Práctica Git - 2ª Parte

Crear tags (etiquetas)

Partimos de la práctica anterior con 4 archivos en nuestra carpeta de trabajo: ***alimentos.txt***, ***droguería.txt***, ***notas.txt*** y ***.gitignore***. Hacemos el commit si no lo habíamos hecho para que guarde todos los cambios. Ponemos de mensaje, por ejemplo: ***"añadido archivo droguería txt y manzanas"***

Vamos a crear una etiqueta. Esto se hace normalmente cuando tenemos ya una versión más o menos sólida. Y justo después de hacer el commit que queremos etiquetar:

```
> git tag v1.0 -m "Primera version"
> git show v1.0
```

El último comando lo que hace es mostrarnos la etiqueta junto con el commit etiquetado. Y también nos muestra las diferencias con los dos commits anteriores. Nos guarda además como usuario etiquetador y nuestro email y más información.

También podemos etiquetar cualquier commit anterior simplemente poniendo su identificador. Provemos con el identificador del commit anterior (añadidos huevos entre dos líneas). Esta vez no añadimos mensaje alguno:

```
> git tag v0.0 ef36
> git show v0.0
commit ef365ce7982f89a12aa0b3e1c15179956255af8d
Author: Nacho Rodriguez <irodriguezn@gmail.com>
Date:   Sun Nov 6 22:43:36 2016 +0100

    añadido huevos entre dos lineas

diff --git a/alimentos.txt b/alimentos.txt
index 0515622..74cbd13 100644
--- a/alimentos.txt
+++ b/alimentos.txt
@@ -1,2 +1,3 @@
 agua
+huevos
 peras
```

Si nos hemos equivocado o nos arrepentimos, por lo que sea, y queremos borrar alguna etiqueta también podemos. Con ***git tag*** (a secas) vemos las etiquetas que tenemos. Y con la opción ***-d*** borramos la que queremos:

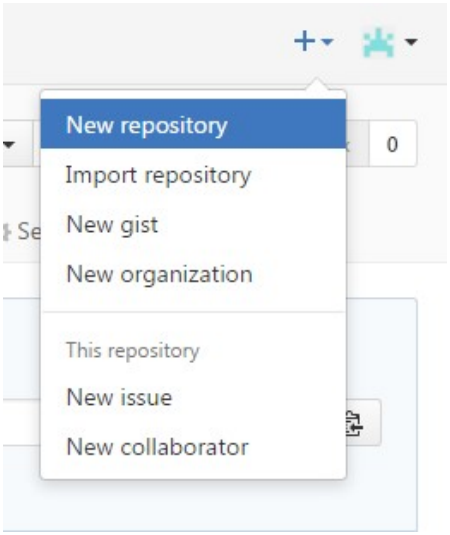
```
> git tag
v0.0
v1.0
> git tag -d v0.0
Deleted tag 'v0.0' (was ef365ce)
```

Trabajar en remoto

Una de los aspectos más interesantes de Git es el poder compartir el código con más gente y así poder colaborar varias personas en el mismo proyecto. Normalmente utilizaremos un repositorio remoto alojado en un hosting de control de versiones como Bitbucket o Github.

Como ya tenemos cuenta creada en ambas páginas vamos a subir el repositorio con el que hemos estado trabajando a ambos sitios. En caso de no tener cuenta, registrarse es muy fácil y gratuito.

Vamos a comenzar con github: ***https://github.com***



En primer lugar creamos un repositorio y lo llamamos lista (si ya está creado por alguna clase anterior entráis al repositorio y en ***settings*** se puede borrar para empezar de cero de nuevo).

Para ello vamos a la parte de arriba a la derecha de la pantalla, al signo "+" y lo creamos.

Y copiamos la url del repositorio. En mi caso:

https://github.com/irodriguezn/lista

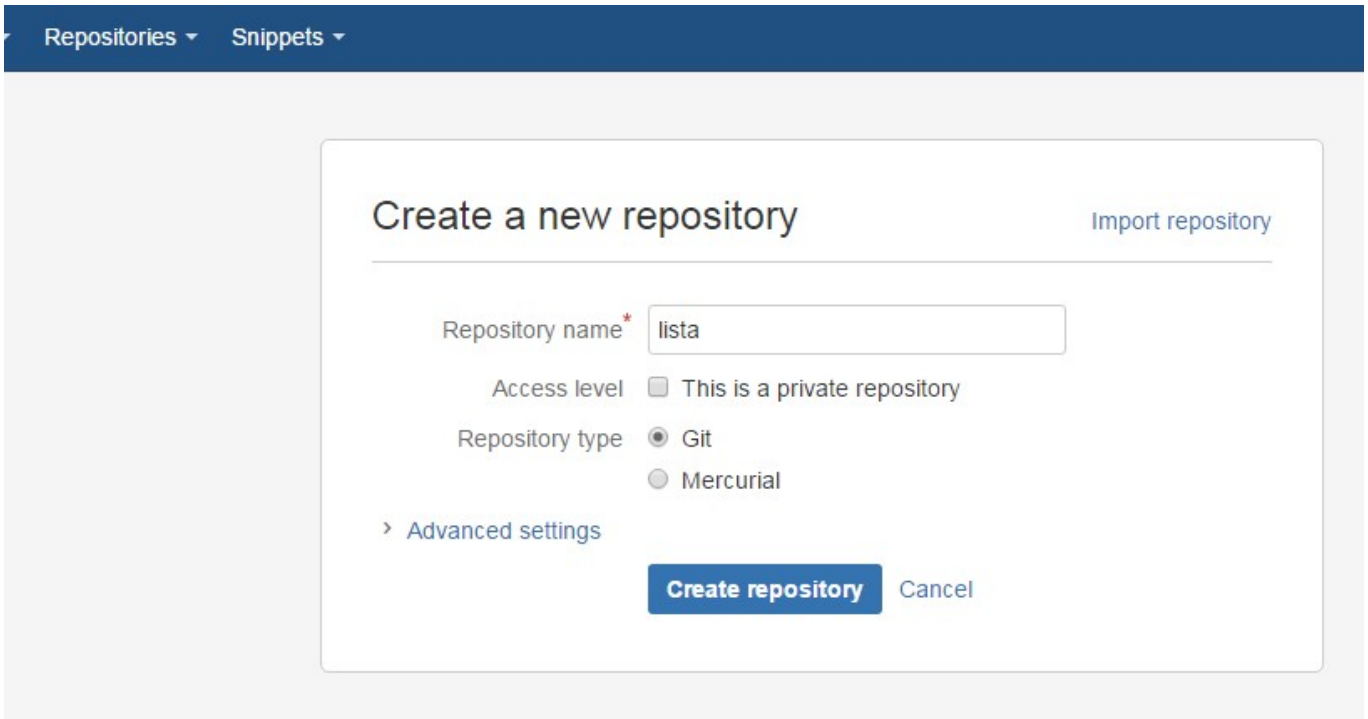
Vincular repositorio local con remotos

Ahora volvemos al **Git CMD** y ponemos lo siguiente:

```
> git remote add github https://github.com/irodriguezn/lista
```

Por supuesto cada uno pondrá la url de su repositorio.

Ahora hacemos lo mismo en bitbucket: ***https://bitbucket.org***



Y hacemos lo mismo que antes. Copiamos la url y ponemos:

```
> git remote add bitbucket https://bitbucket.org/irodriguezn/lista
```

Con estos comandos hemos vinculado nuestro repositorio local con dos repositorios remotos, pero aún no hemos subido nada. Por supuesto, estamos hablando siempre

de un proyecto concreto en una carpeta concreta. Podemos tener otros proyectos en otras carpetas en el disco duro vinculados con los mismos nombres que hemos usado aquí (bitbucket y github) y no tienen nada que ver unos con otros. Para ver los repositorios remotos vinculados a este proyecto hacemos:

```
> git remote
bitbucket
github
```

Ahora vamos a subir el repositorio local a ambos sitios. Para ello escribimos:

```
> git push bitbucket master
Username for 'https://bitbucket.org': irodriguezn
Password for 'https://irodriguezn@bitbucket.org':
> git push github master
Username for 'https://github.com': irodriguezn
Password for 'https://irodriguezn@github.com':
```

Si ahora vamos al navegador y observamos en los repositorios remotos veremos que, efectivamente ha subido todo correctamente.

A partir de ahora ya podemos volver a hacer un ***push*** siempre que queramos y si estamos en otro ordenador o le damos permiso a otra persona para subir y bajar cosas, podemos hacer un ***pull*** para obtener la última versión del repositorio.

Trabajar desde otro equipo: Clonar un proyecto

Vamos a simular que estamos en otro ordenador, nosotros o cualquier otra persona y vamos a bajarnos la última versión y a hacer cambios. Para ello en primer lugar vamos a clonar el proyecto en otra carpeta. Es decir, en lugar de irnos a otro ordenador, simplemente vamos a irnos a otra ruta del disco duro. Pero, en realidad sería igual desde otro equipo.

Nos vamos a la carpeta padre de nuestro proyecto original:

```
> cd ..
```

Y clonamos el proyecto escribiendo la url del repositorio que queremos clonar, por ejemplo de bitbucket:

```
> git clone https://bitbucket.org/irodriguezn/lista/src lista_clonada
```

Si hacemos un dir, veremos que ha creado la carpeta ***lista_clonada***

```
> dir
 El volumen de la unidad C no tiene etiqueta.
 El número de serie del volumen es: E8CB-E67F

 Directorio de C:\practica

07/11/2016  23:10    <DIR>          .
07/11/2016  23:10    <DIR>          ..
07/11/2016  23:09    <DIR>          lista
07/11/2016  23:10    <DIR>          lista_clonada
               0 archivos              0 bytes
               4 dirs   6.251.106.304 bytes libres
```

Nos metemos en la carpeta lista clonada (cd lista_clonada) y observamos que tenemos la carpeta oculta ***.git*** y los dos archivos subidos del repositorio. Por

supuesto no aparece ni el ***.gitignore*** ni el ***notas.txt*** porque no están añadidos al repositorio ni local ni remoto.

Si hacemos un ***git log --oneline*** veremos que tenemos los commits que teníamos en el repositorio original:

```
> git lo
6325f1f añadido archivo drogueria.txt y manzanas
ef365ce añadido huevos entre dos lineas
0438c6c Segundo commit
4b4f8e9 commit inicial
```

Si hacemos un ***git status*** vemos que no hay cambios, que está todo limpio:

```
> git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean
```

Si hacemos un ***git tag*** veremos que no hay nada, porque las etiquetas no se suben, por defecto, a un repositorio remoto haciendo ***push***. Ya veremos más adelante cómo subirlas si queremos.

Ahora hacemos un ***git remote***:

```
> git remote
origin
```

origin es el nombre que se pone por defecto al vínculo creado cuando clonamos. Si no nos gusta podemos renombrarlo haciendo:

```
> git remote rename origin bitbucket
```

Y si volvemos a hacer un ***git remote*** veremos que ha cambiado.

Pero el vínculo sólo lo tenemos con bitbucket. Si queremos vincular también con github tenemos que hacer igual que antes:

```
> git remote add github https://github.com/irodriguezn/lista
```

Ahora abrimos el archivo ***drogueria.txt*** y añadimos dos líneas nuevas: ***desodorante*** y ***escoba***, guardamos y cerramos.

Si hacemos ***un git status***, veremos que está modificado, así que vamos a hacer un ***commit*** con la opción -a directamente:

```
> git commit -a -m "añadidos desde otro ordenador desodorante y escoba"
[master e3b5c8c] añadidos desde otro ordenador desodorante y escoba
 1 file changed, 2 insertions(+)
```

Y ahora vamos a hacer un ***push*** a los repositorios remotos:

```
> git push bitbucket master
> git push github master
```

Esta vez sí que nos pide el usuario y la contraseña.

Si vamos a los sitios veremos que se han actualizado correctamente. Habrá que actualizar la página y puede que volver a pulsar source.

Ahora vamos a volvernos al proyecto original. A la carpeta lista, haciendo ***cd .. y cd lista***.

Si vemos el contenido de nuestro archivo por supuesto será el que había. Si ahora queremos obtener los cambios realizados por otros programadores o por nosotros mismos desde otro sitio tenemos que hacer un pull desde cualquier repositorio, por ejemplo:

```
> git pull bitbucket master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From https://bitbucket.org/irodriguezn/lista
 * branch            master     -> FETCH_HEAD
   6325f1f..e3b5c8c  master     -> bitbucket/master
Updating 6325f1f..e3b5c8c
Fast-forward
 drogueria.txt | 2 ++
 1 file changed, 2 insertions(+)
```

Si ahora abrimos el archivo veremos que lo tenemos actualizado.

Hacer un push de etiquetas

Como hemos dicho antes, las etiquetas por defecto no se suben. Si queremos hacerlo tenemos que hacer:

> git push github --tags

Si quisieramos subirlas a bitbucket haríamos lo mismo, claro.

Ahora para poder importarlas desde el otro proyecto nos volveríamos a él y haríamos:

> git pull github --tags

Eliminar elementos de la configuración global

Acabamos como empezamos, por la configuración global. Imaginemos que nos hemos equivocado añadiendo el email y hemos puesto:

```
> git config --global user.mail "pepe"
```

Para poder borrar esa entrada hacemos:

```
> git config --global --unset user.mail
```

Otra manera, nada recomendable sería abrir el archivo de configuración directamente mediante:

```
> git config --global --edit
```

O simplemente haciendo doble clic en él desde el explorador:

C:\Usuarios\Usuario\.gitconfig