

Parte 1: Configuración Inicial

1. Configurar Git en tu equipo

Ejecuta los siguientes comandos en la terminal:

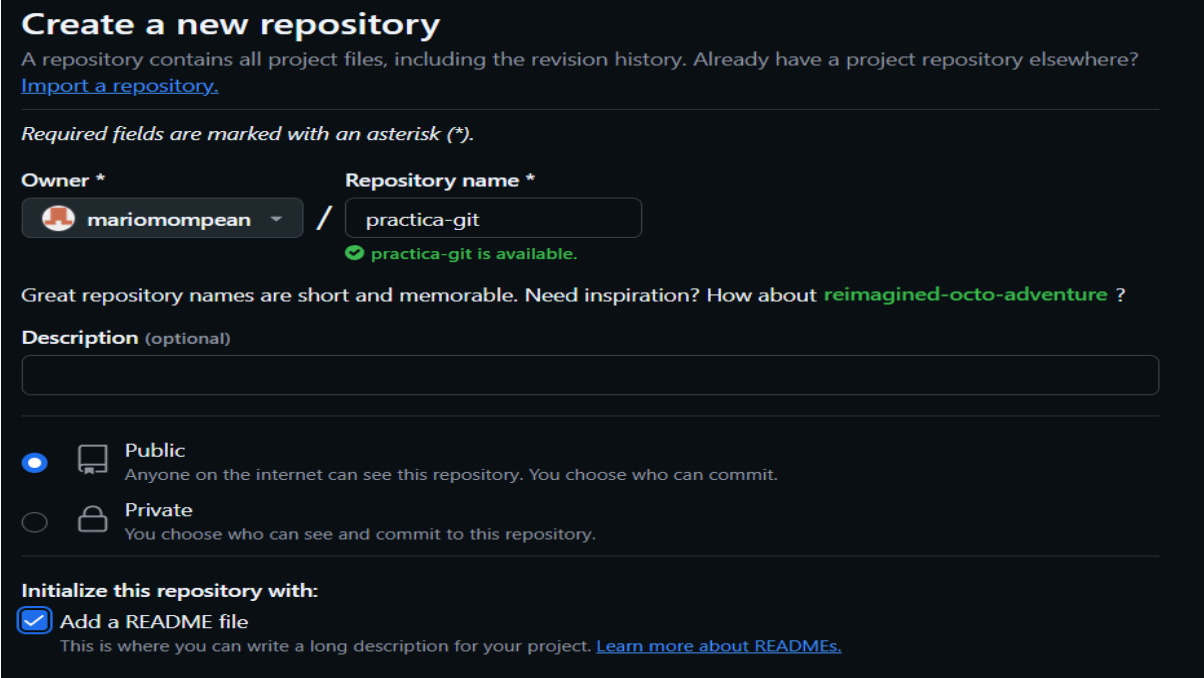
```
git config --global user.name Mario
```

```
git config --global user.email mariomompean9@gmail.com
```

```
C:\xampp\htdocs\despliegue\practica-git>git config --global user.name Mario
C:\xampp\htdocs\despliegue\practica-git>git config --global user.email mariomompean9@gmail.com
```

2. Crear un nuevo repositorio en GitHub


1. Accede a [GitHub](#).
2. Haz clic en **"New Repository"**.
3. Asigna un nombre al repositorio y selecciona la opción **"Initialize this repository with a README"**.




Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Required fields are marked with an asterisk ().*


Owner *  **mariomompean** / **Repository name ***

 **practica-git is available.**

Great repository names are short and memorable. Need inspiration? How about **reimagined-octo-adventure** ?

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

3. Clonar el repositorio en tu equipo

```
git clone https://github.com/mariomompean/practica-git.git
cd practica-git
```

```
C:\xampp\htdocs\despliegue\practica-git>git clone https://github.com/mariomompean/practica-git.git
Cloning into 'practica-git'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
C:\xampp\htdocs\despliegue\practica-git>cd practica-git
```

Parte 2: Trabajo Individual con Git

1. Crear un archivo de prueba y hacer un commit

```
echo "# Proyecto Web" > index.html
git add index.html
git commit -m "Añadir archivo index.html"
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>echo "Proyecto Web" > index.html
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git add index.html
C:\xampp\htdocs\despliegue\practica-git\practica-git>git commit -m "Primer commit del archivo index.html"
[main 5ae4dfa] Primer commit del archivo index.html
1 file changed, 1 insertion(+)
create mode 100644 index.html
```

2. Subir cambios al repositorio remoto

```
git push origin main
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311 bytes | 311.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/mariomompean/practica-git.git
573f765..5ae4dfa  main -> main
```

3. Actualizar el repositorio local con los cambios remotos

```
git pull origin main
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git pull origin main
From https://github.com/mariomompean/practica-git
* branch          main          -> FETCH_HEAD
Already up to date.
```

Parte 3: Trabajo Colaborativo

1. Crear una nueva rama para una funcionalidad

git branch nueva-funcionalidad

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git branch nueva-funcionalidad
```

git checkout nueva-funcionalidad

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git checkout nueva-funcionalidad
Switched to branch 'nueva-funcionalidad'
```

O también:

git checkout -b nueva-funcionalidad

2. Realizar cambios en la nueva rama y hacer commit

echo "<h1>Hola mundo</h1>" > index.html

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>echo "<h1>Hola mundo</h1>" > index.html
```

git add index.html

git commit -m "Añadir título en index.html"

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git add index.html
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git commit -m "Añadir título en index.html"
[nueva-funcionalidad b5c8c2c] Añadir título en index.html
1 file changed, 1 insertion(+), 1 deletion(-)
```

3. Subir la rama al repositorio remoto

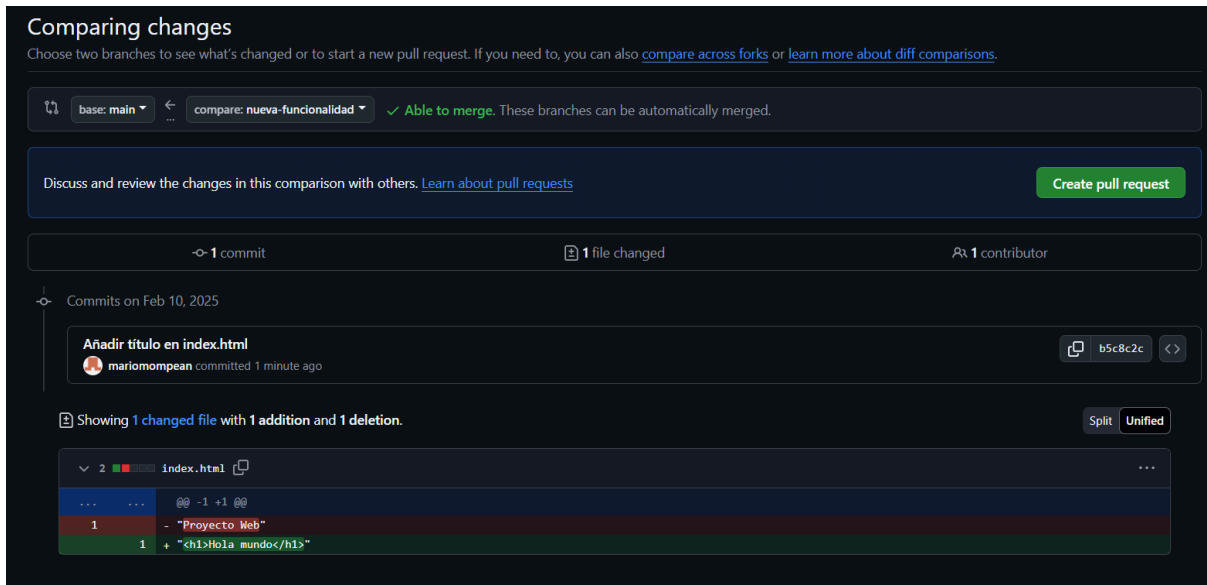
git push origin nueva-funcionalidad

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git push origin nueva-funcionalidad
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 317 bytes | 317.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'nueva-funcionalidad' on GitHub by visiting:
remote:   https://github.com/mariomompean/practica-git/pull/new/nueva-funcionalidad
remote:
To https://github.com/mariomompean/practica-git.git
 * [new branch]   nueva-funcionalidad -> nueva-funcionalidad
```

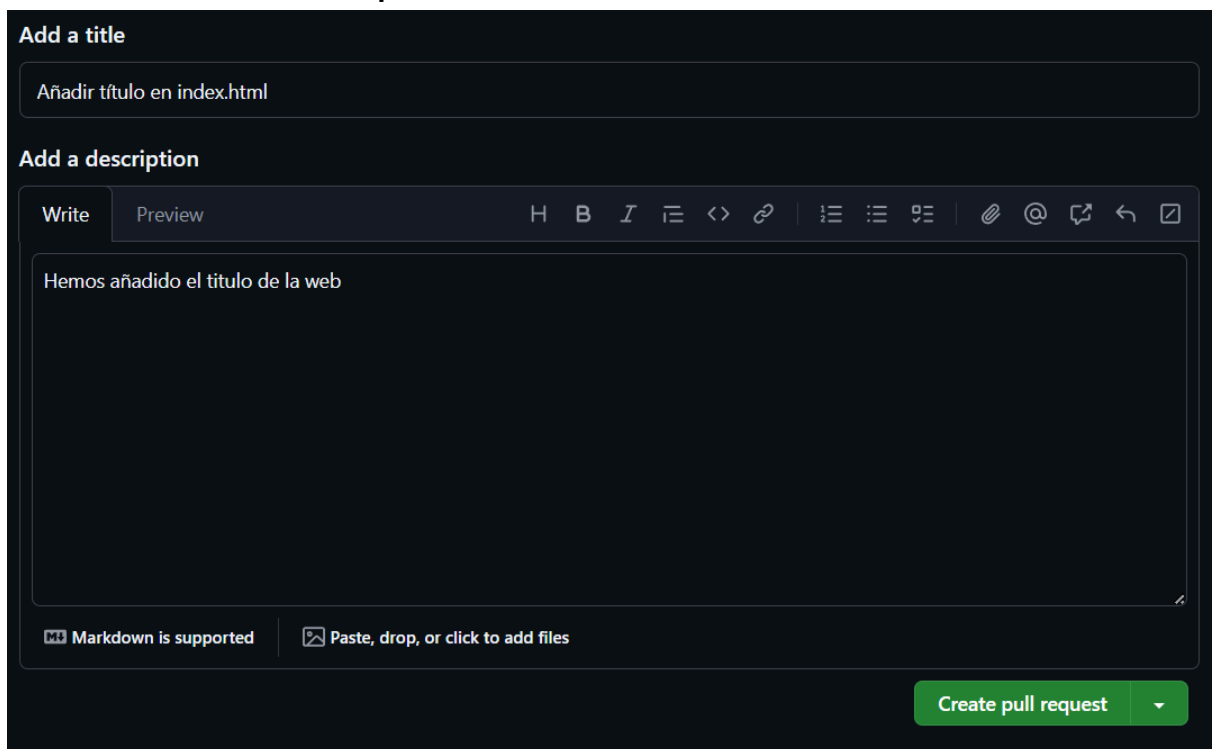
4. Realizar un Pull Request (PR) en GitHub

Un Pull Request (PR) es una solicitud para fusionar los cambios de una rama en otra. Para crear un PR:

1. Accede al repositorio en GitHub.
2. Ve a la pestaña **"Pull Requests"** y haz clic en **"New Pull Request"**.
3. Selecciona la rama que contiene los cambios y compárala con **main**.



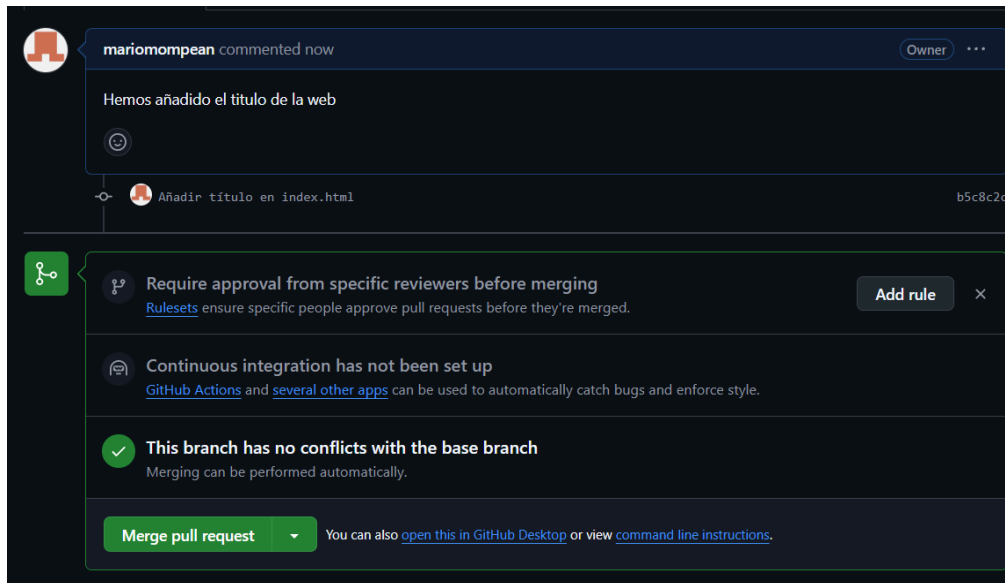
4. Revisa los cambios antes de enviarlos.
5. Escribe una descripción detallada del PR explicando los cambios realizados.
6. Haz clic en **"Create Pull Request"**.



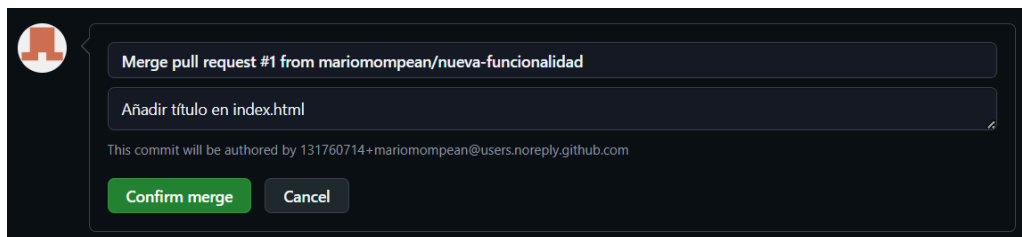
7. Espera revisiones y feedback de los compañeros o del profesor.

5. Revisar y fusionar cambios

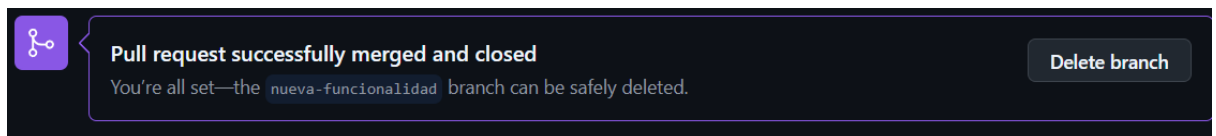
1. Otro compañero revisa el PR y lo aprueba.



2. Si todo está correcto, se fusiona con la rama **main**.



3. Si hay sugerencias o cambios requeridos, deben realizarse antes de la aprobación.
4. Una vez aprobado, hacer clic en **"Merge Pull Request"** y eliminar la rama si ya no es necesaria.



6. Actualizar la rama local con los cambios aprobados

git checkout main

git pull origin main

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\xampp\htdocs\despliegue\practica-git\practica-git>git pull origin main
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 927 bytes | 463.00 KiB/s, done.
From https://github.com/mariomompean/practica-git
* branch          main          -> FETCH_HEAD
   5ae4dfa..d6e6192  main        -> origin/main
Updating 5ae4dfa..d6e6192
Fast-forward
 index.html | 2 +
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Parte 4: Resolución de Conflictos

1. Generar un conflicto (de forma intencionada)

1. Dos alumnos editan la misma línea de `index.html` en diferentes ramas y hacen commits.

Alumno 1:

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git branch alumno1
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git checkout alumno1
Switched to branch 'alumno1'
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>echo "<h1>Hola Alumno 1</h1>" > index.html
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git add index.html
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git commit -m "Commit del alumno 1.html"
[alumno1 ab674dd] Commit del alumno 1.html
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git push origin alumno1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 310 bytes | 310.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'alumno1' on GitHub by visiting:
remote:      https://github.com/mariomompean/practica-git/pull/new/alumno1
remote:
To https://github.com/mariomompean/practica-git.git
* [new branch]      alumno1 -> alumno1
```

Alumno 2:

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git branch alumno2
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git checkout alumno2
Switched to branch 'alumno2'
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>echo "<h1>Hola Alumno 2</h1>" > index.html
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git add index.html
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git commit -m "Commit del alumno 2.html"
[alumno2 afbe86e] Commit del alumno 2.html
1 file changed, 1 insertion(+), 1 deletion(-)
```

2.

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git push origin alumno2
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311 bytes | 311.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'alumno2' on GitHub by visiting:
remote:      https://github.com/mariomompean/practica-git/pull/new/alumno2
remote:
To https://github.com/mariomompean/practica-git.git
* [new branch]      alumno2 -> alumno2
```

Primero fusionamos la del alumno 1.

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\xampp\htdocs\despliegue\practica-git\practica-git>git merge alumno1
Updating d6e6192..ab674dd
Fast-forward
 index.html | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Y ahora nos dará el conflicto.

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

C:\xampp\htdocs\despliegue\practica-git\practica-git>git merge alumno1
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

2. Resolver el conflicto

1. Git avisará del conflicto.

```
<<<<<< HEAD (Current Change)
"<h1>Hola Alumno 2</h1>"
=====
"<h1>Hola Alumno 1</h1>"
jesus
>>>>>> alumno1 (Incoming Change)
```

2. Edita manualmente el archivo afectado, manteniendo los cambios correctos.

```
<> index.html X
C: > xampp > htdocs > despliegue > practica-git > practica-git > <> index.html > ...
You, 8 minutes ago | 1 author (You)
1  "<h1>Hola Alumno 2</h1>"
2
```

3. Una vez resuelto, ejecutar:

```
git add index.html
git commit -m "Resolver conflicto en index.html"
git push origin main
```

```
C:\xampp\htdocs\despliegue\practica-git\practica-git>git add index.html

C:\xampp\htdocs\despliegue\practica-git\practica-git>git commit -m "Resolver conflicto en index.html"
[main 5efbe43] Resolver conflicto en index.html

C:\xampp\htdocs\despliegue\practica-git\practica-git>git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 505 bytes | 505.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/mariomompean/practica-git.git
 d6e6192..5efbe43  main -> main
```

Parte 5: Entrega Final

Cada equipo debe entregar un repositorio con al menos:

- Un **README.md** documentado.
- Un historial de commits bien estructurado.
- Uso de ramas y Pull Requests.
- Al menos un conflicto resuelto correctamente.

Entrega

1. Enviar el enlace del repositorio a la plataforma de entrega.

Evaluación

- **Creación y clonación del repositorio:** 20%
- **Uso correcto de commits y ramas:** 30%
- **Trabajo colaborativo y Pull Requests:** 30%
- **Resolución de conflictos y estructura del repositorio:** 20%