

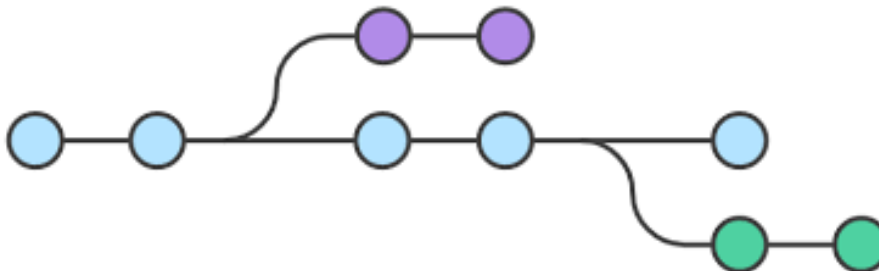
**Objetivo:** Comprender a nivel repositorio como se crean las ramas, la unión(merge) y como resolver conflictos que generan estas.

## Temas

Working with repositories on GitHub(Branches,Merge,Conflicts)

## Introducción

Con ayuda de VSC se crea un archivo .html donde se simula el repositorio main y las ramas paralelas a este mismo.



# Inicializar la rama Master

```
user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git
$ git init
Initialized empty Git repository in C:/Xideral/Tarea1/Git/.git/

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (master)
$ git add .

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (master)
$ git status
On branch master

No commits yet

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

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (master)
$ git commit -m "primer commit"
[master (root-commit) 7df205b] primer commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (master)
$ git branch -M main

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (main)
$ git remote add origin https://github.com/arturo7-7/Git.git

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 211 bytes | 211.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/arturo7-7/Git.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

## Comandos utilizados

`git init` -> Crea el repositorio.

`git add .` -> Alista los los archivos en el área de preparación.

`git status` -> Muestra el estado actual del repositorio.

`git commit -m "mensaje"` -> Compromete los cambios en el repositorio.

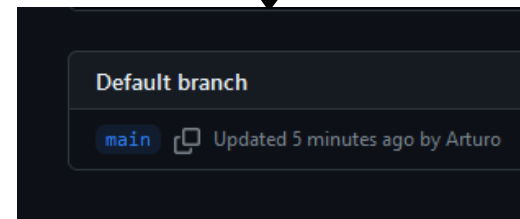
`git branch -M` ->Renombrar la rama actual.

`git remote add origin https://git...` ->Crear repositorio remoto

## Explicación

Se inicializó el repositorio , teniendo como defecto la rama (master), se renombra por "main".

Se crea un repositorio remoto con el URL de mi cuenta personal de Git Hub. El repositorio local se pasa al remoto.



# Cración de Rama

```
To https://github.com/arturo7-7/Git.git
7df205b..ffc3391  main -> main

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (main)
$ git branch experimento

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (main)
$ git checkout
M      index.html
Your branch is up to date with 'origin/main'.

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (main)
$ git add .

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (main)
$ git commit "modificacion en nueva rama"
error: pathspec 'modificacion en nueva rama' did not match any file(s) known to git

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (main)
$ git commit -m "modificacion en nueva rama"
[main 4fb269d] modificacion en nueva rama
1 file changed, 1 insertion(+)
```

```
user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (main)
$ git checkout experimento
Switched to branch 'experimento'

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (experimento)
$
```

## Comandos utilizados

`git branch ramaNombre` -> Genera una nueva rama.  
`git checkout ramaNombre` -> Movernos de rama

## Explicación

Se crea una nueva rama llamada "experimento", se hace la preparación y el commit de cambios en el .html, también se agregaron capturas de pantallas al repositorio conforme se realiza el flujo de comandos.

## Merge

```
user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (main)
$ git merge experimento
Merge made by the 'ort' strategy.
  Movernos_rama_creada.png | Bin 0 -> 11105 bytes
  RamaLocal.png            | Bin 0 -> 47865 bytes
  2 files changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 Movernos_rama_creada.png
  create mode 100644 RamaLocal.png

user@DESKTOP-006N4RP MINGW64 /c/Xideral/Tarea1/Git (main)
$ git branch -d experimento
Deleted branch experimento (was 7cb346c).
```

## Comandos utilizados

`git merge NombreRama` -> Une las ramas

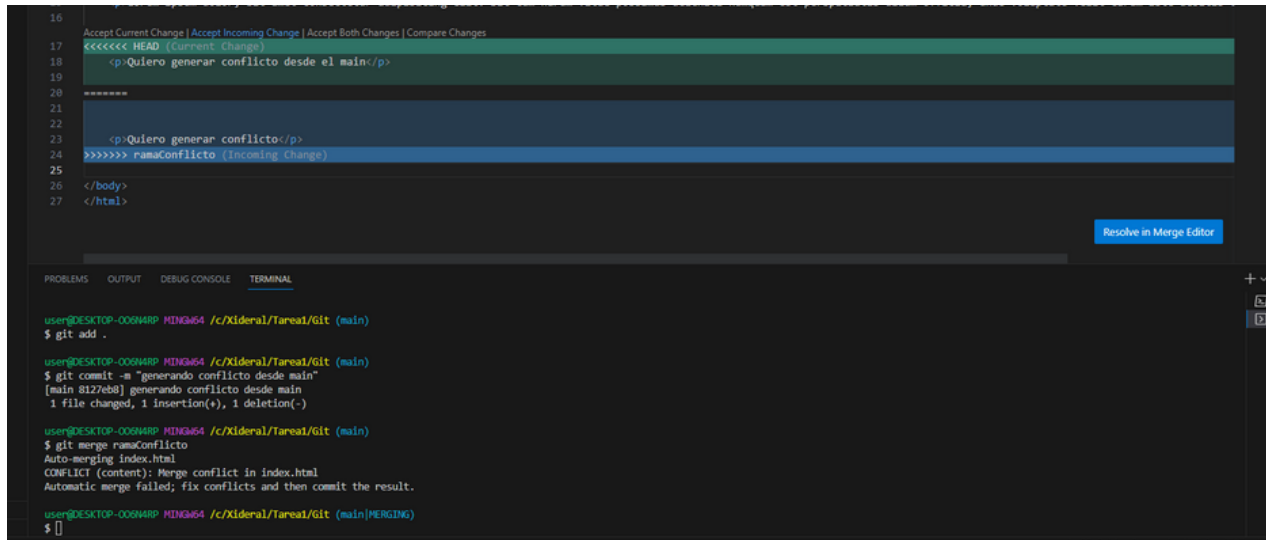
## Explicación

Después de que se ha realizado la unión de las ramas y se ha resuelto cualquier conflicto, es común que se proceda a borrar la rama que se utilizó para la unión.

Beneficios:

- Mantener un árbol de ramas limpio
- Evitar confusión:
- Conservar la estructura del repositorio
- Evitar posibles conflictos futuros

# Resolución de conflicto



```
16
17 Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
18 <----- HEAD (Current Change)
19 <p>Quiero generar conflicto desde el main</p>
20
21 =====
22
23 <p>Quiero generar conflicto</p>
24 >>>>>> ramaConflicto (Incoming Change)
25
26 </body>
27 </html>
```

Resolve in Merge Editor

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
user@DESKTOP-009H4RP MINGW64 /c:/Xideral/Tarea1/Git (main)
$ git add .

user@DESKTOP-009H4RP MINGW64 /c:/Xideral/Tarea1/Git (main)
$ git commit -m "generando conflicto desde main"
[main 8127eb8] generando conflicto desde main
1 file changed, 1 insertion(+), 1 deletion(-)

user@DESKTOP-009H4RP MINGW64 /c:/Xideral/Tarea1/Git (main)
$ git merge ramaConflicto
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.

user@DESKTOP-009H4RP MINGW64 /c:/Xideral/Tarea1/Git (main|MERGING)
$ []
```

El conflicto se produce cuando se intenta editar el mismo elemento en diferentes ramas del proyecto.

Para solucionar estos conflictos, es necesario tomar decisiones claras sobre qué cambios deben mantenerse y cómo se deben combinar las ediciones enfrentadas. Existen varias opciones para resolver los conflictos:

- Aceptar la rama entrante: Esta opción implica tomar todos los cambios realizados en la rama que se está fusionando y descartar los cambios conflictivos de la rama actual.
- Aceptar ambos cambios: En ciertos casos, ambas ediciones de las ramas en conflicto pueden ser relevantes y necesarias para el proyecto.
- Aceptar el cambio concurrente: Cuando las ediciones conflictivas en ambas ramas no son mutuamente excluyentes, sino que pueden coexistir sin problemas, se puede optar por aceptar ambas ediciones sin realizar cambios adicionales.