

TUTORIAL GIT FORK, BRANCH , BRANCH PER FEATURE , MERGE , PULL RESQUEST

Un fork es una copia exacta de un proyecto que ya tiene una primera versión acabada y que falta por ejemplo adaptaciones multiplataforma , o utilidades específicas.

En el siguiente ejemplo utilizo el proyecto CalculaFork , Ya que su autor Pere creo una Issus , para que se desarrollara la función de la división y el testeo de esa clase.

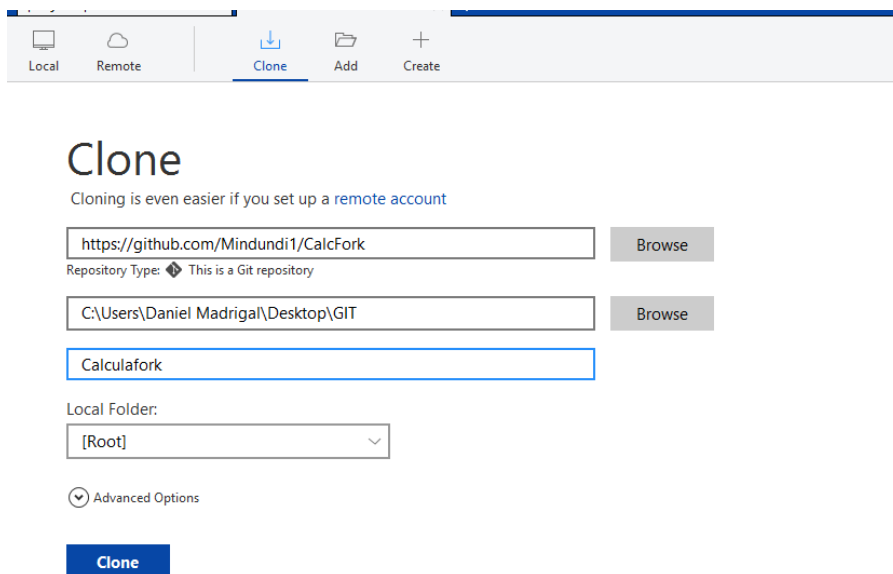
① 3 Open ✓ 1 Closed		Author ▾	Labels ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
①	Log4net	#5 opened 21 hours ago by charliesr					
①	Función de División	#4 opened 21 hours ago by charliesr					
①	Crear funcionalidad Multiplicación en la calculadora	#3 opened 2 days ago by perezamora					

En primer lugar para poder asignar una Issus debemos ser colaboradores del proyecto , una ves seamos colaboradores podemos asignarlos la Issus o si somos los dueños del repositorio podemos asignársela a cualquiera de los colaboradores:

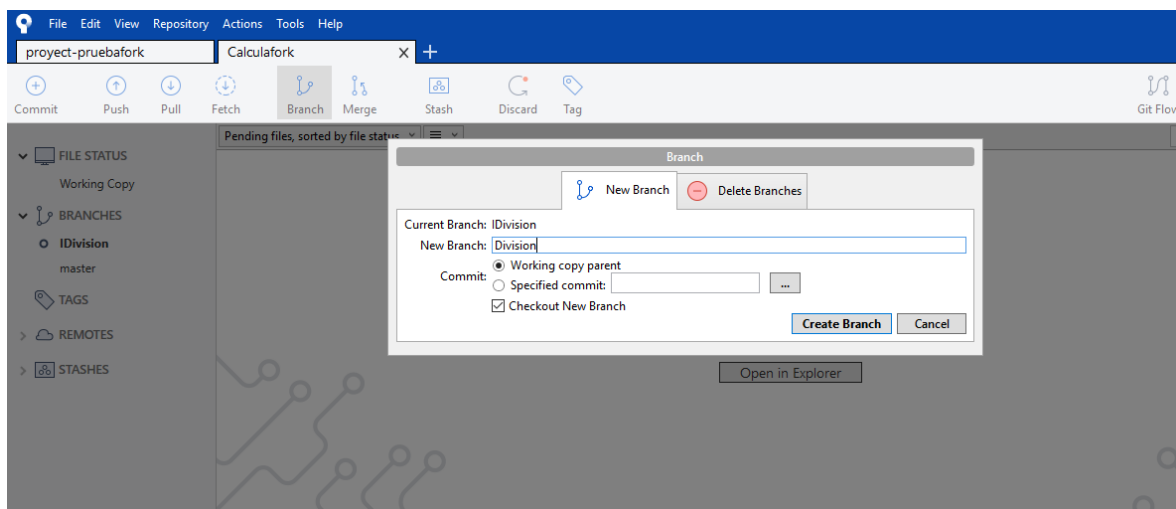
The screenshot shows a GitHub issue page for the repository 'CalculaFork'. The issue title is 'Añadir clase alumno a nuestro proyecto hello world'. The issue is assigned to 'danielMT1490'. The 'Assignees' dropdown menu is open, showing a list of users: 'danielMT1490' and 'Mindundi1'. 'Mindundi1' is selected. The dropdown menu also includes a 'Filter people' search bar, a 'Clear assignees' button, and a 'Milestone' section with 'No milestone' selected. There is also a 'Notifications' section with an 'Unsubscribe' button. The issue is labeled 'help wanted' and '2 participants' are listed at the bottom.

Para esta demostración he hecho un fork con la cuenta mindundi1 del proyecto del Pere , trabajare con Branch y Branch per feature , en este caso hare dos ramas una donde hare la interface y otra donde escribiré la clase División y otra con el test , ya que Branch per feature nos dice que se utilice una rama por característica dentro de una rama de integración continua.

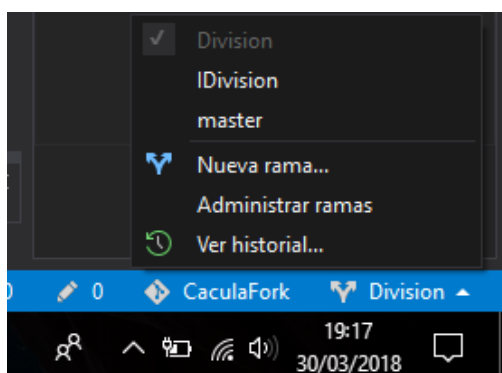
Clonare el repositorio remoto con sourcetree , hare algún ejemplo también con la extesion de visual Studio también.



Crearemos un par de ramas una para la interface y otra para la clase división así vemos como insertarlas en la master tanto con sourcetree como con visual



Visual nos reconocerá las ramas automáticamente cuando abramos el proyecto podemos elegir con cual trabajar.



Ejecutando los comandos `git push set—upstream origin IDivision` subimos a la nube la Rama que hemos Insertado. Hacemos lo mismo con la rama Division asi veremos que nos aparecen en la pagina de git.

```
MINGW64: c:/Users/Daniel Madrigal/Desktop/GIT/CaculaFork
$ cd CaculaFork/

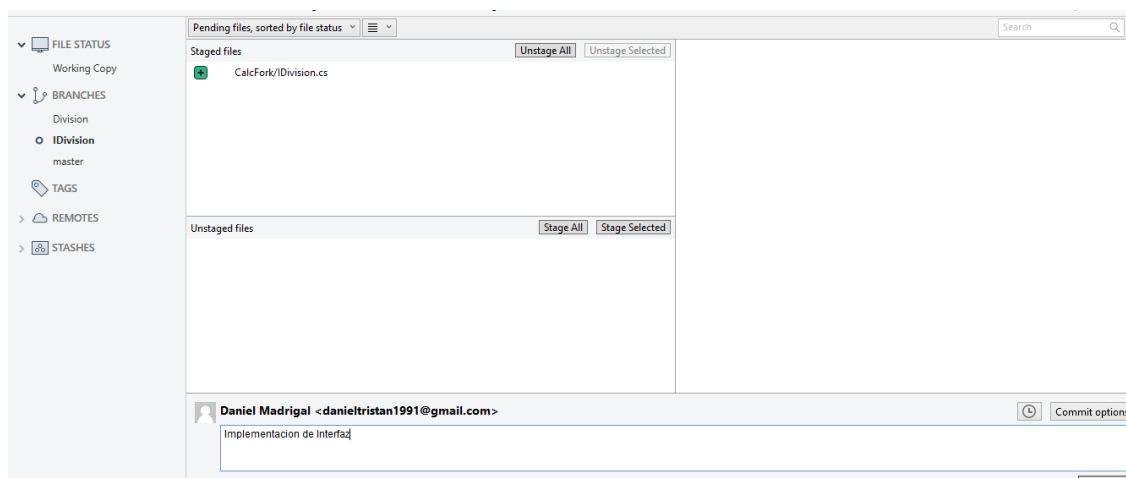
Daniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CaculaFork (IDivision)
$ git push
fatal: The current branch IDivision has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin IDivision

Daniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CaculaFork (IDivision)
$ git push --set-upstream origin IDivision
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 531 bytes | 531.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/Mindundi1/CalcFork
 * [new branch]      IDivision -> IDivision
Branch 'IDivision' set up to track remote branch 'IDivision' from 'origin'.

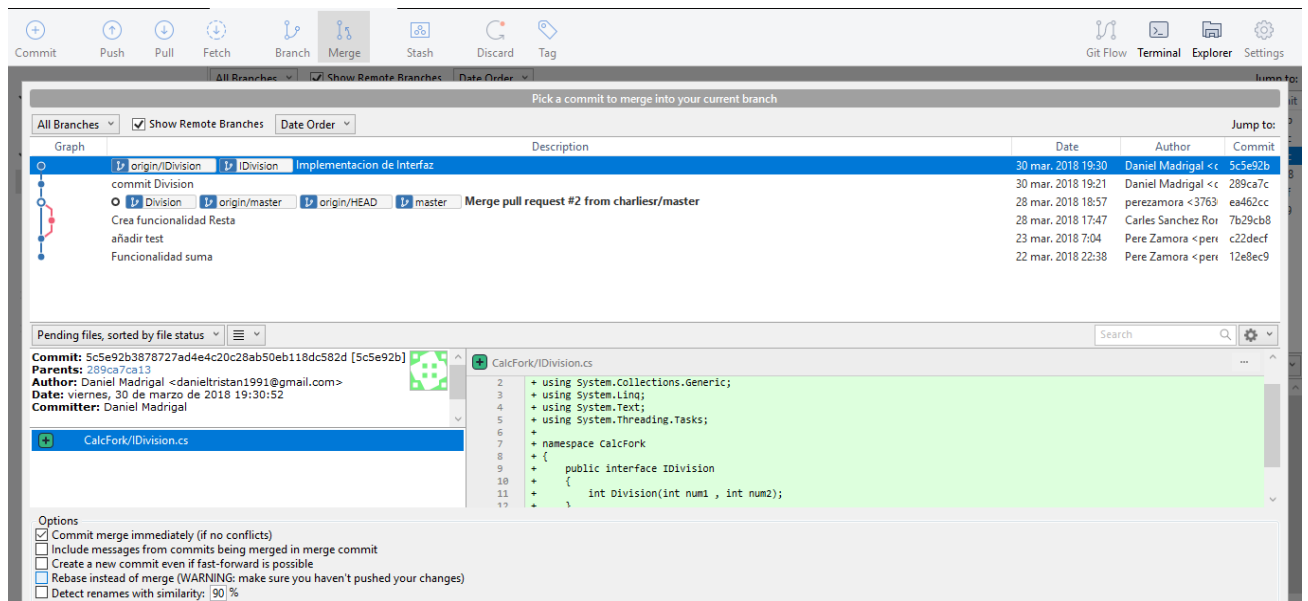
Daniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CaculaFork (IDivision)
$
```

Implementamos la interfaz y hacemos el commit para poder insertar la rama en la master.



Ahora voy a insertar la rama IDivision en la rama Division para poder implementar la interfaz

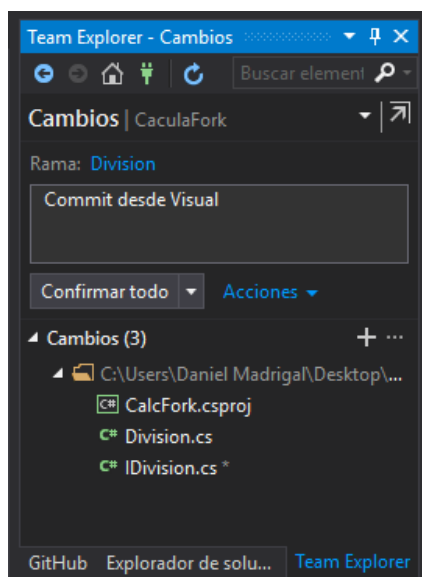
Seleccionamos la rama Division en sourcetree y escogemos la opción merge , dentro seleccionamos la rama IDivision y hacemos la combinación



Nos muestra el nuevo código a instertar .

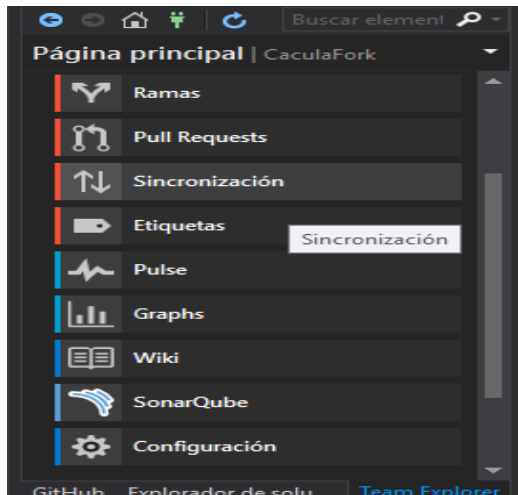
En este caso vamos a hacer el commit de la rama Division desde Visual Studio .

Dentro de Team Explorer/Pagina Principal /Cambios podemos hacer los commits:

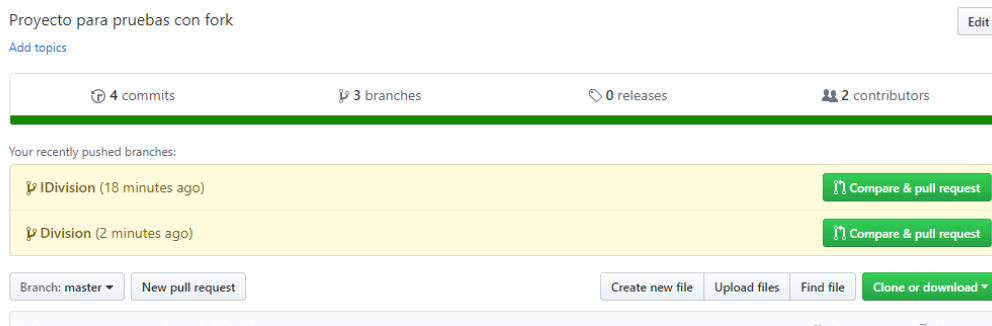


Y para hacer el push desde visual de la rama Division tenemos que hacerder al campo sincronizar.

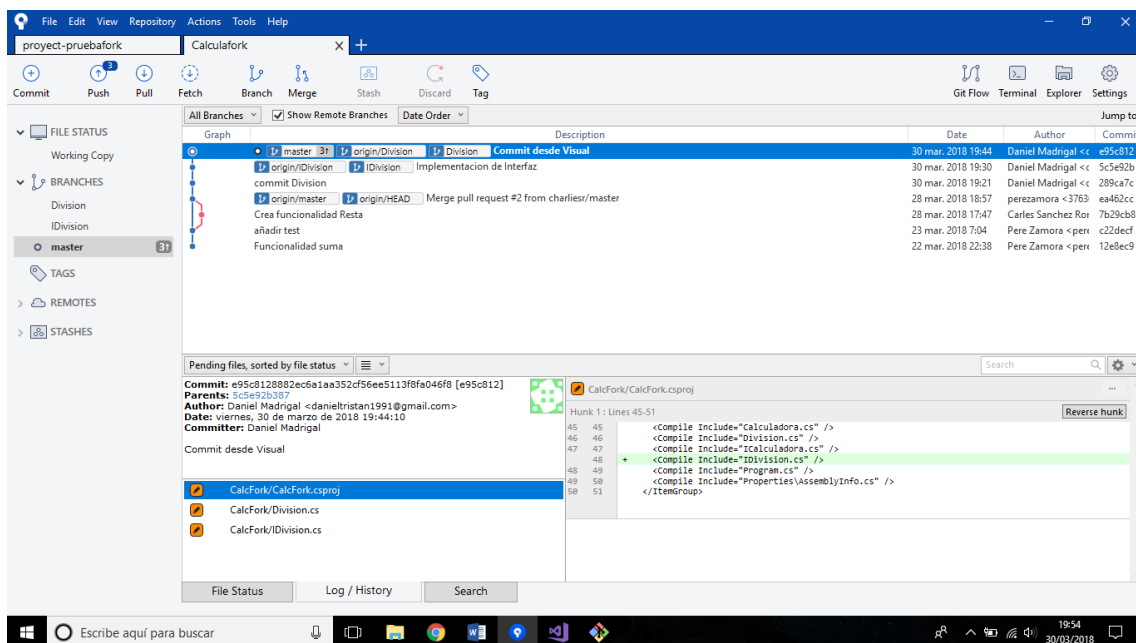
Al tener la extensión de SonarLint me aparta la opción de conectar con SonarQube .



Ya tenemos nuestras dos ramas en actualizadas en el repositorio de la nube.



Podemos hacer un merge a la ramam master para insertar los cambios y desde esta rama haremos un pull request al repositorio danielMT1490 que nos debe dar su opinión y decidirá si insertar los cambios en su proyecto.



Como vemos en la rama master tenemos todos cambios preparado para hacer el push.

Esta vez haremos el push desde línea de comandos cambiando a la rama master ya que estábamos en la Rama Division

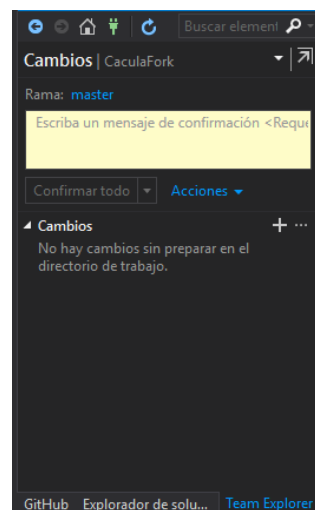
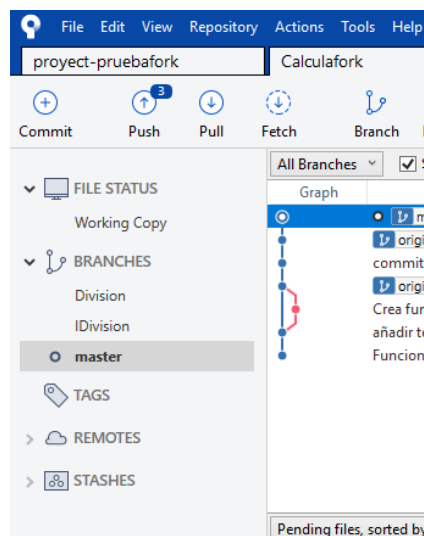
```
MINGW64/c/Users/Daniel Madrigal/Desktop/GIT/CaculaFork
aniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CaculaFork (IDivision)
git push
counting objects: 4, done.
delta compression using up to 4 threads.
compressing objects: 100% (4/4), done.
writing objects: 100% (4/4), 500 bytes | 500.00 KiB/s, done.
total 4 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
to https://github.com/Mindundil/CalcFork
 289ca7c..5c5e92b IDivision -> IDivision

aniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CaculaFork (IDivision)
git push --set-upstream origin master
total 0 (delta 0), reused 0 (delta 0)
to https://github.com/Mindundil/CalcFork
 ea462cc..e95c812 master -> master
branch 'master' set up to track remote branch 'master' from 'origin'.

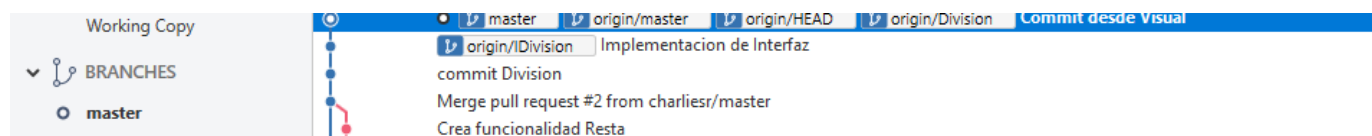
aniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CaculaFork (master)
git push
everything up-to-date

aniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CaculaFork (master)
```

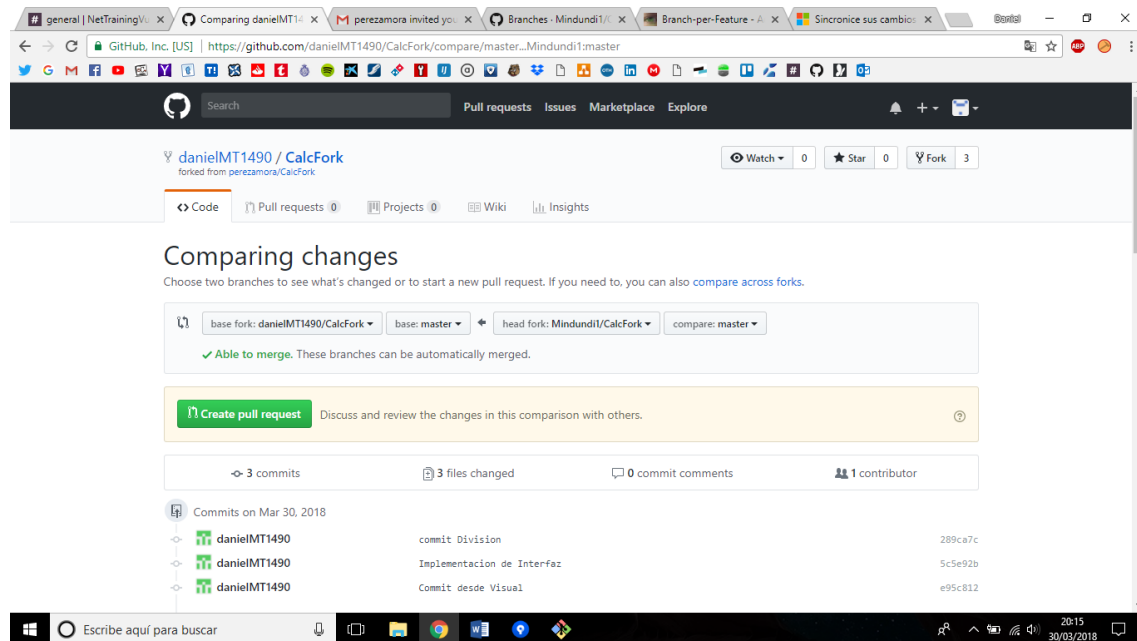
Ya no nos aparecen los cambios ni en Visual ni en SourceTree



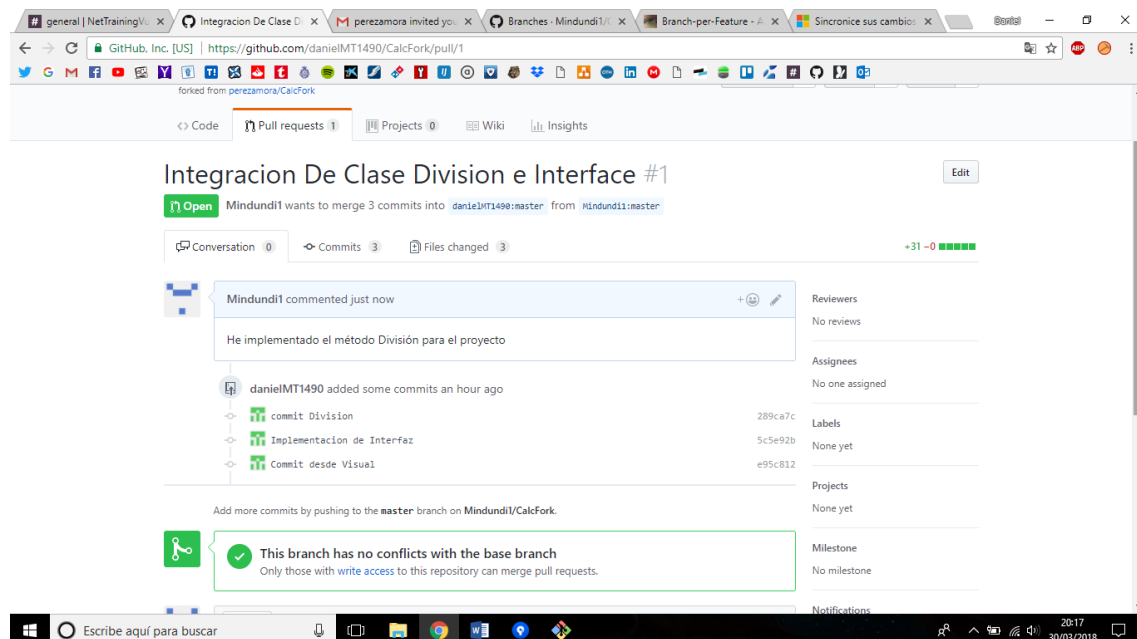
Ahora que la rama master esta actualizada podemos eliminar la otras ramas podemos hacerlo directamente desde la nube o local .



Por la parte de Mindundi1 ya esta todo hecho así que vamos a hacer el pull request desde GitHub.

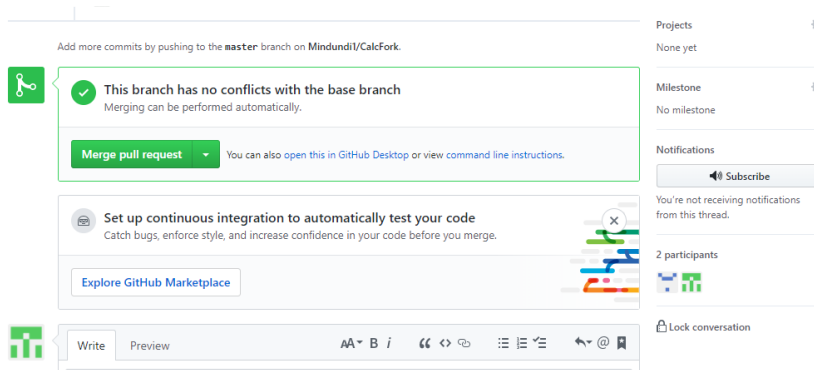


Git compara el código en busca de posibles errores en este caso Mindundi1 lo ha hecho bien y el código puede combinarse sin problemas .



Confirmamos y enviamos la solicitud de integración a danielMT1490 ahora veremos las posibilades dentro del repositorio local desde donde hicimos el fork

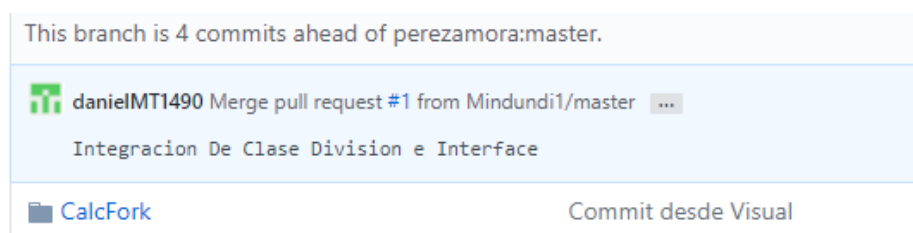
Como el codifigo es correcto y no genera conflictos GitHub nos da la opcion de hacer un merge con nuestro proyecto sin problema pero como no nos fiamos de Mindundi revisaremos el codigo.



Podemos ver tanto los commits que ha hecho como el código pudiendo poner comentarios a las lines de código etc.

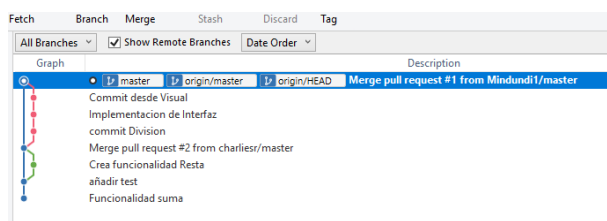


Un vez aceptado vemos como nos aparece el merge en nuestro proyecto cuando entramos en el repositorio local:



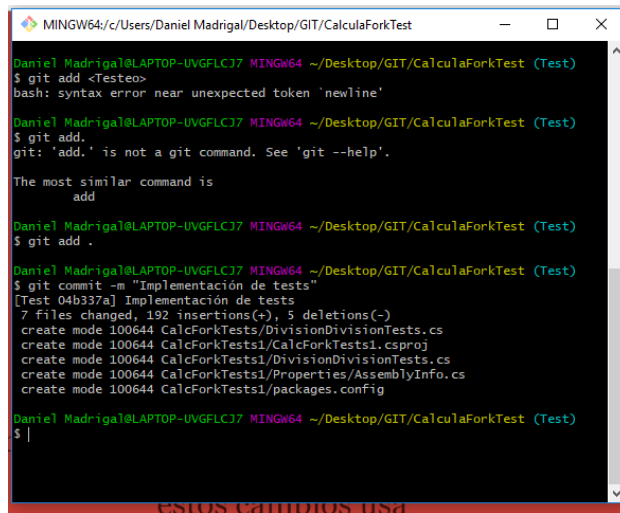
Ahora desde la cuenta de danielMT1490 hare el test y espero que lo integre en su proyecto.

Cuando me clono el proyecto puedo ver los fork y pull que ha tenido el proyecto en este caso de Carlos y de Mindundi 1 .



Una vez integrado el test procederé a hacer el commit y el push y el pull esta vez desde línea de comandos.

Añadimos cambios y hacemos commit



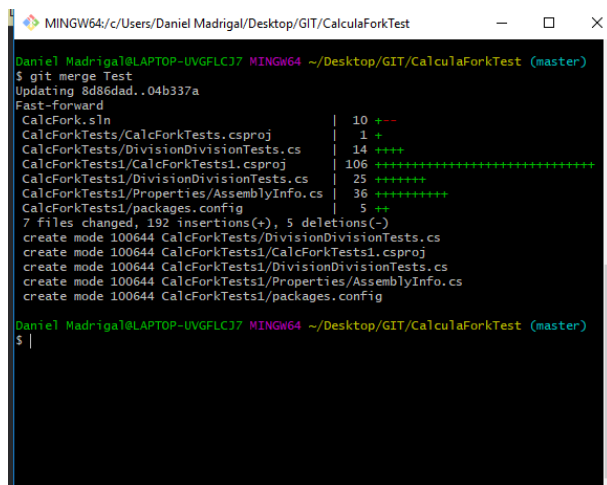
```
MINGW64/c/Users/Daniel Madrigal/Desktop/GIT/CalculaForkTest
Daniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CalculaForkTest (Test)
$ git add <Testeo>
bash: syntax error near unexpected token `newline'

Daniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CalculaForkTest (Test)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
    add

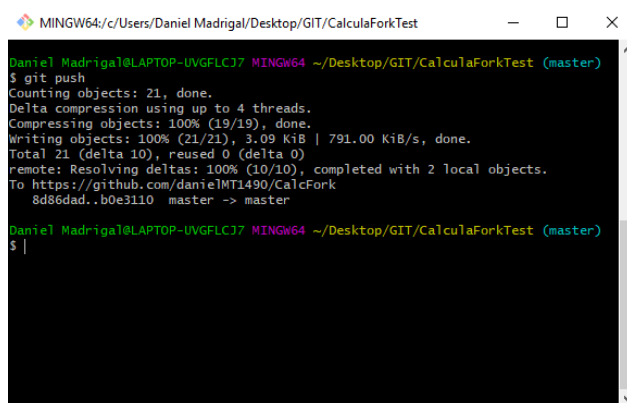
Daniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CalculaForkTest (Test)
$ git add .
Daniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CalculaForkTest (Test)
$ git commit -m "Implementación de tests"
[Test 04b337a] Implementación de tests
7 files changed, 192 insertions(+), 5 deletions(-)
create mode 100644 CalcForkTests/DivisionDivisionTests.cs
create mode 100644 CalcForkTests1/CalcForkTests1.csproj
create mode 100644 CalcForkTests1/DivisionDivisionTests.cs
create mode 100644 CalcForkTests1/Properties/AssemblyInfo.cs
create mode 100644 CalcForkTests1/packages.config
Daniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CalculaForkTest (Test)
$ |
```

Insertamos en la rama master los cambios



```
MINGW64/c/Users/Daniel Madrigal/Desktop/GIT/CalculaForkTest
Daniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CalculaForkTest (master)
$ git merge Test
Updating 8d86dad..04b337a
Fast-forward
 CalcFork.sln | 10 +--
 CalcForkTests/CalcForkTests.csproj | 1 +
 CalcForkTests/DivisionDivisionTests.cs | 14 +++
 CalcForkTests1/CalcForkTests1.csproj | 106 +++++
 CalcForkTests1/DivisionDivisionTests.cs | 25 +++++
 CalcForkTests1/Properties/AssemblyInfo.cs | 36 +++++
 CalcForkTests1/packages.config | 5 ++
 7 files changed, 192 insertions(+), 5 deletions(-)
 create mode 100644 CalcForkTests/DivisionDivisionTests.cs
 create mode 100644 CalcForkTests1/CalcForkTests1.csproj
 create mode 100644 CalcForkTests1/DivisionDivisionTests.cs
 create mode 100644 CalcForkTests1/Properties/AssemblyInfo.cs
 create mode 100644 CalcForkTests1/packages.config
Daniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CalculaForkTest (master)
$ |
```

Hacemos el push



```
MINGW64/c/Users/Daniel Madrigal/Desktop/GIT/CalculaForkTest
Daniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CalculaForkTest (master)
$ git push
Counting objects: 21, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (19/19), done.
Writing objects: 100% (21/21), 3.09 KiB | 791.00 KiB/s, done.
Total 21 (delta 10), reused 0 (delta 0)
remote: Resolving deltas: 100% (10/10), completed with 2 local objects.
To https://github.com/danielMT1490/CalculaFork
 8d86dad..b0e3110 master -> master
Daniel Madrigal@LAPTOP-UVGFLCJ7 MINGW64 ~/Desktop/GIT/CalculaForkTest (master)
$ |
```

Y realizo el pull request desde Github a Pere.

