



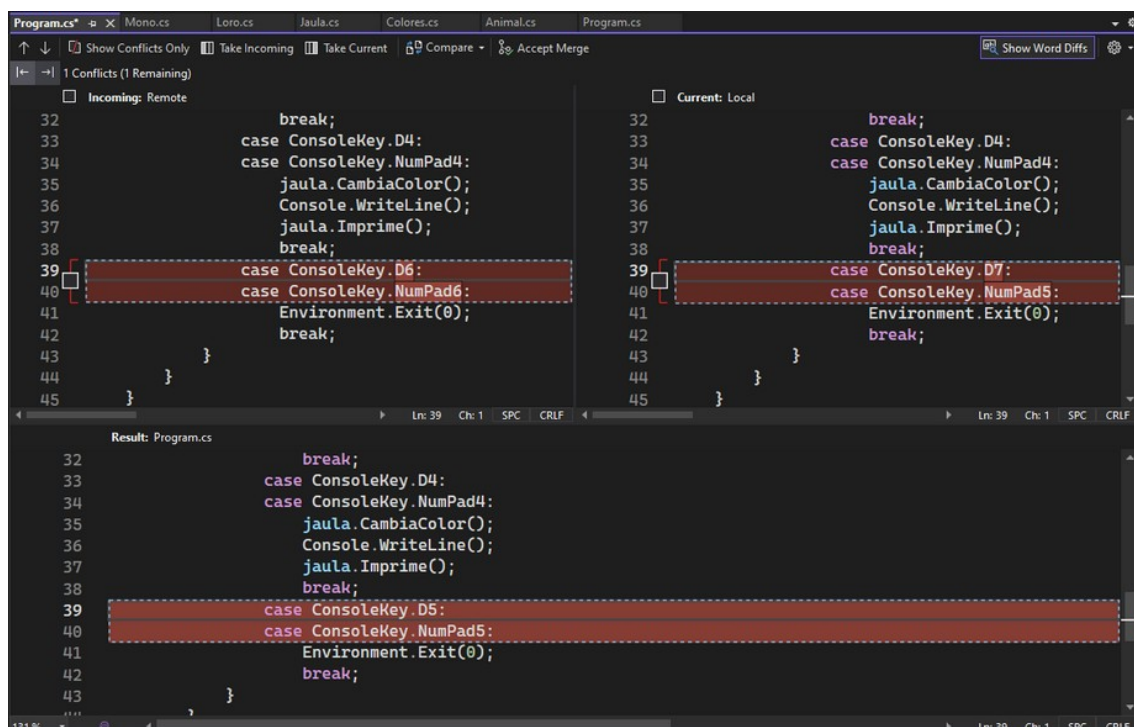
Gastón Bermúdez Rochietti
Miriam Guerra Guerra
Fabio Acosta Fuentes

Introducción:

Nuestro proyecto consiste en crear una aplicación con C#. En la aplicación hay una clase herencia de animal, con loros y monos. Un arraylist llamado jaula. Y un menú con el que interactuamos con los distintos componentes. El repositorio creado se encuentra en [este enlace](#).

Conflictos en GitHub – Rama Máster:

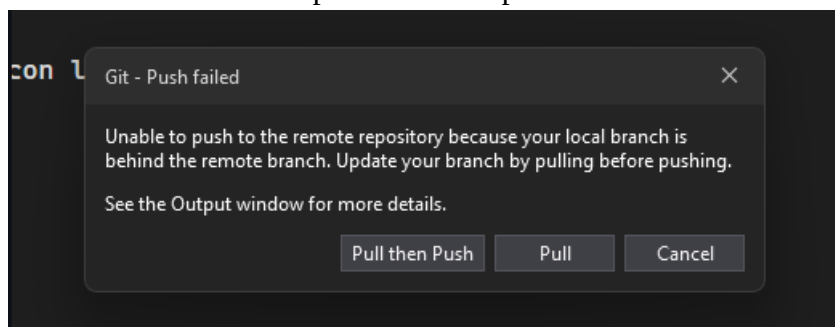
Probamos a experimentar con GitHub, creando situaciones de conflicto, en la rama Master. Cambiando el código en el repositorio local y después intentando unificarlo con el repositorio remoto:



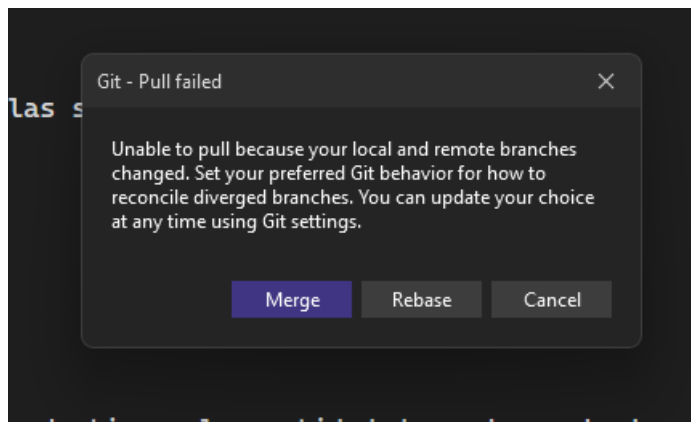
En la captura se están modificando las opciones de un switch según lo que queremos que ocurra cuando se pulsa una tecla determinada del teclado. En el cuadrado de la izquierda está el repositorio remoto, en la derecha el local y abajo está el resultado final que vamos a aprobar para los cambios, después de haber corregido el código.

Algunos problemas que surgieron antes de llegar al punto anterior:

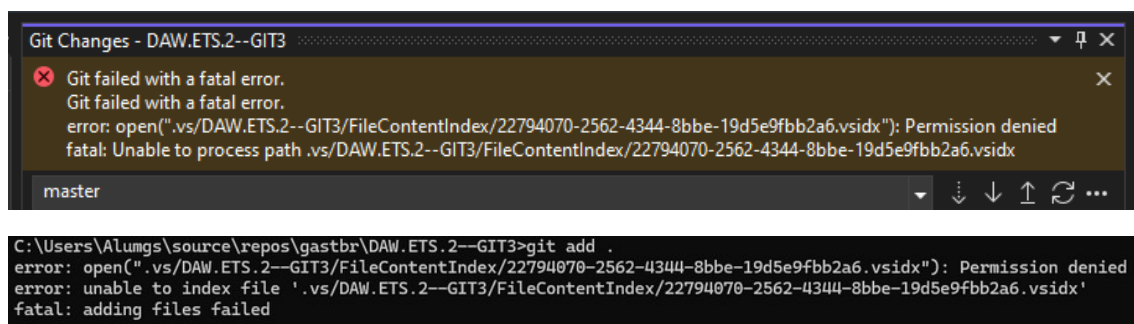
Antes de realizar un push teníamos que realizar un pull, para actualizar los cambios que hubiese hecho otro compañero en el repositorio remoto.



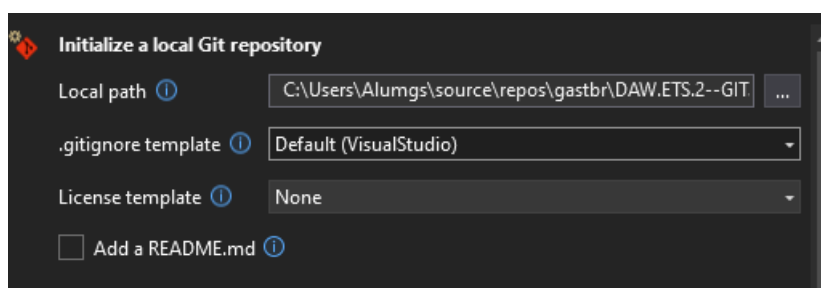
En caso de no hacer un pull antes de empezar a trabajar, ocurría este otro conflicto que notificaba que tanto la rama en remoto como la local tenían cambios y por lo tanto debías comprobar qué cambios habían para poder unificar todo a una solución, como aparece en la primera captura.



También tuvimos un problema para conectar la cuenta de GitHub con el repositorio porque había otra cuenta en el Visual Studio y no nos dejaba eliminarla.



Cuando por fin pudimos trabajar con el repositorio en local, nos mostraba este mensaje para guardar el repositorio en local.



Conflictos en GitHub – Rama Subramas:

```

C:\Users\34680\source\repos\gastbr\DAW.ETS.2--GIT3>git branch Miriam

C:\Users\34680\source\repos\gastbr\DAW.ETS.2--GIT3>git checkout Miriam
Switched to branch 'Miriam'

C:\Users\34680\source\repos\gastbr\DAW.ETS.2--GIT3>git status
On branch Miriam
nothing to commit, working tree clean

C:\Users\34680\source\repos\gastbr\DAW.ETS.2--GIT3>git push
fatal: The current branch Miriam has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin Miriam

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

C:\Users\34680\source\repos\gastbr\DAW.ETS.2--GIT3> git push --set-upstream origin Miriam
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'Miriam' on GitHub by visiting:
remote:      https://github.com/gastbr/DAW.ETS.2--GIT3/pull/new/Miriam
remote:
To https://github.com/gastbr/DAW.ETS.2--GIT3
 * [new branch]      Miriam -> Miriam
branch 'Miriam' set up to track 'origin/Miriam'.

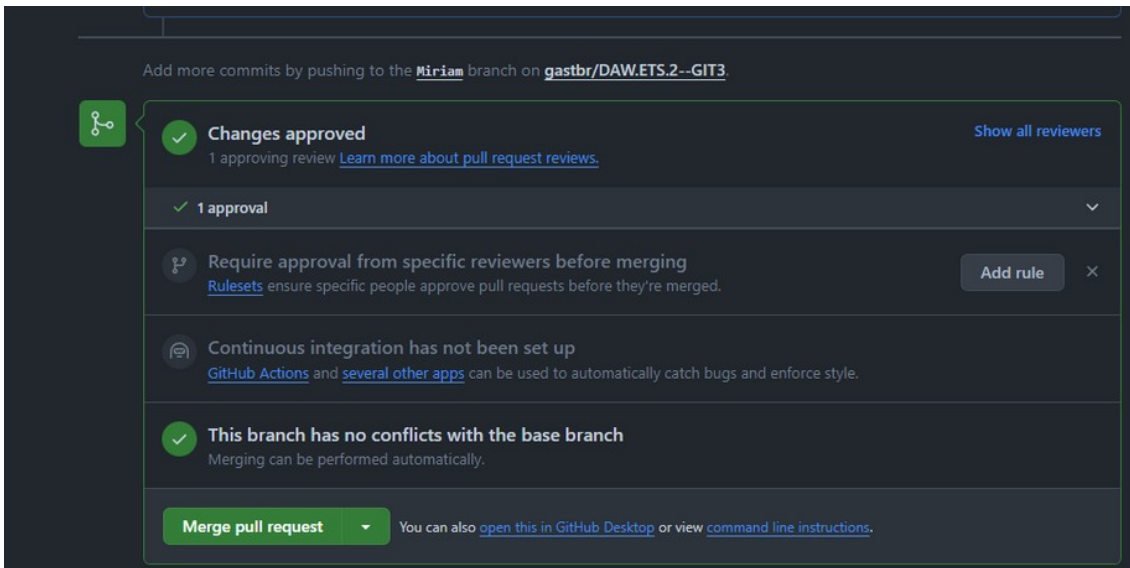
C:\Users\34680\source\repos\gastbr\DAW.ETS.2--GIT3>git status
On branch Miriam
Your branch is up to date with 'origin/Miriam'.

nothing to commit, working tree clean

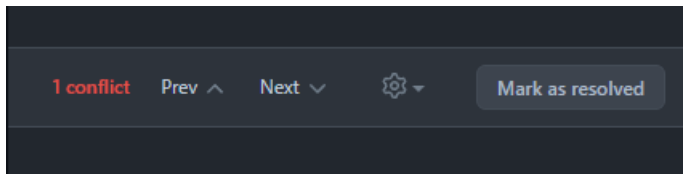
C:\Users\34680\source\repos\gastbr\DAW.ETS.2--GIT3>

```

Creamos la rama secundaria, en la imagen superior se aprecian los comandos para crear la nueva rama y poder trabajar en ella en vez de alterar el código directamente en la rama máster.



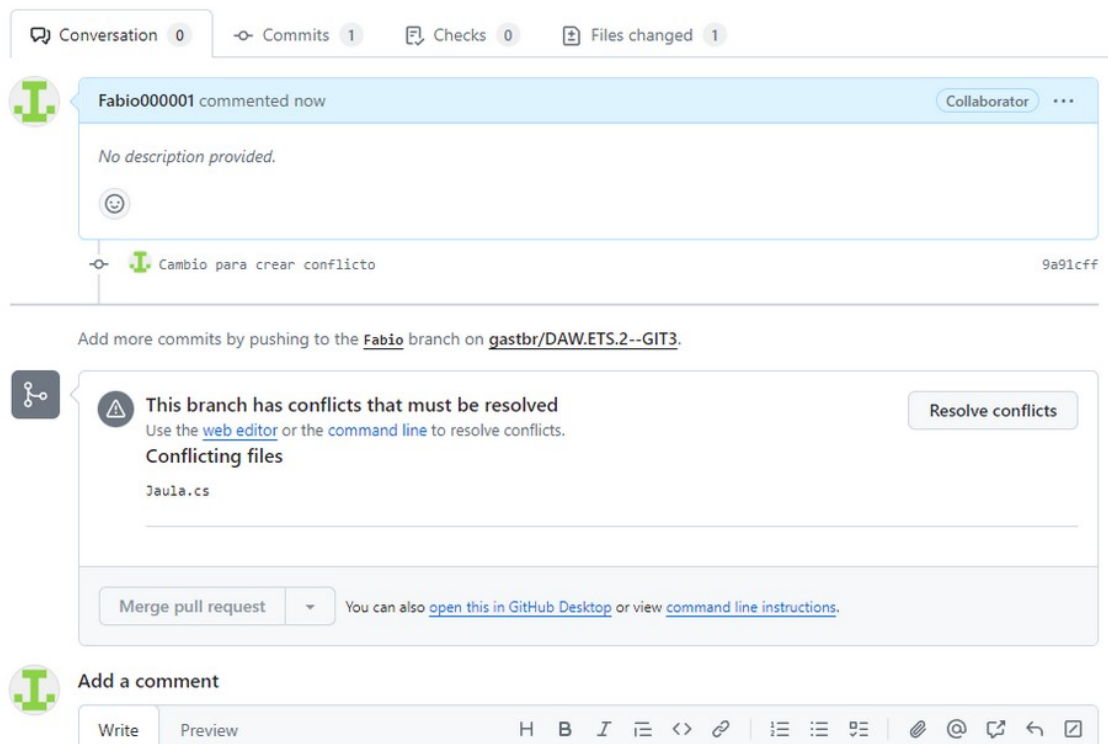
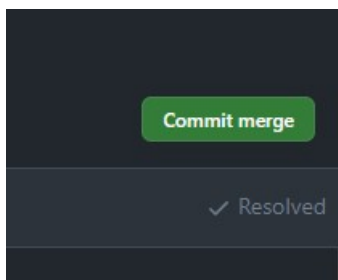
Cuando aprobamos los cambios producidos en una rama, pero esos cambios no tienen conflictos con la rama máster. Aparece un mensaje notificando que no hay problemas de conflictos entre ambas ramas y permite aceptar la petición de merge pull.



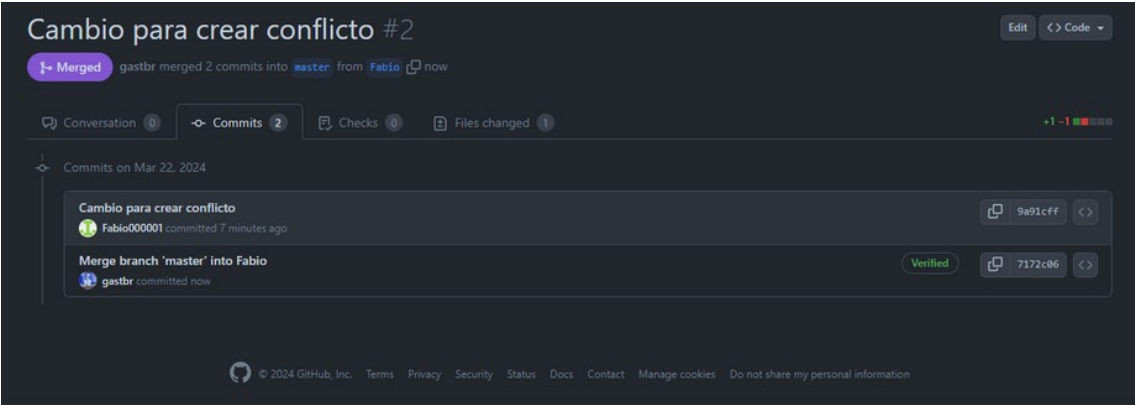
Sin embargo, cuando hay conflicto aparece un mensaje de notificación como el de la imagen superior.

```
87         animales.Clear();
88     <<<<<< Fabio
89         Console.WriteLine("La jaula no tiene nada.");
90         =====
91         Console.WriteLine("Has vaciado la jaula.");
92     >>>>>> master
93     }
```

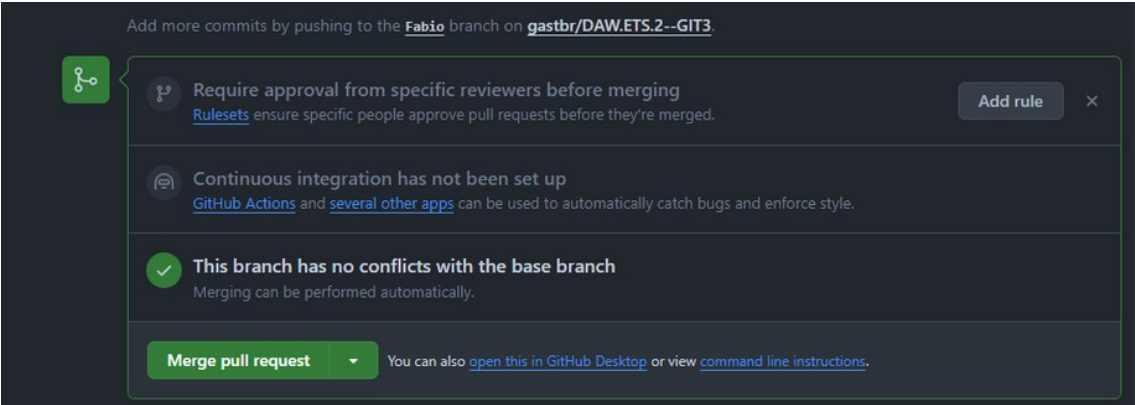
En esta imagen se aprecian las diferencias entre la rama secundaria y la rama máster, hay una diferencia en uno de los comentarios de consola. Una vez solucionado el conflicto será solucionado y podrán realizarse el commit, como se aprecia en la imagen inferior.



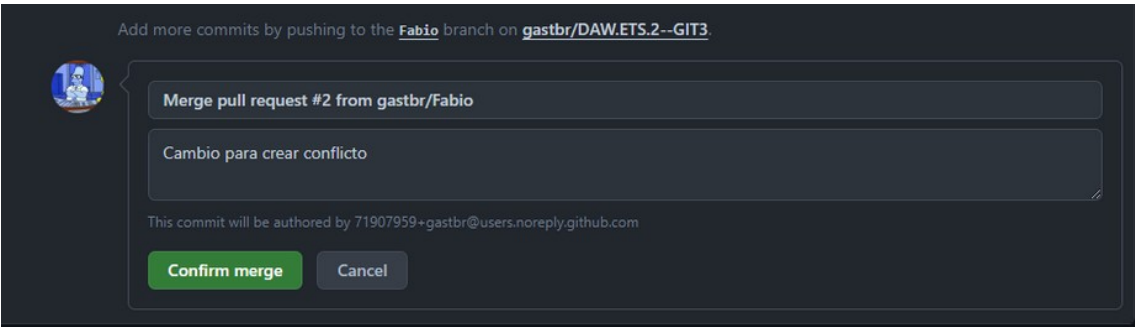
En esta imagen se ve un tablón de mensajes que muestra los conflictos que hay en una rama específica y en qué archivo se encuentran los conflictos.



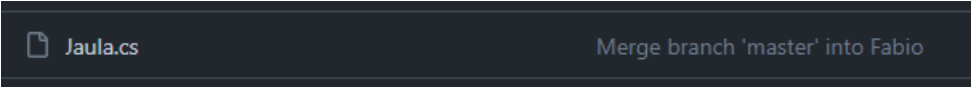
Cuando se realizan los commit se registran de la manera que se ve en la imagen superior. Mostrando detalles de los comentarios que se han puesto en cada commit, y qué ramas han sido las afectadas.



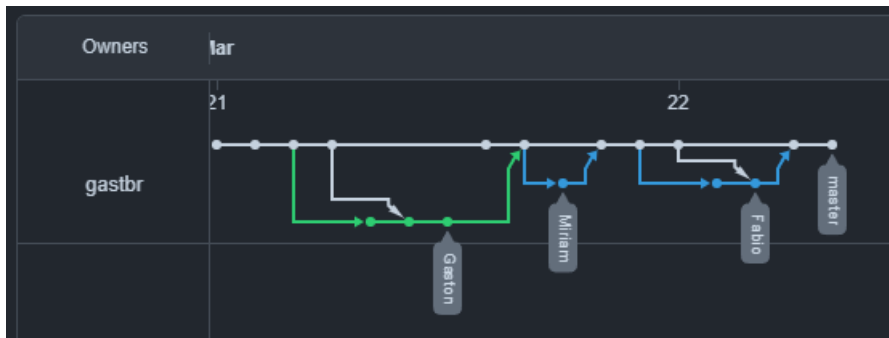
Cuando no hay problemas de conflictos, se puede realizar un merge.



Se puede escribir un mensaje y confirmar el merge.



El archivo que ha recibido un merge tendrá un comentario sobre que rama ha alterado la máster.



En esta imagen se aprecia la rama máster y las otras ramas, se pueden ver los cambios realizados y como ha sido alterada cada rama.