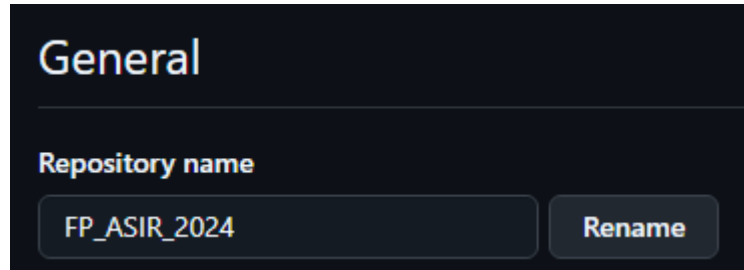
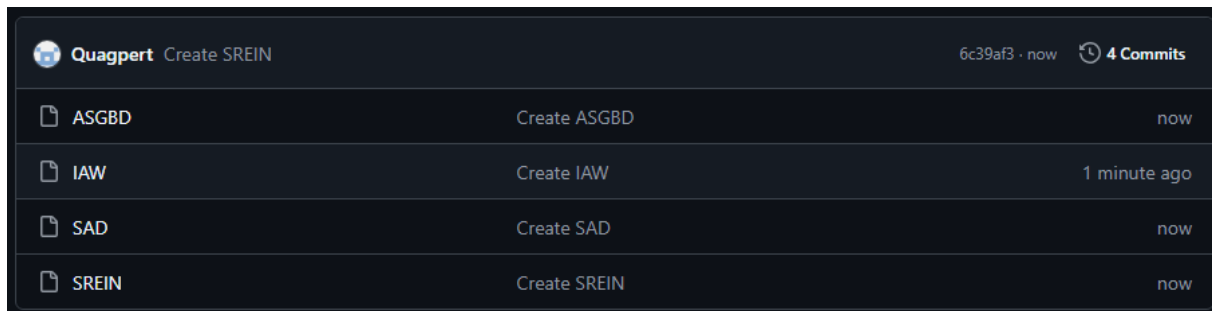


Tarea GIT Introducción

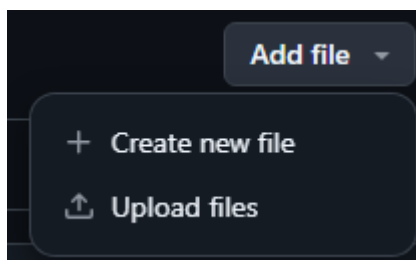
1. Creamos un repositorio local en [GitHub](#).



2. Dentro del repositorio creado anteriormente, añadimos directorios de las asignaturas.



3. Dentro de las carpetas, añadimos un archivo del tema.



1. Crea 3 archivos con cualquier contenido: archivo1.txt, archivo2.txt y archivo3.txt. Estos archivos están contenidos en sus respectivos directorios directorio1, directorio2 y directorio3.

```
C:\Users\Alumno_ciclo\IAW>mkdir directorio1 directorio2 directorio3
```

```
C:\Users\Alumno_ciclo\IAW>echo "ola" > directorio1/archivo1.txt
```

```
C:\Users\Alumno_ciclo\IAW>echo "ola" > directorio2/archivo2.txt
```

```
C:\Users\Alumno_ciclo\IAW>echo "ola" > directorio3/archivo3.txt
```

2. Inicializa un repositorio local

```
C:\Users\Alumno_ciclo\IAW>git init
Initialized empty Git repository in C:/Users/Alumno_ciclo/IAW/.git/
```

3. Añade al stage los archivos de los directorios 1 y 2.

```
C:\Users\Alumno_ciclo\IAW>git add directorio1/archivo1.txt directorio2/archivo2.txt
```

4. Comprueba con git status que el stage tiene lo que quieres.

```
C:\Users\Alumno_ciclo\IAW>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   directorio1/archivo1.txt
    new file:   directorio2/archivo2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    directorio3/
```

5. Crea un commit con esos dos archivos.

```
C:\Users\Alumno_ciclo\IAW>git commit -m "Añadir archivo1 y archivo2"
[master (root-commit) 7c32ce2] Añadir archivo1 y archivo2
 2 files changed, 2 insertions(+)
 create mode 100644 directorio1/archivo1.txt
 create mode 100644 directorio2/archivo2.txt
```

6. Realiza una modificación en el texto del archivo 1/1.txt y otra en el texto del archivo 2/2.txt.

```
C:\Users\Alumno_ciclo\IAW>echo "hola" >> directorio1/archivo1.txt
C:\Users\Alumno_ciclo\IAW>echo "hola" >> directorio2/archivo2.txt
```

7. Añade al stage únicamente las modificaciones del archivo 2/2.txt.

```
C:\Users\Alumno_ciclo\IAW>git add directorio2/archivo2.txt
```

8. Haz git status e interpreta la información que te da.

```
C:\Users\Alumno_ciclo\IAW>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   directorio2/archivo2.txt

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:   directorio1/archivo1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        directorio3/
```

Vemos que la modificación del archivo 2 está añadido en el stage, mientras que el del archivo 1 no. El archivo 3 no está añadido al stage.

9. Crea un commit con las modificaciones que has hecho en el archivo 2/2.txt y con el archivo 3/3.txt.

```
C:\Users\Alumno_ciclo\IAW>git commit -m "Modificar archivo2 y añadir archivo3"
[master 509dce7] Modificar archivo2 y añadir archivo3
3 files changed, 3 insertions(+)
create mode 100644 directorio3/archivo3.txt
```

10. Haz git status y comprueba que el stage está vacío y que en el working directory está el cambio del archivo 1/1.txt.

```
C:\Users\Alumno_ciclo\IAW>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   directorio1/archivo1.txt
        modified:   directorio2/archivo2.txt
        new file:   directorio3/archivo3.txt
```

11. Desversiona el archivo 1/1.txt (es decir, dile a Git que no lo versione), pero sin eliminarlo.

```
C:\Users\Alumno_ciclo\IAW>git rm --cached directorio1/archivo1.txt
rm 'directorio1/archivo1.txt'
```

12. Elimina el archivo 2/2.txt.

```
C:\Users\Alumno_ciclo\IAW>git rm directorio2/archivo2.txt
rm 'directorio2/archivo2.txt'
```

13. Haz git status y comprueba que la eliminación de los dos archivos está en el stage.

```
C:\Users\Alumno_ciclo\IAW>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:   directorio1/archivo1.txt
        deleted:   directorio2/archivo2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        directorio1/
```

14. Crea un commit que almacene la eliminación de esos dos últimos archivos.

```
C:\Users\Alumno_ciclo\IAW>git commit -m "Eliminar archivo1 y archivo2"
[master 54e6f74] Eliminar archivo1 y archivo2
2 files changed, 4 deletions(-)
delete mode 100644 directorio1/archivo1.txt
delete mode 100644 directorio2/archivo2.txt
```

15. Haz git status y comprueba que todo está como esperas (el archivo 1/1.txt está sin versionar y el stage está sin cambios).

```
C:\Users\Alumno_ciclo\IAW>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    directorio1/

nothing added to commit but untracked files present (use "git add" to track)
```

16. Configura el repositorio para que ignore la carpeta 1.

```
C:\Users\Alumno_ciclo\IAW>echo "directorio1/" >> .gitignore
C:\Users\Alumno_ciclo\IAW>git add .gitignore
```

17. Haz git status y comprueba la diferencia.

```
C:\Users\Alumno_ciclo\IAW>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    directorio1/
```