

Informe Laboratorio 3. Control de versiones con Git y GitHub.

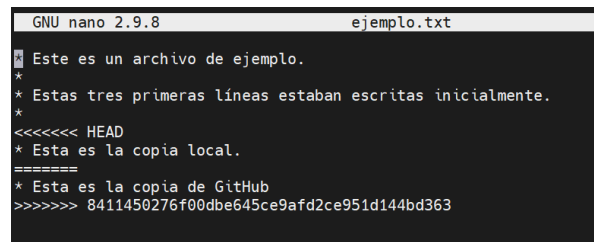
Este laboratorio representa un cambio temático significativo respecto a los anteriores, trasladando el foco del manejo del clúster al aprendizaje de herramientas de control de versiones. La inclusión de Git y GitHub en el temario me parece muy acertada, ya que son herramientas fundamentales en el desarrollo de software y la investigación colaborativa. He realizado el laboratorio detenidamente, revisando los enlaces disponibles y aprovechando esta oportunidad para familiarizarme a fondo con estas herramientas tan relevantes en el ámbito informático. El laboratorio permite descubrir una forma profesional de gestionar versiones de archivos, superando ampliamente métodos tradicionales como el intercambio de correos electrónicos. Un aspecto especialmente valioso es que, tras finalizarlo, se dispone de una cuenta de GitHub completamente configurada y lista para utilizarse en proyectos futuros.

La documentación proporcionada es muy completa, permitiendo seguir la mayoría de pasos satisfactoriamente. Además, el laboratorio combina trabajo desde el navegador y desde la terminal Linux, ofreciendo una visión integral de ambos entornos. La configuración de autenticación por SSH, aunque requiere tiempo inicial, resulta esencial para facilitar el trabajo posterior con múltiples operaciones evitando introducir contraseñas constantemente.

En la sección de modificación de ficheros *fastq*, decidí crear un *script* file-cut.sh automatizado que funciona para cualquier usuario (extrayendo automáticamente el número de alumno), en lugar de limitarlo únicamente a mi usuario. Esta decisión me permitió crear código más reutilizable, además de repasar conceptos de *bash* mediante la inclusión de comprobaciones de argumentos y validación de existencia de archivos.

La actividad extra de pull request resultó especialmente enriquecedora, ya que permitió interactuar con compañeros y experimentar en la práctica flujos de trabajo colaborativo similares a los empleados en entornos reales.

En cuanto a los aspectos a mejorar, considero que debería recalcarse más la importancia de conectarse al repositorio mediante SSH en lugar de HTTPS desde el inicio, especificando claramente los beneficios en términos de productividad al evitar autenticaciones repetidas. Por otra parte, la principal dificultad surgió al intentar visualizar los **símbolos de conflicto** (captura de la derecha), ya que el comando `git pull origin main` por sí solo no funcionaba. Fue necesario ejecutar primero `git config pull.rebase false` para especificar que se deseaba realizar un merge.



```
GNU nano 2.9.8          ejemplo.txt
* Este es un archivo de ejemplo.
*
* Estas tres primeras líneas estaban escritas inicialmente.
*
<<<<<< HEAD
* Esta es la copia local.
=====
* Esta es la copia de GitHub
>>>>>> 8411450276f00dbe645ce9afd2ce951d144bd363
```

Esta situación requirió considerable tiempo de depuración, por lo que sería recomendable añadir directrices más claras en este apartado del laboratorio. Por último, sería beneficioso incluir más explicación teórica sobre la diferencia entre Git (sistema de control de versiones) y GitHub (plataforma de alojamiento), así como conceptos de token, que pueden resultar confusos inicialmente.

El laboratorio cumple satisfactoriamente su objetivo de introducir Git y GitHub como herramientas profesionales de control de versiones. Las competencias adquiridas son directamente aplicables tanto en entornos académicos como laborales, especialmente en proyectos colaborativos de bioinformática donde el control de versiones es esencial para la reproducibilidad y el trabajo en equipo.

Componentes del grupo para la actividad extra en orden:

1. Antonio José Manuel Martínez - 2. Antonio Enrique Coll Meseguer - 3. María Sánchez Vivo - 4. Laura Llamas López - 5. Álvaro Máximo González López

Pull request realizado: <https://github.com/AntonioJoseManuelMartinez/lab-git/pull/4>