

PRÁCTICA 1: ENTORNO DE DESARROLLO

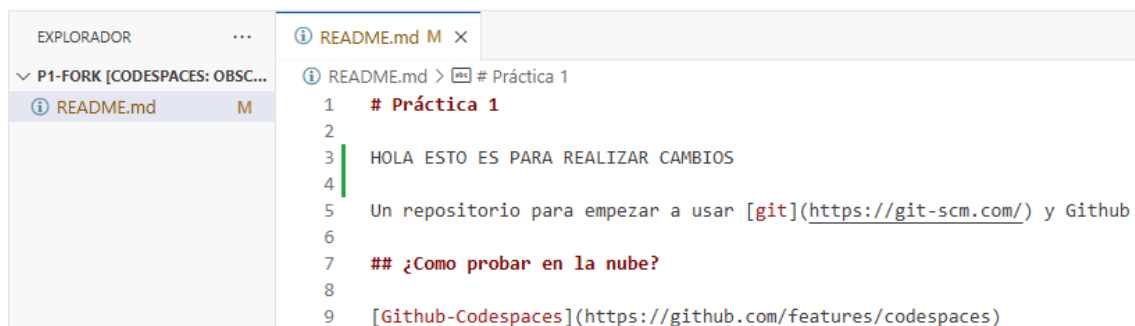
Después de realizar el fork del repositorio, lo primero que nos pedían era realizar un git clone que se utiliza para clonar (descargar) un repositorio remoto a tu máquina local. Si trabajase desde la terminal del ordenador, este comando me permitiría clonar el repositorio en el ordenador pero, como he decidido que iba a trabajar desde un nuevo Codespace dentro del repositorio, este comando no será necesario porque el Codespace ya está vinculado al fork y tendré acceso directo al contenido.

A continuación, ejecuto en la terminal el comando `git status` que muestra el estado actual del repositorio actual. Como no he cambiado nada, al ejecutarlo nos dice que nos encontramos en la rama main (por ahora no hay otra creada), y nos informa de que todos los archivos están actualizados con la fecha actual y, debido a esto, que no hay nada para hacer un commit.

```
@candelaparadinas →/workspaces/p1-fork (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Para ver qué pasa, he decidido añadir el texto “HOLA ESTO ES PARA REALIZAR CAMBIOS” en el README.md que se encuentra seguido a una línea verde que indica que se ha producido un cambio.



Ahora, al ejecutar el `git status`, nos sigue informando de que estamos en la rama main pero nos avisa de que se han producido cambios en el archivo README.md y no se han añadido.

```
● @candelaparadinas →/workspaces/p1-fork (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

Para ello, realizamos un `git add README.md` para actualizar el archivo. (Este comando también sirve para añadir archivos).

```
@candelaparadinas →/workspaces/p1-fork (main) $ git add README.md
```

Y continuamos con el comando `git commit` para confirmar los cambios añadidos al área de preparación con un mensaje descriptivo, en mi caso, “Actualización texto añadido”. Nos informa, una vez ejecutado, de que 1 archivo ha cambiado y ha habido dos inserciones.

```
@candelaparadinas →/workspaces/p1-fork (main) $ git commit -m "Actualización texto añadido"
[main 4cd7c31] Actualización texto añadido
1 file changed, 2 insertions(+)
```

Por último, utilizamos el comando `git push origin main` para enviar los cambios confirmados al repositorio remoto. Una vez ejecutado, nos informa: en la primera línea, identificación de objetos que necesitan ser transferidos al repositorio remoto; en la segunda línea, `git` cuenta cuántos objetos realmente serán enviados o procesados; en la tercera, `git` está optimizando la transferencia de datos utilizando delta compression; en la cuarta, `git` está comprimiendo los objetos que se van a transferir al repositorio remoto; en la quinta línea, `git` está escribiendo los datos comprimidos en el repositorio remoto; y, por último, indica la URL del repositorio remoto al que se están enviando los cambios.

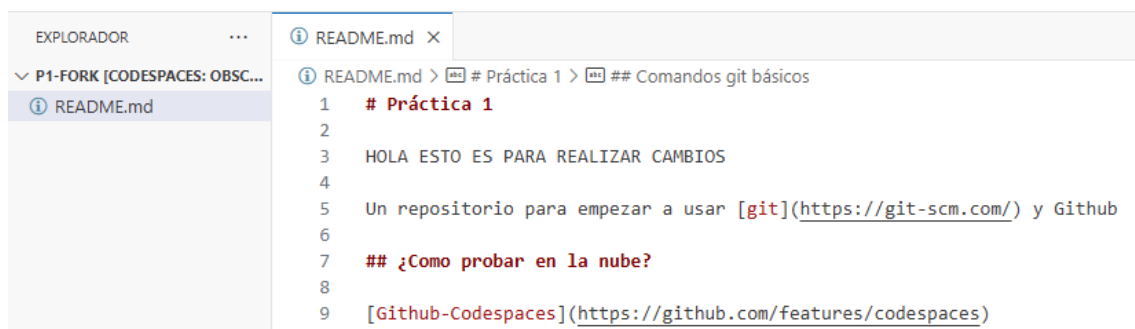
```
@candelaparadinas →/workspaces/p1-fork (main) $ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 346 bytes | 346.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/candelaparadinas/p1-fork
07720b5..4cd7c31 main -> main
```

Una vez completada la actualización del documento, ejecuto el comando `git status` y vemos como ya nos indica que no hay nada para commit.

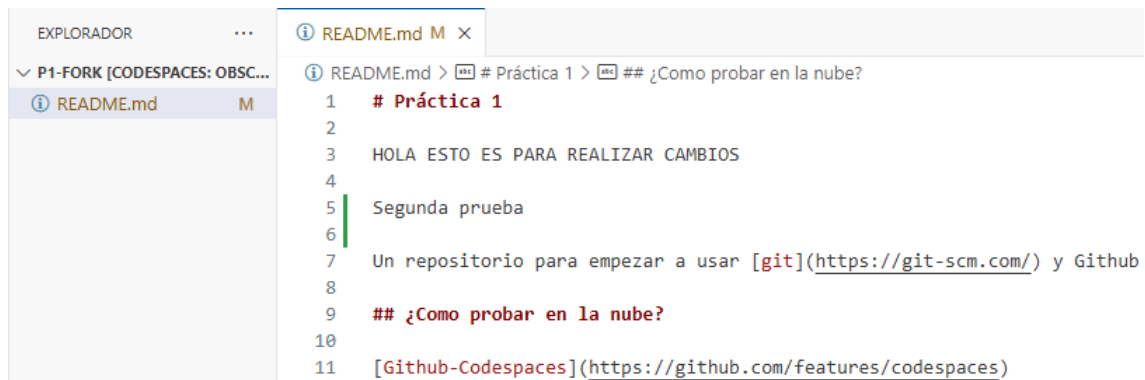
```
@candelaparadinas →/workspaces/p1-fork (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Además, la frase ya no tiene a su izquierda la línea verde que tenía antes de ser actualizada.



Para terminar, nos encontramos con el comando `git checkout` que se encarga de cambiar de rama o restaurar archivos a un estado previo. En este momento, solo tenemos la rama `main` y podríamos crear una nueva para movernos a ella con el comando `git checkout -b nombre-nueva-rama` pero, creo que, en este caso es más interesante restaurar el archivo `README.md` al último estado confirmado. Para ello, utilizamos el comando `git checkout HEAD -- archivo.txt`. He decidido añadir la frase “Segunda prueba”.

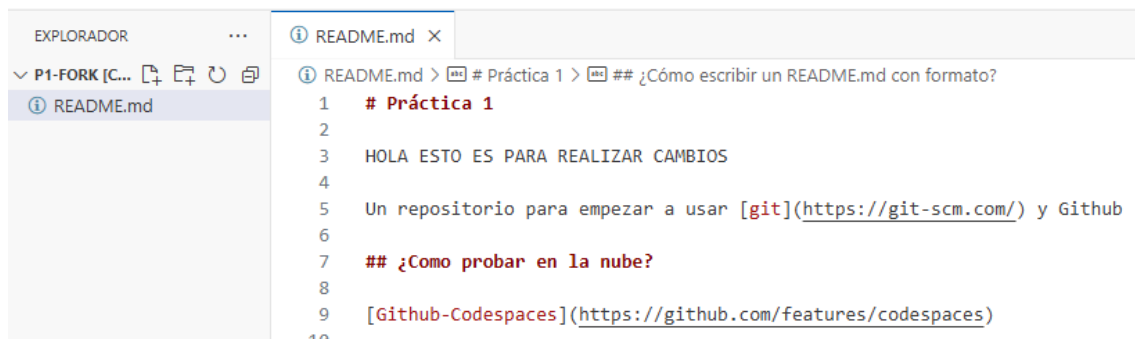


The screenshot shows the VS Code interface. The Explorer on the left shows a file named 'README.md' with a status icon. The Editor on the right shows the content of the file, which is a list of changes:

```
1 # Práctica 1
2
3 HOLA ESTO ES PARA REALIZAR CAMBIOS
4
5 Segunda prueba
6
7 Un repositorio para empezar a usar [git](https://git-scm.com/) y Github
8
9 ## ¿Como probar en la nube?
10
11 [Github-Codespaces](https://github.com/features/codespaces)
```

Ejecutamos este comando y vemos como se elimina la frase y el archivo vuelve al estado anterior porque no habíamos actualizado los cambios.

```
@candelaparadinas →/workspaces/p1-fork (main) $ git checkout HEAD -- README.md
```



The screenshot shows the VS Code interface after running the command. The Explorer on the left shows the 'README.md' file. The Editor on the right shows the content of the file, which is now:

```
1 # Práctica 1
2
3 HOLA ESTO ES PARA REALIZAR CAMBIOS
4
5 Un repositorio para empezar a usar [git](https://git-scm.com/) y Github
6
7 ## ¿Como probar en la nube?
8
9 [Github-Codespaces](https://github.com/features/codespaces)
```