

Laboratorio Corrección de Conflictos

Laboratorio sencillo

Paso 1: Configurar el entorno

Vamos a configurar el entorno creando las respectivas carpetas e iniciando el nuevo repositorio con el **git-init** y nos metemos dentro con el **cd**

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/Laboratorio_correccion_de_conflictos/laboratorio_sencillo_CDC (master)
$ git init git-laboratorio
Reinitialized existing Git repository in C:/Users/Uli/Downloads/Laboratorios_local/Laboratorios_local/Laboratorio_correccion_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio/.git/

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/Laboratorio_correccion_de_conflictos/laboratorio_sencillo_CDC (master)
$ cd git-laboratorio

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/Laboratorio_correccion_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (master)
```

Ahora solo debemos crear un primer commit, con el **echo**, el **add** y el **commit -m** (como estamos haciendo hasta ahora):

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/Laboratorio_correccion_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (master)
$ echo "Esta es la linea original" > archivo.txt

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/Laboratorio_correccion_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (master)
$ git add archivo.txt
warning: in the working copy of 'archivo.txt', LF will be replaced by CRLF the next time Git touches it

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/Laboratorio_correccion_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (master)
$ git commit -m "Primer commit: archivo.txt creado con una linea inicial"
[master (root-commit) 00ecb7b] Primer commit: archivo.txt creado con una linea inicial
1 file changed, 1 insertion(+)
create mode 100644 archivo.txt

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/Laboratorio_correccion_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (master)
```

Paso 2: Crear ramas para provocar el conflicto

Ahora vamos a proceder a crear una rama con el **checkout -b** :

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorio_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (master)
$ git checkout -b rama1
Switched to a new branch 'rama1'
```

Procedemos a **modificar el archivo.txt** y a guardar los cambios correctamente:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorio_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (rama1)
$ echo "Linea agregada en rama1" >> archivo.txt

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorio_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (rama1)
$ git add archivo.txt
warning: in the working copy of 'archivo.txt', LF will be replaced by
it

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorio_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (rama1)
$ git commit -m "Agregada linea en rama1"
[rama1 ceb15fd] Agregada linea en rama1
1 file changed, 1 insertion(+)
```

Ahora tenemos que volver a la rama **main** para poder crear la **rama2** y guardarla correctamente:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorio_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (rama1)
$ git checkout master
Switched to branch 'master'

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorio_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (master)
$ git checkout -b rama2
Switched to a new branch 'rama2'

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorio_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (rama2)
$ echo "Linea agregada en rama2" >> archivo.txt

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorio_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (rama2)
$ git add archivo.txt
warning: in the working copy of 'archivo.txt', LF will be replaced by
it
```

Paso 3: Provocar el conflicto

Ahora debemos volver a la rama main y realizar un merge a rama1:

(Merge sirve básicamente para migrar los datos guardados de una rama en otra)

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorio_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio
● $ git checkout master
Switched to branch 'master'

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorio_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio
● $ git merge rama1
Updating 00ecb7b..ceb15fd
Fast-forward
 archivo.txt | 1 +
 1 file changed, 1 insertion(+)
```

Podemos comprobar que no existe ningún conflicto entre ellos

En cambio si hacemos merge desde **main** a **rama2**, git nos avisara de un conflicto:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorio_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (master)
⊗ $ git merge rama2
Auto-merging archivo.txt
CONFLICT (content): Merge conflict in archivo.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Paso 4: Resolver el conflicto

Para resolver este conflicto debemos utilizar **git status** para ver el estado de nuestro repositorio, si está guardando correctamente o falta algo por guardar o arreglar:

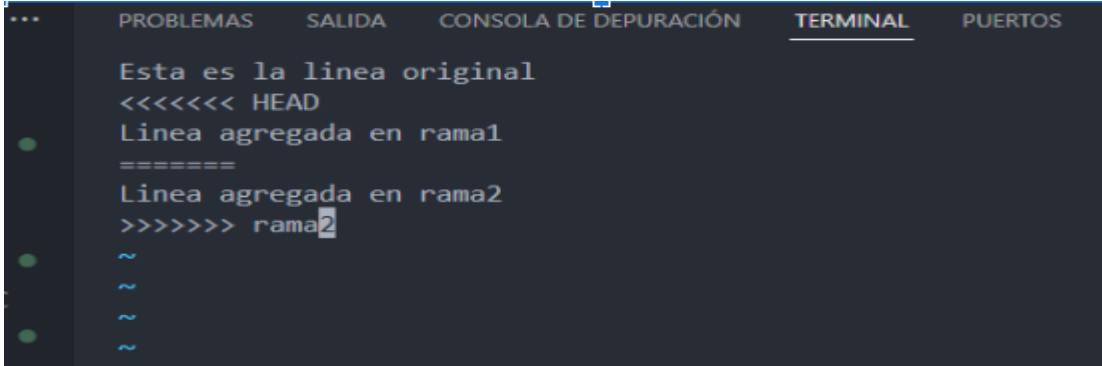
```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorio_de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (master|MERC)
● $ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   archivo.txt

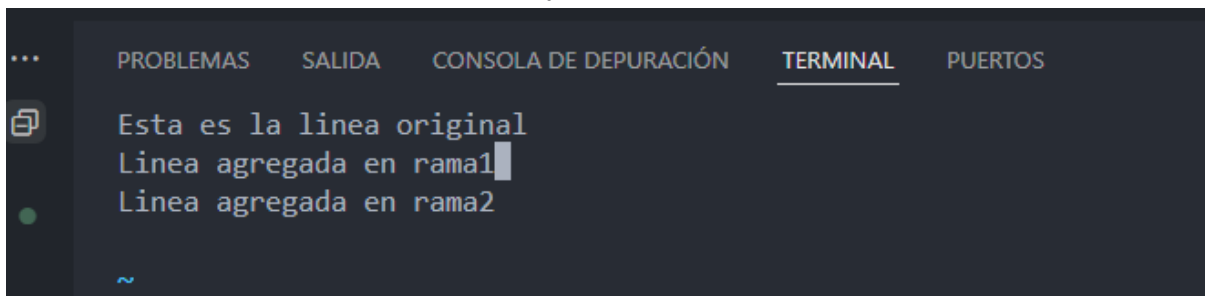
no changes added to commit (use "git add" and/or "git commit -a")
```

Ahora vamos a abrir el **archivo.txt** utilizando el **vim** y comprobaremos el estado del archivo.txt:

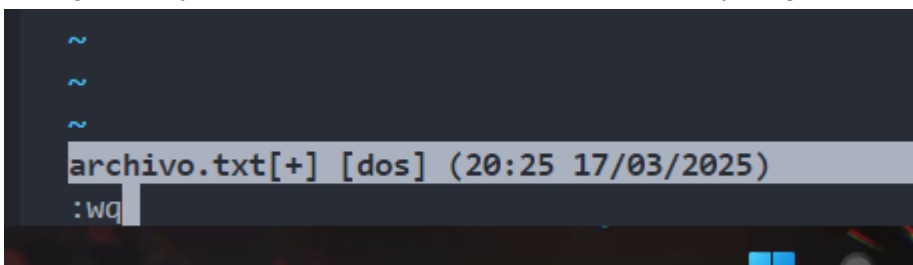
```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_1
de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (master|MERGING)
● $ vim archivo.txt
```



Ahora debemos modificar el **archivo.txt** y para poder manipularlo deberemos utilizar **i**:



Para guardar y salir del archivo usaremos la tecla **ESC** y luego **:wq**



Ahora solo debemos añadir y guardar el conflicto resuelto entre **rama1** y **rama2**:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_1
de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (master|MERGING)
● $ git add archivo.txt

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_1
de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (master|MERGING)
● $ git commit -m "Conflicto resuelto entre rama1 y rama2"
[master fd92bb7] Conflicto resuelto entre rama1 y rama2
```

Paso 5: Verificación

Vamos a verificar como han quedado como han quedado las ramas después de los merges utilizando el **git log --graph --oneline**:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/
de_conflictos/laboratorio_sencillo_CDC/git-laboratorio (master)
$ git log --graph --oneline
* fd92bb7 (HEAD -> master) Conflicto resuelto entre rama1 y rama2
|\
| * b84ede8 (rama2) Agregada linea en rama2
* | ceb15fd (rama1) Agregada linea en rama1
|/
* 00ecb7b Primer commit: archivo.txt creado con una linea inicial
```

Paso 6: Limpiar

Por último eliminamos las ramas con el **git branch -d**

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratori
de_conflictos/laboratorio_sencillo_CDC/git-laborat
$ git branch -d rama1
Deleted branch rama1 (was ceb15fd).

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratori
de_conflictos/laboratorio_sencillo_CDC/git-laborat
$ git branch -d rama2
Deleted branch rama2 (was b84ede8).
```

Con este laboratorio, he visto y experimentado la resolución de conflictos en Git de una manera práctica

Laboratorio avanzado

Paso 1: Configurar el entorno

Vamos a configurar el entorno del siguiente laboratorio utilizando el **git init** y metiéndonos dentro del repositorio:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorio
de_conflictos/laboratorio_avanzado_CDC
● $ git init git-laboratorio-avanzado
Initialized empty Git repository in C:/Users/Uli/Do
ratorio_correccion_de_conflictos/laboratorio_avanza

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorio
de_conflictos/laboratorio_avanzado_CDC
● $ cd git-laboratorio-avanzado/

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorio
de_conflictos/laboratorio_avanzado_CDC/git-laborato
○ $
```

Ahora vamos a proceder a crear el **archivo-complejo.txt** con varias líneas:

Realizamos los mismos pasos de siempre, con la diferencia del **echo -e**, este “**-e**” sirve para **detectar los saltos de línea de un texto escrito en varias líneas**:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/Laboratorio_co
de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (master)
● $ echo -e "Línea 1: Este es el inicio\nLínea 2: Trabajo in
icial\nLínea 3: Pendiente de cambios\nLínea 4: Más traba
jo pendiente\nLínea 5: Fin del archivo" > archivo-complejo.txt

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/Laboratorio_co
de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (master)
● $ git add archivo-complejo.txt
warning: in the working copy of 'archivo-complejo.txt', LF will be replaced by CRLF the next
touches it

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/Laboratorio_co
de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (master)
● $ git commit -m "Primer commit: archivo-complejo.txt creado con varias lineas"
[master (root-commit) 1409381] Primer commit: archivo-complejo.txt creado con varias lineas
1 file changed, 7 insertions(+)
create mode 100644 archivo-complejo.txt
```

Paso 2: Crear las ramas y hacer los cambios

Vamos a crear la ramaA y modificar el archivo-complejo.txt:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorio
de_conflictos/laboratorio_avanzado_CDC/git-laborato
$ git checkout -b ramaA
Switched to a new branch 'ramaA'

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorio
de_conflictos/laboratorio_avanzado_CDC/git-laborato
● $ vim archivo-complejo.txt
```

Ya lo he modificado:

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

sed -i 's/Trabajo inicial/Trabajo realizado en ramaA/' a
rchivo-complejo.txt
sed -i 's/Pendiente de cambios/Trabajo completado en ram
aA/' archivo-complejo.txt
~
~
```

Y procedemos a guardar los cambios correctamente:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_1
de_conflictos/laboratorio_avanzado_CDC/git-laboratorio
● $ git add archivo-complejo.txt
warning: in the working copy of 'archivo-complejo.txt'
touches it

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_1
de_conflictos/laboratorio_avanzado_CDC/git-laboratorio
● $ git commit -m "Modificado archivo en ramaA"
[ramaA 024953a] Modificado archivo en ramaA
1 file changed, 4 insertions(+), 7 deletions(-)
```

Volvemos a la rama **master** y creamos y modificamos la **ramaB**:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_1
de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (ramaB)
● $ git checkout master
Switched to branch 'master'
```

Nos metemos con el **vim** y lo modificamos:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_1
de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (ramaB)
● $ vim archivo-complejo.txt
```

```
... PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
Línea 1: Este es el inicio
Línea 2: Trabajo in
icial
● Línea 3: Pendiente de cambios
Línea 4: Más traba
jo pendiente
... ● Línea 5: Fin del archivo
... ● ~
```

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
sed -i 's/Trabajo inicial/Trabajo realizado en ramaB/' a
rchivo-complejo.txt
sed -i 's/Pendiente de cambios/Trabajo completado en ram
aB/' archivo-complejo.txt
sed -i 's/Más trabajo pendiente/Tarea adicional completa
da en ramaB/' archivo-complejo.txt
~
```

Ahora solo falta guardar los cambios:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_1
de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (ramaB)
● $ git add archivo-complejo.txt

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_1
de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (ramaB)
● $ git commit -m "Modificado archivo en ramaB"
[ramaB ffad86b] Modificado archivo en ramaB
1 file changed, 6 insertions(+), 7 deletions(-)
```


Paso 3: Provocar múltiples conflictos

Volvemos de nuevo a la rama **master**:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local
de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado
$ git checkout master
Switched to branch 'master'
```

Y hacemos **merge** en **ramaA**:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local
de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (master)
$ git merge ramaA
Updating 1409381..024953a
Fast-forward
 archivo-complejo.txt | 11 ++++-----
 1 file changed, 4 insertions(+), 7 deletions(-)
```

Vemos que esto **no causará conflictos**, ya que solo estamos combinando **ramaA** con **master**

En cambio si ahora hacemos **merge** en **ramaB** vemos que **si se generan conflictos**:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local
de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (master)
$ git merge ramaB
Auto-merging archivo-complejo.txt
CONFLICT (content): Merge conflict in archivo-complejo.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Paso 4: Revisar y resolver los conflictos

Primero vamos a revisar el repositorio con el **git status** y veremos que Git nos avisa de que el **archivo-complejo.txt** tiene conflictos:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/Laboratorio_de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   archivo-complejo.txt

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

Veamos el conflicto con el **vim**:

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS
<<<<<<< HEAD
sed -i 's/Trabajo inicial/Trabajo realizado en ramaA/' a
rchivo-complejo.txt
sed -i 's/Pendiente de cambios/Trabajo completado en ram
aA/' archivo-complejo.txt
=====
sed -i 's/Trabajo inicial/Trabajo realizado en ramaB/' a
rchivo-complejo.txt
sed -i 's/Pendiente de cambios/Trabajo completado en ram
aB/' archivo-complejo.txt
sed -i 's/Más trabajo pendiente/Tarea adicional completa
da en ramaB/' archivo-complejo.txt
>>>>>>> ramaB
~
```

Procedamos a arreglarlo manualmente:

```
...  PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

Línea 1: Este es el inicio
Línea 2: Trabajo realizado en ramaA y ramaB
● Línea 3: Trabajo completado en ramaA y ramaB
Línea 4: Tarea adicional completada en ramaB
Línea 5: Fin del archivo
~
~
```

Ahora procedemos a guardar los cambios:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (master)
● $ git add archivo-complejo.txt

Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (master)
● $ git commit -m "Conflictos resueltos entre ramaA y ramaB"
[master 7d19842] Conflictos resueltos entre ramaA y ramaB
```

Paso 5: Verificación y limpieza

Verificamos con el **git log**:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (master)
● $ git log --graph --oneline
* 7d19842 (HEAD -> master) Conflictos resueltos entre ramaA y ramaB
| \
| * ffad86b (ramaB) Modificado archivo en ramaB
* | 024953a (ramaA) Modificado archivo en ramaA
| /
* 1409381 Primer commit: archivo-complejo.txt creado con varias líneas
```

Por último borramos las ramas con el **git branch -d**:

```
Uli@DESKTOP-0330PVB MINGW64 ~/Downloads/Laboratorios_local/Laboratorios_local/de_conflictos/laboratorio_avanzado_CDC/git-laboratorio-avanzado (master)
● $ git branch -d ramaA && git branch -d ramaB
Deleted branch ramaA (was 024953a).
Deleted branch ramaB (was ffad86b).
```

Este laboratorio simula un escenario real en el que múltiples desarrolladores hacen cambios en distintas partes de un archivo y cómo resolver los conflictos resultantes.

Los conflictos están distribuidos en varias líneas, lo que hace que sea más interesante y cercano a situaciones del mundo real.

Este laboratorio te ayudará a fortalecer tus habilidades para manejar conflictos más complejos en Git.