



Sistema de control de versiones

Luis Octavio Ramírez Fernández

26 de febrero de 2015

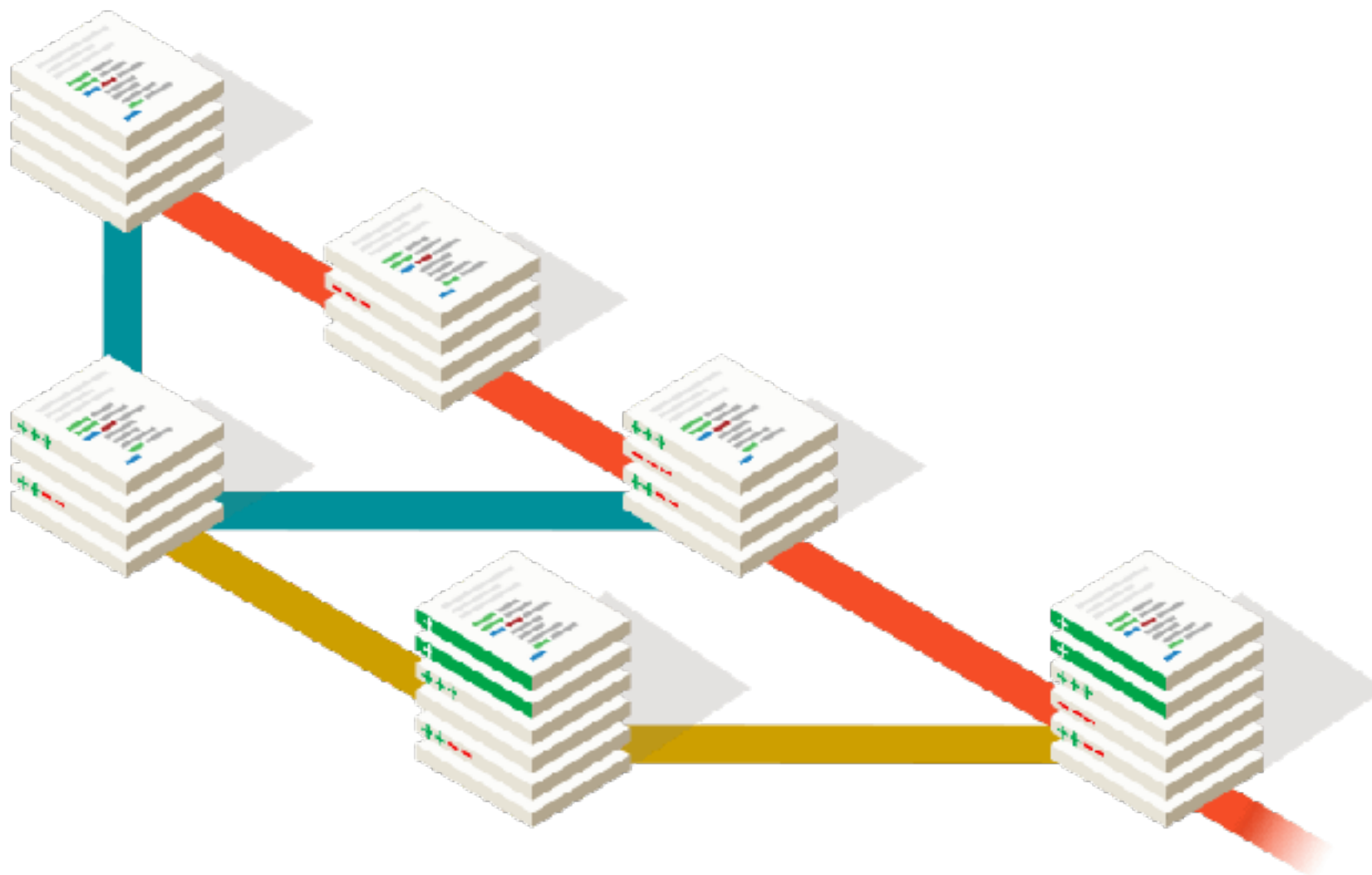


¿Qué es un sistema de versioning?

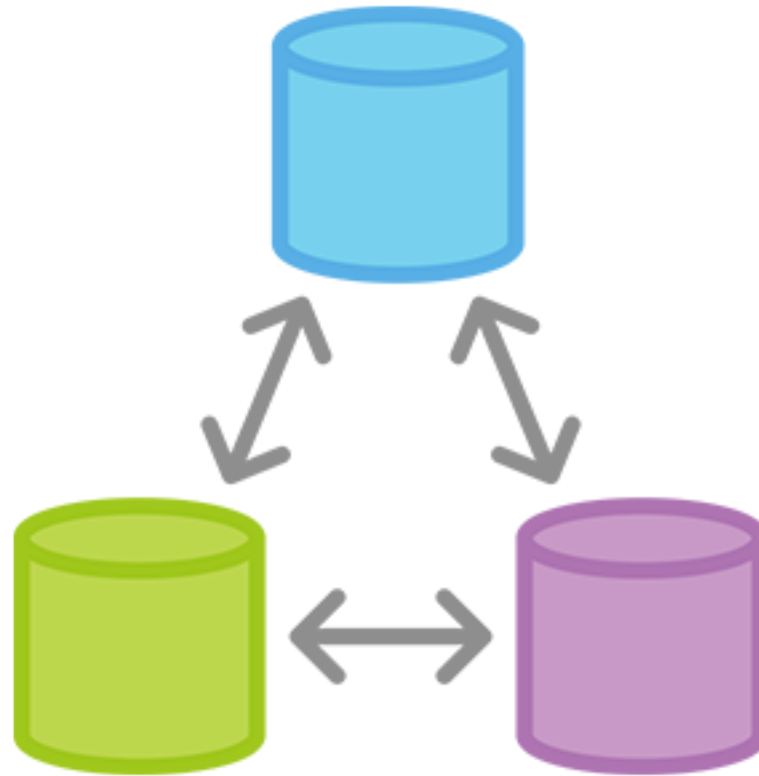
El control de versiones es un sistema que registra los cambios realizado a un archivo o conjunto de archivos a través del tiempo, de manera que puedas llamar una versión específica tiempo después.



git










SparkleShare








SparkleShare Setup

Where's your project hosted?

 **On my own server**
Everything under my control

 **Bitbucket**
Free code hosting for Git and Mercurial

 **Github**
Collaborate on projects hosted on Github

 **Gitorious**
Open source infrastructure for hosting open source projects

Address:
ssh://[Softpedia]:10.9.0.47
ssh://[user@]hostname[:port]

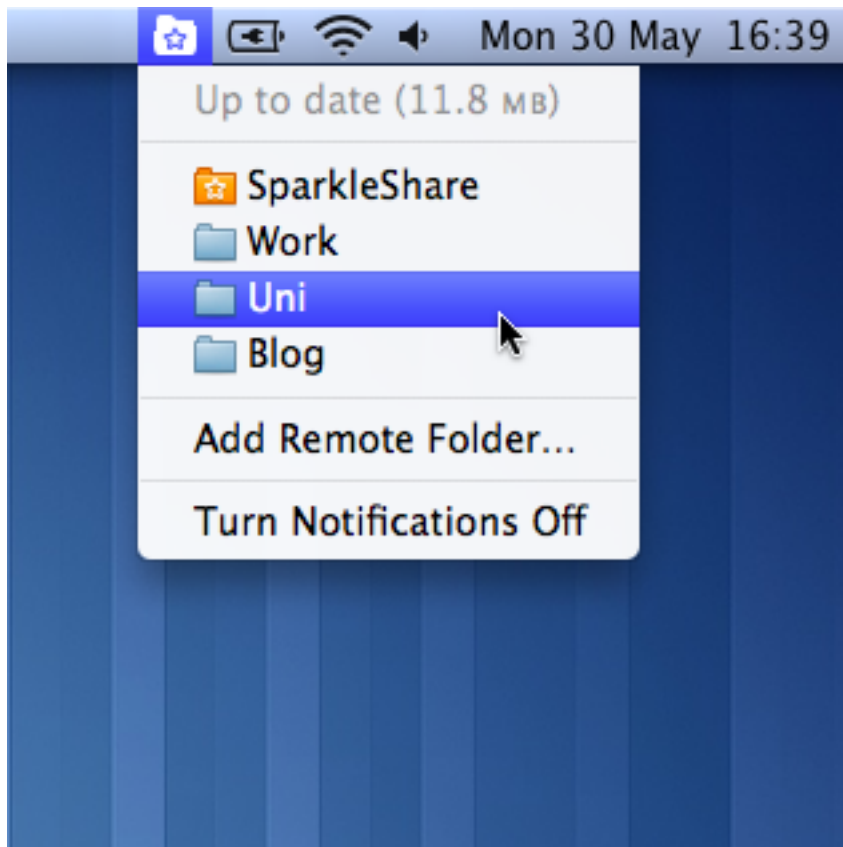
Remote Path:
/Softpedia/Softpedia Test
/path/to/project

☐ Fetch prior revisions

Add **Cancel**



Servidor Sparkle Share





- Repositorio central
- Instala el cliente
- Firmas digitales
- Validan firma digitales





SparkleShare Setup

Where's your project hosted?

 **On my own server**
Everything under my control

 **Bitbucket**
Free code hosting for Git and Mercurial

 **Github**
Collaborate on projects hosted on Github

 **Gitorious**
Open source infrastructure for hosting open source projects

Address:
ssh://[Softpedia]:10.9.0.47
ssh://[user@]hostname[:port]

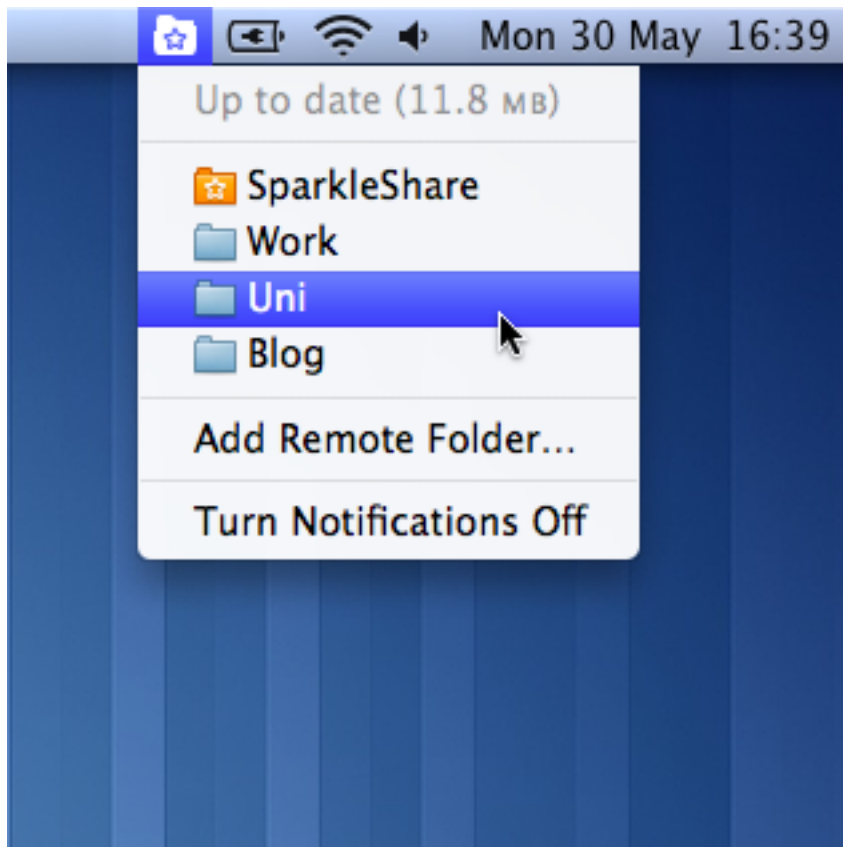
Remote Path:
/Softpedia/Softpedia Test
/path/to/project

☐ Fetch prior revisions

Add Cancel



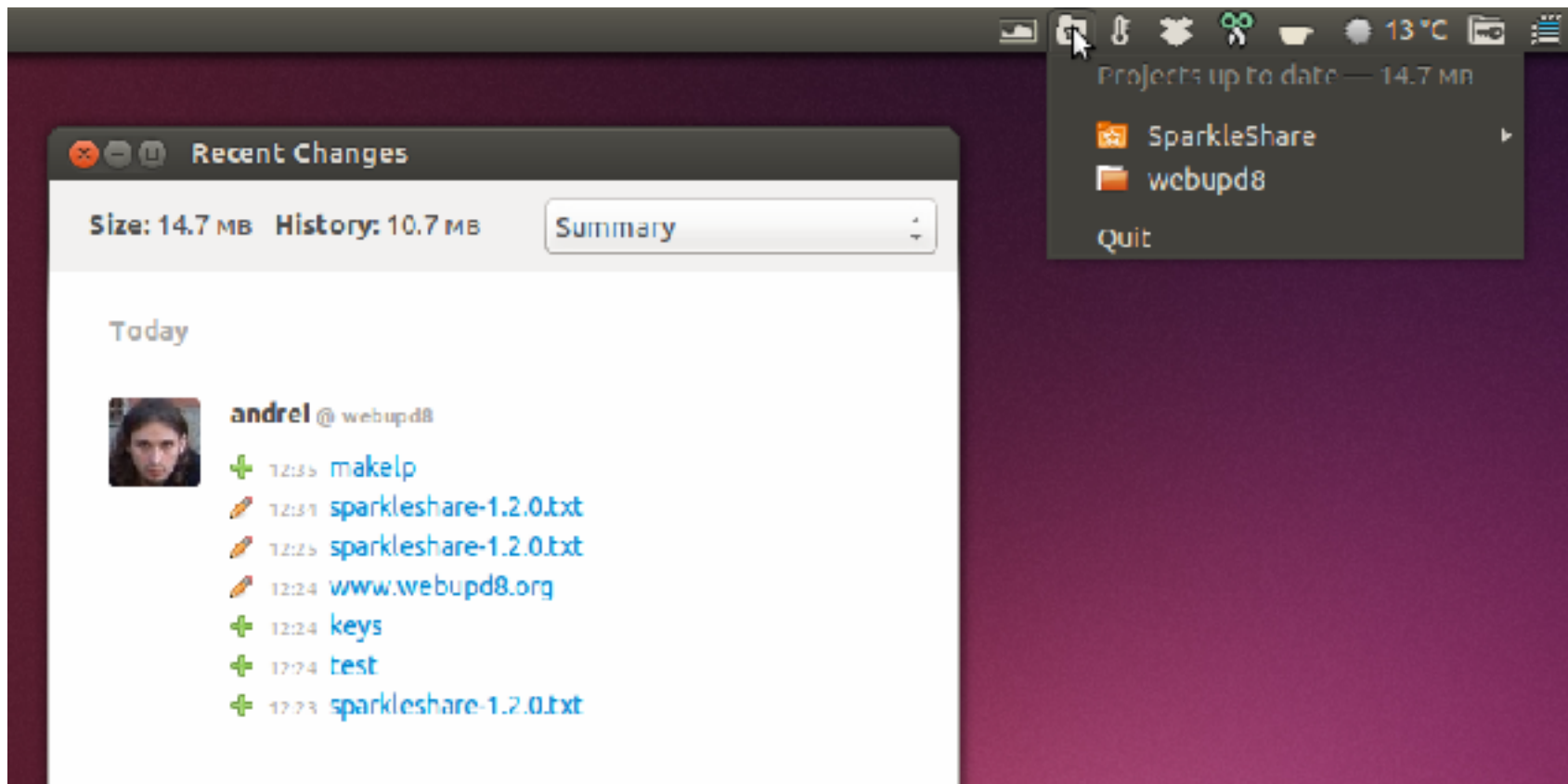
Servidor Sparkle Share



- Repositorio central
- Instala el cliente
- Firmas digitales
- Validan firma digitales



Servidor Sparkle Share





Flujo de trabajo con GIT



Configuración básica

```
git config --global user.name "Juan Pérez"  
git config --global user.email juan@example.com  
git config --global color.ui auto
```



Iniciando con git

Crear un repositorio local

```
$ git init
```

Agregar archivos a git

```
$ git add archivo
```

```
$ git add .
```




Trabajando con git

Estado del repositorio

\$ git status

Confirmar cambios

\$ git commit -m "Mis cambios"

Revisando confirmaciones

\$ git log



Trabajando con git

```
git log -p
```

```
git log - - stat
```

```
git log --pretty=online
```

```
git log --all --pretty=format:"%h %cd %s (%an)" --  
since='7 days ago'
```



Trabajando con git

```
git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
```

```
git log --decorate
```

```
git log --graph --pretty=format:'%C(yellow)%h%Creset \n-%C(yellow)%d%Creset %s %Cgreen(%cr)%C(bold blue)<%an>%Creset' \n--abbrev-commit --date=short
```



Trabajando con git

Crear la rama en el punto actual. Es necesario hacer checkout a la misma.

```
$ git branch <nombre>
```

Crea la rama a partir del commit dado. Es necesario hacer checkout.

```
$ git branch <nombre> <COMMIT>
```





Trabajando con git

Crear rama en el punto actual y hacerle checkout

```
$ git checkout -b <nombre>
```

Crear la rama a partir del commit dado y hacerle checkout.

```
$ git checkout -b <nombre> <COMMIT>
```

Renombrar la rama.

```
$ git branch -m <actual> <nuevo>
```



Trabajando con git

No toca los cambios locales

```
$ git checkout <COMMIT>
```

```
$ git checkout <rama>
```

Sobrescribe los cambios locales

```
$ git checkout -f <COMMIT>
```



Trabajando con git

Fusiona la rama indicada en la rama actual

```
$ git merge <nombre>
```

Muestra los ficheros conflictivos y las diferencias

```
$ git diff
```




Trabajando con git

Abortar el proceso y volver a la situación anterior al intento de merge

```
$ git reset --hard HEAD
```

Deshacer si ya se había confirmado con git commit

```
$ git reset --hard ORIG_HEAD
```



Trabajando con git

Abortar el proceso y volver a la situación anterior al intento de merge

```
$ git reset --hard HEAD
```

Deshacer si ya se había confirmado con git commit

```
$ git reset --hard ORIG_HEAD
```

```
$ git reset --hard HEAD~1
```

```
$ git reset --hard HEAD~2
```



Trabajando con git

Sobrescribe cambios locales sin preguntar

```
$ git checkout <COMMIT> -- <path>
```

```
$ git checkout <COMMIT> -- ./<path>
```



Trabajando con git

\$ git tag -s <nombre> -m <mensaje>

Mostrar lista de tags

\$ git tag

Mostrar lista y descripción

\$ git tag -n



Trabajando con git

Clonar y hacer checkout del HEAD de la rama actual

```
$ git clone <ruta al repositorio>
```

```
$ git pull
```

```
$ git push origin <branch>
```

```
$ git push --all
```



Trabajando con git

```
$ git remote
```

```
$ git remote -v
```

```
$ git remote add rama ssh://usuario@servidor/  
repo.git
```



Trabajando con git

```
$ mkdir repositorio  
$ cd repositorio  
$ git init - -bare
```



Referencias

1. GIT, <http://git-scm.com>
2. Github, <https://github.com>
3. Pro Git, <http://git-scm.com/book/en/v2>
4. Sparkle Share, <http://sparkleshare.org>



¿Preguntas?



¡ Gracias !



Contacto:

Luis Octavio Ramírez Fernández.
lr Ramirez@igg.unam.mx