

Práctica Manejo Básico de GIT

PREREQUISITOS:

- Confirmar que tienes GIT instalado en la máquina (`git --version` te puede ser útil)
- Confirma que tienes una cuenta GITHUB, sino créate una <https://github.com/>

PRÁCTICA:

Vamos a seguir una serie de pasos para crear un repositorio

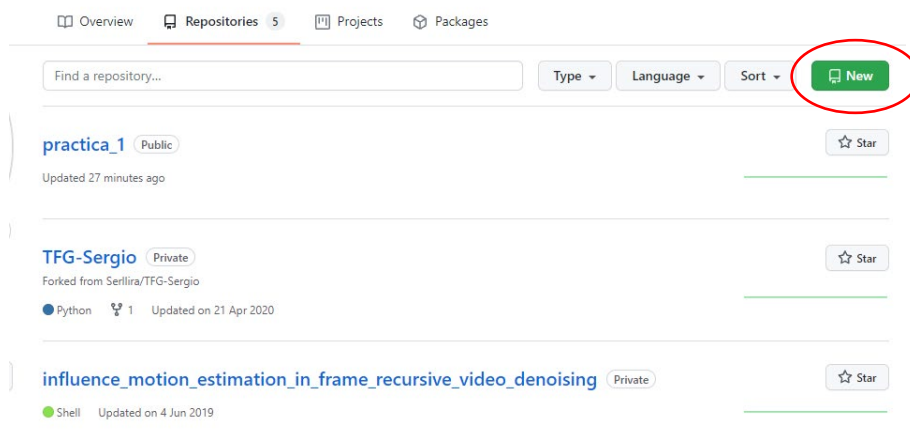
PARTE 1: Crear el repositorio de trabajo

Esta parte se podría realizar de dos maneras distintas. Una de ellas sería crear el repositorio en GITHUB y luego clonarlo (`git clone URL`) **o, como haremos en clase, trabajar desde un inicio en la máquina local para subirlo a un repositorio remoto.**

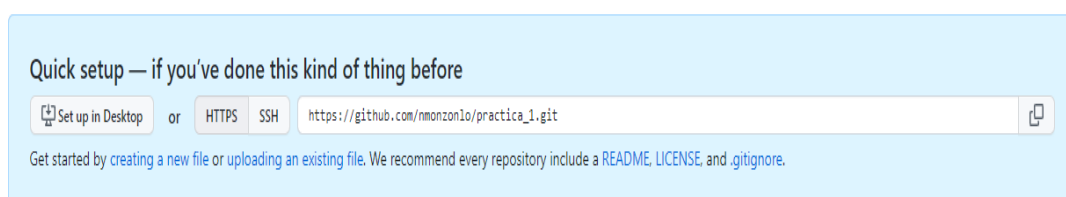
1. Creamos en local una carpeta cuyo nombre sea PRACTICA_GIT.
2. Inicializamos el repositorio (`"git init"`).
3. Incluiremos un fichero README cuyo texto sea "This is a README file".
4. Añadimos el repositorio dicho fichero, lo "commiteamos" con el mensaje "Initial Commit" y revisamos que todo está correcto mediante `"git status"`.

En este punto ya tenemos todo listo en local, pero debemos poder subirlo al repositorio remoto (nada de esto tiene sentido si nuestro trabajo queda siempre en local).

Iremos entonces a nuestra cuenta en GITHUB y crearemos el repositorio con el nombre PRACTICA_GIT.



Entre otras cosas, esto nos permite obtener una URL que podremos "linkar" con nuestro repositorio local. Por ejemplo, durante el desarrollo de este ejemplo



NOTA: En el momento de crear el repositorio, nos indica si queremos añadir un fichero gitignore. Digamos que sí a la versión JAVA.

Crearemos una rama principal y subiremos el repositorio a remoto.

1. `git branch -M main`
2. `git remote add origin https://github.com/nmonzonlo/practica_1.git`
3. `git push -u origin main`

Es probable que tengan algún problema utilizando el repositorio en clases, en este enlace obtendrán la manera de arreglarlo: <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>

A continuación, realizaremos los siguientes pasos

1. Comprobar el estado del repositorio.
2. Crear un fichero `indice.txt` con el siguiente contenido:
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
Capítulo 3: Repositorios remotos
3. Comprobar de nuevo el estado del repositorio.
4. Añadir el fichero a la zona de intercambio temporal.
5. Volver a comprobar una vez más el estado del repositorio.
6. Realizar un commit de los últimos cambios con el mensaje “*Añadido índice del libro.*” y ver el estado del repositorio.
7. Cambiar el fichero `indice.txt` para que contenga lo siguiente:
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
Capítulo 3: Gestión de ramas
Capítulo 4: Repositorios remotos
8. Mostrar los cambios con respecto a la última versión guardada en el repositorio.
9. Añadir el fichero a la zona de intercambio temporal.
10. Hacer un commit de los cambios con el mensaje “*Añadido capítulo 3 sobre gestión de ramas.*”.
11. Mostrar los cambios de la última versión del repositorio con respecto a la anterior.
12. Cambiar el mensaje del último commit por “*Añadido capítulo 3 sobre gestión de ramas al índice.*”
13. Volver a mostrar los últimos cambios del repositorio.

PARTE 2: Historial de cambios

1. Mostrar el historial de cambios del repositorio.
2. Crear la carpeta `capítulos` y crear dentro de ella el fichero `capitulo1.txt` con el siguiente texto.

Git es un sistema de control de versiones ideado por Linus Torvalds.

3. Añadir los cambios a la zona de intercambio temporal.
4. Hacer un commit de los cambios con el mensaje “*Añadido capítulo 1.*”

5. Volver a mostrar el historial de cambios del repositorio.
6. Crear el fichero `capitulo2.txt` en la carpeta `capítulos` con el siguiente texto.
El flujo de trabajo básico con Git consiste en: 1- Hacer cambios en el repositorio. 2- Añadir los cambios a la zona de intercambio temporal. 3- Hacer un commit de los cambios.
7. Añadir los cambios a la zona de intercambio temporal.
8. Hacer un commit de los cambios con el mensaje “*Añadido capítulo 2.*”
9. Mostrar las diferencias entre la última versión y dos versiones anteriores.
10. Crear el fichero `capitulo3.txt` en la carpeta `capítulos` con el siguiente texto.
Git permite la creación de ramas lo que permite tener distintas versiones del mismo proyecto y trabajar de manera simultánea en ellas.
11. Añadir los cambios a la zona de intercambio temporal.
12. Hacer un commit de los cambios con el mensaje “*Añadido capítulo 3.*”
13. Mostrar las diferencias entre la primera y la última versión del repositorio.
14. Añadir al final del fichero `indice.txt` la siguiente línea:
Capítulo 5: Conceptos avanzados
15. Añadir los cambios a la zona de intercambio temporal.
16. Hacer un commit de los cambios con el mensaje “*Añadido capítulo 5 al índice.*”.
17. Mostrar quién ha hecho cambios sobre el fichero `indice.txt`.

PARTE 3: Deshacer cambios

1. Eliminar la última línea del fichero `indice.txt` y guardarlo.
2. Comprobar el estado del repositorio.
3. Deshacer los cambios realizados en el fichero `indice.txt` para volver a la versión anterior del fichero.
4. Volver a comprobar el estado del repositorio.
5. Eliminar la última línea del fichero `indice.txt` y guardarlo.
6. Añadir los cambios a la zona de intercambio temporal.
7. Comprobar de nuevo el estado del repositorio.
8. Quitar los cambios de la zona de intercambio temporal, pero mantenerlos en el directorio de trabajo.
9. Comprobar de nuevo el estado del repositorio.
10. Deshacer los cambios realizados en el fichero `indice.txt` para volver a la versión anterior del fichero.
11. Volver a comprobar el estado del repositorio.
12. Eliminar la última línea del fichero `indice.txt` y guardarlo.
13. Eliminar el fichero `capitulos/capitulo3.txt`.
14. Añadir un fichero nuevo `capitulos/capitulo4.txt` vacío.
15. Añadir los cambios a la zona de intercambio temporal.
16. Comprobar de nuevo el estado del repositorio.
17. Quitar los cambios de la zona de intercambio temporal, pero mantenerlos en el directorio de trabajo.
18. Comprobar de nuevo el estado del repositorio.
19. Deshacer los cambios realizados para volver a la versión del repositorio.
20. Volver a comprobar el estado del repositorio.

21. Eliminar la última línea del fichero indice.txt y guardarlo.
22. Eliminar el fichero capitulos/capitulo3.txt.
23. Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje "*Borrado accidental.*"
24. Comprobar el historial del repositorio.
25. Deshacer el último commit, pero mantener los cambios anteriores en el directorio de trabajo y la zona de intercambio temporal.
26. Comprobar el historial y el estado del repositorio.
27. Volver a hacer el commit con el mismo mensaje de antes.
28. Deshacer el último commit y los cambios anteriores del directorio de trabajo volviendo a la versión anterior del repositorio.
29. Comprobar de nuevo el historial y el estado del repositorio.

PARTE 4: RAMAS

1. Crear una nueva rama bibliografía y mostrar las ramas del repositorio.
2. Crear el fichero capitulos/capitulo1.txt y añadir el texto siguiente
En este capítulo veremos cómo usar GitHub para alojar repositorios en remoto.
3. Añadir los cambios a la zona de intercambio temporal.
4. Hacer un commit con el mensaje "*Añadido capítulo 1.*"
5. Mostrar la historia del repositorio incluyendo todas las ramas.
6. Cambiar a la rama bibliografía.
7. Crear el fichero bibliografia.txt y añadir la siguiente referencia
"Chacon, S. and Straub, B. Pro Git. Apress."
8. Añadir los cambios a la zona de intercambio temporal (*staging area*).
9. Hacer un commit con el mensaje "*Añadida primera referencia bibliográfica.*"
10. Mostrar la historia del repositorio incluyendo todas las ramas.
11. Fusionar la rama bibliografía con la rama master.
12. Mostrar la historia del repositorio incluyendo todas las ramas.
13. Eliminar la rama bibliografía.
14. Mostrar de nuevo la historia del repositorio incluyendo todas las ramas.
15. Crear la rama bibliografía.
16. Cambiar a la rama bibliografía.
17. Cambiar el fichero bibliografia.txt para que contenga las siguientes referencias:
Scott Chacon and Ben Straub. Pro Git. Apress.
Ryan Hodson. Ry's Git Tutorial. Smashwords (2014)
18. Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje "*Añadida nueva referencia bibliográfica.*"
19. Cambiar a la rama master.
20. Cambiar el fichero bibliografia.txt para que contenga las siguientes referencias:
Chacon, S. and Straub, B. Pro Git. Apress.
Loeliger, J. and McCullough, M. Version control with Git. O'Reilly.
21. Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje "*Añadida nueva referencia bibliográfica.*"
22. Fusionar la rama bibliografía con la rama master.
23. Resolver el conflicto dejando el fichero bibliografia.txt con las referencias:

Chacon, S. and Straub, B. Pro Git. Apress.

Loeliger, J. and McCullough, M. Version control with Git. O'Reilly.

Hodson, R. Ry's Git Tutorial. Smashwords (2014)

24. Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje "*Resuelto conflicto de bibliografía.*"
25. Mostrar la historia del repositorio incluyendo todas las ramas.