

Apunte Git

Organización del Computador 2

Primer Cuatrimestre 2017

1. Instalacion

Desde Linux abrimos una terminal y actualizamos las listas de paquetes y dependencias para luego estar seguros de bajar la última versión de *Git* (o al menos la última en las listas de apt-get)

```
> sudo apt-get update
> sudo apt-get install git
```

En la mitad de la instalación *apt-get* les va a preguntar si quieren bajar los archivos a su disco. Ponen que sí (*aka: y*)

```
> Do you want to continue? [y/n] y
```

2. Configuracion

Cada vez que hagan *push* o *pull* *Git* va a pedirles que se identifiquen (en el caso de los *pull* es solo para repositorios privados). Para que no les pregunte esto todo el tiempo pueden guardar sus credenciales en la configuración de git

Supongamos que nuestro usuario en <https://git.exactas.uba.ar> es *master-of-the-universe* y nuestro email es *mi-email@dc.uba.ar*

```
> git config --global user.name "master-of-the-universe"
> git config --global user.email "mi-email@dc.uba.ar"
```

3. Usando Git

Hay una gran gama de colores para usar Git con diferencias sutiles, sin embargo en Orga 2 predicamos el uso de practicas *colorblind friendly* por lo que solo vamos a necesitar de unas pocas y fáciles de diferenciar.

3.1. Clonacion

También conocido como: Bajando a mi pc un repositorio remoto creado en <https://git.exactas.uba.ar>

```
> git clone https://git.exactas.uba.ar/usuario-que-creo-el-repo/nombre-del-repositorio
```

Esto crea una carpeta llamada *nombre-del-repositorio* en el directorio donde esten parados al momento de ejecutar *git clone*

Nota: La persona que crea el repo tiene que darle acceso unicamente a sus compañeros de TP

3.2. Actualizacion de la version local

Tambien conocido como: Actualizar lo que tenemos en nuestra PC con lo que hay en el repositorio remoto

```
> git pull
```

3.3. Actualizacion de la version remota

Tambien conocido como: Actualizar la versión del repositorio remoto con lo que estuvimos haciendo en nuestra PC

Nota: Siempre es recomendable actualizar nuestra version local **antes** de actualizar la version en el repositorio. O sea, hagan un *git pull* antes de hacer lo que sigue a continuacion.

```
> git commit -a m"Un mensaje explicando que cosa estan comiteando: David is the sexiest man alive"
> git push
```

El parámetro *-a* es para indicar que se marquen todos los archivos modificados o borrados. Los archivos nuevos (los que crearon en su carpeta local y que no existen en el repositorio remoto) no van a ser afectador por el commit.

3.4. Agregacion de archivos locales

```
> git add ruta-al-archivo-desde-donde-estoy-ejecutando
```

Existen otros gustos de *git add* que podrían serles utiles. El siguiente ejemplo agrega todos los cpp que están en la carpeta *algun-directorio* el cual a su vez esta en el directorio donde estamos ejecutando

```
> git add algun-directorio/*.cpp
```

4. Conflictos

Generalmente los conflictos ocurren cuando actualizan la versión local del repositorio (*git pull*) y alguien mas modifico cosas que estaban modificando. Estas cosas se llaman *conflictos* (por si quieren googlearlo)

```
> git mergetool
```

Esto va a abrir alguna aplicación que tengamos instalada para solucionar conflictos. Pueden googlear e instalarse la que mas les guste.

Solo por nombrar algunas: meld, opendiff, kdiff3, tkdiff, xxdiff, tortoisemerge, gvimdiff, diffuse, ecmerge, p4merge, araxis, vimdiff, emerge.

Nota: Inmediatamente despues de arreglar los conflictos en nuestra versión local, es necesario comitearlos y pushearlos al repositorio remoto.

5. Reset

Alguna vez puede pasar que rompemos tanto, pero tanto, nuestra versión local que simplemente queramos eliminar su existencia y volver a bajar una versión limpia del repositorio. Hoy se resetea fuerte.

```
> git reset --hard
> git pull
```

Aun peor, despues de una noche de descontrol, encontramos archivos nuevos que no sabemos de donde vienen o para que estan. Para borrar toda esa barbarie:

```
> git clean -df
```

Esto borra tantos archivos *-f* como directorios *-d* que no están en el repositorio remoto ni tampoco marcados para ser subidos. Usar con responsabilidad.



Figura 1: Resolviendo conflictos