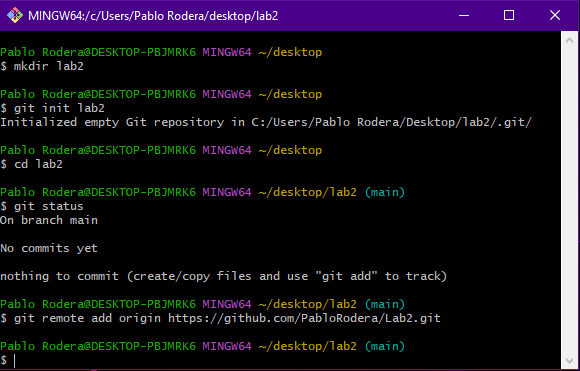
# **Sprint 1 – Lab 2 Gestión de repositorios Git**

[LINK AL REPOSITORIO DE LA PRÁCTICA](https://github.com/PabloRodera/Lab2)

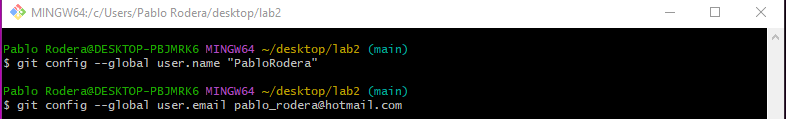
[LINK A MI REPOSITORIO DEL BOOTCAMP DONDE REALIZO TODO LO RELACIONADO CON ESTE](https://github.com/PabloRodera/Bootcamp)

(Por si queréis echar un vistazo, ya que lo uso casi diariamente para realizar el bootcamp)

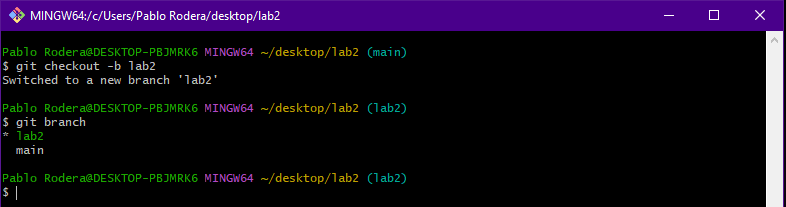
Después de crearnos la cuenta en GitHub y crear un repositorio específico para esta práctica, creamos una carpeta local en la cual iniciamos nuestro repositorio local con el comando **“git init (nombre de la carpeta)”**, después con **"git** **status”** mostramos el estado actual de la rama y, por último, para enlazar nuestro repositorio local con el remoto, usamos el comando **“git remote add origin (url del repositorio remoto)”.**



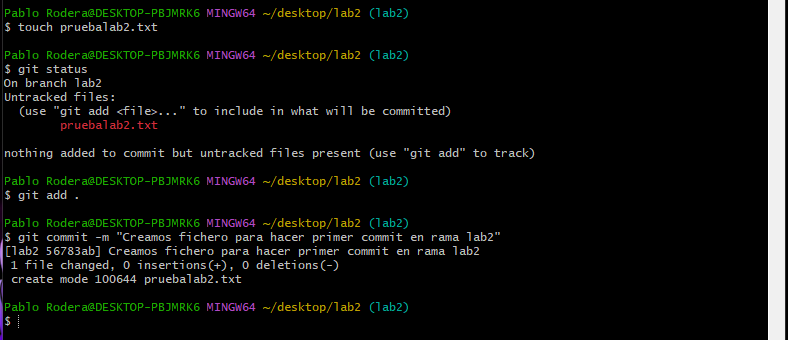
Configuramos nuestro mail de nuestra cuenta de GitHub y nuestro nombre de usuario a nivel global en nuestro sistema para que GitHub nos reconozca y cualquier trabajo que hagamos con Git en nuestro sistema sea bajo nuestro usuario, dando lugar al correcto funcionamiento de Git.



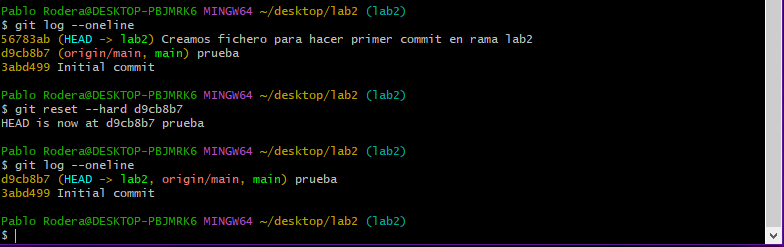
Creamos la nueva rama como se nos pide en el laboratorio:



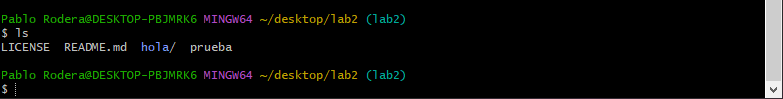
Creamos un fichero .txt de prueba en esta nueva rama y realizamos un commit:



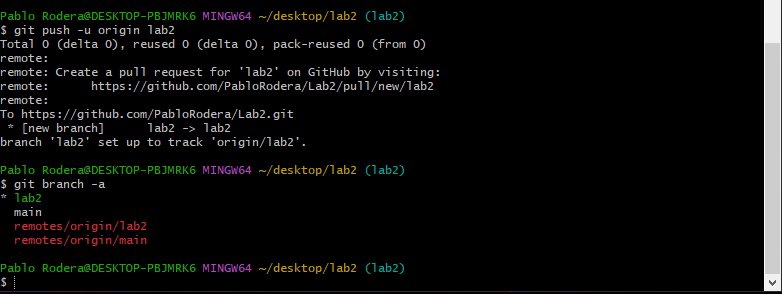
El laboratorio nos pide que eliminemos el commit realizado anteriormente, lo eliminamos y volvemos al estado anterior de la siguiente forma:



Lo que realizamos arriba básicamente es mostrar el historial de commits (he realizado uno anteriormente para comprobar que todo funcionaba correctamente), y con el hash del commit al que queremos volver, usamos el comando **“git reset --hard (hash del commit AL QUE QUEREMOS VOLVER, NO DEL QUE QUEREMOS ELIMINAR)”** (está erróneo en los apuntes), con el parámetro **--hard** también eliminaremos los cambios a nivel local, si realizamos un ls, veremos que el archivo **pruebalab2.txt** no está:

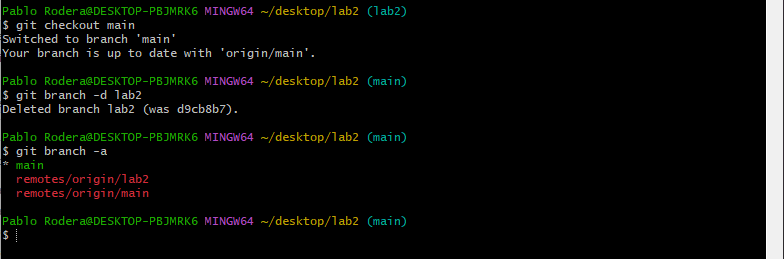


El laboratorio nos pide que eliminemos la rama que hemos creado tanto a nivel local como remota, para ello, ya que la rama actualmente solo está en local, vamos a empujarla a remoto para poder realizar correctamente el ejercicio, esto lo hacemos con el siguiente comando:

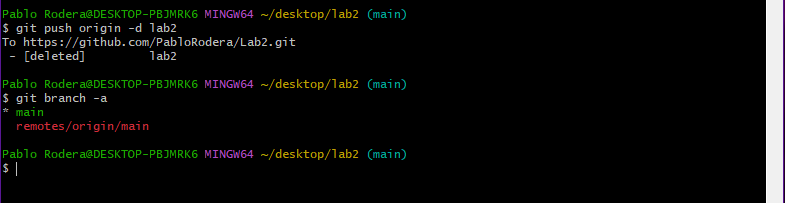


Como vemos, con el comando “**git branch –a**" nos muestra tanto las ramas locales como las remotas, indicándonos que se ha empujado al repositorio remoto correctamente.

Ahora procedemos a eliminar esta rama tanto local como remota, cambiaremos a la rama “main”, ya que no podemos eliminar la rama “lab2” si estamos actualmente en ella, primero la eliminaremos de manera local, para ello, usamos el siguiente comando:



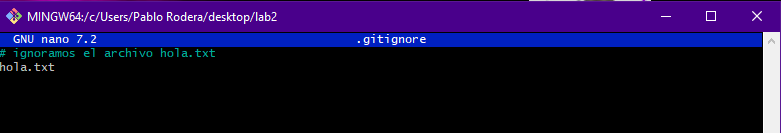
Como podemos observar, se ha eliminado del local, pero no del remoto, para eliminarla del remoto usaremos el siguiente comando:



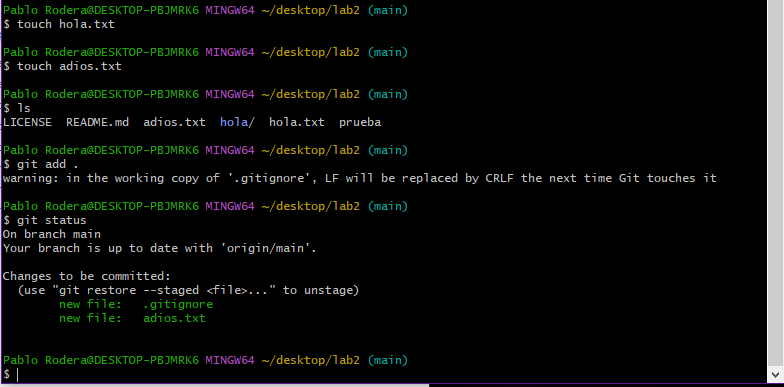
Ahora la rama ya está eliminada totalmente.

Lo siguiente que nos piden en el laboratorio es configurar nuestro Git para ignorar archivos, esto lo realizaremos mediante el archivo **.gitignore**, un archivo de texto que le dice a Git qué archivos o carpetas ignorar en un proyecto. Generalmente se coloca en el directorio raíz de un proyecto. También puedes crear un archivo global **.gitignore**, y cualquier entrada en ese archivo se ignorará en todos tus repositorios de Git.

Vamos a realizar una prueba sencilla en la que nuestro Git ignore el archivo **“hola.txt”**, para ello editamos nuestro archivo **.gitignore** y ponemos lo siguiente:



Ahora cuando hagamos un “**git add .**” se ignorará el archivo **hola.txt**, vamos a realizar una prueba creando este archivo y otro diferente:



Como podemos ver, en el **“git status”** no aparece como cambio pendiente el archivo **“hola.txt**” ya que está siendo ignorado correctamente.

Por último, nos registramos en [www.gitkraken.com](http://www.gitkraken.com) y lo asociamos al repositorio creado:

