

Pasar de carpeta normal a **ACTIVE DIRECTORY**:

```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas
$ git init
Initialized empty Git repository in c:/GIT/Pruebas/.git/
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (master)
$ |
```

Hago un html dentro de la carpeta

Pongo “git status” y me dice que no se ha guardado lo ultimo que he echo.

Con el “**git branch -m main**” se cambia lo del (master) a (main)

```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (master)
$ git status
On branch master

No commits yet

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

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

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (master)
$ git branch -m main
```

Con “**git add .**” se sube todo lo que no esté subido al repositorio, si quisiera que solo se hubiera subido un archivo hay que marcar solo ese.

Ponemos “git status” y ya nos dice que esta bien

```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git add .

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Prueba.html
```

```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git add Prueba.html

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git status
on branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Prueba.html
```

Agregar ponemos commit que es como guardarlo definitivamente.

```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git commit -m "Primer commit"
[main (root-commit) 087cfa3] Primer commit
 1 file changed, 11 insertions(+)
 create mode 100644 Prueba.html
```

```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git log
commit 087cfa3380573c0fcf2f223e85d4284f25fd8122 (HEAD -> main)
Author: Sofia Saenz <sofia.saenz@salesianoslosboscos.com>
Date:   Wed Dec 3 08:51:23 2025 +0100

  Primer commit

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git log --graph
* commit 087cfa3380573c0fcf2f223e85d4284f25fd8122 (HEAD -> main)
  Author: Sofia Saenz <sofia.saenz@salesianoslosboscos.com>
  Date:   Wed Dec 3 08:51:23 2025 +0100

  Primer commit

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git log --graph --pretty=oneline
* 087cfa3380573c0fcf2f223e85d4284f25fd8122 (HEAD -> main) Primer commit

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git log --graph --decorate --all --pretty=oneline
* 087cfa3380573c0fcf2f223e85d4284f25fd8122 (HEAD -> main) Primer commit
```

```

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git config --global alias.tree "log --graph"

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git tree
* commit 087cf3380573c0fcf2f223e85d4284f25fd8122 (HEAD -> main)
  Author: Sofia Saenz <sofia.saenz@salesianoslosboscos.com>
  Date:   Wed Dec 3 08:51:23 2025 +0100

  Primer commit

```

Modifico un fichero:

```

Prueba.html X

C: > GIT > Pruebas > Prueba.html > html > body > h2

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Document</title>
7  </head>
8  <body>
9    <h1>Hola mundo</h1>
10   <h2>Adios mundo</h2>
11 </body>
12 </html>

```

lo guardo:

```

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Prueba.html

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:   Prueba.html

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git add .

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Prueba.html

```

Pero si me he equivocado meto esto y me vuelve a como estaba antes:

```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git checkout Prueba.html
Updated 0 paths from the index
```

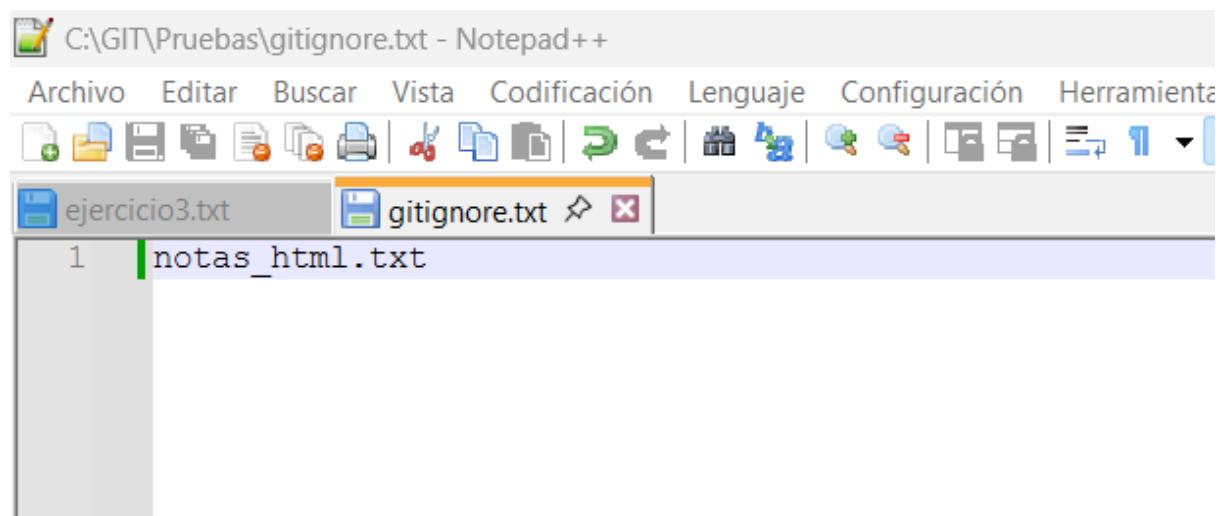
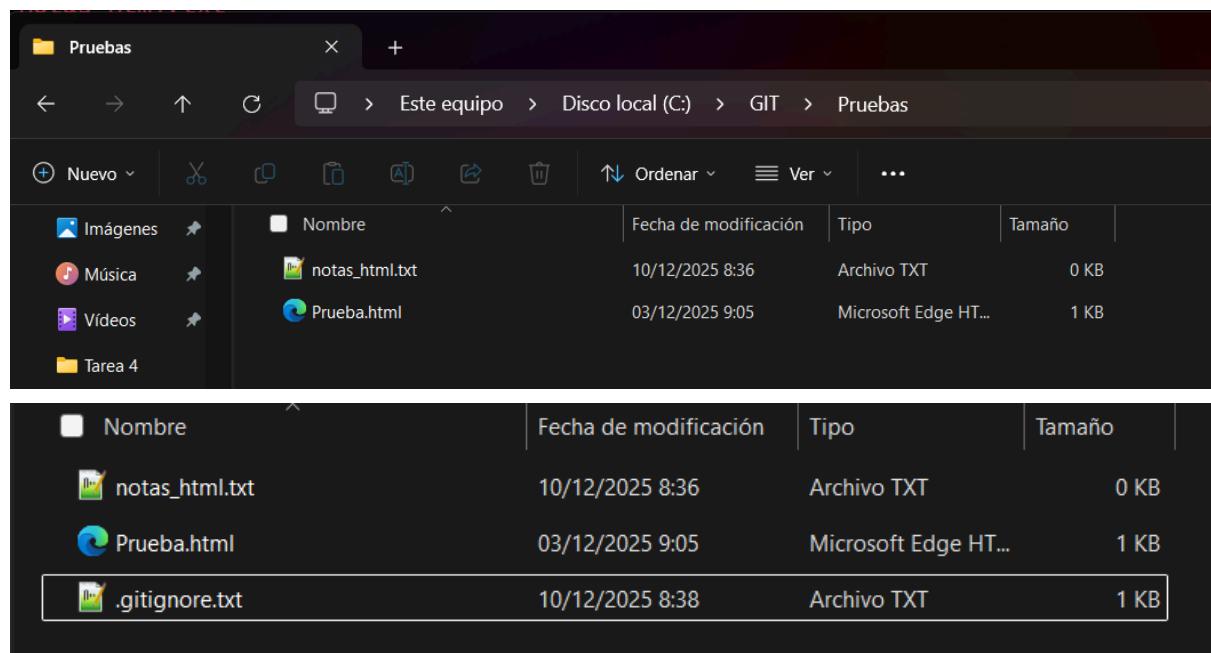
```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git commit -m "Segundo commit"
[main e905a29] Segundo commit
 1 file changed, 3 insertions(+), 1 deletion(-)

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git tree
* commit e905a299065279c56ad736d0bfa192ac2f453bdd (HEAD -> main)
| Author: Sofia Saenz <sofia.saenz@salesianoslosboscos.com>
| Date:   Wed Dec 3 09:07:15 2025 +0100
|
| Segundo commit
|
* commit 087cf3380573c0fcf2f223e85d4284f25fd8122
| Author: Sofia Saenz <sofia.saenz@salesianoslosboscos.com>
| Date:   Wed Dec 3 08:51:23 2025 +0100
|
| Primer commit
```

Cuando me salga END hay que darle a “q” para salir.

```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    notas_html.txt

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



```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    gitignore.txt
    notas_html.txt

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

```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git add .gitignore.txt
```

```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git commit -m "Incluyo gitignore"
[main 5557d95] Incluyo gitignore
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore.txt
```

Debería de haber ocultado notas_html.txt pero no lo hizo.

```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    notas_html.txt

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

```
Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git tree
* ac172fa2982984b6c942ae1f0e8c3a6fc2657208 (HEAD -> main) Todo subido
* 5557d95ddaeefcb05915b00d30fa2804de54d10c3 Incluyo gitignore
* e905a299065279c56ad736d0bfa192ac2f453bdd Segundo commit
* 087cfa3380573c0fcf2f223e85d4284f25fd8122 Primer commit

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git reflog
ac172fa (HEAD -> main) HEAD@{0}: commit: Todo subido
5557d95 HEAD@{1}: commit: Incluyo gitignore
e905a29 HEAD@{2}: commit: Segundo commit
087cfa3 HEAD@{3}: commit (initial): Primer commit

Alumno04@PC04 MINGW64 /c/GIT/Pruebas (main)
$ git checkout
```

git branch login → Crear una nueva rama llamada login.

git switch login → Cambiar a esa rama.

git diff main → Para ver diferencias que tienes desde login con main. (El comando lo mande desde login).

git merge main → Para traer los cambios que hay en main a login.

git stash → Para guardar por si quieres trabajar en otra rama pero no se guarda del todo.

No es como un commit, es como un guardado distinto.

Para borrar un stash se usa git stash drop.

git branch -d (nombre de la rama) → Así elimino ramas.

tag → Para hacer chekouts usando texto en vez de el código.