

Git: software de control de versiones

¿Cómo usarlo?

Daniel Lubián Arenillas

10 de febrero de 2018



"FINAL".doc



FINAL.doc!



FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.##\$%WHYDID
ICOMETOGRADSCHOOL?????.doc

JOSE CUNHA © 2012



Wikipedia

`git` 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.

Para instalarlo:

<https://git-scm.com/>





Es especialmente potente con archivos de texto

- Trabajos con Matlab, Fortran, Python, C, GMAT ... **cualquier código fuente.**
- Documentos hechos con **L^AT_EX**.
- Cualquier otro archivo de texto (p.ej: datos de resultados):
.txt, .csv, .dat ...

→ Si lo puedes abrir bien con el **notepad.exe**, va perfecto.

Con archivos distintos pierdes funcionalidades, por lo que pierde algo de sentido. Requiere ser mucho más organizado.

- Catia, Excel, Word,...



- Un histórico de versiones bien organizado:
 - **qué,**
 - **quién,**
 - **cuándo,** y
 - **por qué.**
- Poder **volver atrás** fácilmente.
- Poder introducir nuevos cambios **sin perder lo anterior.**
- Poder **comparar versiones** de forma visual y detallada.
- **Mezclar cambios** introducidos por varias personas a la vez, sin riesgos.
- Y mucho más...



- **init** : inicializa el VCS
- **status** : cómo están las cosas → estado
- **add** : añade archivos al commit
- **commit** : comete (hace efectivos) esos cambios
- **diff** : muestra las diferencias
- **blame** : muestra el “culpable” de cada línea
- **branch** : crea/maneja ramas del código
- **merge** : mezcla commits, branches, etc
- **checkout** : cambia a otro commit u otra rama



- **clone** : crea una copia de un repo remoto en la que estás
- **push** : “empuja” (sube) los commits del repo local a un repo remoto
- **pull** : descarga (baja) los commits del repo remoto en el local

(Igual que en ConCORDE)



- Interesantes para guardar el trabajo en otro sitio.
- Para la colaboración son fundamentales: mucho mejor que GDrive o Dropbox.
- Proporcionan una “interfaz gráfica” para manejar todo esto de forma cómoda.
- Añaden nuevas posibilidades: *issues*, tareas, comentarios ...



- <https://github.com/>
- Código cerrado
- Sin pagar **no** hay repositorios privados
- Mucho más conocido



- <https://gitlab.com/>
- Código abierto
- Sin pagar **sí** hay repositorios privados
- ¿?

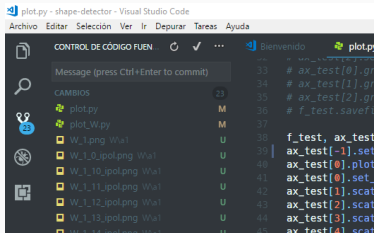
Otros servicios: Bitbucket (<https://bitbucket.org/>)

Para quien no quiera línea de comandos...



Visual Studio Code (Microsoft)

- Editor de código abierto
- Se puede hacer prácticamente todo, pero para algunas cosas es más cómodo ir a la línea de comandos



Atom (Github)

- Editor de código abierto
- Excelente integración con Github

TortoiseGit (para Windows)
ungit (multiplataforma)

Github Desktop (Win, Mac)

GitKraken (multiplataforma)

⋮

<https://git-scm.com/downloads/guis>



Es libre

Es cómodo

Te da seguridad

Es fácil de usar

Es perfecto para nuestros trabajos