

Primera Práctica: Git

En estos ejercicios es necesario que hagas una captura de pantalla por cada punto mostrando que has realizado la acción que te pide. Cuando se haga una pregunta será necesario responder de manera razonada los resultados que aparecen.

Ejercicio 1

1. Crea un directorio y escribir git init
2. Usa git status para ver en qué rama estás.
3. ¿Qué aspecto tiene git log?
4. Crea un archivo
5. ¿Qué aspecto tiene ahora la salida de git status?
6. Añade el archivo al área de preparación
7. ¿Qué aspecto tiene ahora git status?
8. Haz un commit del archivo en el repositorio
9. ¿Qué aspecto tiene ahora git status?
10. Cambiar el contenido del archivo que has creado antes
11. ¿Qué sucede cuando ejecutas el comando git status?
12. Añadir el cambio del archivo
13. ¿Qué sucede cuando ejecutas el comando git status esta vez?
14. Vuelve a cambiar el archivo
15. Haz un commit
16. ¿Qué aspecto tiene ahora tras ejecutar git status? ¿Y el log?
17. Añade y haz un commit con el cambio más reciente

Ejercicio 2

1. Crea un archivo llamado file.txt.
2. Almacena el archivo en el repositorio con un commit.
3. Sobrescribe el contenido de file.txt: `echo 2 > file.txt` para cambiar el estado de tu archivo en el directorio de trabajo.
4. ¿Qué te dice `git diff`?
5. ¿Qué te dice `git diff --staged`? ¿Por qué está en blanco?
6. Ejecuta `git add file.txt` para añadir a stage tus cambios desde el directorio de trabajo.
7. ¿Qué te dice `git diff`?
8. ¿Qué te dice `git diff --staged`?
9. Añade a stage el archivo.
10. Sobrescribe el contenido de file.txt: `echo 3 > file.txt` para cambiar el estado de tu archivo en el directorio de trabajo.
11. ¿Qué te dice `git diff`?
12. ¿Qué te dice `git diff --staged`?
13. Explica lo que está ocurriendo.
14. Ejecuta `git status` y observa qué file.txt está presente dos veces en la salida.
15. Ejecuta `git reset HEAD file.txt` para deshacer el cambio.
16. ¿Qué te dice ahora `git status`?
17. Prepara el cambio y haz un commit.
18. ¿Qué aspecto tiene el registro?
19. Sobrescribe el contenido de file.txt: `echo 4 > file.txt`.
20. ¿Cuál es el contenido de file.txt?
21. ¿Qué nos dice el estado de git?
22. Ejecuta `git checkout file.txt`.
23. ¿Cuál es el contenido de file.txt?
24. ¿Qué nos dice el comando status de git?

Ejercicio 3

1. Usa git branch para ver cuantas ramas tiene tu repositorio
2. ¿En qué rama estás?
3. Usa git branch mybranch para crear una nueva rama llamada mybranch
4. Usa git branch de nuevo para ver la nueva rama creada.
5. Usa git checkout mybranch para cambiar a tu nueva rama.
6. ¿Cómo cambia la salida de git status cuando cambias entre el master y la nueva rama que has creado?
7. ¿Cómo cambia el espacio de trabajo cuando cambias entre las dos ramas?
8. Asegúrate de que estás en tu rama mybranch antes de continuar.
9. Crea un archivo llamado archivo1.txt con tu nombre.
10. Añade el archivo y haz un commit con este cambio.
11. Usa git log --oneline --graph para ver tu rama apuntando al nuevo commit.
12. Vuelve a la rama llamada master.
13. Utiliza git log --oneline --graph y observa cómo el commit que hiciste en la rama mybranch no aparece en la rama master.
14. Crea un nuevo archivo llamado archivo2.txt y commitea ese archivo.
15. Usa git log --oneline --graph --all para ver que tu rama apunta al nuevo commit, y que las dos ramas tienen ahora diferentes commits en ellas.
16. Cambia a tu rama mybranch.
17. ¿Qué ha pasado con tu directorio de trabajo? ¿Puedes ver tu archivo2.txt?
18. Usa git diff mybranch master para ver la diferencia entre las dos ramas.
19. Cambia tu rama a master
20. Haz un merge de la nuevaRama en master
21. ¿Qué tipo de merge se ha realizado? Busca en internet y describe todos los tipos y explica con tus palabras qué hace cada uno
22. ¿Ves los dos archivos?

Ejercicio 4

1. Crea un repositorio en GitHub
2. Clona el repositorio de GitHub en local
3. Crea un archivo cuyo nombre sea EjercicioCuatroGitHub.txt
4. Haz commit de ese archivo en tu repositorio local.
5. Ve a Github y crea un archivo desde el explorador llamado HotFix y almacenalo en el repositorio
6. Haz un push de los cambio del local al repositorio
7. ¿Se han subido tus cambios del local al repositorio remoto?
8. Haz un pull de los cambio del repositorio remoto
9. haz un push de los cambios del local al repositorio remoto
10. ¿Se han subido tus cambios del local al repositorio remoto? ¿Por qué?
11. Crea una rama en el repositorio local
12. Crea un nuevo archivo llamado PullRequest.txt
13. haz commit de ese archivo en la nueva rama
14. Sube el commit al repositorio remoto
15. Crea un pull request en GitHub de esa rama sobre la principal
16. Haz un comentario al pull request
17. Completa el pull request