

UT1 TAREA3 : Añadir / modificar archivos a Git

Realiza las siguientes actividades. Asegúrate de coger capturas de cada paso y de contestar a las preguntas realizadas.

AÑADIR ARCHIVOS A GIT.

1. Crea un nuevo directorio de trabajo llamado proyecto e inicializa un repositorio vacío en ese directorio.

Uso este comando "mkdir proyecto" para crear el directorio de trabajo.

"cd proyecto" para situarse dentro de ese directorio.

2. Crea un nuevo fichero fichero1.txt con la cadena "Creando el fichero 1".

Uso este comando "nano fichero1.txt" para crear el fichero solicitado.

3. Crea un nuevo fichero fichero2.txt con la cadena "Creando el fichero2".

Uso este comando "nano fichero2.txt" para crear el fichero solicitado.

4. Verifica que ambos archivos se encuentran sin seguimiento.

Uso el comando "git status" para comprobar que se encuentra sin seguimiento.

```
DAM!@DESKTOP-3LHBG1B MINGW64 ~/Desktop/Archivos Visual Studios/Lenguaje de marcas
$ cd
```

```
DAM!@DESKTOP-3LHBG1B MINGW64 ~
$ cd desktop/git
```

```
DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git (master)
$ mkdir proyecto
```

```
DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git (master)
$ /cd proyecto
bash: /cd: No such file or directory
```

```
DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git (master)
$ cd proyecto
```

```
DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ nano fichero1.txt
```

```
DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ nano fichero2.txt
```

```
DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git status
On branch master
```

```
No commits yet
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
../ejemplo1/
../ejemplo2/
../escenario1/
../escenario2/
../escenario3/
./
```

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

5. Traslada los cambios al área de preparación (stage).

Uso el comando "git add ." para añadir los ficheros al área de preparación.

6. Crea un snapshot de los cambios que tienes en el stage. ¿De qué color se ha puesto la palabra master en la barra del prompt? ¿Qué nos indica?

Uso el comando "git commit -m ..." para llevar los ficheros al área de repositorio o zona de commit. el "-m ..." es para ponerle un mensaje al commit.

La palabra master es la rama principal donde se mezclarán las restantes ramas creadas más tarde.

```
DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git add .

DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   fichero1.txt
        new file:   fichero2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ../ejemplo1/
        ../ejemplo2/
        ../escenario1/
        ../escenario2/
        ../escenario3/

DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git commit -m "haciendo el primer commit"
[master (root-commit) 6dd67a1] haciendo el primer commit
 2 files changed, 2 insertions(+)
 create mode 100644 proyecto/fichero1.txt
 create mode 100644 proyecto/fichero2.txt
```

MODIFICANDO ARCHIVOS.

1. Modifica el archivo fichero1.txt añadiéndole la cadena "Modificando el fichero 1".

Uso el comando "nano fichero1.txt" para acceder al fichero y modificarlo.

2. Verifica que el archivo 1 se encuentra sin seguimiento. ¿De qué color se ha puesto la palabra master en la barra del prompt? ¿Qué símbolo ha aparecido y qué significa?

Uso comando "git status" para comprobar que fichero1.txt no está en seguimiento.

La palabra master es la rama principal (también llamada main) donde se mezclarán las restantes ramas creadas más tarde.

3. Traslada los cambios al área de preparación (stage).

Uso el comando "git add ." añadido el fichero1.txt al área de preparación, también se podría haber usado el comando "git add fichero1.txt" ya que solo es 1 fichero el que se va a añadir al área de preparación.

4. ¿Puedes indicar las diferencias que hay entre el fichero1.txt que se encuentra en el stage y el del repositorio? ¿Qué comando tienes que utilizar?

Con el comando "git diff --staged" se consigue ver las diferencias entre el área de preparación y el área de repositorio o zona de commit

```
DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ nano fichero1.txt

DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git status
On branch master
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:   fichero1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ../ejemplo1/
        ../ejemplo2/
        ../escenario1/
        ../escenario2/
        ../escenario3/

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

DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
    add

DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git add .

DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git diff

DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git diff --staged
diff --git a/proyecto/fichero1.txt b/proyecto/fichero1.txt
index cf7b6b2..12a8de7 100644
--- a/proyecto/fichero1.txt
+++ b/proyecto/fichero1.txt
@@ -1,3 @@
  Creando el fichero 1
+Modificando el fichero 1
+
```

5. Crea un snapshot de los cambios que tienes en el stage. Utiliza el VSCode para escribir el mensaje. ¿Qué sentencia debes utilizar entonces?

Usando el código "git commit -m"..." realizo el commit y escribo un mensaje como para darle información al commit

```
DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git commit -m "haciendo el segundo commit"
[master 3c0e55b] haciendo el segundo commit
1 file changed, 2 insertions(+)
```

ELIMINANDO ARCHIVOS.

1. Elimina el fichero2.txt.

Usando el comando "rm fichero2.txt" se borra el archivo2.txt

2. Detalla y realiza la secuencia de pasos necesaria para incorporar la eliminación del fichero en el repositorio.

Usando el comando "git add ." pasamos el borrado al área de preparación y con el comando "git commit -m"..." " paso la eliminación del archivo al área de repositorio o zona de commit.

Se podría haber usado el git status para ver el que situación esta el archivo en el área de stage

Si hubiera usado el comando "git rm fichero2.txt" se hubiera hecho todo en 1 solo paso.

```
DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ rm fichero2.txt

DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git add .

DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git commit -m "realizando el 3er commit para borrar el fichero2"
[master 286d539] realizando el 3er commit para borrar el fichero2
1 file changed, 1 deletion(-)
delete mode 100644 proyecto/fichero2.txt
```

RENOMBRANDO ARCHIVOS.

1. Renombra el fichero1.txt a nuevofich.txt. ¿Qué comandos puedes utilizar para realizar esta operación?

Uso el comando "mv fichero1.txt nuevofich.txt" para renombrar el fichero1.txt

2. Justifica y detalla el utilizado y realiza la secuencia de pasos necesaria para incorporar este cambio al repositorio.

Uso el comando "git add ." para pasarlo al área de preparación o stage y el comando "git commit -m"..." " para añadirlo al área de commit y añadirle un mensaje.

Se podría haber usado el git status para ver el que situación esta el archivo en el área de stage

Si hubiera usado el comando "git mv fichero1.txt nuevofich.txt" se hubiera hecho todo en 1 solo paso.

```
DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ mv fichero1.txt nuevofich.txt

DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
    add

DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git add .

DAM!@DESKTOP-3LHBG1B MINGW64 ~/desktop/git/proyecto (master)
$ git commit -m "haciendo el 4o commit, renombrando el fichero1.txt"
[master 24ae37f] haciendo el 4o commit, renombrando el fichero1.txt
1 file changed, 0 insertions(+), 0 deletions(-)
rename proyecto/{fichero1.txt => nuevofich.txt} (100%)
```