**Ejercicio 1**

Se deberá crear un repositorio y realizar una serie de operaciones desde la terminal y responder a las preguntas del cuestionario, en el que se pregunta por comandos utilizados en ciertos pasos.

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

1. Crear un repositorio:

**mkdir git-practica**

Para decirle que en esa carpeta se controle los cambios de los archivos y se inicialice el repositorio:

**git init**

1. Crear un archivo git-nuestro.md con el contenido:

**code .**

**creo el archivo git-nuestro.md en code y escribo lo siguiente:**

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

**Guardo y vuelvo a git bash**

1. Añadir git-nuestro.md al staging área

**git add git-nuestro.md**

1. Mover lo que hay en el staging area al repositorio

**git commit -m "commit inicial"**

**[master (root-commit) 98716b1] commit inicial**

1. **file changed, 10 insertions(+)**

**create mode 100644 git-nuestro.md**

1. Crear una rama llamada “styled”

**Creo la nueva rama sin modificar el contenido de las áreas**

**git branch styled**

1. Listar las ramas que hay en el repositorio

**git branch**

**\* master**

**styled**

1. Moverse a la rama “styled”

**git checkout styled**

**Switched to branch 'styled'**

1. Comprobar que se está en la rama correcta

**git branch**

**master**

**\* styled**

1. Modificar en el archivo git-nuestro.md:

code git-nuestro.md

\*Git\* nuestro que estás 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`

1. Añadir los cambios al staging area y luego pasarlos al repositorio

* **Para saber el estado actual:**

**git status**

**On branch styled**

**Changes not staged for commit:**

**(use "git add <file>..." to update what will be committed)**

**(use "git restore <file>..." to discard changes in working directory)**

**modified: git-nuestro.md**

**no changes added to commit (use "git add" and/or "git commit -a")**

* **Añado el archivo al staging área git-nuestro.md**

**git add git-nuestro.md**

* **Muevo lo que hay en el staging area al repositorio**

**git commit -m "Primer commit en branch styled"**

**[styled 2c3346b] Primer commit en branch styled**

**1 file changed, 10 insertions(+), 10 deletions(-)**

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

**git reset --hard HEAD~1**

**HEAD is now at 98716b1 commit inicial**

1. **Rehacer el último commit (el que acabamos de deshacer)**

* **Primero miramos lo que ha pasado en nuestro repositorio**

git reflog

**98716b1 (HEAD -> styled, master) HEAD@{0}: reset: moving to HEAD~1**

**2c3346b HEAD@{1}: commit: Primer commit en branch styled**

**98716b1 (HEAD -> styled, master) HEAD@{2}: checkout: moving from master to styled**

**98716b1 (HEAD -> styled, master) HEAD@{3}: commit (initial): commit inicial**

* **Vamos a la dirección hash 2c3346b y recuperamos lo borrado**

**$ git reset 2c3346b**

**Unstaged changes after reset:**

**M git-nuestro.md**

* **Hago git status para ver lo que sucede**

**$ git status**

**On branch styled**

**Changes not staged for commit:**

**(use "git add <file>..." to update what will be committed)**

**(use "git restore <file>..." to discard changes in working directory)**

**modified: git-nuestro.md**

**no changes added to commit (use "git add" and/or "git commit -a")**

* **Añado el archivo al staging área git-nuestro.md**

**git add git-nuestro.md**

* **Muevo lo que hay en el staging area al repositorio**

**git commit -m "Primer commit en branch styled"**

**$ git commit -m "Commit en branch styled recuperando git-nuestro.md"**

**[styled 6629906] Commit en branch styled recuperando git-nuestro.md**

**1 file changed, 10 insertions(+), 10 deletions(-)**

1. **Hacer un merge con ‘master’ (styled absorbe a master)**

**git merge master**

**Already up to date.**

1. Volver a la rama master

**git checkout master**

**Switched to branch 'master'**

1. Crear una nueva rama llamada “htmlify”

**Creo la nueva rama sin modificar el contenido de las áreas**

**git branch htmlify**

1. Cambiar a la rama htmlify

**git checkout htmlify**

**Switched to branch 'htmlify'**

1. Modificar en el archivo git-nuestro.md:

**code 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>

1. Hacer un commit

* **Añado el archivo al staging área git-nuestro.md**

**git add git-nuestro.md**

* **Muevo lo que hay en el staging area al repositorio**

**git commit -m "Primer commit en branch htmlify"**

**[htmlify 2b26bdb] Primer commit en branch htmlify**

**1 file changed, 8 insertions(+), 10 deletions(-)**

1. **Hacer un merge de “htmlify” en “styled” (styled absorbe a htmlify)**

* **Voy a la rama styled**

**git checkout styled**

**Switched to branch 'styled'**

* **Hago git status para ver lo que sucede**

**git status**

**On branch styled**

**nothing to commit, working tree clean**

* **Hago merge para que styled absorba a htmlify**

**git merge htmlify**

**Merge made by the 'ort' strategy.**

**git-nuestro.md | 18 ++++++++----------**

**file changed, 8 insertions(+), 10 deletions(-)**

1. Si hay conflictos, deberemos resolverlos quedándonos con el contenido de la rama “styled”.

* **Para solucionar, debo editar el archivo git-nuestro.md y dejar el contenido que había en styled**

\*Git\* nuestro que estás 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`

* **Añado el archivo al staging área git-nuestro.md**

**git add git-nuestro.md**

* **Muevo lo que hay en el staging área al repositorio**

**git commit -m "Nuevo commit en branch styled"**

**[styled e13867d] Nuevo commit en branch styled**

* + - 1. **file changed, 10 insertions(+), 8 deletions(-)**

1. **Desde “master”, hacer un merge con “styled”**

* **Voy a la rama master**

**git checkout master**

**Switched to branch 'master'**

* **Hago merge para que master absorba a styled**

**git merge styled**

**Updating 98716b1..e13867d**

**Fast-forward**

**git-nuestro.md | 20 ++++++++++----------**

**1 file changed, 10 insertions(+), 10 deletions(-)**

1. Crear una rama “title” y cambiarse a esa rama

**git checkout -b title**

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

**code git-nuestro.md**

* **Añado el archivo al staging área git-nuestro.md**

**git add git-nuestro.md**

* **Muevo lo que hay en el staging área al repositorio**

**git commit -m "Nuevo commit en branch title"**

**[title fb3a465] Nuevo commit en branch title**

**1 file changed, 1 insertion(+)**

1. **Volver a la rama master**

**git checkout master**

**Switched to branch 'master'**

1. **Dibujar el diagrama**

**git log --graph**

1. **Hacer un merge “no fast-forward” de “title” en “master” (master absorbe a title)**

* **El merge no fast-forward hace un commit**

**git merge --no-ff title**

**Merge made by the 'ort' strategy.**

**git-nuestro.md | 1 +**

**1 file changed, 1 insertion(+)**

1. **Deshacer el merge (sin perder los cambios del working copy)**

**git reset HEAD~1**

**Unstaged changes after reset:**

**M git-nuestro.md**

1. **Descartar los cambios**

**git restore git-nuestro.md**

**HEAD is now at 6903bb0 Comit branch master**

1. **Eliminar la rama “title”**

**git branch -D title**

**Deleted branch title (was c7714cc).**

1. **Rehacer el merge que hemos deshecho**

* **Miro qué está pasando en el sistema**

**git reflog**

* **Reseteo con –hard dirigiéndome al id del commit**

**git reset --hard 5f7ceb9**

**HEAD is now at 5f7ceb9 Nuevo commit en branch master**

1. **Volver a master y eliminar el resto de ramas**

**$ git branch -D htmlify styled**

**Deleted branch htmlify (was 81be524).**

**Deleted branch styled (was 81be524).**

1. **Volver al commit inicial cuando se creó el poema**

* **Miro qué está pasando en el sistema**

**git reflog**

* **Miro qué hash tiene y voy**

**git reset --hard 2c3346b**

**Note: switching to '2c3346b'.**

**You are in 'detached HEAD' state. You can look around, make experimental**

**changes and commit them, and you can discard any commits you make in this**

**state without impacting any branches by switching back to a branch.**

**If you want to create a new branch to retain commits you create, you may**

**do so (now or later) by using -c with the switch command. Example:**

**git switch -c <new-branch-name>**

**Or undo this operation with:**

**git switch -**

**Turn off this advice by setting config variable advice.detachedHead to false**

**HEAD is now at 2c3346b Primer commit en branch styled**

1. **Volver al estado final, cuando pusimos título al poema**

**git reset --hard 6629906**

**Previous HEAD position was 2c3346b Primer commit en branch styled**

**HEAD is now at 6629906 Commit en branch styled recuperando git-nuestro.md**

1. **Crear los siguientes tags:**

**inicial: en el primer commit styled:**

* **Miro qué hash tiene el primer commit styled**

**git checkout 98716b1**

**git tag -a Inicial**

**Styled: modificación del paso 10:**

* **Miro qué hash tiene el paso 10 styled**

**git checkout 2c3346b**

**git tag -a htmlify**

**Htmlify modificación del paso 17-18:**

* **Miro qué hash tiene el paso 17-18 htmlify**

**git checkout fb3a465**

**git tag -a title**

**Title modificación del paso 30:**

* **Miro qué hash tiene el paso 30 Title**

**git checkout 5f7ceb9**

**git tag -a rehacer merge**

1. **Ir al tag htmlify**

* **Listo las etiquetas**

**git tag**

* **Voy al tag htmlify**

**git checkout htmlify**

* **Verifico estar en el tag htmlify**

**$ git status**

**HEAD detached at htmlify**

**nothing to commit, working tree clean**

# Ejercicio 2: Github, Forks y Pull Requests (opcional)

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).

Cada alumno deberá introducir su cita en una posición aleatoria entre dos frases ya existentes (por favor no la pongáis todos al final o al principio, que luego me toca resolver conflictos!).

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