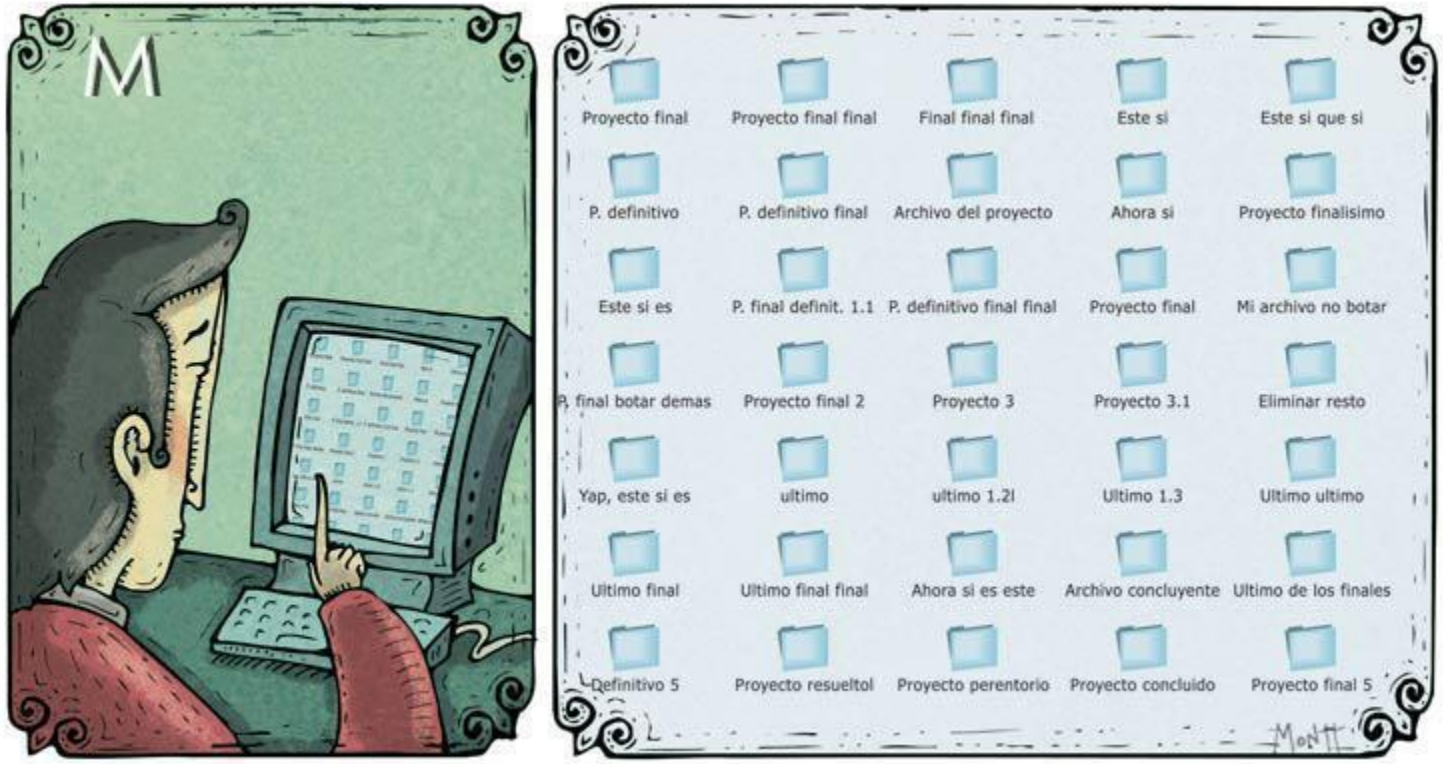
Control de versiones

Gestión de cambios sobre los elementos de algún producto o una configuración del mismo.

— Wikipedia

Control de versiones manual



Qué es un SCV?

Software para gestionar el historial de versiones de un proyecto.

Ventajas de un SCV

- Copias de seguridad
- Deshacer cambios
- Historial de cambios
- Diferentes versiones del proyecto

Cuando usar un SCV

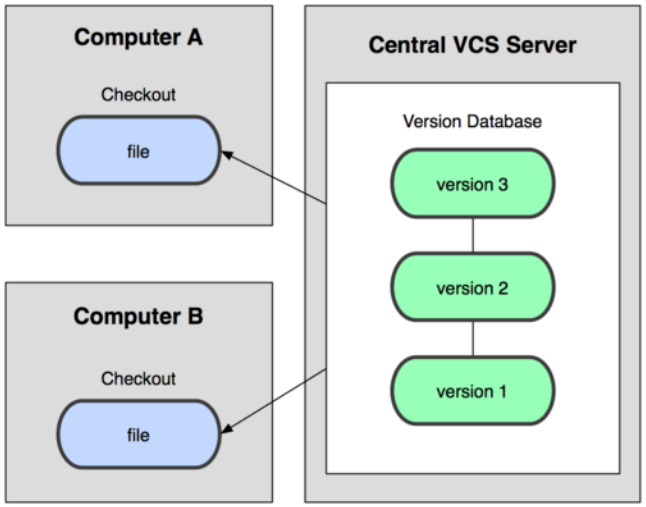
- Trabajos,
- Tesis,
- Documentación,
- Traducciones,
- Software
- ...
- Trabajando en equipo
- Trabajando solo

Algunos SCV

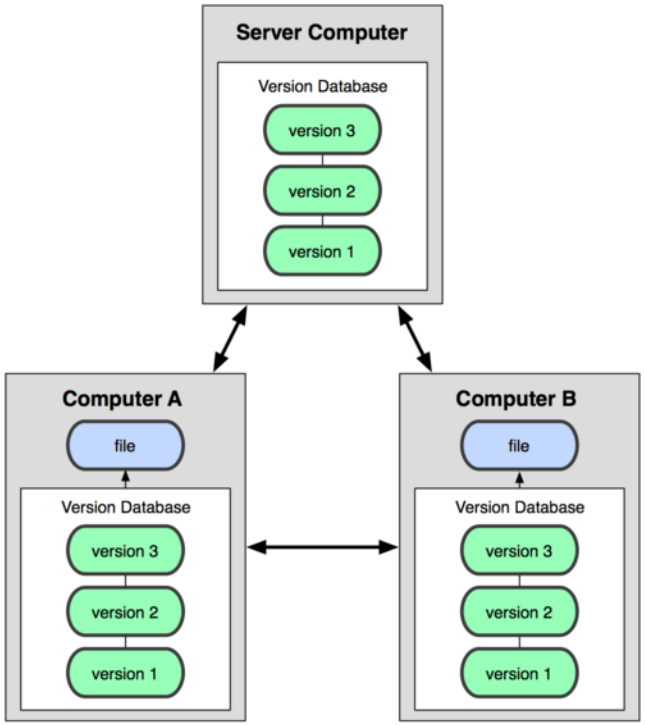
Caracterización

- Formas de colaborar
  - Exclusiva
  - Colaborativa
- Flujos de trabajo
  - Centralizado
  - Gestor de Integraciones
  - Dictador y Tenientes
- Arquitecturas de almacenamiento
  - Centralizados
  - Distribuidos

Centralizados



Distribuidos



Terminología

Estructurales

- Repository, Working copy, Working version, Workspace

Conceptuales

- Trunk, Branch, Tag
- Tip, Head
- Revision, Change, Changelist
- Stage

Acciones

- Commit, Checkout, Export, Import
- Conflict, Resolve, Merge, Rebase
- Cherry Pick, Revert, Stash, Patch
- Fork, Clone, Push, Pull

GIT

Sistema de control de versiones distribuido, diseñado y desarrollado por Linus Torvalds para el kernel Linux.

"I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'git'."

— Linus Torvalds

Ventajas

- Distribuido
- Rápido
- Eficiente (branches)
- Seguro (reflog)
- Flexible
- Local
- Pequeño
- Limpio (solo .git)
- GitHub

Desventajas

- Curva de aprendizaje
- Número comandos
- Significado comandos (usuarios subversion)

Comandos

Local Commands

- git config
- git init
- **git add**
- **git commit**
- **git status**
- git tag
- git log

Branchy Commands

- **git checkout**
- **git branch**
- **git merge**
- git rebase

Remotey Commands

- git remote
- **git fetch**
- **git pull**
- **git clone**
- **git push**

Patchy Commands

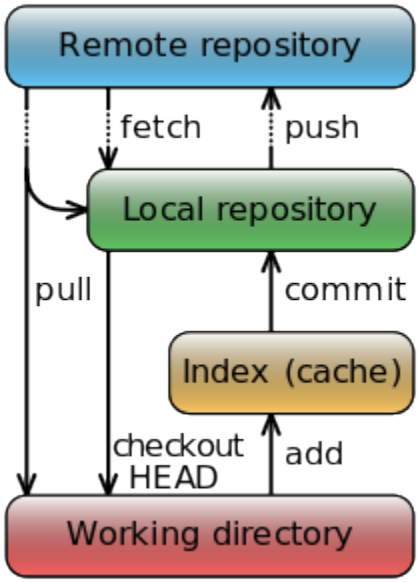
- git diff
- git apply
- git format-patch
- git am

Branching

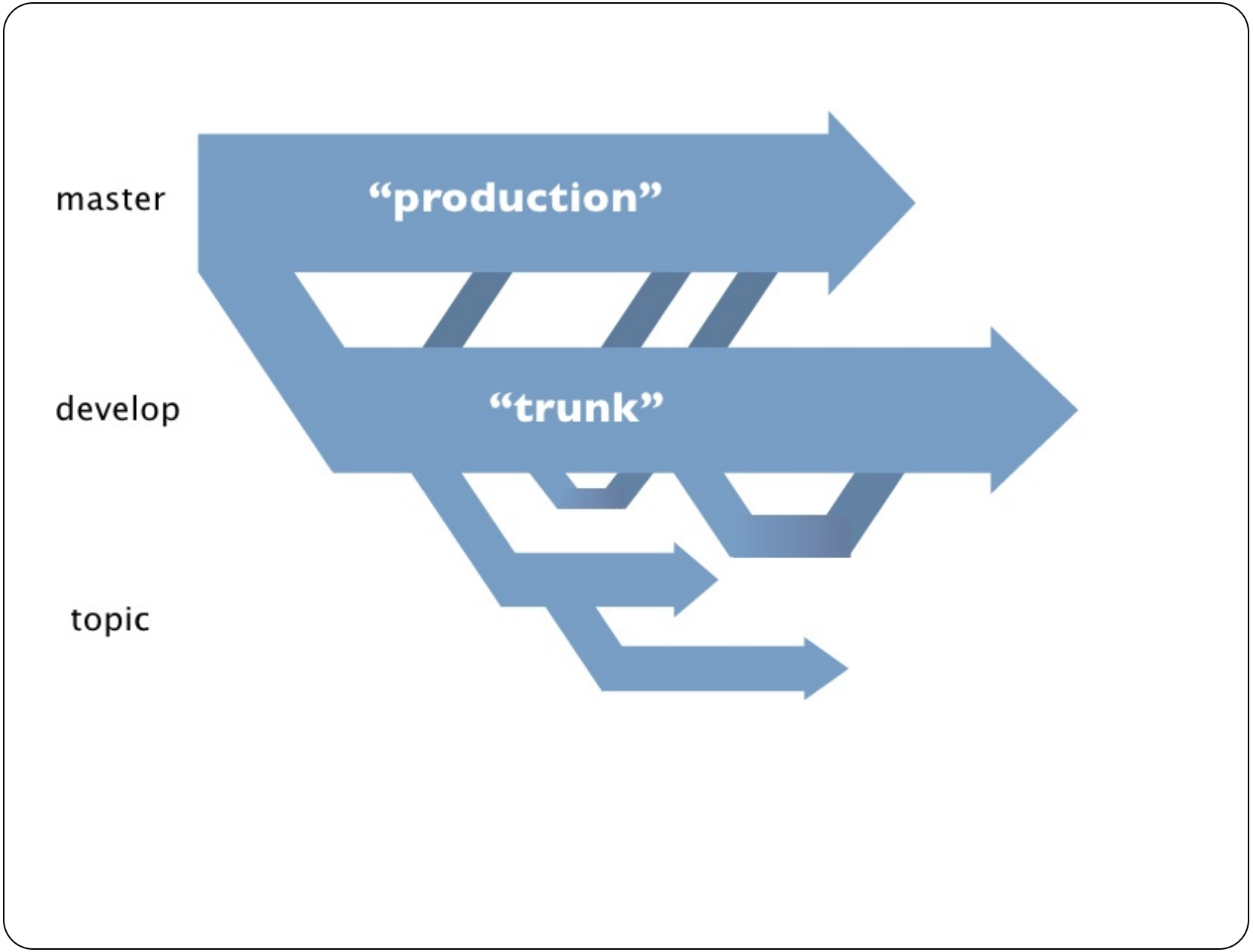
En Git es común trabajar con múltiples ramas.

La facilidad de git para el uso de ramas nos permite bifurcar una versión concreta del proyecto para empezar a trabajar en una nueva característica, o un bug que haya que solucionar.

Estructura y flujos



Convención


master
"production"
develop
"trunk"
topic

Doit

Me lo contaron y lo olvidé; lo vi y lo entendí; lo hice y lo aprendí.

— Confucio

Cheat Sheet

GitHub

Sistema web para alojar proyectos utilizando Git

Características

- Wiki
- Página web
- Gráficos del desarrollo y de sus bifurcaciones.

- Funcionalidades de Red social.

Sofware

Linux

- apt-get install git-core

Window

- http://code.google.com/p/msysgit
- Git Extensions (https://github.com/gitextensions/gitextensions)

Mac

- http://code.google.com/p/git-osx-installer

Fuentes

- http://es.wikipedia.org/wiki/Control_de_versiones
- http://git-scm.com/book
- http://www.blog.sergiorus.com/post/968247237/git-sistema-de-control-de-versiones-distribuido
- http://byte.kde.org/~zrusin/git/
- http://www.slideshare.net/chacon/getting-git

Video

- http://www.slideshare.net/chacon/getting-git

Filminas

QR Code