Laura Moreno Fernández
Febrero 2024

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

1. Instalo git desde https://git-scm.com/download/windows
2. Verifico la instalación con git --version
3. Conecto con mi cuenta de GitHub con:
   git config --global user.name Lauralunaress
   git config --global user.email info@lauramorenofernandez.com
4. Genero una clave SHH con:
   ssh-keygen -t rsa -b 4096 -C info@lauramorenofernandez.com
5. La guardo en C:\Users\laura\.ssh\id_rsa y establezco la contraseña
6. Configuro la clave en GitHub desde el apartado SSH keys
7. Verifico que la conexión funciona con ssh -T git@github.com
8. Obtengo la respuesta “Hi Lauralunaress! You've successfully authenticated, but GitHub does not provide shell access.”
9. Creo la carpeta donde quiero el repositorio e inicializo:
   Laura Moreno@LauraSEOmalaga MINGW64 ~/Git Laura
   $ git init
   Initialized empty Git repository in C:/Users/laura/Git Laura/.git/

2) Crear un archivo git-nuestro.mdcon 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

Creo y guardo en formato markdown con VSC

3) Añadir git-nuestro.md al staging area

1. git add git-nuestro.md
2. Compruebo que esté bien agregado con git status

4) Mover lo que hay en el staging area al repositorio

1. Laura Moreno@LauraSEOmalaga MINGW64 ~/Git Laura (master)
   $ git commit -m “Subida primera version”
   [master (root-commit) 2da9551] Subida primera version
   1 file changed, 12 insertions(+)
   create mode 100644 git-nuestro.md
2. Compruebo el commit en el log con git log

5) Crear una rama llamada “styled”

```
git branch styled
```

6) Listar las ramas que hay en el repositorio

1. git branch (para locales)
2. git branch -a (para locales y remotas)

7) Moverse a la rama “styled”

```
git checkout styled
```

8) Comprobar que se está en la rama correcta

1. Se puede utilizar el comando git branch. Este comando lista todas las ramas locales y resalta con un asterisco (*) la rama en la que te encuentras actualmente. Sin embargo, esto no es muy conveniente si tienes muchas ramas
2. Mejor comprobar con git status, que devuelve On branch styled y nothing to commit, working tree

clean

9) Modificar en el archivo 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`

Abro el archivo en VCS con code git-nuestro.md, modifico y guardo.

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

1. git add git-nuestro.md
2. Compruebo que esté bien agregado con git status
3. Laura Moreno@LauraSEOmalaga MINGW64 ~/Git Laura (styled)
   $ git commit -m "Subida segunda version"
   [styled c0558a0] Subida segunda version
   1 file changed, 9 insertions(+), 11 deletions(-)

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

1. git reset --hard HEAD~1
2. Abro y compruebo que el archivo sea el de la versión anterior

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

1. Visualizo el historial de commits con git reflog
2. Localizo el hash del commit
3. git reset --hard c0558a0
4. Commit con los cambios recuperados:
   git commit -m "Recuperación Segunda version"

13) Hacer un merge con 'master' (styled absorbe a master)

1. Me aseguro de en qué rama estoy con git branch
2. Cambio a master con git checkout master
3. Mergeo con git merge styled
4. Compruebo con git branch

14) Volver a la rama master

```
git checkout master
```

15) Crear una nueva rama llamada "htmlify"

```
git branch htmlify
```

16) Cambiar a la rama htmlify

```
git checkout 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 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

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

Abro el archivo en VCS con code git-nuestro.md, lo modifico y guardo

18) Hacer un commit

1. git add git-nuestro.md
2. Compruebo que esté bien agregado con git status
   Laura Moreno@LauraSEOmalaga MINGW64 ~/Git Laura (htmlify)
   $ git commit -m "Subida tercera version"
   [htmlify f14354d] Subida tercera version
   1 file changed, 10 insertions(+), 10 deletions(-)

19) Hacer un merge de “htmlify” en “styled” (styled absorbe a htmlify)

1. Ir a styled con git checkout styled
2. Hacer el merge con git merge htmlify

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

No hubo

21) Desde “master”, hacer un merge con “styled”

1. Cambio a master con git checkout master
2. Fusiono styled en master con git merge styled
3. Verifico que todo está bien:
   $ git status
   On branch master
   nothing to commit, working tree clean

   $ git log
   commit f14354d1f6ce7055367f7cae0ffd11f1791d386b (HEAD -> master, styled, htmlify)
   Author: Lauralunaress <info@lauramorenofernandez.com>
   Date: Sat Feb 10 14:22:51 2024 +0100

      Subida tercera version

   commit c0558a0b0927ab15f9c607c91461b4c669b7c0af
   Author: Lauralunaress <info@lauramorenofernandez.com>
   Date: Sat Feb 10 13:53:27 2024 +0100

      Subida segunda version

   commit 2da95512dc95661c77ad04d9789113e5b462f19b
   Author: Lauralunaress <info@lauramorenofernandez.com>
   Date: Sat Feb 10 13:32:41 2024 +0100

      Subida primera version

22) Crear una rama “title” y cambiarse a esa rama

23) Añadir un título (a tu gusto) al archivo git-nuestro.mdy 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 (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