

Git y GitHub

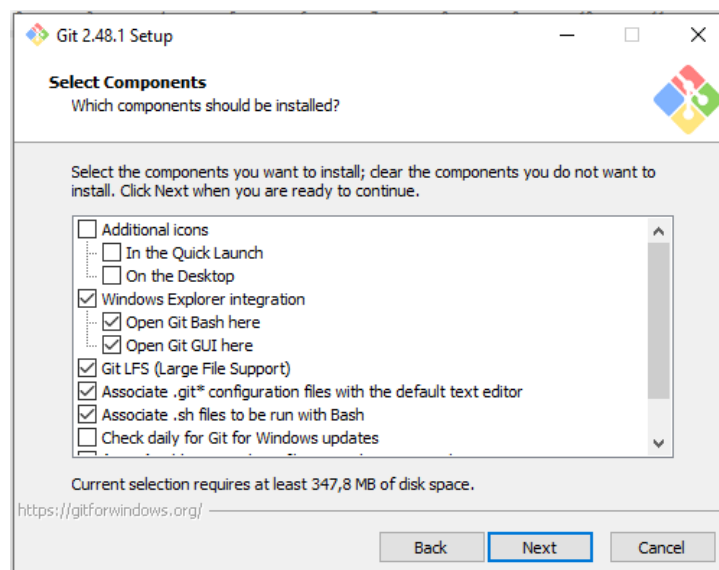
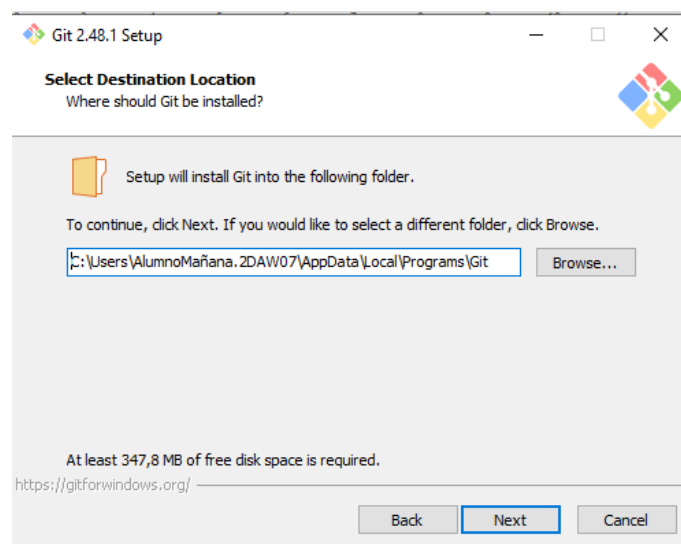
Marta López de los Bueis 2ºDAW

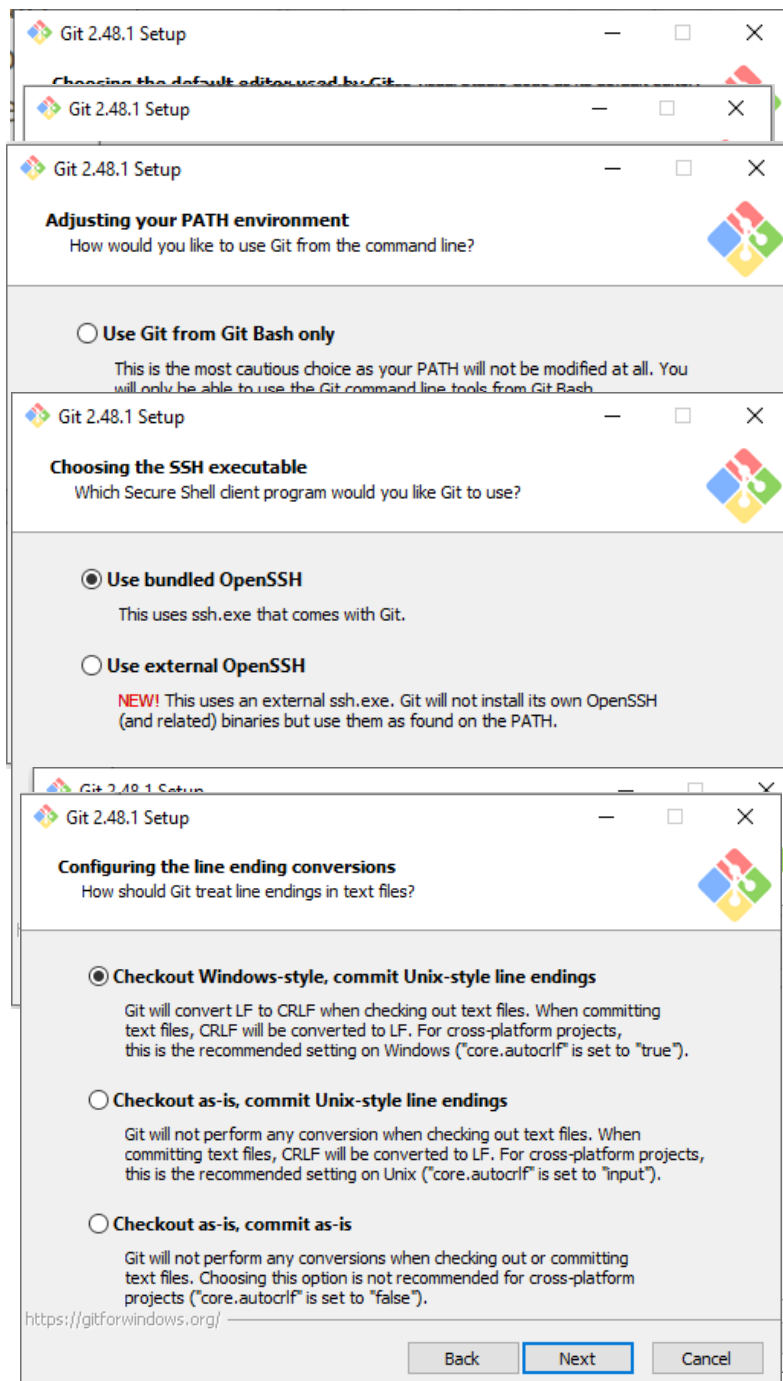
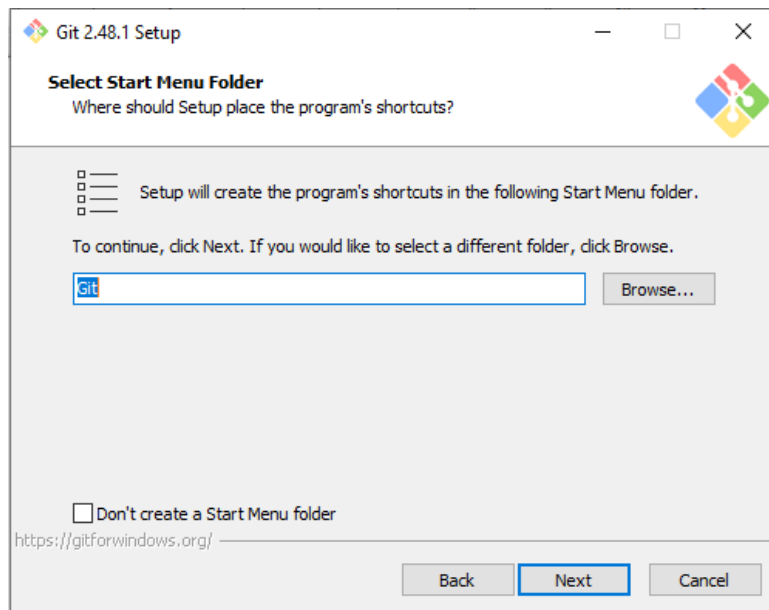
Git es un sistema de control de versiones, permite gestionar cualquier cambio en un proyecto, en el caso de ser necesario, se podrían revertir los cambios realizados.

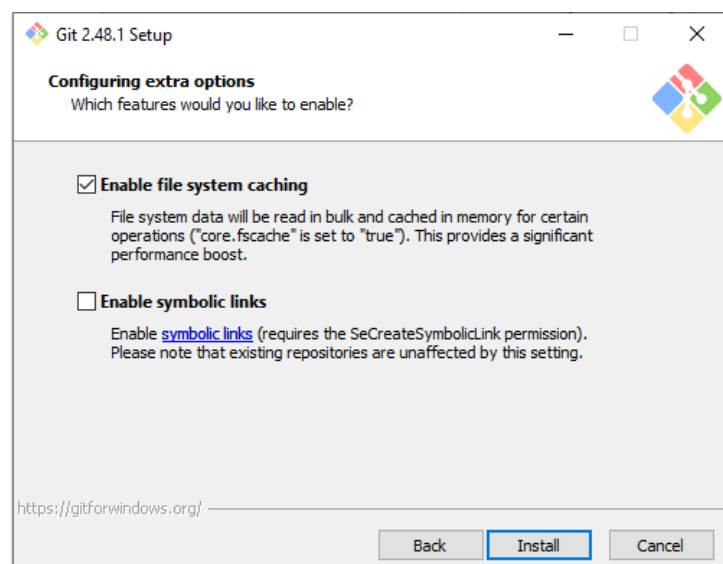
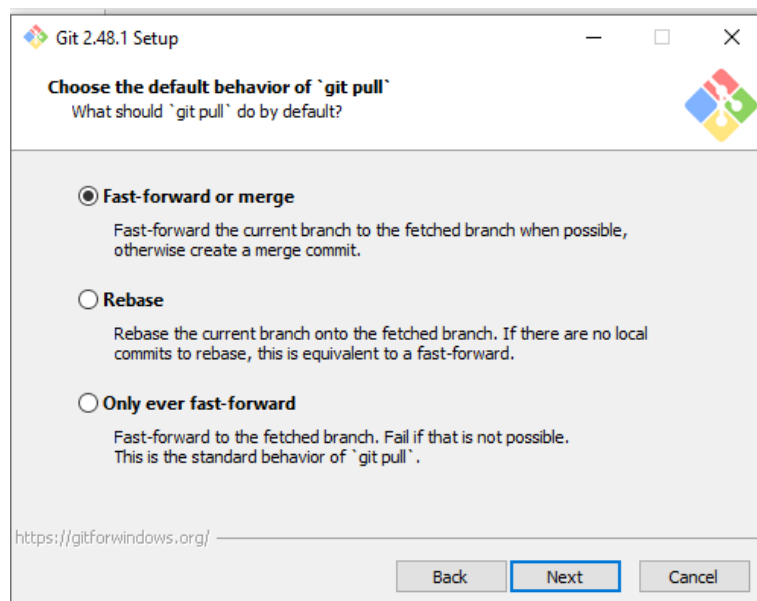
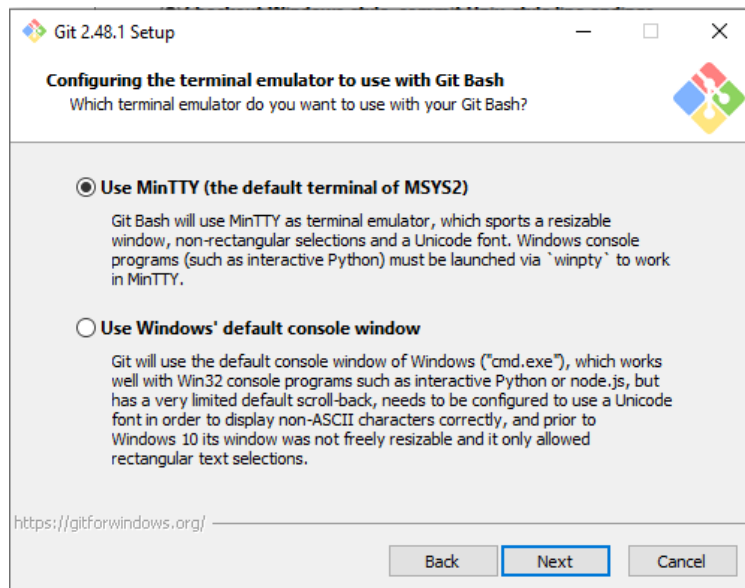
Git se utiliza de forma local y GitHub de forma remota.

1. Instalación de Git (local)

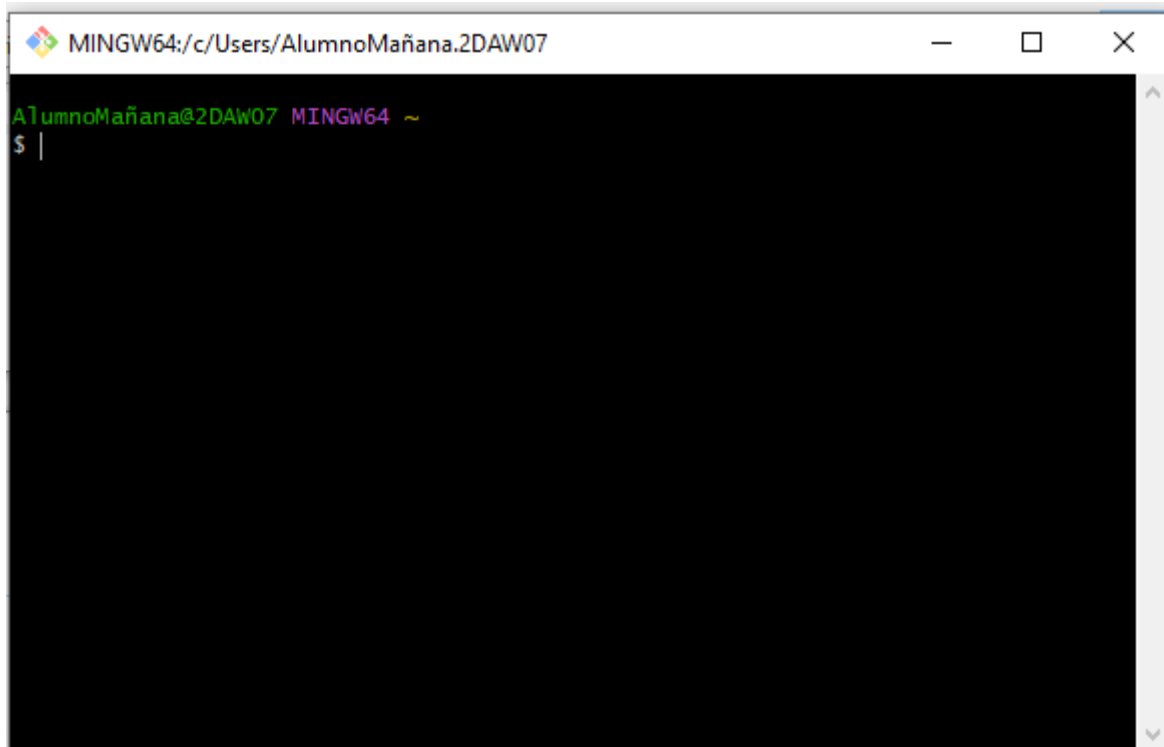
Descargamos de la pagina de git el .exe. Al ejecutarlo comienza la instalación:





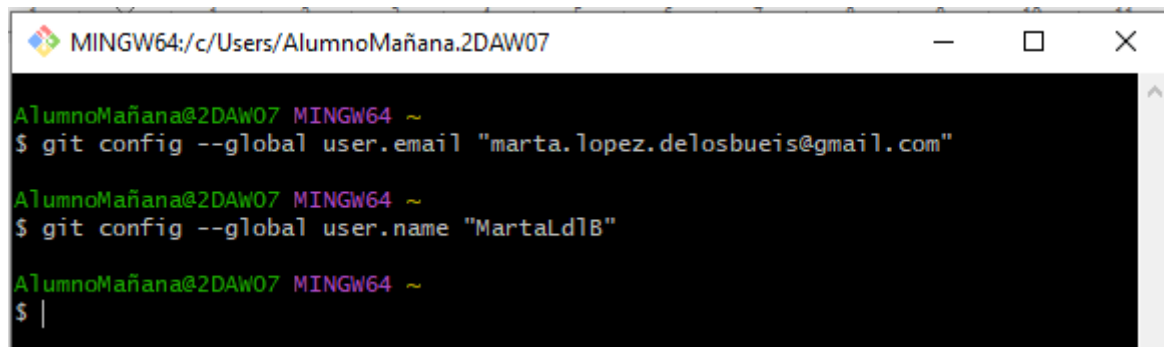


Una vez instalado podemos acceder a su terminal

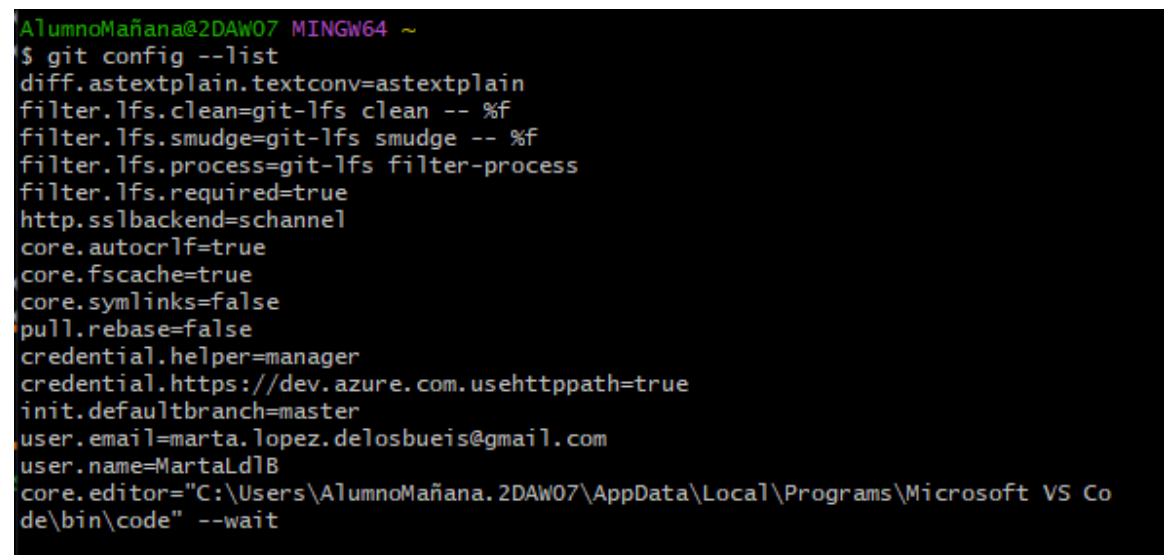


```
MINGW64:/c:/Users/AlumnoMañana.2DAW07
AlumnoMañana@2DAW07 MINGW64 ~
$ |
```

2. Configuramos el usuario y la contraseña



```
MINGW64:/c:/Users/AlumnoMañana.2DAW07
AlumnoMañana@2DAW07 MINGW64 ~
$ git config --global user.email "marta.lopez.delosbueis@gmail.com"
AlumnoMañana@2DAW07 MINGW64 ~
$ git config --global user.name "MartaLd1B"
AlumnoMañana@2DAW07 MINGW64 ~
$ |
```



```
AlumnoMañana@2DAW07 MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=marta.lopez.delosbueis@gmail.com
user.name=MartaLd1B
core.editor="C:\Users\AlumnoMañana.2DAW07\AppData\Local\Programs\Microsoft VS Co
de\bin\code" --wait
```

El comando `git config --global -e` abre el archivo de configuración global de Git en un editor de texto. Esto permite modificar la configuración de Git a nivel global (para todos los repositorios en el sistema).

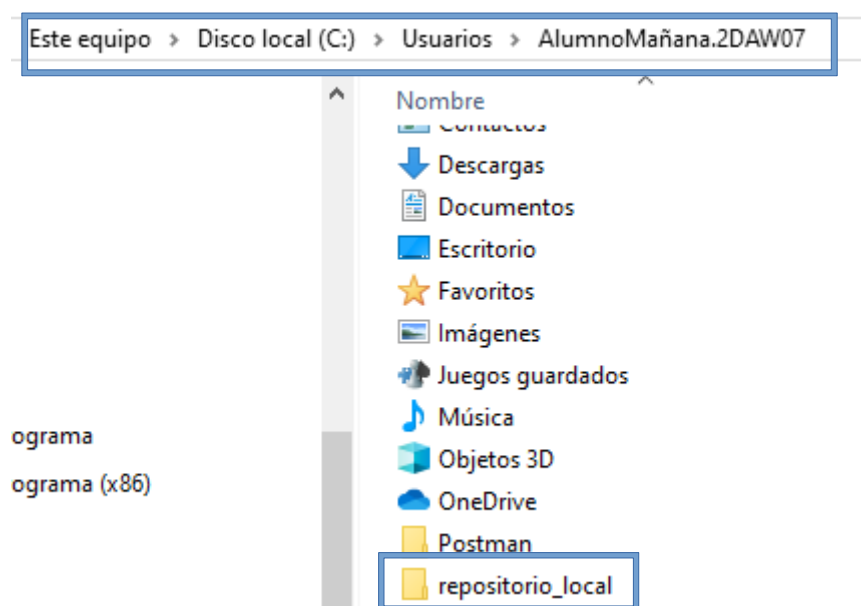
```
AlumnoMañana@2DAW07 MINGW64 ~  
$ git config --global -e  
hint: Waiting for your editor to close the file... |
```

Una vez ejecutado este comando se abre VisualStudio, con un editor de texto llamado `.gitconfig`

```
.gitconfig x  
C: > Users > AlumnoMañana.2DAW07 > .gitconfig  
1 [user]  
2   email = marta.lopez.delosbueis@gmail.com  
3   name = MartaLdlB  
4 [core]  
5   editor = \"C:\\Users\\AlumnoMañana.2DAW07\\AppData\\Local\\Programs\\Microsoft VS Code\\bin\\code\" --wait  
6
```

3. Hacemos una carpeta que será nuestro repositorio local


```
AlumnoMañana@2DAW07 MINGW64 ~  
$ mkdir ./repositorio_local
```



Creamos un archivo en la carpeta que hemos creado, aun no es un repositorio local

```
AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local
$ touch prueba.php



AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local
$
```

» Usuarios » AlumnoMañana.2DAW07 » repositorio_local	
Nombre	Fecha de modif
 prueba.php	21/02/2025 9:27

El comando `git init` inicializa un nuevo repositorio de Git en un directorio. Al ejecutarlo, Git crea una carpeta oculta llamada `.git`, donde almacena toda la información necesaria para el control de versiones, como el historial de cambios y configuraciones del repositorio.

```
AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local
$ git init
Initialized empty Git repository in C:/Users/AlumnoMañana.2DAW07/repositorio_local/.git/

AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$
```

» Usuarios » AlumnoMañana.2DAW07 » repositorio_local	
Nombre	Fecha de modificación
 .git	21/02/2025 9:28
 prueba.php	21/02/2025 9:27

Con `git status` muestra el estado actual del repositorio.

```
AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git status
On branch master

No commits yet

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

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

Para añadir los archivos que queramos a nuestro repositorio usamos `git add <nombre del archivo>` o `git add .`

`git add <nombre del archivo>` nos añade unicamente el archivo que le decimos al área de preparación (staging).

`git add .` Nos añade todos los archivos que hayan sido modificados al área de preparación (staging).

```
AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git add .

AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git status
On branch master

No commits yet

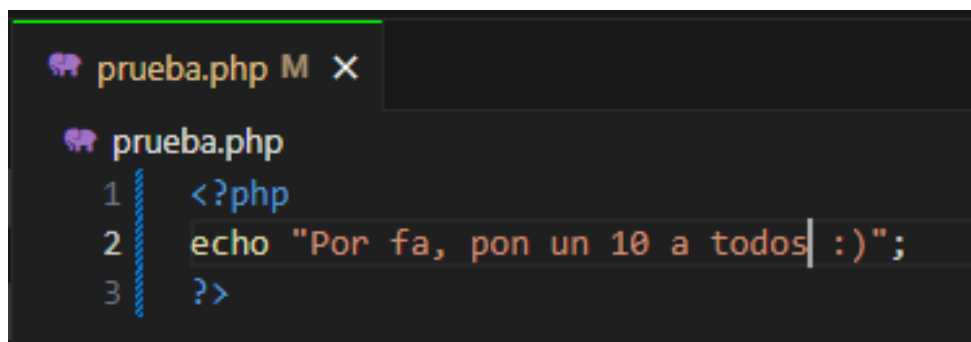
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   prueba.php
```

Ahora está a la espera de hacer el commit.

Al hacer commit se almacena un historial del repositorio para guardar permanentemente los cambios, los commits suelen ir acompañados de un mensaje.

```
AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git commit -m "1º commit"
[master (root-commit) 46febf5] 1º commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 prueba.php
```

Modificamos el archivo que creamos anteriormente



```
prueba.php M X
prueba.php
1  <?php
2  echo "Por fa, pon un 10 a todos :)";
3  ?>
```

Comprobamos el status y vemos que indica que hay cambios para el commit

```
AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (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:   prueba.php

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

Con git diff podemos ver los cambios realizados

```
AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git diff
warning: in the working copy of 'prueba.php', LF will be re
t touches it
diff --git a/prueba.php b/prueba.php
index e69de29..37efc14 100644
--- a/prueba.php
+++ b/prueba.php
@@ -0,0 +1,3 @@
+<?php
+echo "Por fa, pon un 10 a todos :)";
+?>
\ No newline at end of file
```

Eliminar un archivo del repositorios

Creamos un archivo y lo añadimos al repositorios

```
AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ touch archivo_eliminar.php

AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git add archivo_eliminar.php
```

```
AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   archivo_eliminar.php
```



```

AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git reset HEAD archivo_eliminar.php

AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        archivo_eliminar.php

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

```

```

AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git commit -m "El segundo commit que hará que saquemos un 10"
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        archivo_eliminar.php

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

AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        archivo_eliminar.php

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

```

El comando `git checkout` se usa para cambiar entre ramas o restaurar archivos en un repositorio de Git.

```

AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git checkout .
Updated 0 paths from the index

```

El comando `git log` muestra el historial de commits en un repositorio de Git. Permite ver los cambios realizados, quién los hizo y cuándo.

```

AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git log
commit b91faa1a6ed7023664c35905ee8fecf346c7ab25 (HEAD -> master)
Author: MartaLd1B <marta.lopez.delosbueis@gmail.com>
Date:   Fri Feb 21 09:52:46 2025 +0100

    El cambio que hará que tengamos un 10

commit 46febf59db493c0cb202a61019ccad422cb8bf84
Author: MartaLd1B <marta.lopez.delosbueis@gmail.com>
Date:   Fri Feb 21 09:44:51 2025 +0100

    1º commit

```

Para volver a un commit anterior copiamos el identificador que nos muestra el log

```
commit b91faa1a6ed7023664c35905ee8fecf346c7ab25 (HEAD -> master)
```

```
AlumnoMañana@2DAW07 MINGW64 ~/repositorio_local (master)
$ git checkout b91faa1a6ed7023664c35905ee8fecf346c7ab25
Note: switching to 'b91faa1a6ed7023664c35905ee8fecf346c7ab25'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at b91faa1 El cambio que hará que tengamos un 10
```