site...'. Below the header, there is a paragraph describing Git as a free and open source distributed version control system. To the right of this text is a diagram showing a branching model with stacks of boxes representing code and colored lines (red, blue, yellow) representing branches. Below the description, there is a 'Try Git' button. Further down, there are four circular icons with text: 'About' (gears), 'Documentation' (book), 'Downloads' (download arrow), and 'Community' (speech bubbles). At the bottom left, there is a small image of the 'Pro Git' book cover with text about its availability. At the bottom right, there is a monitor displaying the 'Latest source Release 2.9.3' and a 'Downloads for Windows' button. Below the monitor are links for 'Windows GUIs', 'Tarballs', 'Mac Build', and 'Source Code'.

git --distributed-is-the-new-centralized

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

 **Learn Git in your browser for free with Try Git.**

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

 **Pro Git** by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Latest source Release
2.9.3
Release Notes (2016-08-12)
[Downloads for Windows](#)

 [Windows GUIs](#)  [Tarballs](#)
 [Mac Build](#)  [Source Code](#)

Figura 3. Livro GIT <https://git-scm.com/>.