

Introducción a Git & GitHub

Control de versiones o Gestión de la configuración

Prof. Sebastian Saaibi & David Cardozo¹

¹Física

Lectura 6 Herramientas Computacionales
Universidad de los Andes

13 de marzo de 2015

- 1 Introducción: Control de Versiones
- 2 Control de versiones con Git
- 3 Repositorios distribuidos: Github

Control de Versiones

Control de versiones o gestión de la configuración (VCS) (SCC) (SCM)

Sistemas para registrar la historia de los cambios que van sufriendo un conjunto de ficheros.

Funciona para:

- Poder saber cuándo se modificó y quién lo hizo, cada fichero de un proyecto
- Poder reconstruir el estado de los ficheros al estado que tenían en algún momento del pasado.
- Poder coordinar los cambios que realiza un conjunto de desarrolladores sobre los ficheros de un proyecto.

Sistema de control de versiones (VCS) Herramienta de software que ayuda a gestionar el control de versiones: *Ej: SVN, Git*

git I

Git es n sistema de control de versiones distribuido.

Repositorio o repo

Guarda la historia completa de un subconjunto de ficheros del proyecto

Github ofrecen hospedaje para git en la red

- Útiles para colaborar entre varios miembros de un equipo de desarrollo.
- Útil para desarrolladores individuales para hacer backups del repositorio
- Portafolio de presencia y soporte en la web.

git: el repositorio o repo

El repo

Donde git guarda la historia completa de algún subconjunto de ficheros del proyecto

Vamos a inicializar el repo de un proyecto.

```
mkdir testgit  
cd testgit  
git init
```

Figura: Iniciando git

El repositorio queda inicializado, pero vacío, en:

testgit/.git/

git: *tracked files*

Es el subconjunto de ficheros del proyecto que forman parte del repositorio. Sus versiones se van guardando cada vez que se compromete su estado (se hace commit)

add command

git add '*.txt' añade recursivamente los ficheros acabados en txt al conjunto de *tracked files*

No todos los ficheros del proyecto tienen que formar parte de los *tracked files*.

En **.gitignore** se especifican los ficheros que no deben añadirse.

.gitignore

En la raíz del proyecto el fichero **.gitignore** especifica los ficheros que no deben añadirse al repo.

un comentario

**.aux # ignorar los ficheros que acaben en .aux*

carpeta/ # ignorar todos los ficheros del subdir carpeta/

git: commit de ficheros

Cuando estás contento con el estado del proyecto haces commit para que se refleje en el repo el estado actual de los *tracked files*

¿Qué permite commit?

Recuperar en el futuro el estado (foto) comprometido de los tracked files.

NO ES UNA COPIA DE BACKUP DE ARCHIVOS

Es muy importante añadir un texto que explique el estado que se compromete.

```
git commit -m 'msg'
```


git: hands on

```
git config --global core.editor 'gedit'
mkdir gittest
cd gittest; touch a.txt; touch b.txt
git init
git add a.txt
git status
git b.txt
git commit
git status
echo "nueva linea" > a.txt
git status
git diff
git commit -a
git status
git log
```

Github: Introducción

Permite almacenar gratuitamente todos los repos que quieras (siempre que sean públicos) Al principio generas pareja clave pública/privada para autenticare en el servicio github con ssh.

Push / Pull

Usando git, haces **push** de tu repo a github, creando allí una réplica. Otros desarrolladores pueden hacer *push* de sus cambios y **pull** de los de otros.

Preparación para usar github

- Creamos una cuenta gratis en github
- Informamos a git del nombre de usuario y del correo para que cada commit quede identificado:

```
git config --global user.name 'username-de-github'  
git config --global user.email 'david@correo.com'
```

Creación de las parejas de claves para identificación:

```
cd ~/.ssh  
ssh-keygen -t rsa -C david@correo.com  
<Enter>  
#entre una contraseña pass123  
chmod 0600 *
```

git configuración

Creamos un repositorio remoto en [https://github.com/](https://github.com/HerramientasComputacionales) **HerramientasComputacionales**

```
git remote add origin git@github.com:username/<Herramientas...>.git  
git push origin master
```