

Introducción a Git

Parte primera

Antonio Maurandi López

3 de abril de 2017

UNIVERSIDAD DE
MURCIA

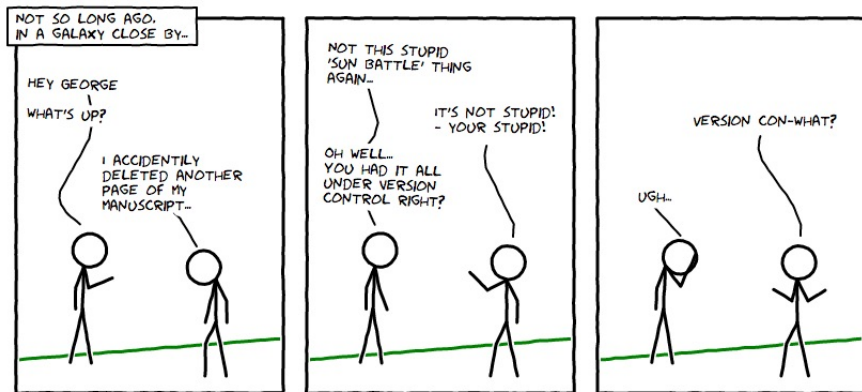


¿Qué es Git?

Una sesión básica de Git

GUIs para Git

Version control



¿Qué es Git?

Definición

Git (pronunciado “guit”) es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.



Cómo funciona

1. Git modela sus datos como un cto de instantáneas de un mini sistema de archivos
2. Cada vez que confirmas un cambio, o guardas el estado de tu proyecto en Git, Git hace una *foto* del aspecto de todos lo archivos en ese momento, y guarda una referencia a esa instantánea
3. Si los archivos no se han modificado, Git no almacena el archivo de nuevo, hace un enlace al archivo anterior idéntico

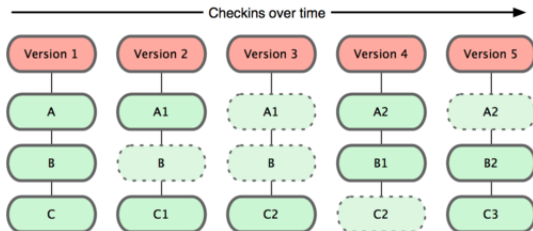


Figura 1: Git almacena la información como instantáneas del proyecto

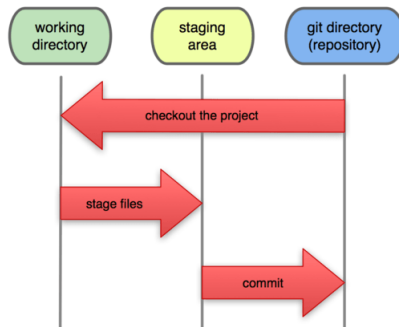
Particularidades

- Casi cualquier operación es local
 - hay muy poco que no puedas hacer si estás desconectado
- Git tiene integridad: `checksum`
 - no puedes perder información o sufrir corrupción de archivos sin que Git lo detecte...
- Git generalmente sólo añade información
 - es muy difícil conseguir que el sistema haga algo que no se pueda deshacer
 - puedes perder o estropear cambios que no has confirmado todavía

Los tres estados

1. **confirmado** o `committed`: los datos están almacenados de manera segura en tu base de datos local
2. **modificado** o `modified`: has modificado el archivo pero todavía no lo has confirmado a tu base de datos
3. **preparado** o `staged` : has marcado un archivo modificado en su versión actual para que vaya en tu próxima confirmación

Local Operations



Flujo de trabajo básico

1. Modificar archivos en el directorio de trabajo
2. Preparar archivos
3. Confirmar cambios

Flujo de trabajo básico en remoto

1. Traer la última versión
2. Modificar archivos en el directorio de trabajo
3. Preparar archivos
4. Confirmar cambios
5. *Subir* versión

Instalación

- GNU/Linux: `$ apt-get install git`
- MacOX: <http://sourceforge.net/projects/git-osx-installer/>
- Wiiiindous: <http://msysgit.github.com/>

Configuración

- usuario: ~/.gitconfig
- repositorio .git/config

Cada nivel sobrescribe los valores del nivel anterior, por lo que los valores de .git/config tienen preferencia sobre los de /etc/gitconfig

```
$ git config --global user.name "Antonio Maurandi"  
$ git config --global user.email amaurand@um.es
```

Comprobar configuración y acceso remoto

- `$ git config --list`

```
$ git config --list
user.name=Antonio Maurandi López
user.email=amaurandi@um.es
gui.editor=gedit
gui.recentrepo=/home/amaurandi/Git/007cursodocint
gui.spellingdictionary=en
gui.recentrepo=/home/amaurandi/Git/gitdemo
...
remote.origin.url=git@gitlab.atica.um.es:amaurandi.um.es/gitdemo.git
...
```

- Si trabajamos en remoto: generar una **SSH key**

- Por consola: `ssh-keygen -t rsa -b 4096 -C 'amaurandi@um.es'`
- Desde Rstudio: Tools/Global Options/Git/SVN, click Create RSA key

Una sesión básica de Git

Una sesión básica de Git. local

1. `git add`: añade ficheros a seguimiento...
2. `git commit -m "comentario"`

Repros en remoto

Hay dos opciones:

1. Está en remoto y la quiero traer (`clone`)
2. Está en local y la quiero *subir*: (`push`)
 - es más liosa o compleja y requiere muchas veces del truco de crearse una con el mismo nombre en remoto, clonarla vacía, copiar en ese directorio los ficheros a subir, y entonces subirla (`push`)

Clonar

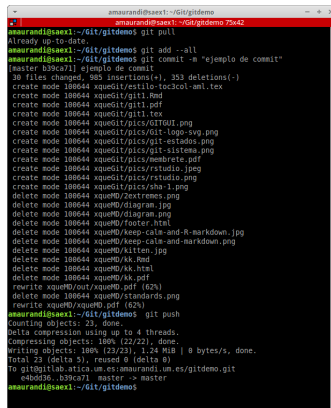
1. Traerse una repro remota:

- `git remote add origin`
`git@gitlab.atika.um.es:amaurandi.um.es/gitdemo.git`

2. Después es lo mismo pero tb hay que bajar cambios posibles (pull) añadir cambios y subirlos con un push:

- `git pull`
- `git add --all`
- `git commit -m "comentario"`
- `git push`

Sesión remota tipo



```
amaurandi@saexi: ~/Git/gitdemo
$
amaurandi@saexi:~/Git/gitdemo$ git pull
Already up-to-date.
amaurandi@saexi:~/Git/gitdemo$ git add --all
amaurandi@saexi:~/Git/gitdemo$ git commit -m "ejemplo de commit"
[master b39ca71] ejemplo de commit
30 files changed, 985 insertions(+), 353 deletions(-)
create mode 100644 xqueGit/estilo-toc3col-aml.tex
create mode 100644 xqueGit/git1.Rmd
create mode 100644 xqueGit/git1.pdf
create mode 100644 xqueGit/git1.tex
create mode 100644 xqueGit/pics/GITGUI.png
create mode 100644 xqueGit/pics/Git-logo-svg.png
create mode 100644 xqueGit/pics/git-estados.png
create mode 100644 xqueGit/pics/git-sistema.png
create mode 100644 xqueGit/pics/membrete.pdf
create mode 100644 xqueGit/pics/rstudio.jpeg
create mode 100644 xqueGit/pics/rstudio.png
create mode 100644 xqueGit/pics/sha-1.png
delete mode 100644 xqueMD/zextremes.png
delete mode 100644 xqueMD/diagram.png
delete mode 100644 xqueMD/footer.html
delete mode 100644 xqueMD/keep-cala-and-R-markdown.png
delete mode 100644 xqueMD/keep-cala-and-markdown.png
delete mode 100644 xqueMD/kitten.jpg
delete mode 100644 xqueMD/kk.Rmd
delete mode 100644 xqueMD/kk.html
delete mode 100644 xqueMD/kk.pdf
rewrite xqueMD/out/xqueMD.pdf (62%)
delete mode 100644 xqueMD/standards.png
rewrite xqueMD/xqueMD.pdf (62%)
amaurandi@saexi:~/Git/gitdemo$ git push
Counting objects: 23, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (22/22), done.
Writing objects: 100% (23/23), 1.24 MiB | 0 bytes/s, done.
Total 23 (delta 5), reused 8 (delta 0)
To git@gitlab.atica.un.es:amaurandi.un.es/gitdemo.git
 e4bdd36..b39ca71 master -> master
amaurandi@saexi:~/Git/gitdemo$
```

Figura 3: pincha para agrandar imagen aquí

Más sentencias y opciones: https://es.wikipedia.org/wiki/Git#.C3.93rdenes_b.C3.A1sicas

commit: comentarios



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT
MESSAGES GET LESS AND LESS INFORMATIVE.

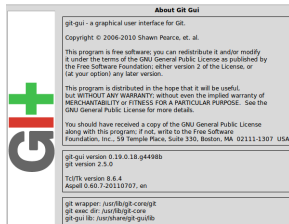
Figura 4: xkcd: git commit -m "usar comentarios descriptivos"

GUIs para Git


Con GIT GUI

El comando `add` es `staged changes`, es decir *staging* hacemos *add*

- Ventana Unstaged Changes, marcando ficheros aquí y con `Ctrl+T` los pasamos a Staged Changes (*will commit*), es decir hacemos un *add* de esos ficheros.
- Después hay un botón de *commit* y otro de *push*. El commit te solicita siempre un comentario.
- Para hacer un *pull* en GIT GUI
 1. menú Remote: fetch from origin
 2. menu Merge: Local merge



con Rstudio

Merece la pena hacerlo con Rstudio , es más *claro*, y además ofrece una utilidad sencilla para crear la clave ssh.

Uso básico:

1. <http://happygitwithr.com/push-pull-github.html>
2. Identificación http o ssh (preferido)
 - ssh:
<http://happygitwithr.com/ssh-keys.html#ssh-keys>

GitLab: <http://gitlab.atica.um.es/>

gitdemo

★ 0

P 0

SSH

HTTP

git@gitlab.atica.um.es:amaurandi.1

+

8 COMMITS

1 BRANCH

0 TAGS

9.34 MB

ADD CHANGELOG

ADD LICENSE

ADD CONTRIBUTION GUIDE

b39ca714 ejemplo de commit · about an hour ago by Antonio Maurandi López

esto es un repo git

En esta repro podemos encontrar dos documntos:

- Introducción a **GIT**
- Introducción a **Markdown**

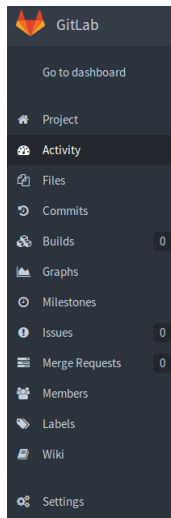
Pretendemos ir de lo sencillo a lo complejo

GitLab: commits

The screenshot shows the GitLab interface for the 'gitdemo' repository. At the top, there are navigation tabs: 'Commits' (8), 'Network', 'Compare', 'Branches' (1), and 'Tags' (0). Below the tabs, the repository name 'gitdemo' is displayed next to a dropdown menu currently set to 'master'. A search icon is located on the right. The main content area lists commits. The first commit is dated '29 Mar, 2017' and has '2 commits'. It features a commit message 'ejemplo de commit' with a preview of the commit text: 'Hacemos un commit después de ordenar todos los files y directorios para que qued...'. The commit hash is 'b39ca714' with a 'Browse Files' link. The second commit is dated '28 Mar, 2017' and has '1 commit'. Its message is 'sobre markdown beamer motivacional' with a preview of the commit text: 'Antonio Maurandi López authored about 16 hours ago'. The commit hash is 'add5dd0c' with a 'Browse Files' link.

Commit Date	Commit Message	Commit Hash
29 Mar, 2017 2 commits	ejemplo de commit Antonio Maurandi López authored about an hour ago Hacemos un commit después de ordenar todos los files y directorios para que qued... Antonio Maurandi López authored about 4 hours ago	b39ca714 Browse Files »
28 Mar, 2017 1 commit	sobre markdown beamer motivacional Antonio Maurandi López authored about 16 hours ago	add5dd0c Browse Files »

GitLab: más opciones



En próximas reuniones

- histórico de confirmaciones
- Deshaciendo cosas : `git commit --amend`
- Creando etiquetas: `git tag`
- Ramas o *branches*
- ramificar y fusionar

Keep calm



Referencias

- <https://git-scm.com/book/es/v1/>
- <http://happygitwithr.com/>
- Using Version Control with RStudio

xkcd



Figura 5: xkcd: unos pocos comandos y ya está