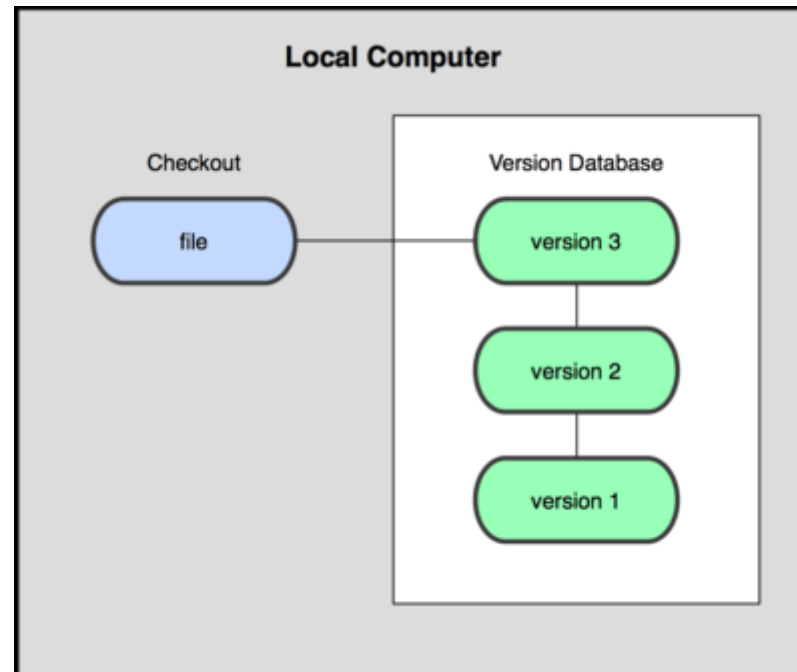

Fundamentos de Git

Obtener un repositorio Git

- Es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que se puedan recuperar versiones específicas.
- Otros nombres con los que se les conoce:
 - VCS del inglés Version Control System
 - SCM del inglés Source Code Management o sistema de control de código fuente.

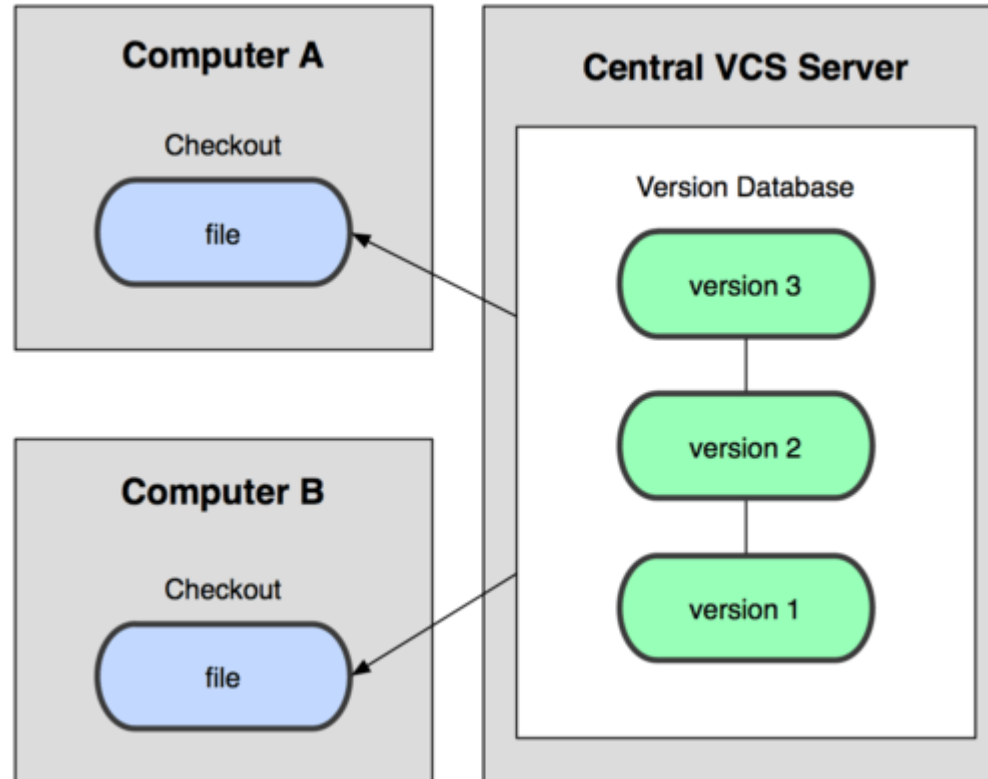
Tipos de sistemas de control de versiones

- **Sistemas de control de versiones locales**
 - Ventajas: Simpleza.
 - Desventajas: No permite la colaboración entre desarrolladores.
 - Revision Control System o RCS es el más popular
 - Parte del Proyecto GNU
 - Se sigue utilizando en:
 - Herramientas wiki como Twiki.
 - Sistemas operativos como Mac OS X.



Tipos de sistemas de control de versiones

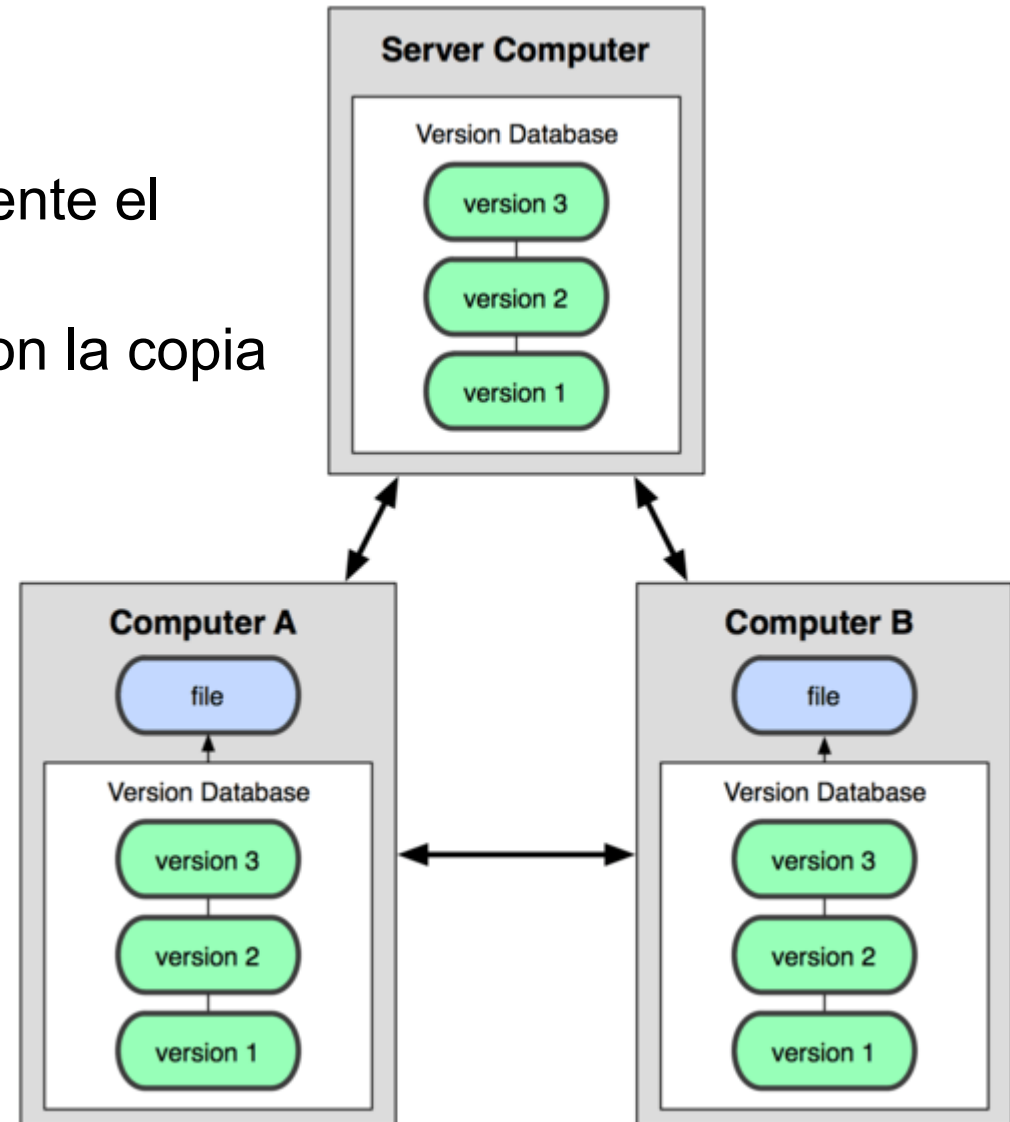
- **Sistemas de control de versiones centralizados o Centralized Version Control Systems o CVCSs**
 - **Ventajas:** Permite la colaboración entre desarrolladores.
 - **Inconvenientes:** Sensible a fallos en el servidor.
 - **CVS o Concurrent Versions System es el más popular.**
 - Es una aplicación GPL que implementa un sistema de control de versiones cliente-servidor.
 - **Otros:** Subversion y Perforce.



Tipos de sistemas de control de versiones

■ Sistemas de control de versiones distribuidos o Distributed Version Control Systems o DVCSs.

- Permite la colaboración entre desarrolladores.
- Los clientes replican completamente el repositorio.
- Se puede restaurar el servidor con la copia del cliente.
- Ejemplos:
 - Git,
 - Bitkeeper,
 - Mercurial
 - Bazaar
 - Darcs

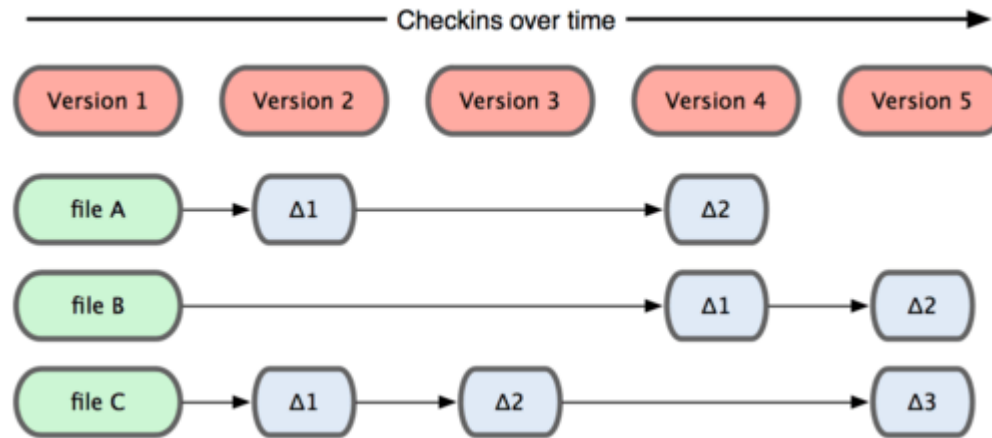


Breve historia de Git

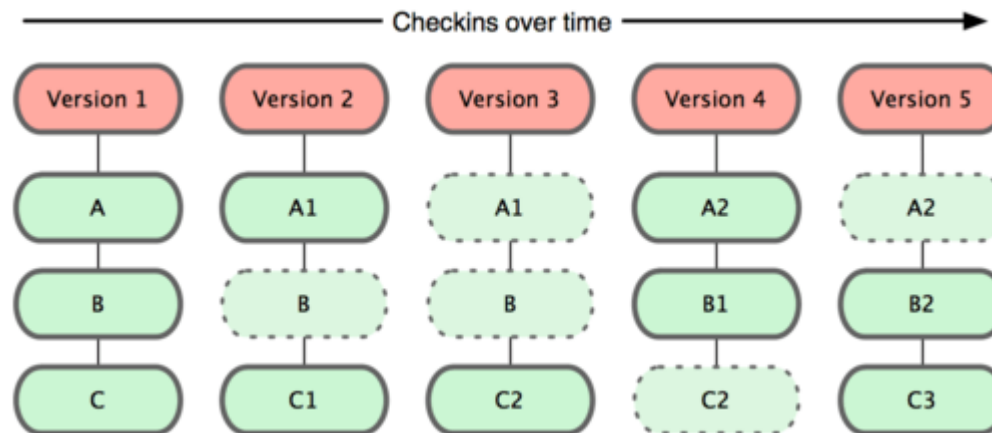
- En 2002, el proyecto del núcleo de Linux empezó a usar un DVCS propietario llamado BitKeeper.
- En 2005 Linus Torvalds diseña Git ante la ruptura de relaciones entre la comunidad que desarrollaba el núcleo de Linux y BitKeeper.
- **Objetivos:**
 - Velocidad.
 - Diseño sencillo.
 - Fuerte apoyo al desarrollo no lineal (miles de ramas paralelas).
 - Completamente distribuido.
 - Capaz de manejar grandes proyectos (como el núcleo de Linux).
 - Fácil de usar.

Fundamentos de Git

- La mayoría de los demás VCSs almacenan la información como una lista de cambios en los archivos respecto a una versión base.



- Git modela sus datos más como un conjunto de instantáneas de un mini sistema de archivos.
 - Si los archivos no se han modificado, Git sólo almacena un enlace a ellos.



Fundamentos de Git

- Casi cualquier operación es local.
 - Muy rápidas.
 - Puedes trabajar sin conexión.
- Git garantiza la integridad de la información.
 - Genera una suma de comprobación o checksum de 42 caracteres alfanuméricos usando hash SHA-1.
`24b9da6552252987aa493b52f8696cd6d3b00373`
 - Git no guarda todo por nombre de archivo, sino por el valor hash de sus contenidos.
- Git generalmente sólo añade información.
 - Difícil borrar información.
 - Difícil hacer algo que no se pueda revertir.

Fundamentos de Git

■ Los tres estados

■ Modificado (modified).

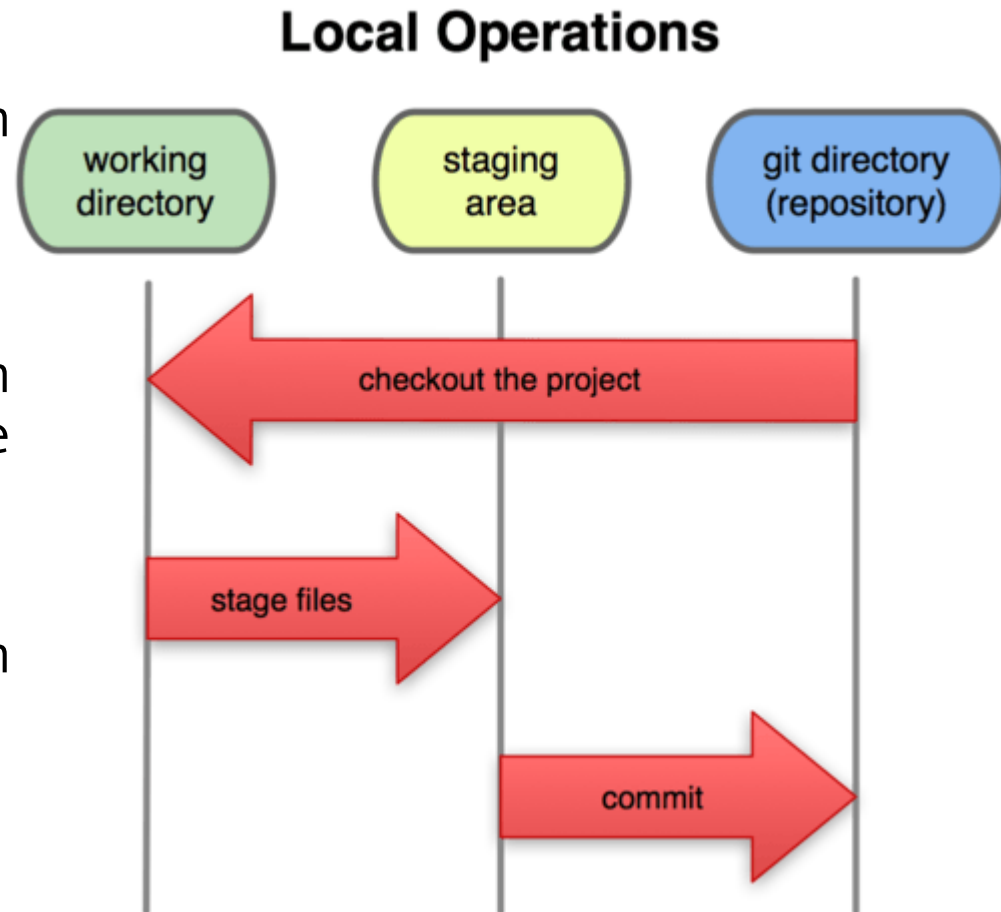
- Los ficheros se modifican en el directorio de trabajo.
- El directorio de trabajo es la copia de la versión del proyecto sobre la que trabajamos.

■ Preparado (staged).

- Los datos modificados son almacenados en archivos en el directorio de Git (.git) para el próximo commit.

■ Confirmado (committed).

- Los datos se almacenan en el directorio de Git (.git), donde Git almacena los metadatos y la base de datos de objetos para tu proyecto.
- Es lo que se copia cuando clonas un repositorio desde otro ordenador.



Instalación de Git

- Instalar Git desde Linux

- En una distribución basada en Debian como Ubuntu

```
# apt-get install git
```

- Instalar en Mac

- Usar el instalador gráfico de Git, que puedes descargar en:

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

- Si tienes MacPorts instalado (<http://www.macports.org>).

```
$ sudo port install git-core +svn +doc +bash_completion +gitweb
```

- Instalar en Windows

- Descarga el archivo exe del instalador desde la página de GitHub, y ejecútalo:

<http://msysgit.github.com/>

Configurar Git por primera vez

- `git config` permite obtener y establecer variables de configuración, que controlan el aspecto y funcionamiento de Git.
- Estas variables pueden almacenarse en tres sitios distintos
 - `/etc/gitconfig`
 - Contiene valores para todos los usuarios del sistema y todos sus repositorios.
 - Opción `--system` lee y escribe específicamente en este archivo.
 - `~/.gitconfig`
 - Específico a cada usuario.
 - Opción `--global` lee y escribe específicamente en este archivo.
 - `.git/config`
 - Específico del repositorio.
- Cada nivel sobrescribe los valores del nivel anterior.
 - Así los valores de `.git/config` tienen preferencia sobre los de `/etc/gitconfig`.

Configurar Git por primera vez

■ Tu identidad

- Debes especificar como mínimo tu nombre y correo.

```
$ git config --global user.name "John Doe"
```

```
$ git config --global user.email
```

- Cuando se hace un commit se usa esta información.
 - Se puede cambiar más adelante.

■ Tu Editor por defecto

- Se utiliza cuando Git necesita que introduzcas un mensaje.

```
$ git config --global core.editor gedit
```

- Por defecto el del sistema, normalmente `vi` o `vim`

■ Tu herramienta de diferencias

- Usada para resolver conflictos de unión (merge)

```
$ git config --global merge.tool vimdiff
```

- Git acepta `kdiff3`, `tkdiff`, `meld`, `xxdiff`, `emerge`, `vimdiff`, `gvimdiff`, `ecmerge`, `opendiff`,...

Configurar Git por primera vez

- Comprobar configuración

```
$ git config --list
user.name=Scott Chacon
user.email=schacon@gmail.com
color.status=auto
color.branch=auto
color.interactive=auto
color.diff=auto
...
```

- Puede que veas claves repetidas,
 - Git lee la misma clave de distintos archivos
 - Por ejemplo, de `/etc/gitconfig` y `~/.gitconfig`
 - Git usa el último valor.
- Ver valor de una clave usar `git config {clave}`

```
$ git config user.name
Pepe Perez
```

Configurar Git por primera vez

- Obtener ayuda

- Hay tres formas

- \$ git help <comando>

- \$ git <comando> --help

- \$ man git-<comando>

- Ejemplo:

- \$ git help config

- Puedes acceder a ellos desde cualquier sitio, incluso sin conexión.