

# Práctica del curso de git, gitHub y Sourcetree

Información de contacto del profesor:

Alberto Casero

[acasero@agbotraining.com](mailto:acasero@agbotraining.com)

Skype: kas.appeal

Twitter: @KasAppeal

## Ejercicio 1

Se deberá crear un repositorio y realizar una serie de operaciones **desde la consola de comandos** sobre el mismo para posteriormente subir el repositorio a Github.

Se deberá entregar a través del formulario de prácticas indicando la URL del repositorio. En el repositorio, deberá existir un archivo readme.md con las respuestas a las siguientes preguntas:

- ¿Qué comando utilizaste en el paso 11? ¿Por qué?
- ¿Qué comando o comandos utilizaste en el paso 12? ¿Por qué?
- El merge del paso 13, ¿Causó algún conflicto? ¿Por qué?
- El merge del paso 19, ¿Causó algún conflicto? ¿Por qué?
- El merge del paso 21, ¿Causó algún conflicto? ¿Por qué?
- ¿Qué comando o comandos utilizaste en el paso 25?
- El merge del paso 26, ¿Podría ser fast forward? ¿Por qué?
- ¿Qué comando o comandos utilizaste en el paso 27?
- ¿Qué comando o comandos utilizaste en el paso 28?
- ¿Qué comando o comandos utilizaste en el paso 29?
- ¿Qué comando o comandos utilizaste en el paso 30?
- ¿Qué comando o comandos usaste en el paso 32?
- ¿Qué comando o comandos usaste en el punto 33?

Los pasos a ejecutar son los siguientes (los pasos en negrita indican que hay una pregunta asociada):

- 1) Crear un repositorio
- 2) Crear un archivo *git-nuestro.md* con el contenido:

```
Git nuestro  
  
Git nuestro que estas en los repos  
  
Comprimidos sean tus commits  
  
Venga a nosotros tu log  
  
En el local como en el remote  
  
Danos hoy nuestro pull de cada día  
  
Perdona nuestros conflictos  
  
Como también perdonamos los de otros geeks  
  
No nos dejes caer en detached HEAD  
  
y líbranos de SVN  
  
git commit --amend
```

- 3) Añadir *git-nuestro.md* al staging area
- 4) Mover lo que hay en el staging area al repositorio
- 5) Crear una rama llamada “styled”
- 6) Listar las ramas que hay en el repositorio
- 7) Moverse a la rama “styled”
- 8) Comprobar que se está en la rama correcta
- 9) Modificar en el archivo *git-nuestro.md*:

```
*Git* nuestro que estas en los repos  
  
Comprimidos sean tus *commits*  
  
Venga a nosotros tu *log*  
  
En el local como en el *remote*  
  
Danos hoy nuestro *pull* de cada día  
  
Perdona nuestros *conflictos*  
  
Como también perdonamos los de otros geeks
```

No nos dejes caer en *\*detached HEAD\**

y líbranos de *\*SVN\**

```
`git commit --amend`
```

10) Añadir los cambios al staging área y luego pasarlos al repositorio

**11) Deshacer el último commit (perdiendo los cambios realizados en el working copy)**

**12) Rehacer el último commit (el que acabamos de deshacer)**

**13) Hacer un merge con 'master' (htmlify absorbe a master)**

14) Volver a la rama master

15) Crear una nueva rama llamada "htmlify"

16) Cambiar a la rama htmlify

17) Modificar en el archivo *git-nuestro.md*:

```
<p><em>Git</em> nuestro que estas en los repos<br />
```

```
Comprimidos sean tus <em>commits</em><br />
```

```
Venga a nosotros tu <em>log</em><br />
```

```
En el local como en el <em>remote</em><br />
```

```
Danos hoy nuestro <em>pull</em> de cada día<br />
```

```
Perdona nuestros <em>conflictos</em><br />
```

```
Como también perdonamos los de otros geeks<br />
```

```
No nos dejes caer en <em>detached HEAD</em><br />
```

```
y líbranos de <em>SVN</em><br />
```

```
<code>git commit --amend</code></p>
```

18) Hacer un commit

**19) Hacer un merge de "htmlify" en "styled" (styled absorbe a htmlify)**

20) Si hay conflictos, deberemos resolverlos quedándonos con el contenido de la rama "styled".

**21) Desde "master", hacer un merge con "styled"**

22) Crear una rama "title" y cambiarse a esa rama

23) Añadir un título (a tu gusto) al archivo *git-nuestro.md* y hacer un commit.

24) Volver a la rama master

- 25) Dibujar el diagrama**
- 26) Hacer un merge “no fast-forward” de “title” en “master” (master absorbe a title)**
- 27) Deshacer el merge (sin perder los cambios del working copy)**
- 28) Descartar los cambios**
- 29) Eliminar la rama “title”**
- 30) Rehacer el merge que hemos deshecho**
- 31) Volver a master y eliminar el resto de ramas
- 32) Volver al commit inicial cuando se creó el poema**
- 33) Volver al estado final, cuando pusimos título al poema**
- 34) Crear los siguientes tags:
  - inicial: en el primer commit
  - styled: modificación del paso 10
  - htmlify: modificación del paso 17-18
  - title: modificación del paso 30
- 35) Ir al tag htmlify

## Ejercicio 2: Github, Forks y Pull Requests

La práctica consiste en hacer un fork del repositorio <https://github.com/kasappeal/nerdquotes.git> y, en el archivo README.md añadir, en formato markdown, una cita (a ser posible con connotaciones frikis) seguido del nombre de su autor (en cursiva).

Tras añadir la línea en el fork, solicitar un pull request al repositorio original (<https://github.com/kasappeal/nerdquotes.git>).

Si el pull request causa conflictos, se rechazará y de manera que el alumno deberá actualizar su repositorio con los últimos contenidos del repositorio forkeado, solucionar los conflictos y volver a realizar otro pull request.

Cada cita deberá llegar el formato siguiente:

-----

> Cita

*\*Nombre del autor\**

Es decir:

- Deberá empezar creando una línea horizontal en markdown: -----
- Una línea en blanco
- La cita, con un sangrado en markdown (empezando con el caracter: >)
- Una línea en blanco
- El nombre del autor en cursiva (poniendo el nombre del autor entre *\*Nombre del autor\**)

Ejemplo:

-----

> Programmer: An organism that turns caffeine and pizza into source code.

*\*Blogger de Niro\**

Notas:

- El nombre del autor puede ser ficticio.
- La cita no tiene por qué estar en inglés.
- Puedes inventar tu propia cita o utilizar una de las siguientes: <http://www.brainyquote.com/quotes/keywords/nerd.html>