

Práctica: Uso de Git en Equipo



Repository Git: <https://github.com/bcarrascosergio/GitEnEquipo.git>

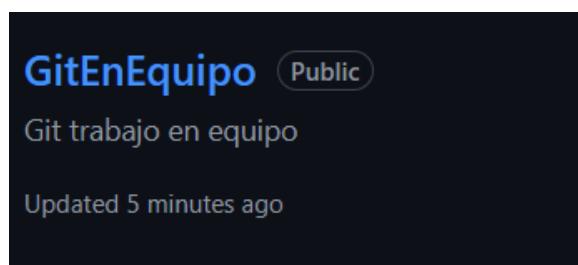
En grupos de 3-4 personas debéis hacer lo siguiente. Todo debe ir con capturas de pantalla en un documento que me vais a entregar.

1. Creación del repositorio

1. Un miembro del equipo debe crear un nuevo repositorio en GitHub/GitLab.

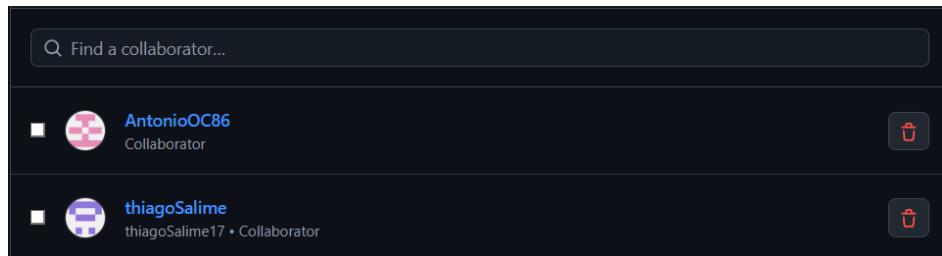
Creamos el repositorio llamado GitEnEquipo y la configuramos como pública.

Sergio se encargó de crear el repositorio público de git.



2. Configurar el repositorio como público.
3. Los demás miembros del equipo deben ser añadidos como colaboradores.

Ahora ya estamos todos como colaboradores en el repositorio.



2. Clonación del repositorio

1. Cada miembro del equipo debe clonar el repositorio en su máquina con:

```
git clone <URL_DEL_REPO>
```

Todos los miembros hemos clonado el repositorio al local.

```
ACER@LPTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo
$ git clone https://github.com/bcarrascosergio/GitEnEquipo.git
Cloning into 'GitEnEquipo'...
warning: You appear to have cloned an empty repository.
```

```
pablo@DESKTOP-1FRGOU6 MINGW64 ~ (master)
$ cd IdeaProjects/gitEnEquipo/

pablo@DESKTOP-1FRGOU6 MINGW64 ~/IdeaProjects/gitEnEquipo (master)
$ git clone https://github.com/bcarrascosergio/GitEnEquipo.git
```

3. Creación de ramas temáticas

1. Se definirán ramas para diferentes tareas del proyecto, por ejemplo:

- documentacion para la documentación.
- funcionalidad-A para una funcionalidad específica.

```
git checkout -b <nombre-de-la-rama>
```

Una vez ya creadas las ramas, tienen que aparecer de esta manera

Antonio se encargó de crear ambas ramas.

```
antho@AOC MINGW64 ~/IdeaProjects/GitEnEquipo (main)
$ git checkout -b "documentación"
Switched to a new branch 'documentación'

antho@AOC MINGW64 ~/IdeaProjects/GitEnEquipo (documentación)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

antho@AOC MINGW64 ~/IdeaProjects/GitEnEquipo (main)
$ git branch
  documentación
* main

antho@AOC MINGW64 ~/IdeaProjects/GitEnEquipo (main)
$ git checkout -b "funcionalidad-A"
Switched to a new branch 'funcionalidad-A'
```

```
ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (main)
$ git pull
From https://github.com/bcarrascosergio/GitEnEquipo
 * [new branch]      documentación    -> origin/documentación
 * [new branch]      funcionalidad-A -> origin/funcionalidad-A
Already up to date.
```

**En esta captura descargamos los cambios en nuestros equipos
(Sergio y Thiago)**

2. Dentro de cada equipo, dos miembros trabajarán sobre la misma rama para generar conflictos intencionados. Debéis indicar quien es quien y donde vais a trabajar en todo momento.

Creamos los ficheros en los cualesharemos el conflicto, los creamos de la siguiente manera:

```
ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (funcionalidad-A)
$ nano funcionalidad.txt

ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (funcionalidad-A)
$ cat funcionalidad.txt
Hola cómo estás???
ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (funcionalidad-A)
$
```

Y lo subiremos al repositorio de la siguiente forma

```
ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (funcionalidad-A)
$ git add funcionalidad.txt
warning: in the working copy of 'funcionalidad.txt', LF will be replaced
by CRLF.
hes it

ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (funcionalidad-A)
$ git commit -a -m "Subiendo funcionalidad.txt"
[funcionalidad-A bd26657] Subiendo funcionalidad.txt
 1 file changed, 1 insertion(+)
 create mode 100644 funcionalidad.txt

ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (funcionalidad-A)
$ git push
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), 316 bytes | 316.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/bcarrascosergio/GitEnEquipo.git
 57ee6c2..bd26657  funcionalidad-A -> funcionalidad-A
```

3. Por cada rama, se debe crear un archivo y debéis modificarlo. Aquellas personas que trabajen sobre la misma rama, deben modificar a la vez el mismo archivo después de crearlo.
4. Confirmar los cambios:

```
git add .
```

```
git commit -m "Añadida nueva funcionalidad en <nOMBRE-de-la-rama>"
```

```
ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (main|MERGING)
$ git add .
error: 'GitEnEquipo/' does not have a commit checked out
fatal: adding files failed

ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (main|MERGING)
$ git commit -a -m "Sergio comentó el fichero principal.txt"
[main da54410] Sergio comentó el fichero principal.txt
```

5. Subir la rama al repositorio remoto:

git push origin <nombre-de-la-rama>

```
ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (main)
$ git push
To https://github.com/bcarrascosergio/GitEnEquipo.git
 ! [rejected]      main -> main (non-fast-forward)
error: failed to push some refs to 'https://github.com/bcarrascosergio/GitEnEquipo.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. If you want to integrate the remote changes,
hint: use 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (main)
$ git pull
Auto-merging principal.txt
CONFLICT (content): Merge conflict in principal.txt.
Automatic merge failed; fix conflicts and then commit the result.

ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (main|MERGING)
$ git add .
error: 'GitEnEquipo/' does not have a commit checked out
fatal: adding files failed

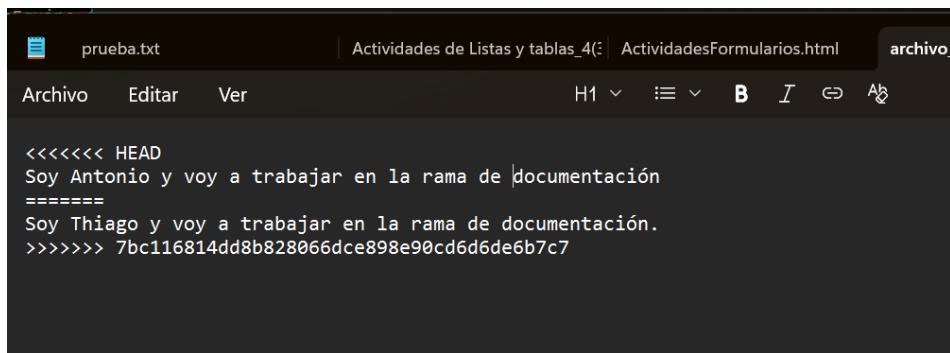
ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (main|MERGING)
$ git commit -a -m "Sergio comentó el fichero principal.txt"
[main da54410] Sergio comentó el fichero principal.txt
 1 file changed, 1 insertion(+)

ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (main)
$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 735 bytes | 735.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/bcarrascosergio/GitEnEquipo.git
 6c00bd3..da54410 main -> main
```

En las capturas de la parte superior trabajaron Thiago y Sergio, y generaron un conflicto. Esto fue en la rama principal (main).

```
antho@AOC MINGW64 ~/IdeaProjects/GitEnEquipo (documentación)
$ git commit -a -m "Antonio comentó el archivo archivo_documentacion.txt"
[documentación 771a5f8] Antonio comentó el archivo archivo_documentacion.txt
 1 file changed, 1 insertion(+)

antho@AOC MINGW64 ~/IdeaProjects/GitEnEquipo (documentación)
$ git push
To https://github.com/bcarrascosergio/GitEnEquipo
 ! [rejected]      documentación -> documentación (fetch first)
error: failed to push some refs to 'https://github.com/bcarrascosergio/GitEnEquipo'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```



En las capturas de la parte superior trabajaron Thiago y Antonio, y generaron un conflicto. Esto fue en la rama documentación.

4. Merge de las ramas al main

1. Volver a la rama main:

```
git checkout main
```

2. Descargar los últimos cambios:

```
git pull origin main
```

3. Fusionar cada rama temática en main mediante pull requests en GitHub/GitLab o localmente con:

```
git merge <nombre-de-la-rama>
```

4. En caso de conflictos, Git los señalará en los archivos afectados. Debéis enviar captura de pantalla del conflicto.

Haremos el git merge documentación en la rama main.

El comando no aparece en las capturas, porque se nos abre directamente la interfaz de nano, el comando se utiliza en la rama que quieras actualizar:

git merge <nombre rama actualizada>

```
Merge made by the 'ort' strategy.
principal.txt | 6 +++++-
 1 file changed, 5 insertions(+), 1 deletion(-)

pablo@DESKTOP-1FRGOU6 MINGW64 ~/IdeaProjects/gitEnEquipo (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 322 bytes | 322.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/bcarrascosergio/GitEnEquipo.git
  da54410..c07d5b1  main -> main

pablo@DESKTOP-1FRGOU6 MINGW64 ~/IdeaProjects/gitEnEquipo (main)
$ |
```

Ahora hacemos el git merge de la funcionalidad-a en la rama main.

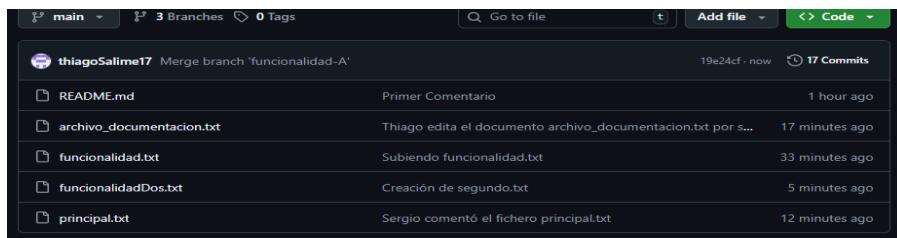
```
Merge made by the 'ort' strategy.
 funcionalidad.txt | 1 +
 funcionalidadDos.txt | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 funcionalidad.txt
 create mode 100644 funcionalidadDos.txt

pablo@DESKTOP-1FRGOU6 MINGW64 ~/IdeaProjects/gitEnEquipo (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 429 bytes | 214.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/bcarrascosergio/GitEnEquipo.git
 c07d5b1..19e24cf  main -> main

pablo@DESKTOP-1FRGOU6 MINGW64 ~/IdeaProjects/gitEnEquipo (main)
$ |
```

Thiago hizo el merge de ambas ramas con main.

Así quedaría la rama main una vez realizados todos los merge.



5. Resolución de conflictos

1. Editar manualmente los archivos con conflictos y elegir qué cambios mantener. No te olvides de la captura de pantalla de esta resolución.
2. Una vez resuelto, añadir los archivos modificados:

git add <archivo>

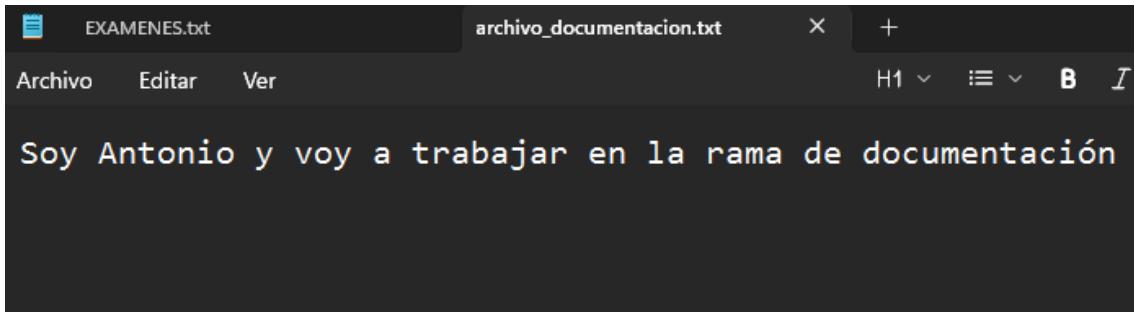
3. Confirmar la resolución del conflicto:

git commit -m "Resuelto conflicto en <archivo>"

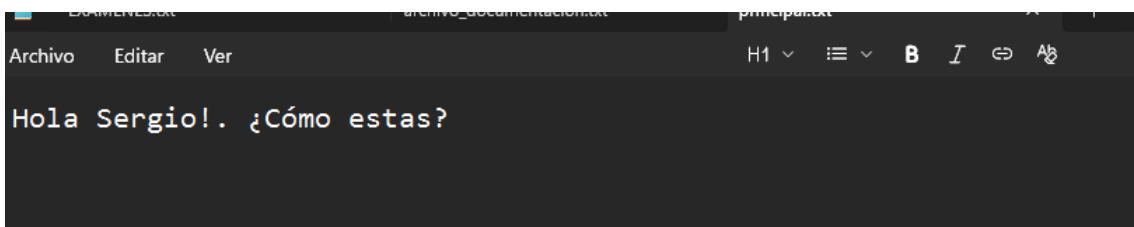
4. Subir los cambios al repositorio remoto:

git push origin main

Archivo de archivo_documentación.txt tenía conflicto y manualmente lo hemos arreglado.



Archivo de principal.txt tenía conflicto y manualmente lo hemos arreglado.



6. Eliminación de ramas innecesarias

Una vez fusionadas todas las ramas en main, se pueden eliminar:

```
git branch -d <nombre-de-la-rama>
```

```
ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (funcionalidad-A)
$ git branch -d documentación
warning: deleting branch 'documentación' that has been merged to
      'refs/remotes/origin/documentación', but not yet merged to HEAD
Deleted branch documentación (was 9d1b053).

ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (funcionalidad-A)
$ git branch
* funcionalidad-A
  main
```

Para eliminar la rama en el remoto:

```
git push origin --delete <nombre-de-la-rama>
```

```
ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (funcionalidad-A)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (main)
$ git push origin --delete documentación
To https://github.com/bcarrascosergio/GitEnEquipo.git
 - [deleted]      documentación

ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (main)
$ |
```

```
ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (main)
$ git branch -d funcionalidad-A
Deleted branch funcionalidad-A (was 02d9d02).

ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (main)
$ git push origin --delete funcionalidad-A
To https://github.com/bcarrascosergio/GitEnEquipo.git
 - [deleted]      funcionalidad-A

ACER@LpTp-Sergi MINGW64 ~/IdeaProjects/GitEnEquipo (main)
$ |
```