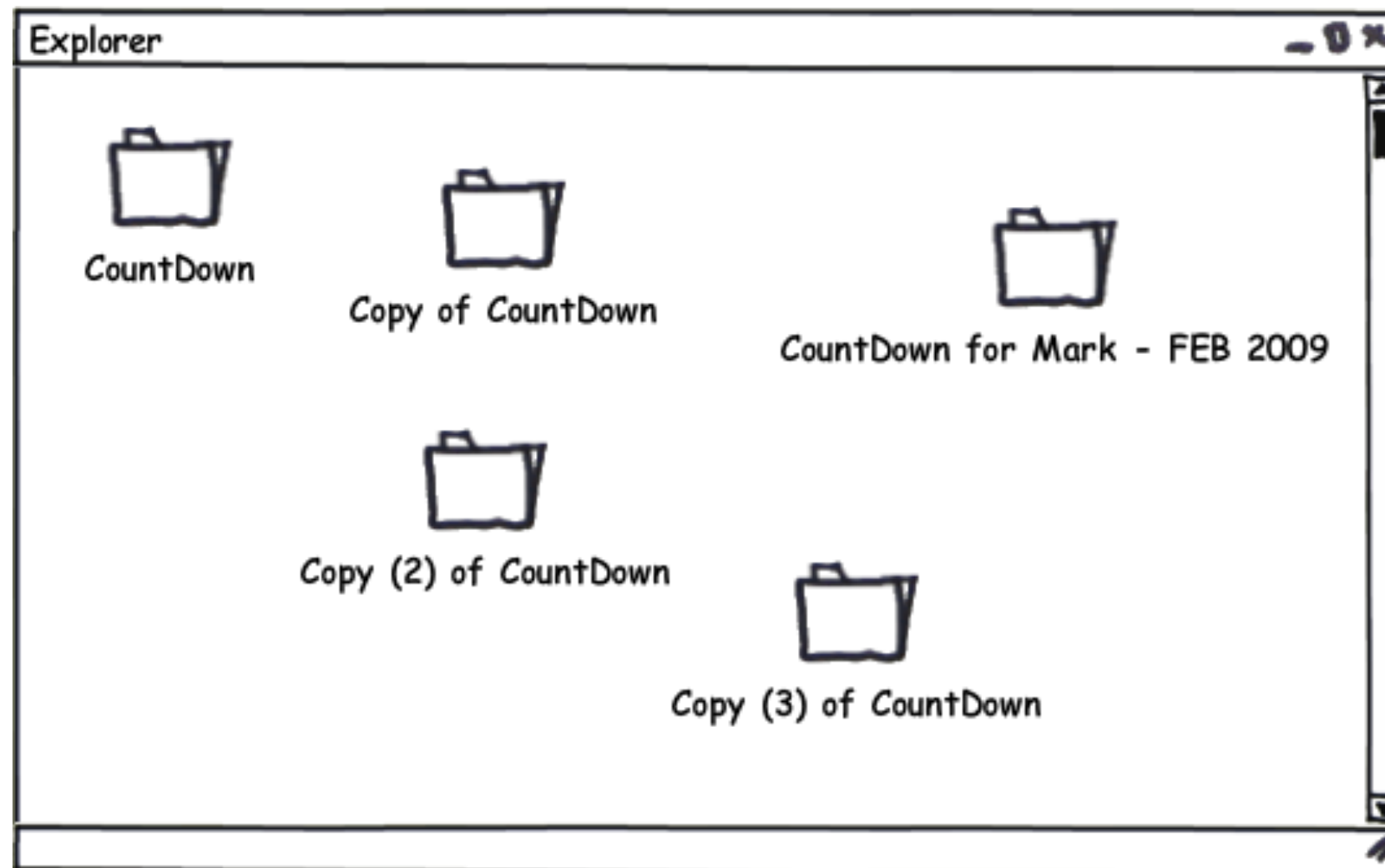


Git

Control de versiones



Antes del control de versiones.

Varias versiones del mismo archivo, usando un esquema numerado. No había ninguna forma de rastrear el historial de cambios, y depende de la integridad de un entorno local.

Sistemas de Control de Versiones VCS's

- » Sistema de control de versiones(VCS): Sistema que registra cambios en un archivo o grupo de archivos sobre una línea de tiempo. Cada cambio se identifica como una revisión asociada a un timestamp y el autor del cambio.
- » Sistemas de control de versiones centralizado(CVCSs): Este sistema usa un repositorio central que sirve como un único punto de entrada para todos los cambios del proyecto.
- » Sistemas de control de versiones distribuidos(DVCSs): Este sistema no depende de un servidor central para almacenar todas las versiones de un proyecto, sino de las copias clonadas por un colaborador. Al clonar una copia de un repositorio, se descarga el historial completo de revisiones del proyecto.

Diagramas SCV's Centralizado

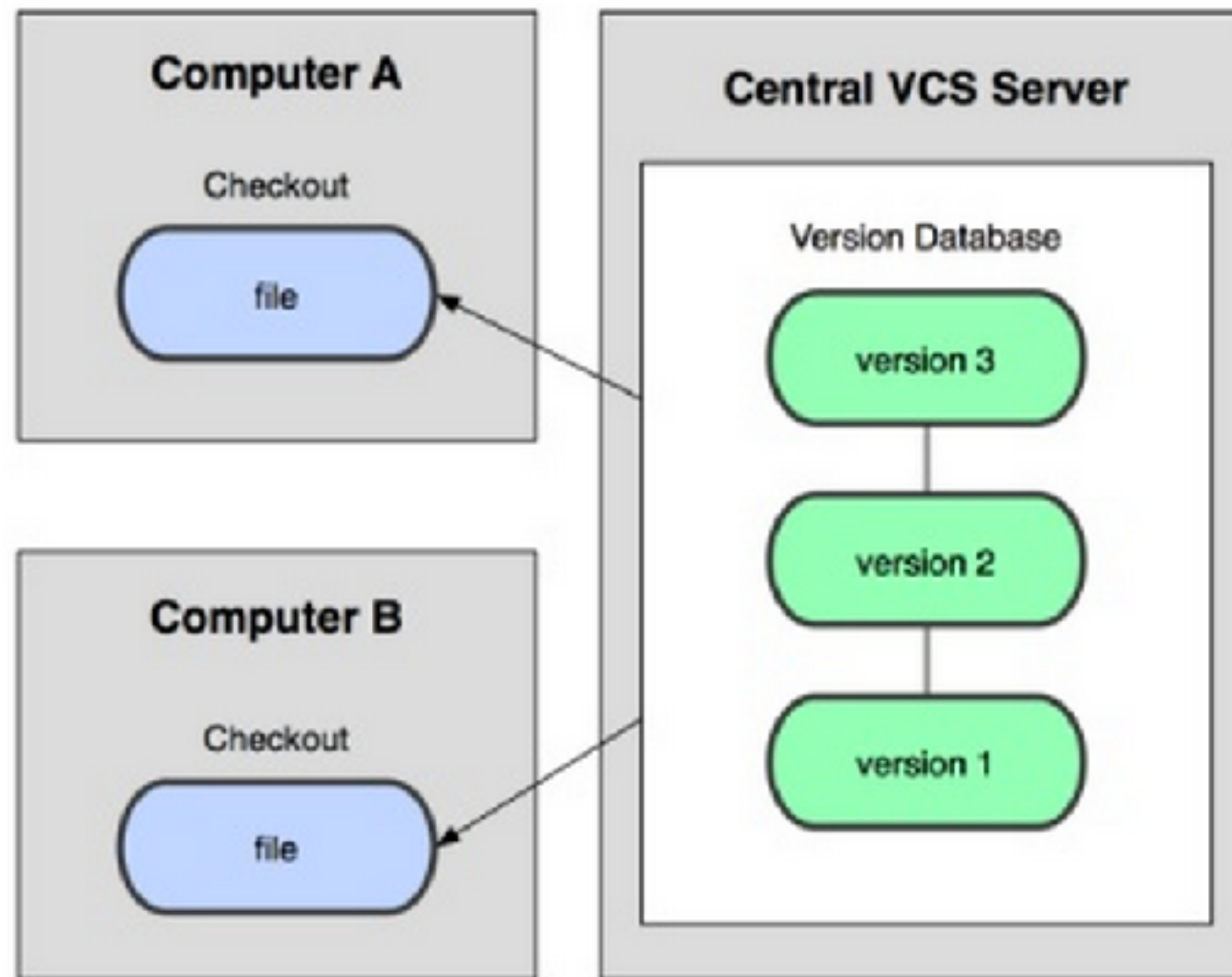
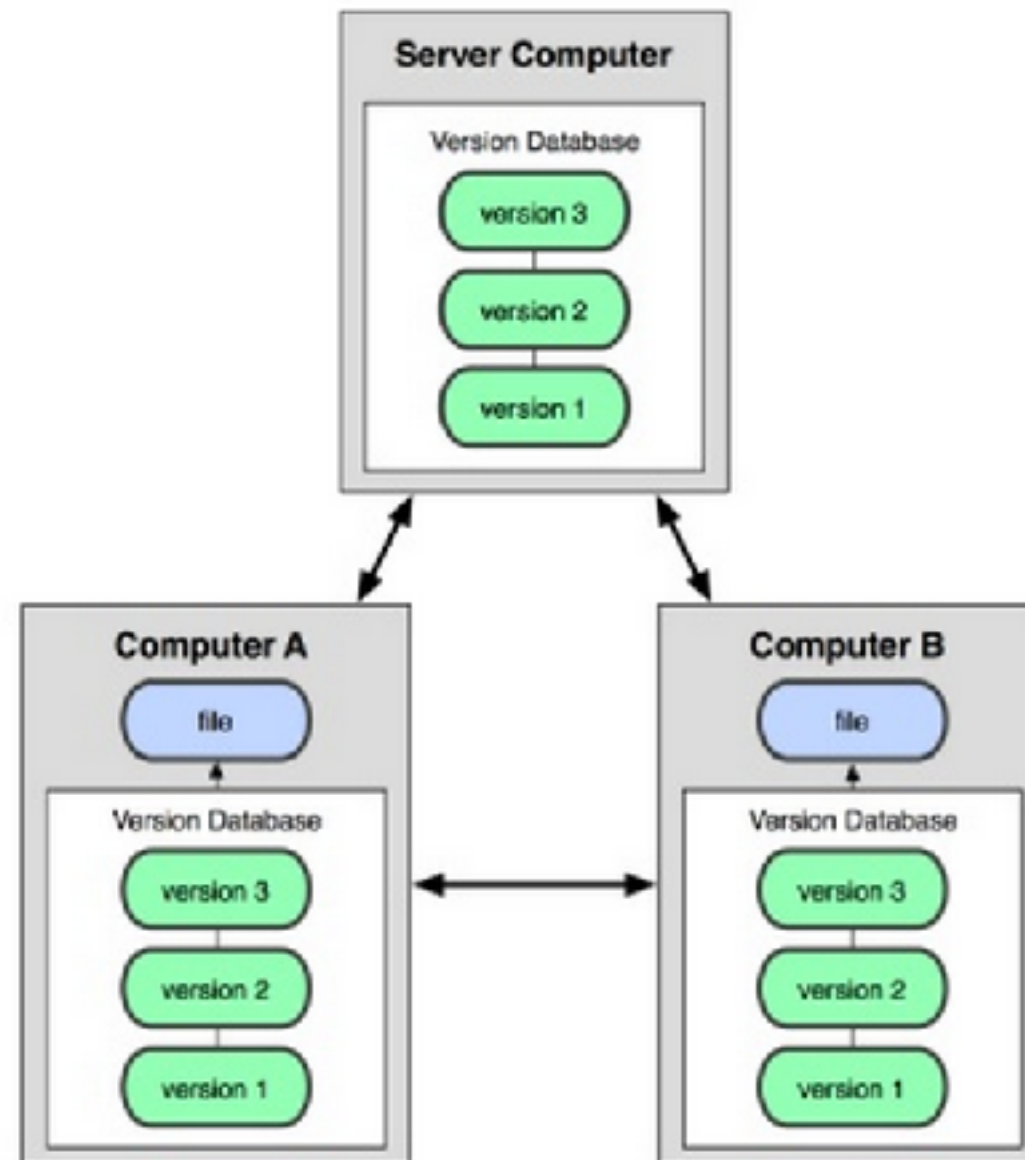


Diagrama SCV's Distribuido



VCS's populares

- » Git - <http://git-scm.com>
- » Subversion - <https://subversion.apache.org>
- » Mercurial - <http://mercurial.selenic.com>
- » Perforce (comercial) - <http://www.perforce.com/>
- » Otros: Bazaar, LibreSource, Monotone

¿Qué es Git?



- Es un sistema de control de versiones distribuido.
- Se desarrolló en el 2005 por Linus Torvalds, está escrito en C, shell, Tcl y perl.
- La última versión estable es la 2.3.7 liberada el 27 de Abril de 2015
- Se destaca por ser eficiente, rápido, seguro y flexible.
- <https://github.com/torvalds>

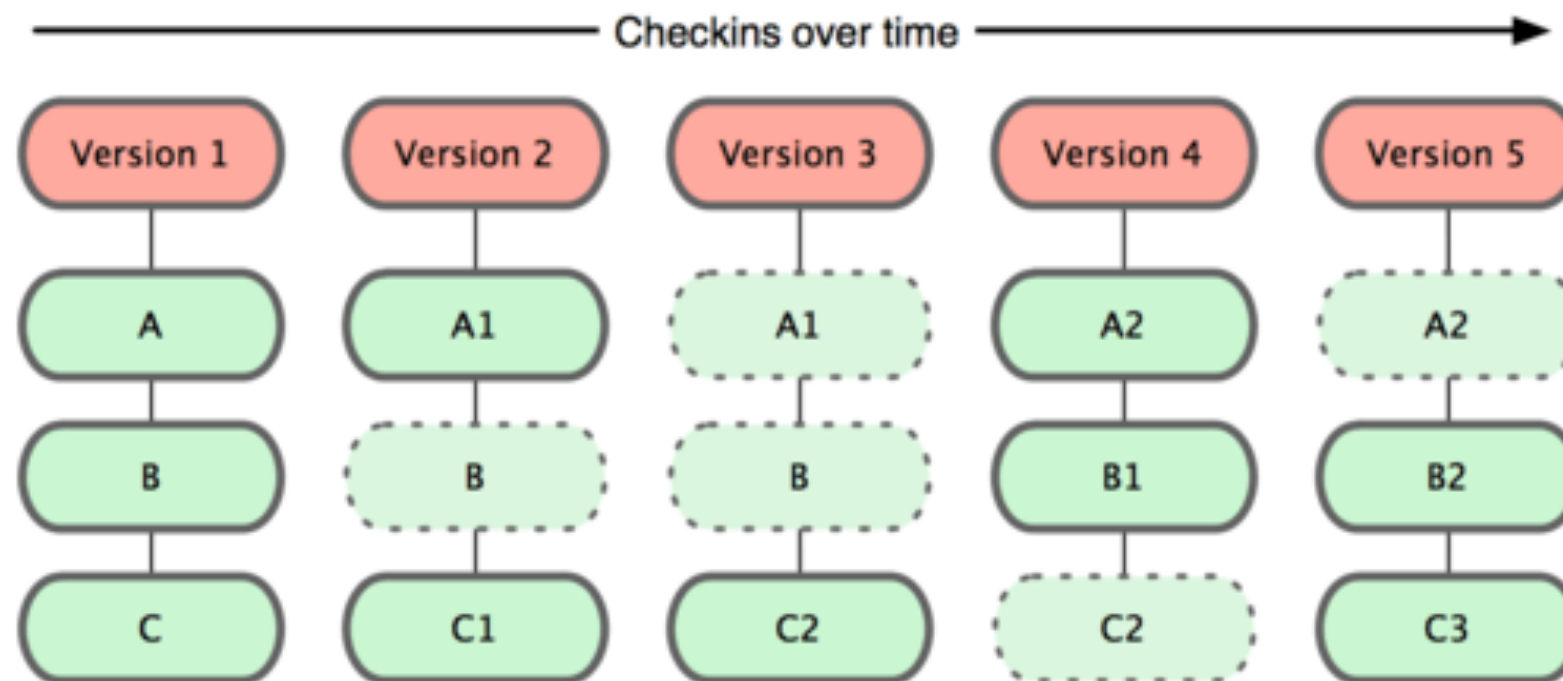


Ventajas de Git sobre SVN

- » Git es más rápido
- » Los repositorios de git son más pequeños que los de SVN
- » Git se diseñó para ser totalmente distribuido desde el comienzo, permitiendo a cada desarrollador tener completo control local.
- » Los branches de Git cargan su historial completo, además son más simplificados que los de SVN.
- » Mejor auditoría de código.
- » Facilidad para resolver conflictos.

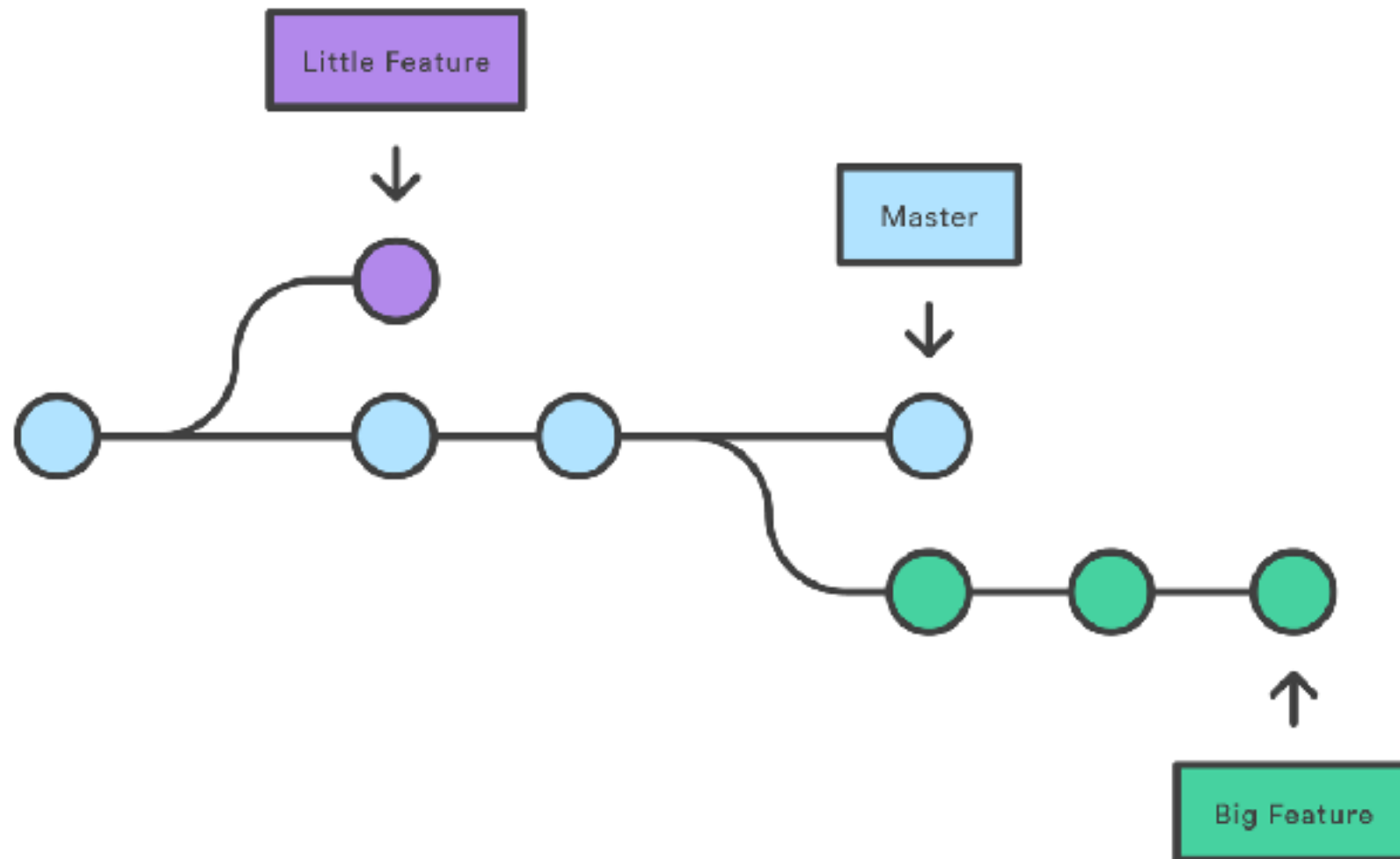
Snapshots

(No diferencias)



- SVN: Lista de archivos con cambios
- GIT: Snapshots de todo el FS. Si no hay cambios, no se guarda el archivo de nuevo.

Git Branching



Instalar GIT

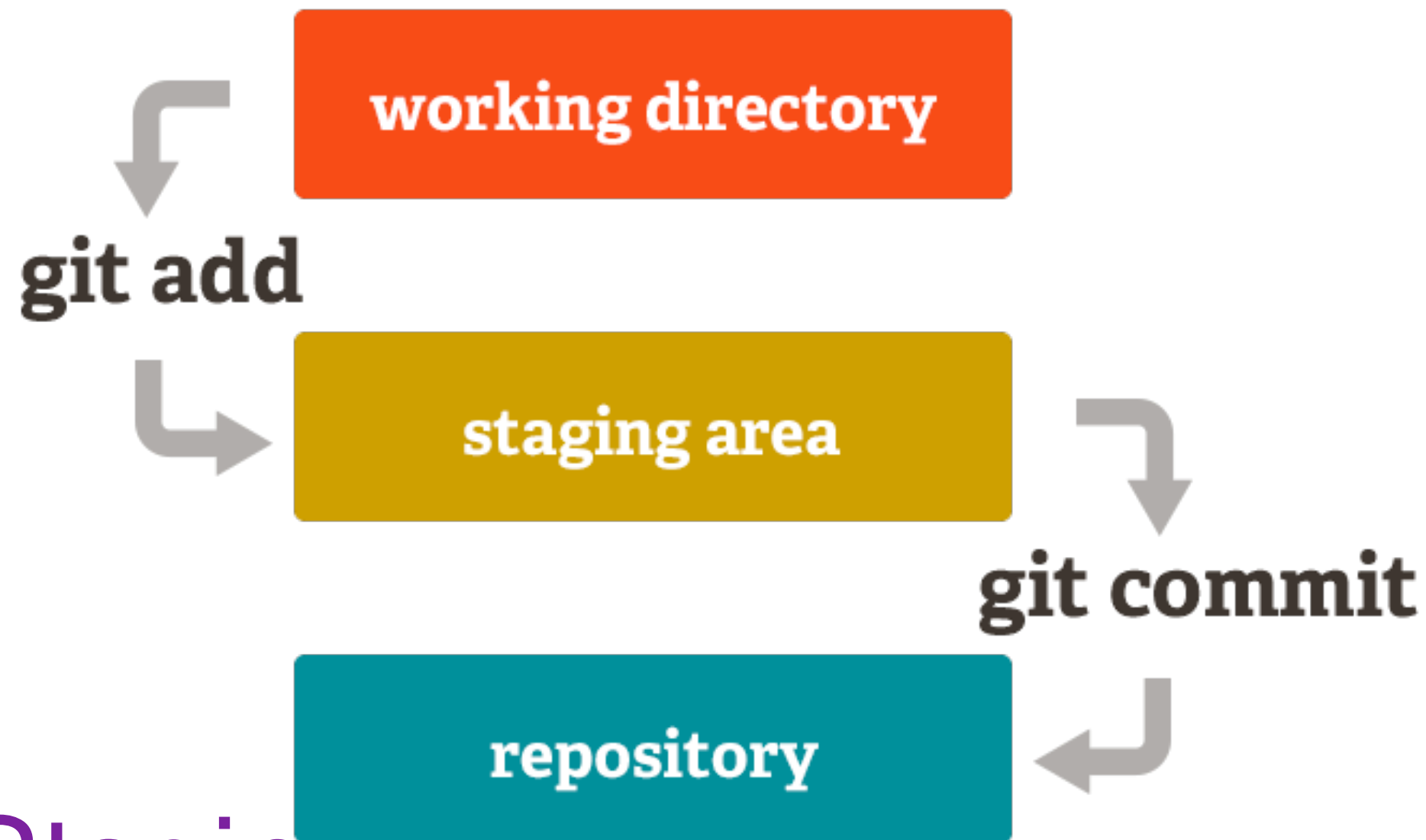
- » Descargar de git-scm
 - » <http://git-scm.com/download>
- » Instalar en Linux
 - » Yum install git
 - » apt-get install git
- » Instalar en Mac
 - » <http://sourceforge.net/projects/git-osx-installer/>
 - » brew install git
- » Instalar en Windows
 - » <http://msysgit.github.io>

Cientes de GUI de Git

- » <https://mac.github.com/>
- » <https://windows.github.com/>
- » <http://www.sourcetreeapp.com/>
- » <https://msysgit.github.io/>

Integración de GIT con Editores

- » NetBeans <https://netbeans.org/kb/docs/ide/git.html>
- » Eclipse <https://eclipse.org/egit/>
- » Sublime <https://sublimegit.net/>
- » Brackets <https://github.com/zaggino/brackets-git>
- » PhpStorm <https://www.jetbrains.com/phpstorm/help/using-git-integration.html>



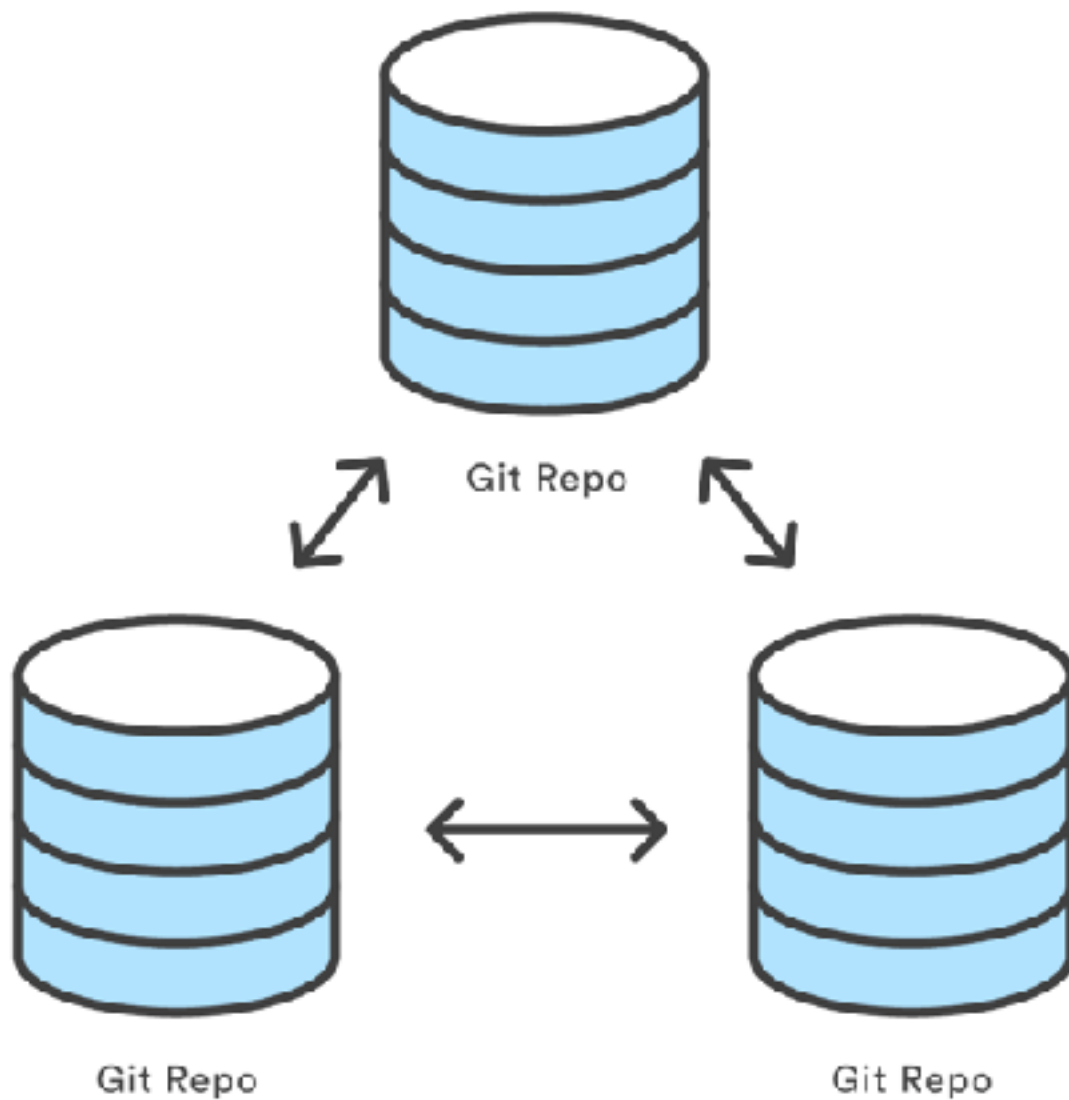
Staging Area

- El directorio .git:

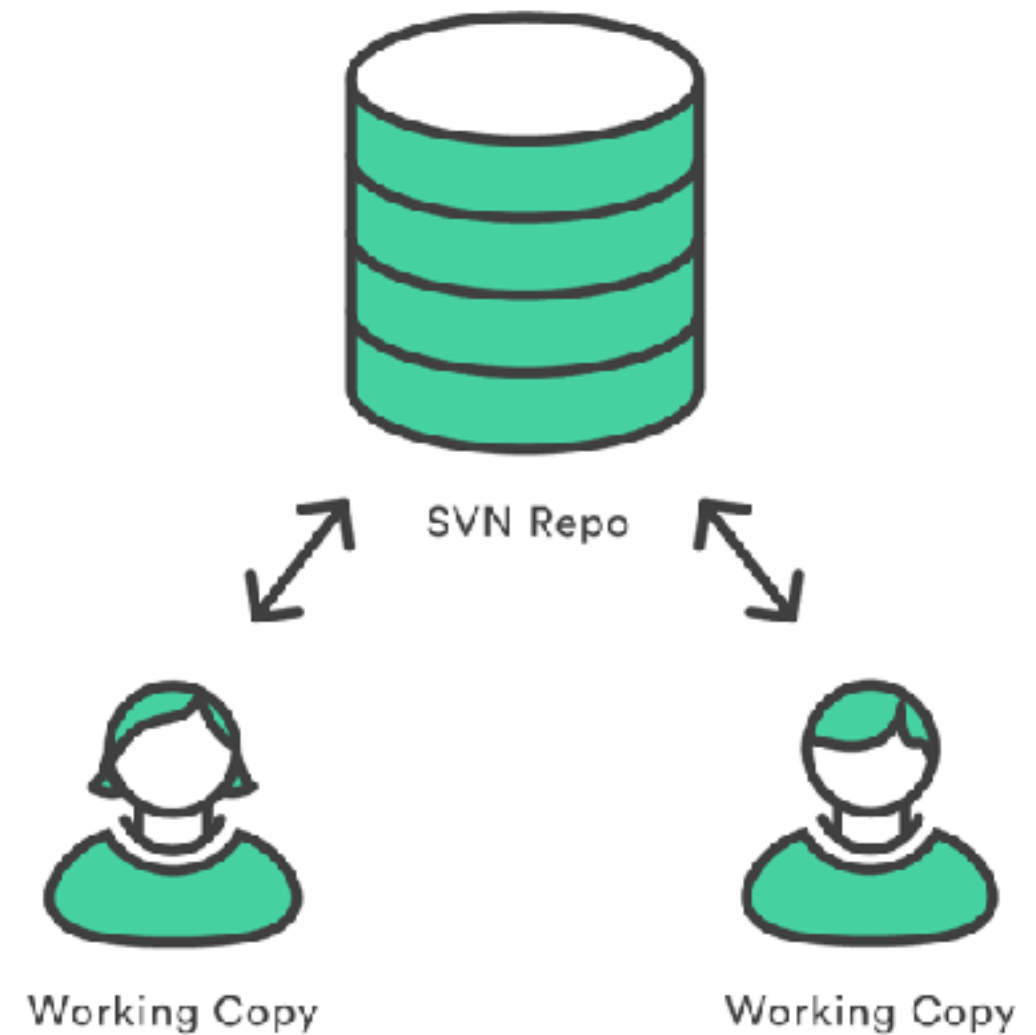
branches COMMIT_EDITMSG config description FETCH_HEAD HEAD hooks index info logs
objects ORIG_HEAD packed-refs refs

Esquema de colaboración

Repo-To-Repo Collaboration



Central-Repo-to-Working-Copy Collaboration



Operaciones Básicas

- » `git config --global user.name <name>`
- » `git config --global user.email <email>`
- » `git init`
- » `git clone <repo> <directory>`
- » `git pull`
- » `git add <file|directory>`
- » `git commit -m "<message>"`
- » `git push`
- » `git status`
- » `git log`
- » `git remote -v`

Más Operaciones Básicas

- » `git branch <newbranch>`
- » `git checkout <existingbranch>`
- » `git checkout -b <newbranch>`
- » `git reset --hard`
- » `git checkout <commit> <file>`
- » `git checkout -- <file>`
- » `git revert <commit>`
- » `git clean`
- » `git commit --amend`
- » `git merge`
- » `git remote add <name> <url>`
- » `git remote rename <old-name> <new-name>`
- » `git fetch <remote> <branch>`
- » `git tag -a <version> -m "<annotation>"`

Resolución de Conflictos



Conceptos



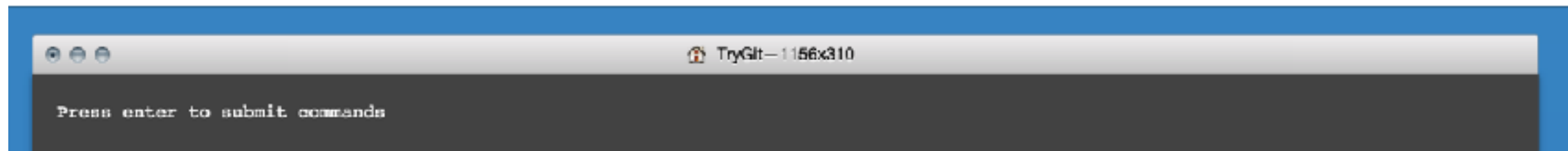
- » Repositorio: Almacenamiento principal de historial de cambios, se puede representar como una cadena de commits.
- » Tags: Las etiquetas apuntan a puntos específicos dentro del historial de revisiones, y se utilizan para marcar hitos en los cambios de código, donde se recomienda seguir la semántica de versionamiento dependiendo de la magnitud de los cambios: Major.Minor.Patch.
- » Branch: Este concepto difiere para los contextos de Git y Subversion, pero en general el “Branching” se refiere a la divergencia de la principal línea de desarrollo para que se pueda continuar con un desarrollo de código paralelo. En svn un branch es una copia de archivos que crea una nueva dirección del workflow de desarrollo. En Git, es simplemente un puntero que se puede mover entre diferentes commits dentro de la cronología de revisiones.
- » Commit: Snapshot del árbol de trabajo en un punto de tiempo
- » Working tree: directorio en el sistema de archivos que está asociado con un repositorio, contiene archivos y subdirectorios.

1.1 · Got 15 minutes and want to learn Git?

Git allows groups of people to work on the same documents (often code) at the same time, and without stepping on each other's toes. It's a distributed version control system.

Our terminal prompt below is currently in a directory we decided to name "octobox". To initialize a Git repository here, type the following command:

➔ `git init`



Práctica - Try Git

<https://try.github.io/levels/1/challenges/1>

Git en el servidor - Hosting de Git

- » Github - <https://github.com/>
- » Assembla - <https://www.assembla.com/>
- » Beanstalk
- » Bitbucket
- » Codebase
- » Cloudforge
- » Gitorious

Referencias Documentación

<http://git-scm.com/documentation>

<https://www.atlassian.com/git/tutorials/>

<http://gitref.org/>

<https://help.github.com/>

<https://git.kernel.org/cgit/>

¿Cómo migrarse de SVN a Git? - <https://www.atlassian.com/git/tutorials/migrating-overview>

GitHub

GitHub Bootcamp ×

1



Set up Git
A quick guide to help you get started with Git.

2



Create repositories
Repositories are where you'll work and collaborate on projects.

3



Fork repositories
Forking creates a new, unique project from an existing one.

4



Work together
Send pull requests, follow friends. Star and watch projects.

Top 10 Lenguajes en GitHub

	Rank			# New Repos Created		
Language	2014	2013	2012	2014	2013	2012
JavaScript	1	1	2	383185	320534	277875
Java	2	3	3	283354	185530	240992
Ruby	3	2	1	259268	228145	310281
C	4	7	4	178891	79223	203992
CSS	5	12	25	175573	18869	3791
PHP	6	4	6	175476	139591	157185
Python	7	5	5	151669	126027	165655
C++	8	6	7	78878	104499	88615
Objective-C	9	8	11	60579	40072	36539
C#	10	10	10	59472	34992	39486

Preguntas

Gracias