

Ejemplos

1. Fusión Fast-forward (git merge --ff)

Pasos Prácticos:

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-fast-forward-merge> git init
Initialized empty Git repository in C:/Users/Camila Lopez/Desktop/6to/prueba-fast-forward-merge/.git/
PS C:\Users\Camila Lopez\Desktop\6to\prueba-fast-forward-merge> echo "# Mi Proyecto" > README.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-fast-forward-merge> git add README.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-fast-forward-merge> git commit -m "Commit inicial en main"
[main (root-commit) 5e43c5b] Commit inicial en main
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-fast-forward-merge> git checkout -b add-description
Switched to a new branch 'add-description'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-fast-forward-merge> echo "Este proyecto es un ejemplo de cómo usar Git." >> README.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-fast-forward-merge> git add README.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-fast-forward-merge> git commit -m "Agregar descripción al README.md"
[add-description 27be7e6] Agregar descripción al README.md
1 file changed, 0 insertions(+), 0 deletions(-)
```

Pregunta:

Desde la rama main se realiza un merge con la rama add-description, que por defecto es fast forward

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-fast-forward-merge> git checkout main
Switched to branch 'main'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-fast-forward-merge> git merge add-description
Updating 5e43c5b..27be7e6
Fast-forward
 README.md | Bin 32 -> 126 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
```

Ahora el puntero main apunta directamente al commit 27be7e6, que es el último commit de add-description.

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-fast-forward-merge> git log --graph --oneline
* 27be7e6 (HEAD -> main, add-description) Agregar descripción al README.md
* 5e43c5b Commit inicial en main
```

2. Fusión No-fast-forward (git merge --no-ff)

Pasos prácticos:

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-no-fast-forward-merge> git init
Initialized empty Git repository in C:/Users/Camila Lopez/Desktop/6to/prueba-no-fast-forward-merge/.git/
PS C:\Users\Camila Lopez\Desktop\6to\prueba-no-fast-forward-merge> echo "# Mi Proyecto" > README.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-no-fast-forward-merge> git add README.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-no-fast-forward-merge> git commit -m "Commit inicial en main"
[main (root-commit) de222e3] Commit inicial en main
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
```

```

PS C:\Users\Camila Lopez\Desktop\6to\prueba-no-fast-forward-merge> git checkout -b add-feature
Switched to a new branch 'add-feature'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-no-fast-forward-merge> echo "Implementando una nueva característica.." >> README.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-no-fast-forward-merge> git add README.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-no-fast-forward-merge> git commit -m "Implementar nueva característica"
[add-feature f03e3ea] Implementar nueva característica
1 file changed, 0 insertions(+), 0 deletions(-)

```

Pregunta:

Desde la rama main se realiza un merge no fast forward con la rama add-description

```

PS C:\Users\Camila Lopez\Desktop\6to\prueba-no-fast-forward-merge> git checkout main
Switched to branch 'main'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-no-fast-forward-merge> git merge --no-ff add-feature
Merge made by the 'ort' strategy.
 README.md | Bin 32 -> 116 bytes
1 file changed, 0 insertions(+), 0 deletions(-)

```

```

Merge branch 'add-feature'
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.

```

Debido al merge se genera un commit de fusión.

```

PS C:\Users\Camila Lopez\Desktop\6to\prueba-no-fast-forward-merge> git log --graph --oneline
* 182f9e0 (HEAD -> main) Merge branch 'add-feature'
|
| * f03e3ea (add-feature) Implementar nueva característica
|/
* de222e3 Commit inicial en main

```

3. Fusión squash (git merge --squash)

Pasos prácticos:

```

PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> git init
Initialized empty Git repository in C:/Users/Camila Lopez/Desktop/6to/prueba-squash-merge/.git/
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> echo "# Mi Proyecto" > README.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> git add README.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> git commit -m "Commit inicial en main"
[main (root-commit) 7de7eb3] Commit inicial en main
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> git checkout -b add-basic-files
Switched to a new branch 'add-basic-files'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> echo "# CÓMO CONTRIBUIR" >> CONTRIBUTING.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> git add CONTRIBUTING.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> git commit -m "Agregar CONTRIBUTING.md"
[add-basic-files 48eae6d] Agregar CONTRIBUTING.md
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 CONTRIBUTING.md
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> echo "# LICENCIA" >> LICENSE.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> git add LICENSE.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> git commit -m "Agregar LICENSE.txt"
[add-basic-files 701ef0f] Agregar LICENSE.txt
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 LICENSE.txt

```

Se realizó un merge squash en main con la rama add-basic-files

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> git checkout main
Switched to branch 'main'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> git merge --squash add-basic-files
Updating 7de7eb3..701ef0f
Fast-forward
Squash commit -- not updating HEAD
CONTRIBUTING.md | Bin 0 -> 40 bytes
LICENSE.txt      | Bin 0 -> 26 bytes
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 CONTRIBUTING.md
create mode 100644 LICENSE.txt

PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> git add .
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> git commit -m "Agregar documentación estándar del repositorio"
[main d15e36f] Agregar documentación estándar del repositorio
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 CONTRIBUTING.md
create mode 100644 LICENSE.txt
```

Visualizamos de manera gráfica como los cambios en la rama add-basic-files se agruparon en uno solo y luego se aplicaron a la rama main

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-squash-merge> git log --graph --oneline
* d15e36f (HEAD -> main) Agregar documentación estándar del repositorio
* 7de7eb3 Commit inicial en main
```

Ejercicios

Identificamos las ramas en nuestro repositorio repoPrueba

```
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git branch
* main
nuevaRama1
```

Visualizamos el historial de commits en cada rama

```
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git log --graph
* commit fff05f77846b9e2f03564953c4e0dd3c5fbf8979 (HEAD -> main)
  Author: Ariana Lopez <ariana.lopez.j@uni.pe>
  Date: Tue Apr 8 23:56:34 2025 -0500
```

Prueba n1

```
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git checkout nuevaRama1
Switched to branch 'nuevaRama1'
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git log --graph
* commit 5a0d32cbd1094dedf7923ee2e670763a80e6c38d (HEAD -> nuevaRama1)
  Author: Ariana Lopez <ariana.lopez.j@uni.pe>
  Date: Tue Apr 8 23:57:34 2025 -0500
```

Prueba n2

```
* commit fff05f77846b9e2f03564953c4e0dd3c5fbf8979 (main)
  Author: Ariana Lopez <ariana.lopez.j@uni.pe>
  Date: Tue Apr 8 23:56:34 2025 -0500
```

Prueba n1

Observando notamos que es posible fusionar la rama main y nuevaRama1 con fast forward, debido a que no hay conflictos en ninguno de los archivos.

```
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git merge --ff nuevaRama1
Updating fff05f7..5a0d32c
Fast-forward
 prueba.txt | Bin 18 -> 46 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git log --graph --oneline
* 5a0d32c (HEAD -> main, nuevaRama1) Prueba n2
* fff05f7 Prueba n1
```

Pregunta:

- ¿En qué situaciones recomendarías evitar el uso de git merge --ff? Reflexiona sobre las desventajas de este método.

En situaciones de equipos colaborativos, es esencial mantener el historial de los commits. Esta información se puede perder usando --ff

Trabajamos en 2 ramas, main y nuevaRama2, y realizamos cambios en cada una.

```
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git checkout -b nuevaRama2
Switched to a new branch 'nuevaRama2'
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> echo "Penultima Prueba">>prueba.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git add .
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git commit -m "Prueba n3"
[nuevaRama2 5f3d13d] Prueba n3
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git checkout main
Switched to branch 'main'
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> echo "Ultima Prueba">>prueba.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git add .
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git commit -m "Prueba n4"
[main 6c683c8] Prueba n4
 1 file changed, 0 insertions(+), 0 deletions(-)
```

Luego, fusionamos ambas ramas con git merge --no-ff

```
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git merge --no-ff nuevaRama2
warning: Cannot merge binary files: prueba.txt (HEAD vs. nuevaRama2)
Auto-merging prueba.txt
```

```
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> code prueba.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git add prueba.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git commit
[main 04de051] Merge branch 'nuevaRama2'
```

Arreglamos los conflictos manualmente

```
Result C:\Users\Camila Lopez\Desktop\6to\repoPrueba\prueba.txt
1 Prueba
2 Nueva Prueba
 Incoming + Current | Remove Incoming | Remove Current
3 Penultima Prueba
4 Ultima Prueba
5
```

Visualizamos el historial de commits, podemos ver como la línea de nuevaRama2 converge a la línea de la rama main en el último commit 04de051

```
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git log --graph --oneline
* 04de051 (HEAD -> main) Merge branch 'nuevaRama2'
| \
| * 5f3d13d (nuevaRama2) Prueba n3
* | 6c683c8 Prueba n4
|/
* 5a0d32c (nuevaRama1) Prueba n2
* fff05f7 Prueba n1
```

Pregunta:

- ¿Cuáles son las principales ventajas de utilizar git merge --no-ff en un proyecto en equipo?
Preserva el historial de commits, tanto los padres del commit como el cambio realizado.
- ¿Qué problemas podrían surgir al depender excesivamente de commits de fusión?
El historial puede volverse complejo y poco legible, estos commits de merge pueden llegar a ser muy ruidosos

Se hicieron varios commits en la rama de feature

```
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git checkout -b feature
Switched to a new branch 'feature'
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> echo "Autor: Ariana Lopez">prueba.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git add .
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git commit -m "Añadir Autor"
[feature 551137b] Añadir Autor
1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> echo "Fecha: 9 de abril">prueba.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git add .
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git commit -m "Añadir Fecha"
[feature 00e1305] Añadir Fecha
1 file changed, 0 insertions(+), 0 deletions(-)
```

Visualizamos el historial de los commits hasta el momento.

```
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git checkout feature
Switched to branch 'feature'
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git log --graph --oneline
* 00e1305 (HEAD -> feature) Añadir Fecha
* 551137b Añadir Autor
* 8f82c7c (main) Merge branch 'nuevaRama2'
| \
| * 5f3d13d (nuevaRama2) Prueba n3
* | 6c683c8 Prueba n4
|/
* 5a0d32c (nuevaRama1) Prueba n2
* fff05f7 Prueba n1
```

Fusionamos la rama main con feature-3 con squash. Luego, visualizamos su historial y notamos que en lugar de los 2 commits que aparecían antes ahora aparece solo uno, el

commit merge.

```
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git checkout main
Switched to branch 'main'
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git merge --squash feature
Updating 8f82c7c..00e1305
Fast-forward
Squash commit -- not updating HEAD
 prueba.txt | Bin 112 -> 192 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Camila Lopez\Desktop\6to\repoPrueba> git log --graph --oneline
* 8f82c7c (HEAD -> main) Merge branch 'nuevaRama2'
|
| * 5f3d13d (nuevaRama2) Prueba n3
| * 6c683c8 Prueba n4
|/
* 5a0d32c (nuevaRama1) Prueba n2
* fff05f7 Prueba n1
```

Preguntas:

- ¿Qué método prefieres en diferentes escenarios y por qué?
En proyectos pequeños o personales prefiero el fast forward, ya que permite mantener un historial lineal y limpio.

Resolver conflictos en una fusión non-fast-forward

En un nuevo repositorio, creamos un archivo index.html y realizamos un commit en main

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> git init
Initialized empty Git repository in C:/Users/Camila Lopez/Desktop/6to/prueba-mege-conflict/.git/
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> echo "<html><body><h1>Proyecto inicial CC3S
2</h1></body></html>" > index.html
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> git add index.html
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> git commit -m "commit inicial del index.htm
l en main"
[main (root-commit) 6376b8f] commit inicial del index.html en main
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html
```

En una nueva rama feature-update editamos el archivo y realizamos un commit en la rama actual.

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> git checkout -b feature-update
Switched to a new branch 'feature-update'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> echo "<p>....</p>" >> index.html
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> git add index.html
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> git commit -m "Actualiza el contenido de in
dex.html"
[feature-update 640969c] Actualiza el contenido de index.html
 1 file changed, 0 insertions(+), 0 deletions(-)
```

Regresamos a main y volvemos a hacer un cambio al archivo.

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> git checkout main
Switched to branch 'main'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> echo "<footer>Contacta aquí example@example
.com</footer>" >> index.html
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> git add index.html
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> git commit -m "Agrega un footer a index.htm
l"
[main 4653443] Agrega un footer a index.html
 1 file changed, 0 insertions(+), 0 deletions(-)
```

Fusionamos ambas ramas con no fast forward, después de resolver los conflictos.

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> git merge --no-ff feature-update
warning: Cannot merge binary files: index.html (HEAD vs. feature-update)
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

```
Result C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict\index.html
1  <html><body><h1>Proyecto inicial CC3S2</h1></body></html>
   Incoming + Current | Remove Incoming | Remove Current
2  <p>.....</p>
3  <footer>Contacta aquí example@example.com</footer>
4
```

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> git add index.html
```

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> git commit
[main a595d15] Merge branch 'feature-update'
```

```
Merge branch 'feature-update'

# Conflicts:
#   index.html
#
```

Verificamos el historial

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-mege-conflict> git log --graph --oneline
* a595d15 (HEAD -> main) Merge branch 'feature-update'
|
| * 640969c (feature-update) Actualiza el contenido de index.html
| * 4653443 Agrega un footer a index.html
|/
* 6376b8f commit inicial del index.html en main
```

Preguntas:

- ¿Qué pasos adicionales tuviste que tomar para resolver el conflicto?
Tuve que abrir el archivo en conflicto y resolver dicho conflicto en el editor.
Luego tuve que añadir la versión corregida del archivo con git add .
luego hacer un git commit para poder finalizar el merge
- ¿Qué estrategias podrías emplear para evitar conflictos en futuros desarrollos colaborativos?
evitar realizar cambios en un mismo archivo en diferentes ramas

Ejercicio: Comparar los historiales con git log después de diferentes fusiones

Pasos prácticos:

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git init
Initialized empty Git repository in C:/Users/Camila Lopez/Desktop/6to/prueba-compare-merge/.git/
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> echo "Version 1.0" > version.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git add version.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git commit -m "Versión agregada"
```

Se tienen 2 ramas cada una con cambios respecto al archivo version.txt

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git checkout -b feature-1
Switched to a new branch 'feature-1'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> echo "Característica 1 agregada" >> version.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git add version.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git commit -m "Agregar característica 1"
[feature-1 0fb6546] Agregar característica 1
1 file changed, 0 insertions(+), 0 deletions(-)
```

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git checkout main
Switched to branch 'main'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git checkout -b feature-2
Switched to a new branch 'feature-2'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> echo "Característica 2 agregada" >> version.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git add version.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git commit -m "Se agrega característica 2"
[feature-2 aa6f5d8] Se agrega característica 2
1 file changed, 0 insertions(+), 0 deletions(-)
```

Fusionamos feature-1 usando fast-forward y feature-2 usando non-fast-forward

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git checkout main
Switched to branch 'main'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git merge feature-1 --ff
Updating e8d0c47..0fb6546
Fast-forward
 version.txt | Bin 28 -> 82 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git merge feature-2 --no-ff
warning: Cannot merge binary files: version.txt (HEAD vs. feature-2)
Auto-merging version.txt
CONFLICT (content): Merge conflict in version.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Los conflictos generados en el merge se arreglan de manera manual

```
Result C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge\version.txt
1  Version 1.0
   Current + Incoming | Remove Current | Remove Incoming
2  Característica 1 agregada
3  Característica 2 agregada
4
```

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> code version.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git add version.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git commit
[main 69063fc] Merge branch 'feature-2'
```

Creamos una rama feature-3, se realizan cambios añadidos y guardados.


```

PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git checkout -b feature-3
Switched to a new branch 'feature-3'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> echo "Caracteristica 3 paso 1" >> version.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git add version.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git commit -m "Caracteristica 3 paso 1"
[feature-3 0b5b044] Caracteristica 3 paso 1
1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> echo "Caracteristica 3 paso 2" >> version.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git add version.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git commit -m "Caracteristica 3 paso 2"
[feature-3 f2a428b] Caracteristica 3 paso 2
1 file changed, 0 insertions(+), 0 deletions(-)

```

Se fusiona la rama main con la rama feature-3 con el método squash, es decir agrupa todos los commits de feature-3 antes aplicarlo en el main.

```

PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git checkout main
Switched to branch 'main'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git merge --squash feature-3
Updating 69063fc..f2a428b
Fast-forward
Squash commit -- not updating HEAD
version.txt | Bin 136 -> 236 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git commit -m "Agregar caracteristica 3 en un commit"
[main 84c617d] Agregar caracteristica 3 en un commit
1 file changed, 0 insertions(+), 0 deletions(-)

```

Visualizamos el historial de commits de manera gráfica

```

PS C:\Users\Camila Lopez\Desktop\6to\prueba-compare-merge> git log --graph --oneline --all
* 84c617d (HEAD -> main) Agregar caracteristica 3 en un commit
* f2a428b (feature-3) Caracteristica 3 paso 2
* 0b5b044 Caracteristica 3 paso 1
/
* 69063fc Merge branch 'feature-2'
\
|
| * aa6f5d8 (feature-2) Se agrega caracteristica 2
* | 0fb6546 (feature-1) Agregar caracteristica 1
|/
* e8d0c47 Versión agregada

```

Ejercicio: Usando fusiones automáticas y revertir fusiones

Pasos prácticos:

```

PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git init
Initialized empty Git repository in C:/Users/Camila Lopez/Desktop/6to/prueba-auto-merge/.git/
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> echo "Linea 1" > file.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git add file.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git commit -m "Agrega linea 1"
[main (root-commit) 1ec35ad] Agrega linea 1
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> echo "Linea 2" >> file.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git add file.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git commit -m "Agrega linea 2"
[main 88cbbc4] Agrega linea 2
1 file changed, 0 insertions(+), 0 deletions(-)

```

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git checkout -b auto-merge
Switched to a new branch 'auto-merge'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> echo "Linea 3" >> file.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git add file.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git commit -m "Agrega linea 3"
[auto-merge 291d62d] Agrega linea 3
1 file changed, 0 insertions(+), 0 deletions(-)
```

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git checkout main
Switched to branch 'main'
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> echo "Footer: Fin del archivo" >> file.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git add file.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git commit -m "Add footer al archivo file.txt"
```

Resolvemos el conflicto en un archivo de manera manual

Result C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge\file.txt

```
1 Linea 1
2 Linea 2
Incoming + Current | Remove Incoming | Remove Current
3 Linea 3
4 Footer: Fin del archivo
5
```

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> code file.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git add file.txt
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git commit
[main 159942c] Merge branch 'auto-merge'
```

Revertimos el último commit, tomado como base el primer padre main

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git revert -m 1 HEAD
[main 3fcd067] Revert "Merge branch 'auto-merge'"
1 file changed, 0 insertions(+), 0 deletions(-)
```

```
PS C:\Users\Camila Lopez\Desktop\6to\prueba-auto-merge> git log --graph --oneline
* 3fcd067 (HEAD -> main) Revert "Merge branch 'auto-merge'"
* 159942c Merge branch 'auto-merge'
| \
| * 291d62d (auto-merge) Agrega linea 3
* | 9cf81c6 Add footer al archivo file.txt
| /
* 88cbbc4 Agrega linea 2
* 1ec35ad Agrega linea 1
```

Preguntas:

- ¿Cuándo usarías un comando como git revert para deshacer una fusión?
Cuando se cometió algún error en el merge ya realizado, entonces se aplica git reset para borrar del historial el commit del merge
- ¿Qué tan útil es la función de fusión automática en Git?
Es útil ya que permite fusionar ramas de manera automática de cumplirse con las condiciones, lo que evita la resolución de conflictos manual.

Ejercicio: Fusión remota en un repositorio colaborativo

Pasos prácticos:


Se clonó el repositorio personal del curso CC3S2. Luego, se creó un rama colaboración, y dentro de ella un archivo colaboración.txt. Los cambios se añadieron y se guardaron mediante un commit.

```
PS C:\Users\Camila Lopez\Desktop\6to> git clone https://github.com/aclj20/CC3S2.git
Cloning into 'CC3S2'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 21 (delta 7), reused 15 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (21/21), 1.99 MiB | 661.00 KiB/s, done.
Resolving deltas: 100% (7/7), done.
PS C:\Users\Camila Lopez\Desktop\6to> cd CC3S2
PS C:\Users\Camila Lopez\Desktop\6to\CC3S2> git checkout -b colaboración
Switched to a new branch 'colaboración'
PS C:\Users\Camila Lopez\Desktop\6to\CC3S2> echo "Colaboración remota" > colaboración.txt
PS C:\Users\Camila Lopez\Desktop\6to\CC3S2> git add colaboracion.txt
PS C:\Users\Camila Lopez\Desktop\6to\CC3S2> git commit -m "Agregar archivo de colaboración"
[colaboración a6bd735] Agregar archivo de colaboración
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 colaboracion.txt
```

Se empujo los cambios a la rama remota

```
PS C:\Users\Camila Lopez\Desktop\6to\CC3S2> git push origin colaboración
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), 327 bytes | 327.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'colaboración' on GitHub by visiting:
remote:   https://github.com/aclj20/CC3S2/pull/new/colaboraci%C3%B3n
remote:
To https://github.com/aclj20/CC3S2.git
 * [new branch]      colaboración -> colaboración
```

Desde la interfaz de github, creamos un pull request para realizar la fusión entre ramas

 **colaboración** had recent pushes 2 seconds ago

[Compare & pull request](#)

Commit message

Merge pull request #2 from aclj20/colaboración

Extended description

Agregar archivo y rama de colaboración

This commit will be authored by ariana.lopez.j@uni.pe.

Confirm merge **Cancel**

Pull request successfully merged and closed

You're all set — the `colaboración` branch can be safely deleted.

Delete branch

Preguntas:

- ¿Cómo cambia la estrategia de fusión cuando colaboras con otras personas en un repositorio remoto?
Cuando colaboramos con más gente es importante mantener un historial claro de cuándo y cómo se realizó una fusión, por ello se prefieren usar herramientas como git merge no fast forward en lugar de git fast forward.
- ¿Qué problemas comunes pueden surgir al integrar ramas remotas?
Pueden surgir conflictos en archivos que han sido modificados en distintas ramas.

Ejercicio final

Se creó un directorio y un archivo inicial llamado bienvenida.txt

```
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git init
Initialized empty Git repository in C:/Users/Camila Lopez/Desktop/6to/repoProyecto/.git/
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> echo "Bienvenido(a)">bienvenida.txt
```

```
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git add bienvenida.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git commit -m "Agrega bienvenida"
[main (root-commit) b63a56d] Agrega bienvenida
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 bienvenida.txt
```

Se creó una nueva rama feature1 y luego se hicieron algunas modificaciones.

```
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git checkout -b feature1
Switched to a new branch 'feature1'
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> echo "Buenos días">>bienvenida.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git add bienvenida.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git commit -m "Agrega buenos días"
[feature1 62efc9a] Agrega buenos días
1 file changed, 0 insertions(+), 0 deletions(-)
```

Fusionamos feature1 con main utilizando git merge --ff

```
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git merge --ff feature1
Updating b63a56d..62efc9a
Fast-forward
 bienvenida.txt | Bin 32 -> 58 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
```

Se creó una nueva rama feature2 y luego se hicieron algunas modificaciones.

```
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git checkout -b feature2
b63a56d
Switched to a new branch 'feature2'
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> echo "Buenos tardes">>bienvenida.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git add bienvenida.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git commit -m "Agrega buenas tardes"
[feature2 64bc9e5] Agregas buenas tardes
1 file changed, 0 insertions(+), 0 deletions(-)
```

Fusionamos feature2 con main utilizando git merge --no-ff

```
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git checkout main
Switched to branch 'main'
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git merge --no-ff feature2
warning: Cannot merge binary files: bienvenida.txt (HEAD vs. feature2)
Auto-merging bienvenida.txt
CONFLICT (content): Merge conflict in bienvenida.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Resolvemos los conflictos manuales

```
Result C:\Users\Camila Lopez\Desktop\6to\repoProyecto\bienvenida.txt
1 Bienvenido(a)
  Current + Incoming | Remove Current | Remove Incoming
2 Buenos días
3 Buenos tardes
4
```

```
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> code bienvenida.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git add .
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git commit
[main c909b79] Merge branch 'feature2'
```

Se creó una rama feature3 e hizo modificaciones


```

PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git checkout -b feature3
Switched to a new branch 'feature3'
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> echo "Buenas Noches">>bienvenida.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git add .
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git commit -m "Agrega buenas noches"
[feature3 08e006a] Agregas buenas noches
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> echo "Fin">>bienvenida.txt
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git add .
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git commit -m "Agrega Fin"
[feature3 ed0196a] Agregas Fin
 1 file changed, 0 insertions(+), 0 deletions(-)

```

Luego, utiliza squash para combinar 2 commit o más en uno solo de la rama dentro de rama en la que te encuentres actualmente

```

PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git checkout main
Switched to branch 'main'
PS C:\Users\Camila Lopez\Desktop\6to\repoProyecto> git merge --squash feature3
Updating c909b79..ed0196a
Fast-forward
Squash commit -- not updating HEAD
 bienvenida.txt | Bin 88 -> 128 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)

```