

GIT

TU JALAS, YO EMPUJO

Carlos Remuzzi / bit.ly/1SHzcMM

FLISOL 2016 LA PAZ

git - *n.* "worthless person," 1946, British slang, a southern variant of Scottish get "illegitimate child, brat," which is related to beget.

~ *Etymology Dictionary, Douglas Harper*

I'm an egotistical b*stard, and I name all my projects after myself. First 'Linux', now 'Git'.

~ *Linus Torvalds*

CONTENIDOS

1. Que cosa es GIT
2. Instalar y configurar GIT
3. Conceptos fundamentales
4. Mas herramientas

1. QUE COSA ES GIT

- SCM - Sistema de manejo de código fuente
- Open Source
- Integridad
- Alto Rendimiento
- Distribuido

SISTEMA DE MANEJO DE CODIGO FUENTE



creado en el 2005 por Linus Torvalds



mantenido por Junio C Hamano

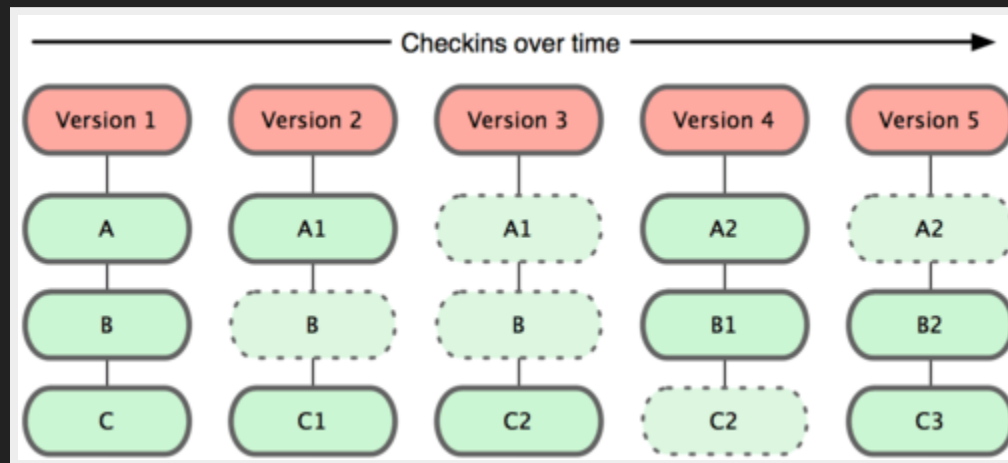
OPEN SOURCE

codigo fuente disponible:

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

INTEGRIDAD

Conjunto de instantáneas de un mini sistema de archivos



INTEGRIDAD

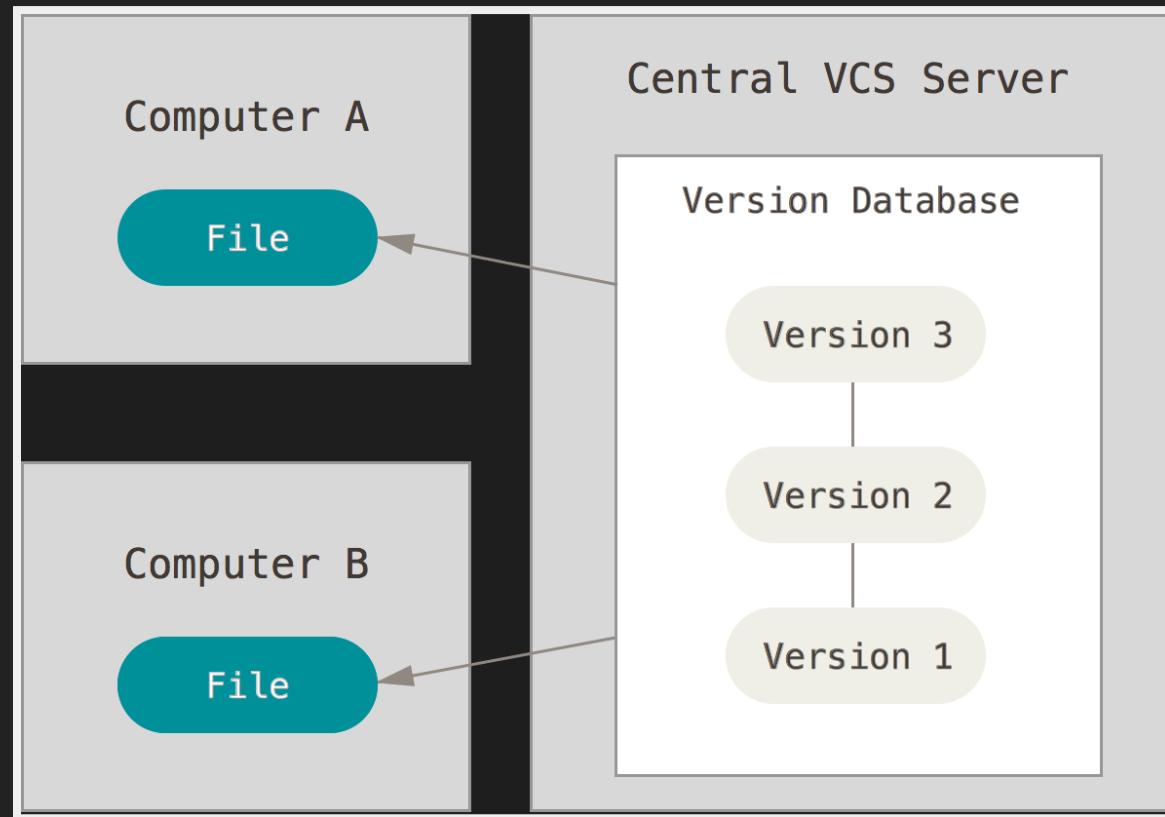
- se calcula un hash de cada instantanea
- SHA-1 checksum, 40 caracteres
- se verifica antes de escribir cualquier dato

ALTO RENDIMIENTO

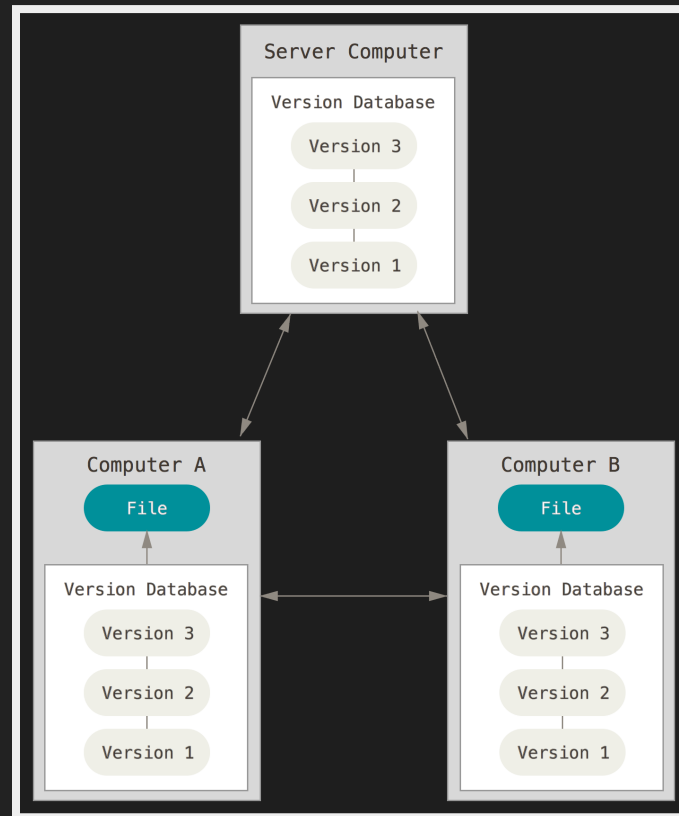


SISTEMA DISTRIBUIDO

SISTEMAS CENTRALIZADOS



SISTEMAS DISTRIBUIDOS



2.1 INSTALACION

LINUX:

```
apt-get install git # Debian / Ubuntu  
yum install git-all # Fedora
```

WINDOWS:

git-scm.com/download/win

LINUX:

git-scm.com/download/mac

2.2 CONFIGURACION

```
$ git config --global user.name "Jimmy Best"  
$ git config --global user.email jimmy@best.com
```

3. CONCEPTOS FUNDAMENTALES

CREA UN REPOSITORIO NUEVO

```
$ git init
```


VERIFICA EL ESTADO DE TU REPOSITORIO

```
$ git status
```

```
On branch master
```

```
Your branch is up-to-date with 'origin/master'.
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
    new file:   addedfile.txt
```

```
Changes not staged for commit:
```

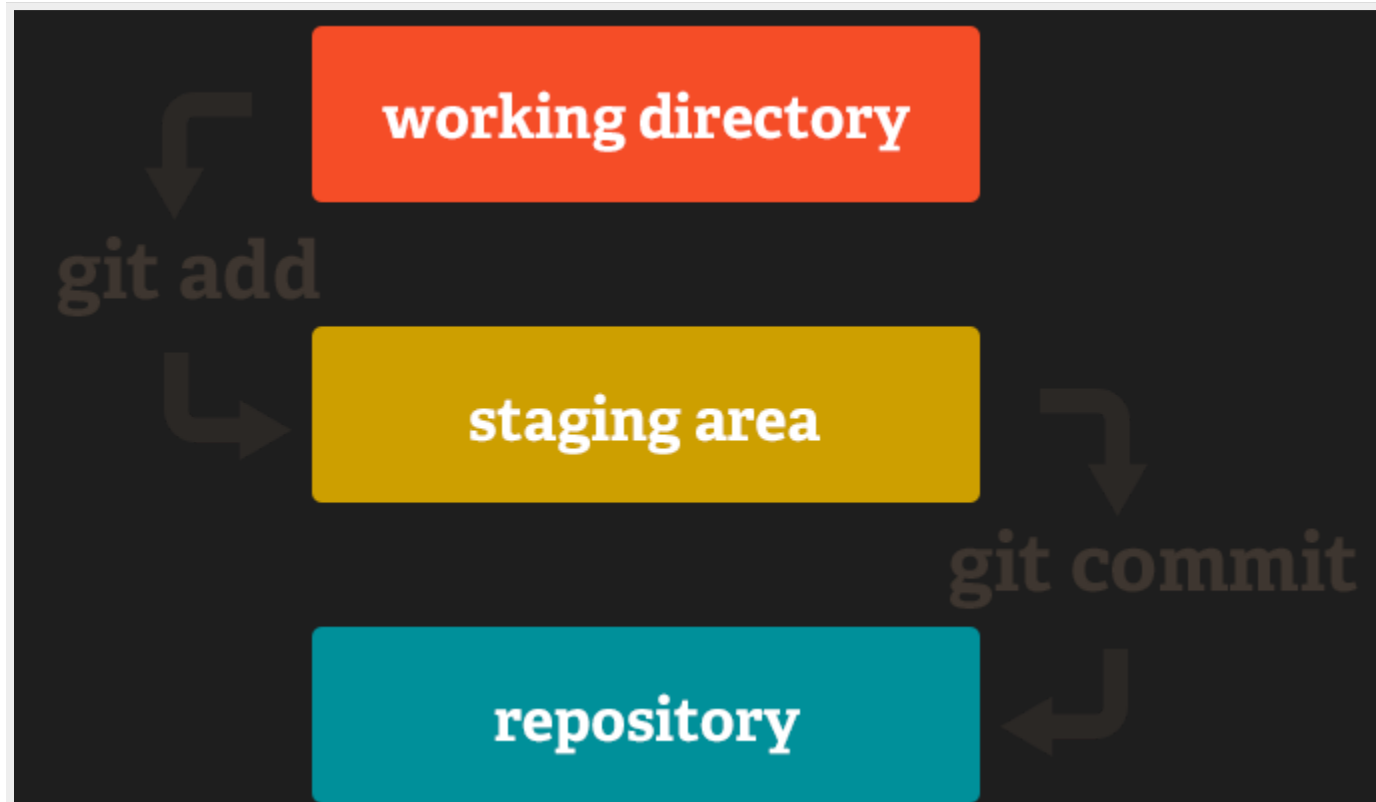
```
(use "git add <file>..." to update what will be committed)
```

```
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
    modified:   index.html
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```



AÑADIR CAMBIOS

```
$ git add
```

HACER COMMIT

```
$ git commit      # la staging area  
$ git commit -a   # todos los archivos
```

VER LA HISTORIA DE TUS COMMITS

```
$ git log
commit 0f1bf4b1b5cb74342de9c1f6280b39ca5a1f79eb
Author: Carlos Remuzzi <cremuzzi@agetic.gob.bo>
Date:   Tue Apr 19 20:22:57 2016 +0000

    adding configuration

    </cremuzzi@agetic.gob.bo>
```

CONFIGURAR UN REPOSITORIO REMOTO

```
$ git remote add URL_DEL_REPOSITORIO
```

ENVIAR A REPOSITORIO REMOTO

```
$ git push -u origin master
```

JALAR DEL REPOSITORIO REMOTO

```
$ git pull
```


CREAR UNA NUEVA RAMA

```
$ git branch NOMBRE_DE_LA_RAMA # crea la rama  
$ git checkout -b NOMBRE_DE_LA_RAMA # crea la rama y entra a la misma
```

CAMBIAR DE RAMA

```
$ git checkout NOMBRE_DE_LA_RAMA
```

UNIR DOS RAMAS

```
$ git merge NOMBRE_DE_LA_SEGUNDA_RAMA
```

4. MAS HERRAMIENTAS

- blame
- cherry-pick
- reset
- rebase
- reflog
- squash & split
- bisect

FIN

lecturas recomendadas

- `git --help`
- pagina web de GIT: git-scm.com
- blog de Junio C Hamano: git-blame.blogspot.com
- GIT avanzado: bit.ly/git-pycon-2015