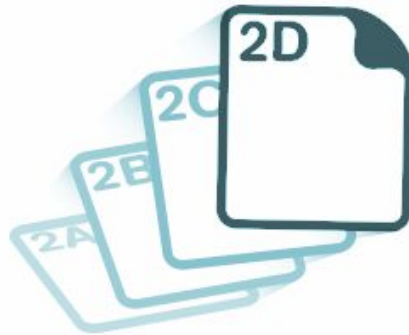


Sistemas de Controle de Versão Modernos



AGENDA

- ◎ Introdução
 - Para que serve controle de versão?
 - Como funciona?
 - Controle de Versão Centralizado x Distribuído
- ◎ Principais Sistemas de Controle de Versão Modernos
 - SVN
 - Git
 - Mercurial
 - Bazaar
- ◎ Comparação entre os Sistemas de Controle de Versão
- ◎ Serviços que fornecem Controle de Versão



Alguns problemas...

- © Sobrescrever o código de outra pessoa por acidente e acabar perdendo as alterações
- © Ter dificuldades em saber quais as alterações efetuadas em um programa, quando foram feitas e quem fez
- © Ter dificuldade em recuperar o código de uma versão anterior que está em produção
- © Ter problemas em manter variações do sistema ao mesmo tempo



Controle de Versão

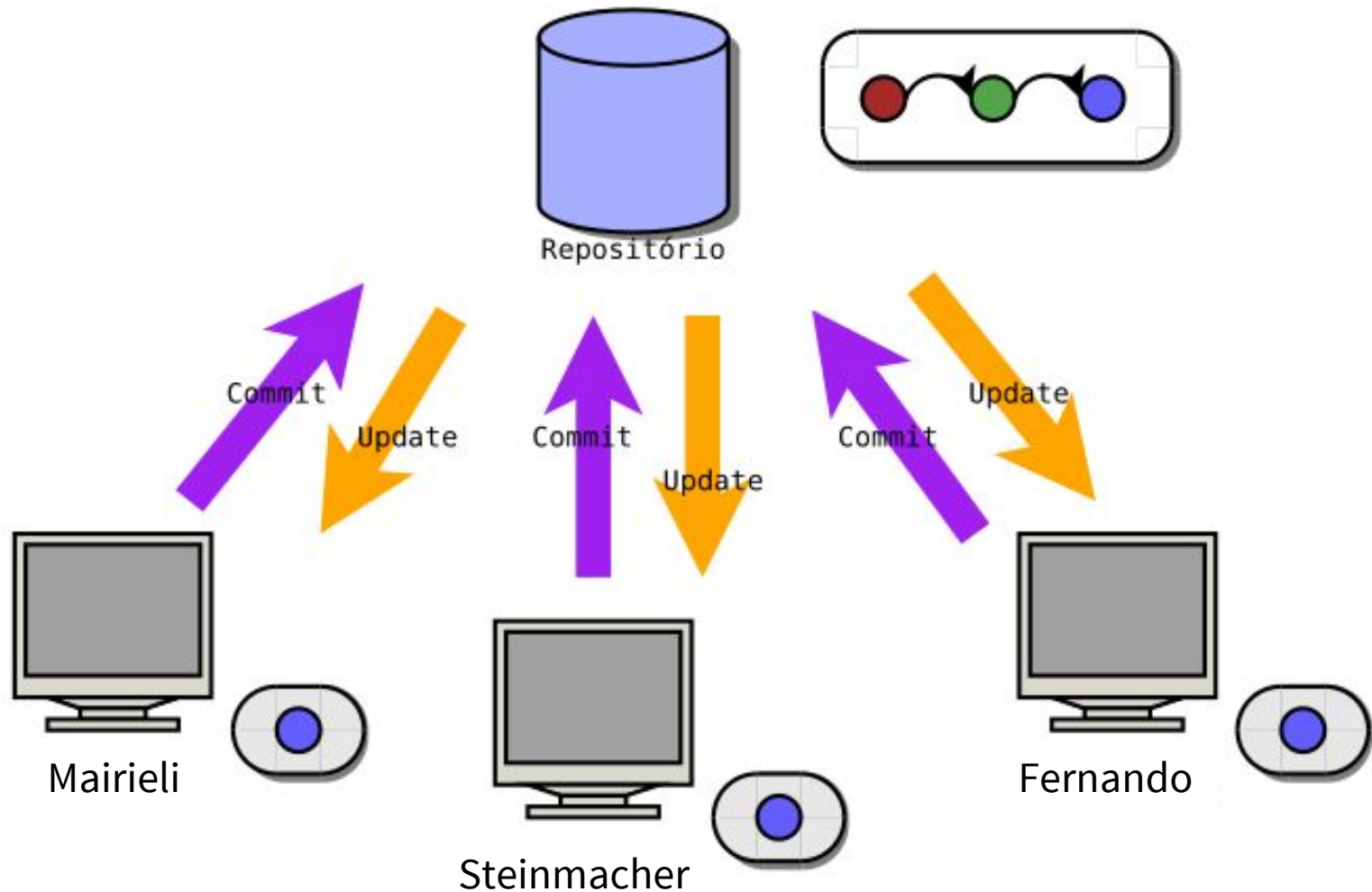
- © RCS foi uma ferramenta de controle de versão muito utilizada, no entanto ela trabalhava apenas com arquivos individuais e não com projetos
- © Controle de Versão é a forma inteligente de manter uma copia segura dos arquivos de um projeto

Para que serve o controle de versão?

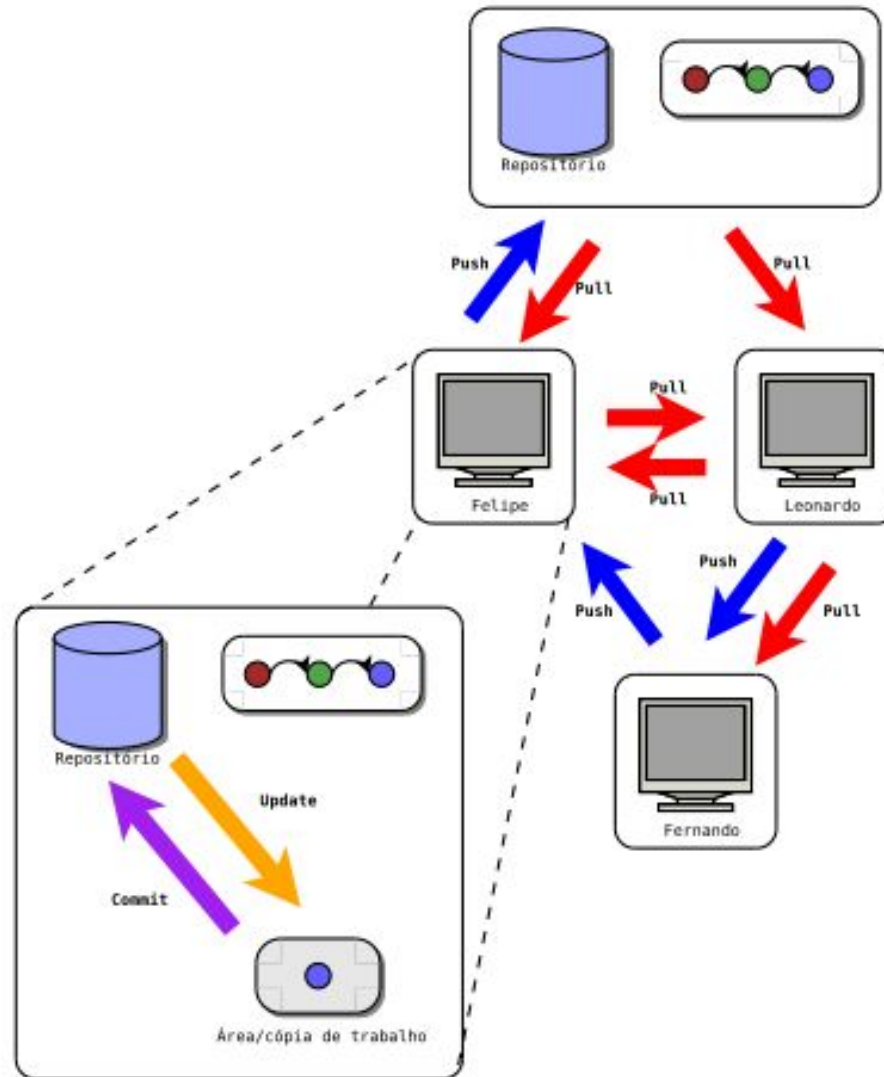
Apoia o desenvolvimento de diversas maneiras:

- ◎ **Histórico** - registra toda a evolução do projeto
- ◎ **Colaboração** - permite o trabalho em paralelo por parte dos desenvolvedores
- ◎ **Variações no projeto** - diferentes linhas de evolução do projeto

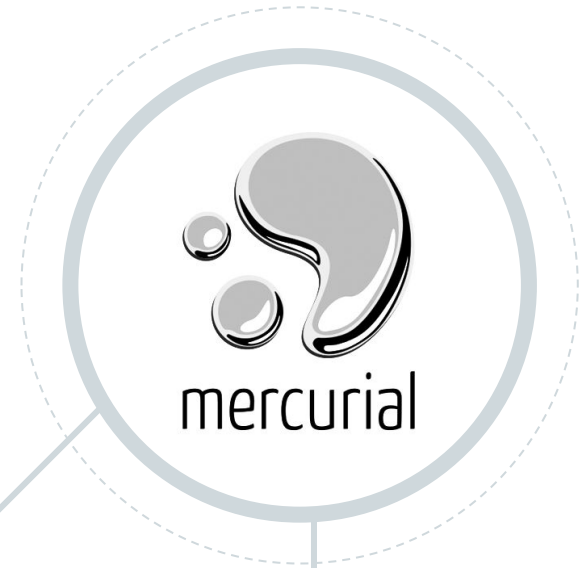
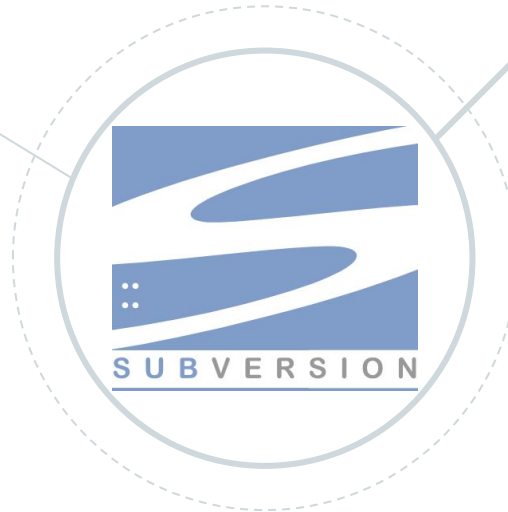
Controle de Versão Centralizado



Controle de Versão Distribuído



Principais Sistemas de Controle de Versão



Subversion



- © SVN Lançado em fevereiro de 2004
- © Substituto do CVS
- © Centralizado
 - Todo histórico de alterações são armazenadas no servidor tornando necessário estar conectado para disponibilizar suas alterações para o Sistema de Controle de Versão.
- © **Open Source**

Git

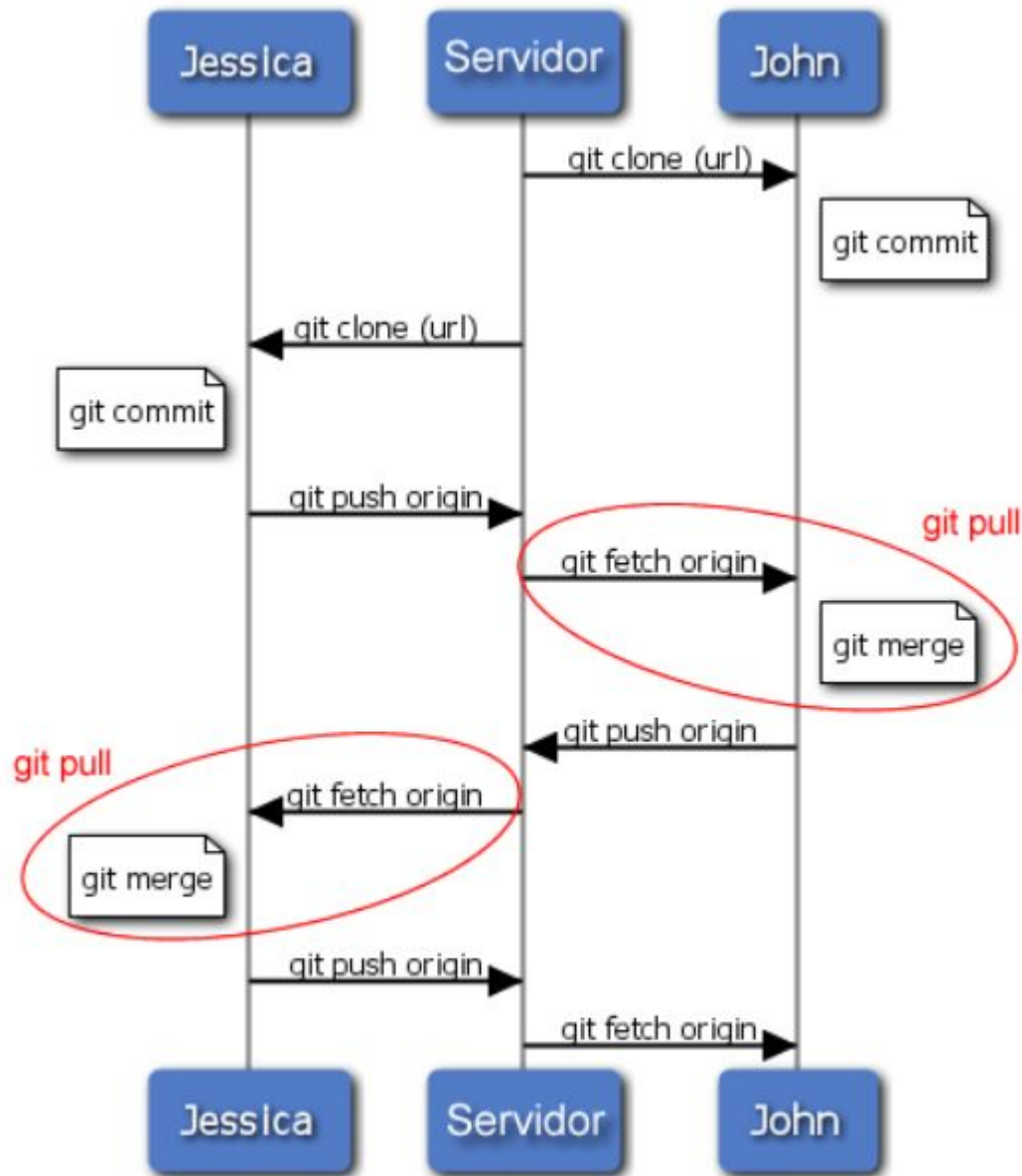


- © Inicialmente projetado e desenvolvido por Linus Torvalds e o time de desenvolvimento do Kernel do Linux
- © Design simples
- © Totalmente distribuído
- © Suporte robusto a desenvolvimento não linear (milhares de branches paralelos)
- © Ênfase em velocidade
- © **Open Source**

Git



- © Snapshots e não diferenças!
 - Diferente do SVN, o Git não guarda só as diferenças do seu arquivo mas sim uma imagem completa dele
- © Integridade de Arquivos
 - Antes de efetivar suas alterações é feita uma validação com checksum
- © Todo o trabalho é local, os commits são sempre feitos no seu repositório local e depois, através do método push envia todos os seus snapshots para um servidor que se responsabiliza de organiza-los.



Mercurial



- © Larry MacVoy torna BitKeeper proprietária e paga
- © Então Matt Mackall lança o Mercurial em Abril de 2005
- © Desenvolvido em Python
- © Distribuido
- © **Open Source**



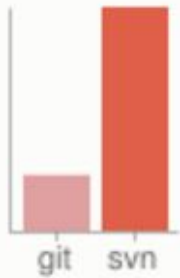
Bazaar

- © Lançado em Março de 2005
- © Mantido pela Canonical
- © Desenvolvido em Python
- © Distribuído
- © Em 2014
 - Emacs migra de Bazaar para Git
 - Bugzilla troca Bazaar pelo Git, declarando:
“Bazaar is almost dead. There are maybe 2-3 commits to trunk every month. The time to fix bugs in Bazaar also seems to be quite long, generally.”

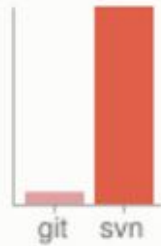


Comparação entre os Sistemas de Controle de Versão

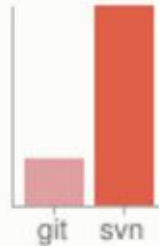
Commit A



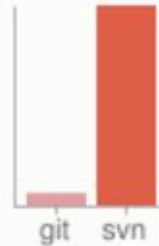
Commit B



Diff Curr



Diff Rec



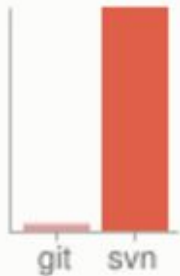
Diff Tags



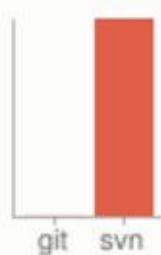
Clone



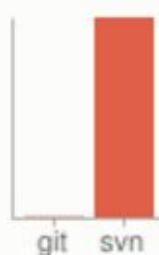
Log (50)



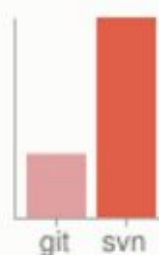
Log (All)



Log (File)



Update



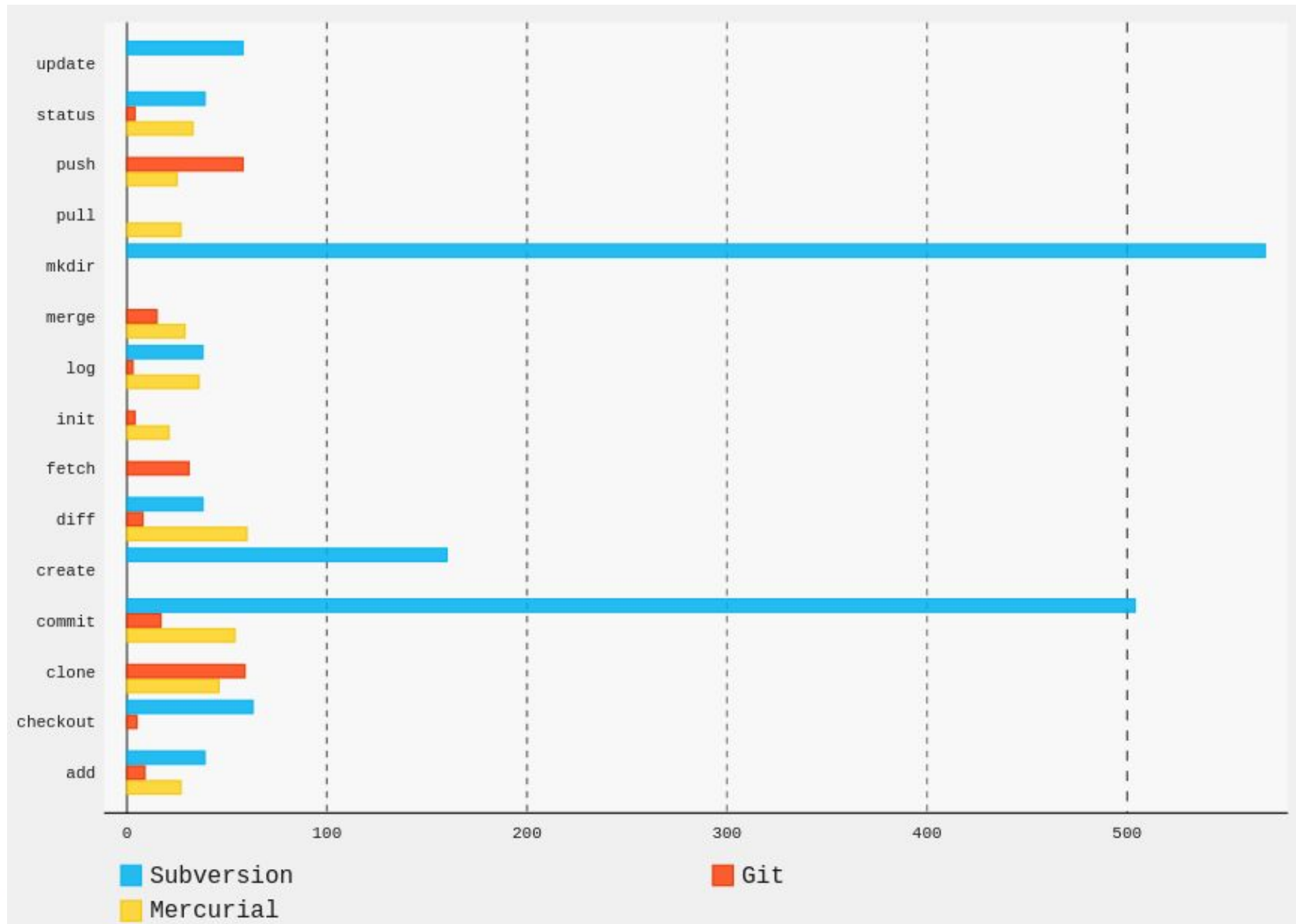
Blame



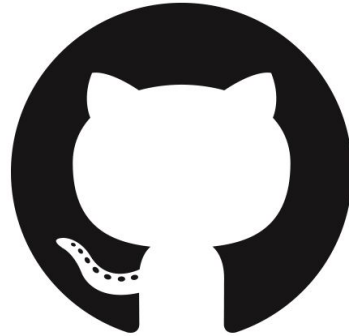
Size



Comparação entre Git, Subversion e Mercurial



Serviços que fornecem Controle de Versão



Sistemas de Controle de Versão Modernos

