UT3 -GIT – Parte IV

Ramas y merges

[1 Ramas uniones y conflictos 2](#_Toc124153061)

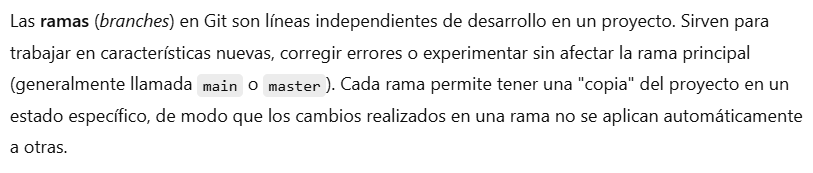
[1.1 Tipos de merge 2](#_Toc124153062)

[1.2 Merge Fast-forward 3](#_Toc124153063)

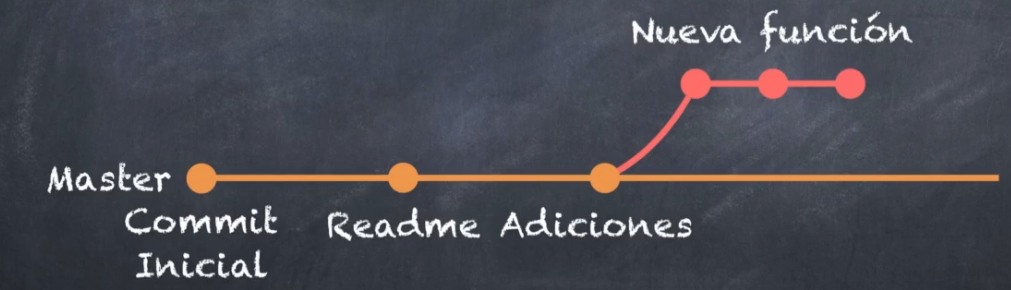
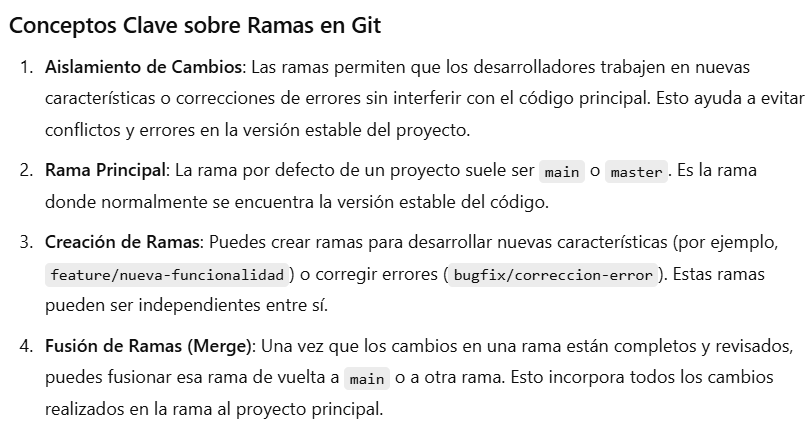
[1.3 Merge unión automática 9](#_Toc124153064)

[1.4 Merges manuales (con conflictos) 13](#_Toc124153065)

# Ramas



Crear una nueva rama se puede ver visualmente como una bifurcación en el camino actual:



En un momento dado puede que queramos que una rama secundaria se integre en la principal a esto se le llama **unión** o **merge:**



## Cambiar el nombre de una rama

Para modificar el nombre de la rama “master” al nombre “main” usamos el siguiente comando:

La opción -m indica que se va a cambiar de nombre a una rama.

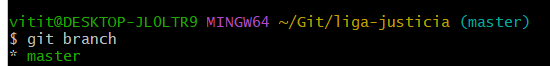
git branch -m master main



## Crear y cambiar de rama

El comando git branch nos indica en que rama estamos trabajando:

git branch



**Crear una rama:**

git branch <nombre-rama>

**Cambiarte a otra rama existente**:

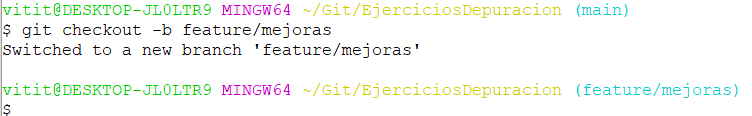
git checkout <nombre-rama>

**Crear y cambiar a una nueva rama al mismo tiempo** (si la rama no existe aún):

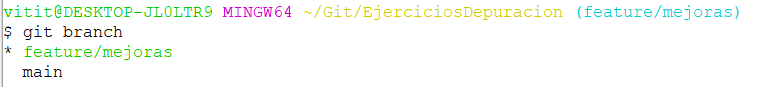
git checkout -b <nombre-nueva-rama>

En un proyecto en el que se quiere añadir una nueva funcionalidad, se suele crear una rama de características con el nombre *feature/nueva-funcionalidad*

Sobre el repositorio de EjerciciosDepuracion crea cambia a una nueva rama denominada **feature/mejoras**



Si consultamos las git Branch vemos las ramas creadas.



## Borrar una rama

Utilizando la opcion -d (delete)

git branch –d <nombre-rama>

# Merges (fusiones de ramas)

Existen distintos tipos de merges:

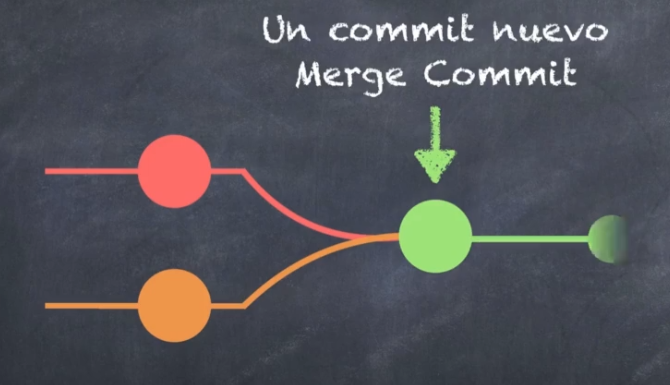
* **Fast-forward**: este tipo de unión se realiza cuando no ha habido cambios en la rama principal y por lo tanto se pueden unir sin problema las dos ramas simplemente agregando los commits de la rama secundaria a la principal:



* **Uniones automáticas**: este tipo de unión se realiza hay cambios en la rama principal pero que no entran en conflicto con los cambios realizados en la rama secundaria, por lo cual git puede realizar la unión de forma automática:



* **Manuales**: lo que git no puede decidir por sí mismo como hacer la unión y lo que hace es preguntarnos como realizar la unión y a continuación realizar un commit que se denomina **Merge Commit**



## Merge Fast-forward

Vamos a hacer cambios en la rama **feature/mejoras** y no realizar cambios en la rama **main**, para posteriormente hacer un merge.

En la clase Ejemplo1While.java declara 2 atributos y un constructor:

**private** **int** suma, num;

**public** Ejemplo1While() {

**this**.suma = 0;

**this**.num = 0;

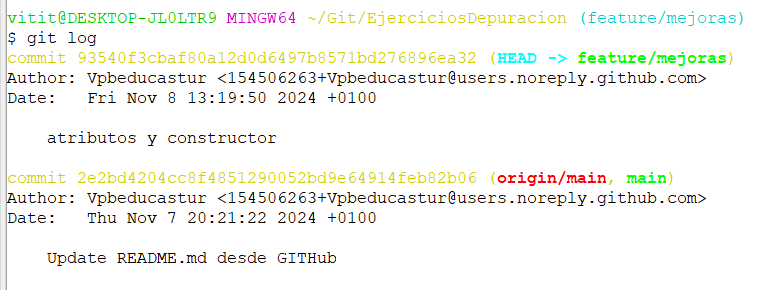
}

Agregamos al stage y haz el commit con el siguiente mensaje:

git add .

git commit -m “**atributos y constructor**”

Si consultamos ahora el histórico vemos que HEAD apunta al ultimo commit realizado en la rama de mejoras y la rama principal está un commit por detrás.



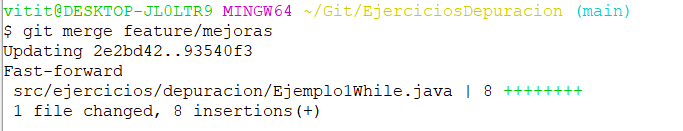
Vamos ahora a hacer la fusión de la rama de mejoras con la rama principal.

* Nos cambiamos a la rama principal
* Ejecutamos el merge:

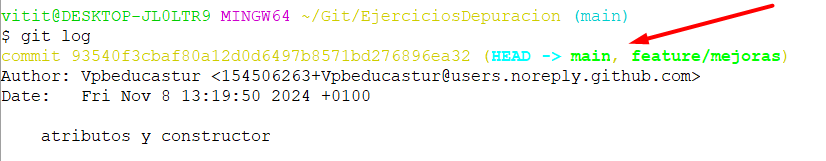
git checkout main

git merge feature/mejoras

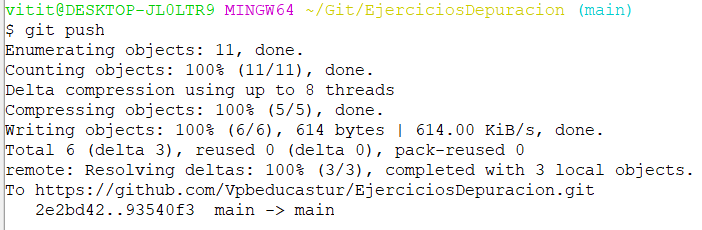
Vemos que se ha realizado un merge fast-forward y no ha habido conflictos. Git ha podido integrar los cambios de la rama mejoras en la rama principal.



Consultamos el histórico y vemos que ambas ramas ahora se encuentran en el mismo punto (HEAD)



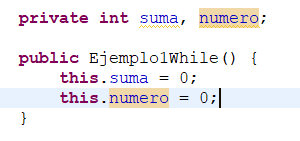
Ya solo nos quedaría subir los cambios al repositorio remoto con un push.



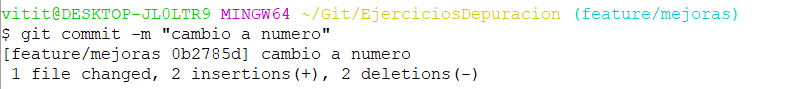
## Merges manuales (con conflictos)

Vamos a hacer cambios que entren en conflicto entre ramas.

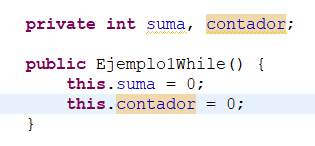
* **Desde la rama feature/mejoras** hacemos un cambio y hacemos commit
  + Renombramos la variable num a numero



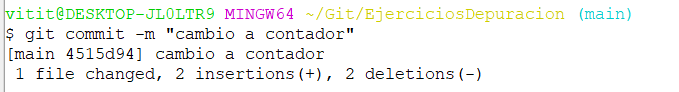
* + Hacemos el commit “cambio a numero”



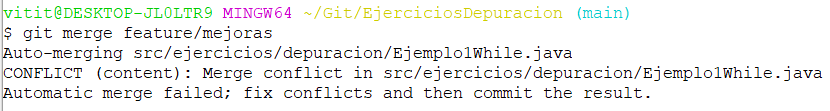
* **Desde la rama main** hacemos un cambio en las mismas línea de codigo y hacemos commit.
  + Renombramos la variable num a contador



* + Hacemos el commit “cambio a contador”



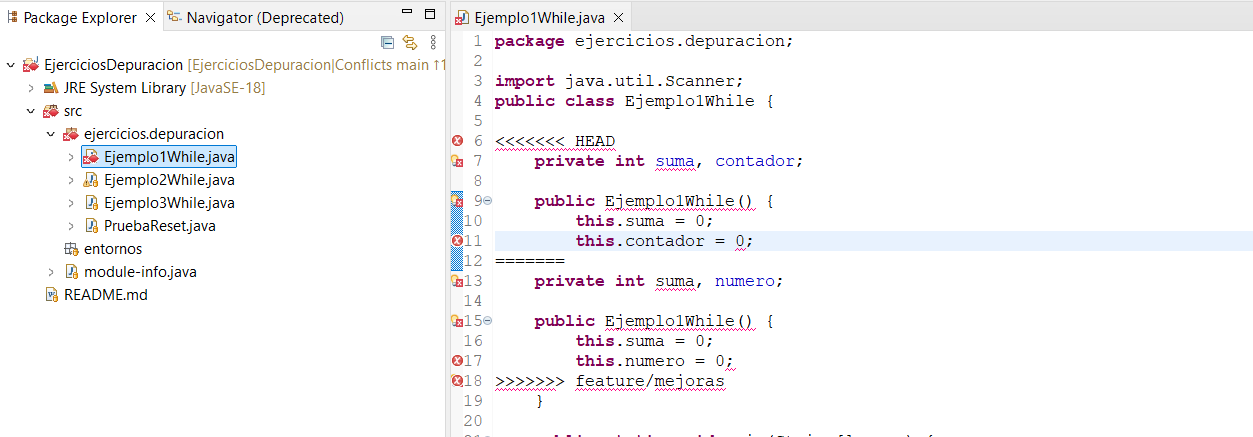
* Hacemos el merge que producirá un conflicto.



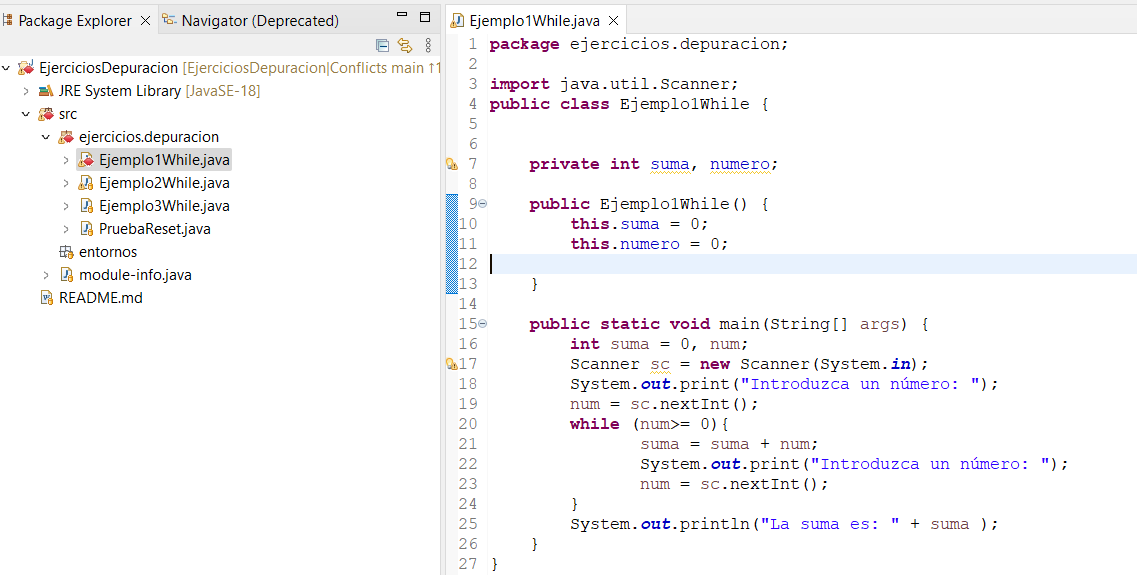
Git nos muestra un error. Nos dice que ha intentado hacer el merge automático (Auto-merging) pero que no es posible realizarlo porque hay conflictos que hay que resolver de manera manual.

Necesitamos abrir el fichero que provoca el conflicto desde un editor.

Si nos fijamos GIT ha modificado el fichero, indicándonos con texto que modificaciones son las ultimas realizadas (HEAD)



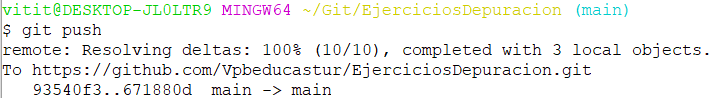
Editamos manualmente el fichero para corregir el conflicto. Vamos a quedarnos con el atributo numero.



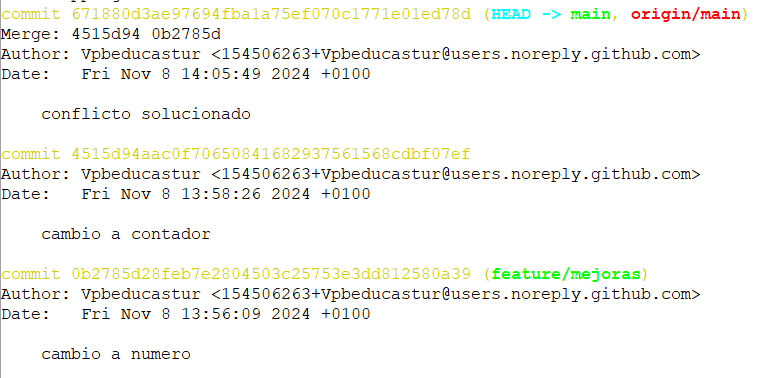
Guardamos el fichero y hacemos el commit con el texto **“conflicto solucionado”**para solucionar finalmente el conflicto. Fijate que la rama está en un estado | MERGING (merge pendiente) hasta que no confirmemos la solucion al conflicto.



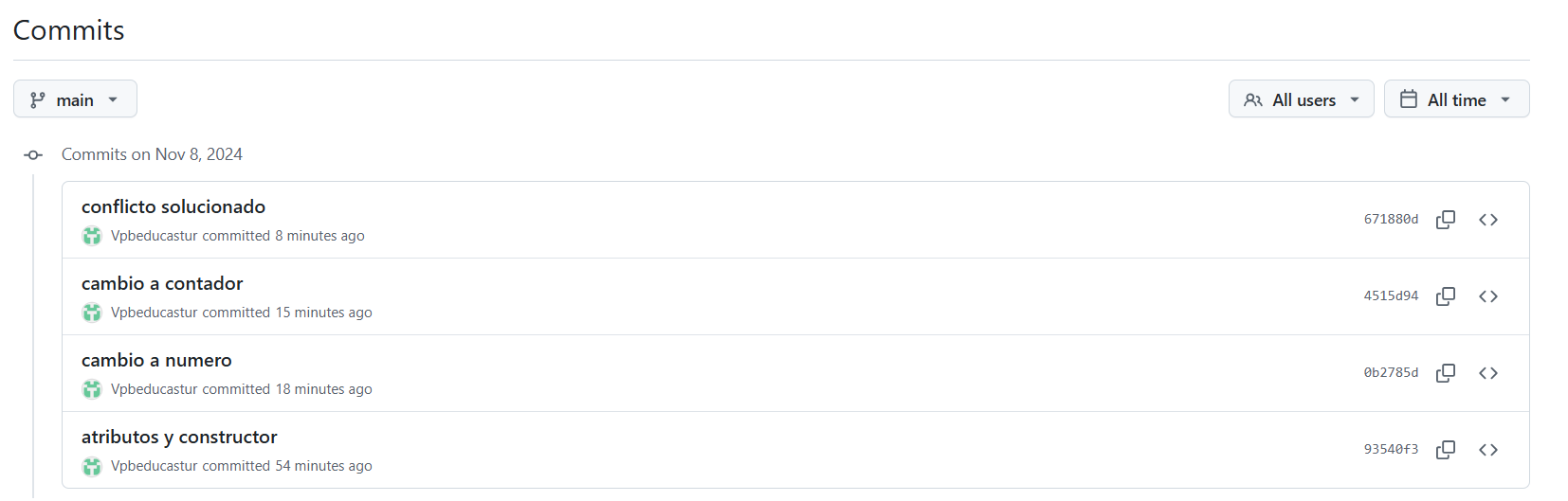
Finalmente subimos todo al repositorio remoto haciendo un push.



Ahora el histórico me muestra que la rama main está en el HEAD y la rama de mejoras sigue estando en el commit de “cambio a numero”



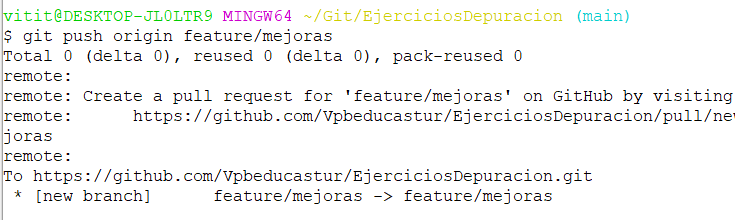
Y desde el repositorio remoto GitHub podemos tambien ver los ultimos commits de la rama main:



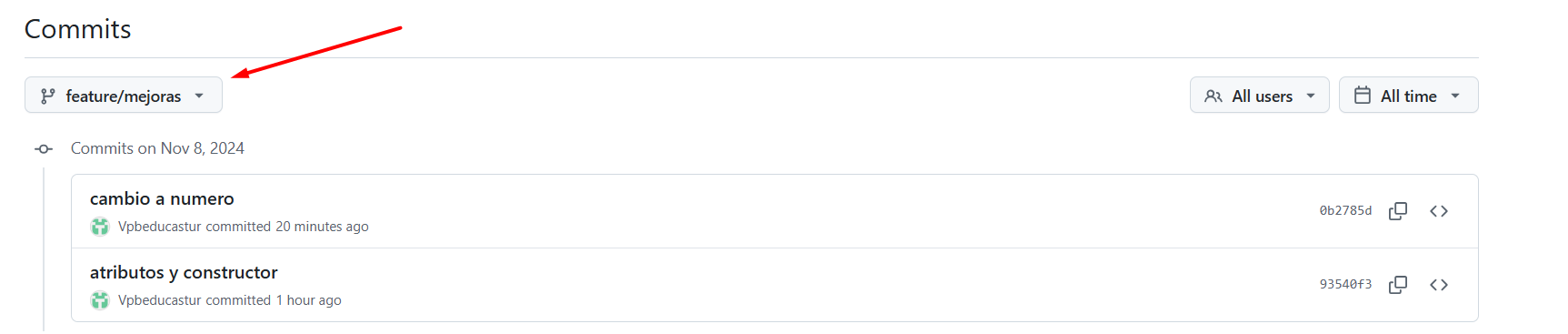
# Subir una rama a Github

Es posible que no hayas subido a remoto la rama de feature/mejoras. Para subirla debes ejecutar:

git push origin nombre-de-la-rama



Ahora podrás consultar la rama desde Github



# Entrega del trabajo realizado hasta el momento

Crea un documento PDF con la siguiente información:

La captura del resultado completo de tu histórico *git reflog*, que debe contener al menos:

1. (HEAD -> main, origin/main) HEAD: commit (merge): conflicto solucionado
2. commit: cambio a contador
3. (feature/mejoras) HEAD: commit: cambio a numero
4. merge feature/mejoras: Fast-forward
5. commit: atributos y constructor
6. commit: gitignore configurado
7. commit: gitkeep añadido
8. reset: moving to ##################
9. reset: moving to HEAD^
10. commit: cambio int por double
11. commit: modificacion pedirNumero
12. commit (initial): primer commit

Accede a Github y obtén las URL del historico de commits de las ramas **principal** y **feature/mejoras**. Asegúrate de que sea público para que el profesor pueda acceder y adjúntalas al documento.

Ejemplo:

<https://github.com/Vpbeducastur/EjerciciosDepuracion/commits/main>

<https://github.com/Vpbeducastur/EjerciciosDepuracion/commits/feature/mejoras>