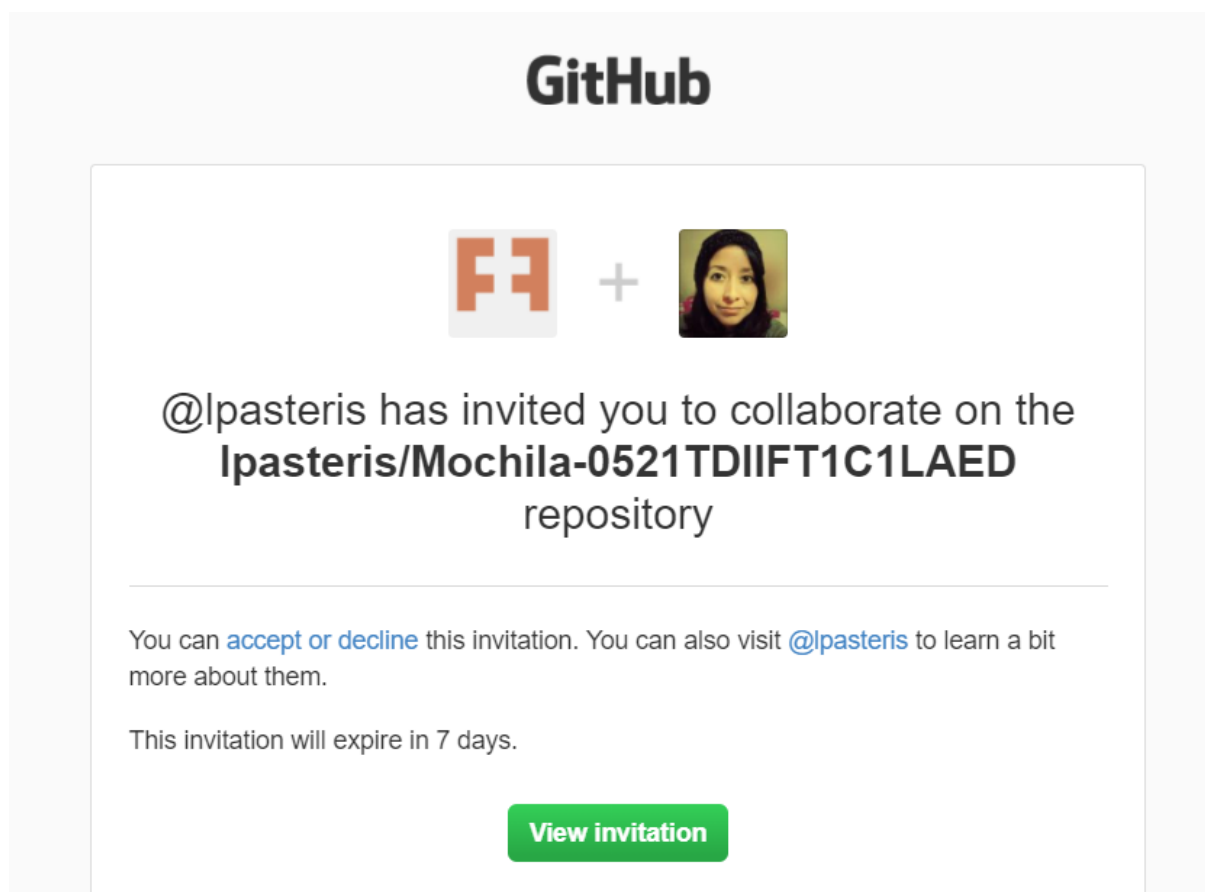


Git + GitHub

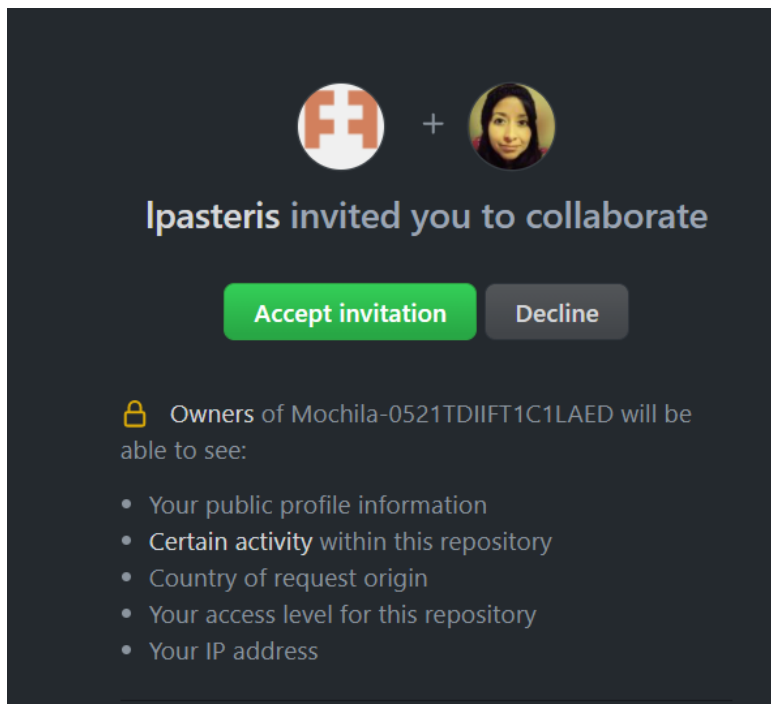
Guia paso a paso para trabajar en la mochila de la cursada.

Primeros pasos

1. Lo primero que tenés que hacer es proporcionarle tu nombre de usuario a tu profesor o profesora para que te agregue como colaborador al repositorio.
2. Una vez que te agregó como colaborador, te llegará un correo electrónico con la invitación:

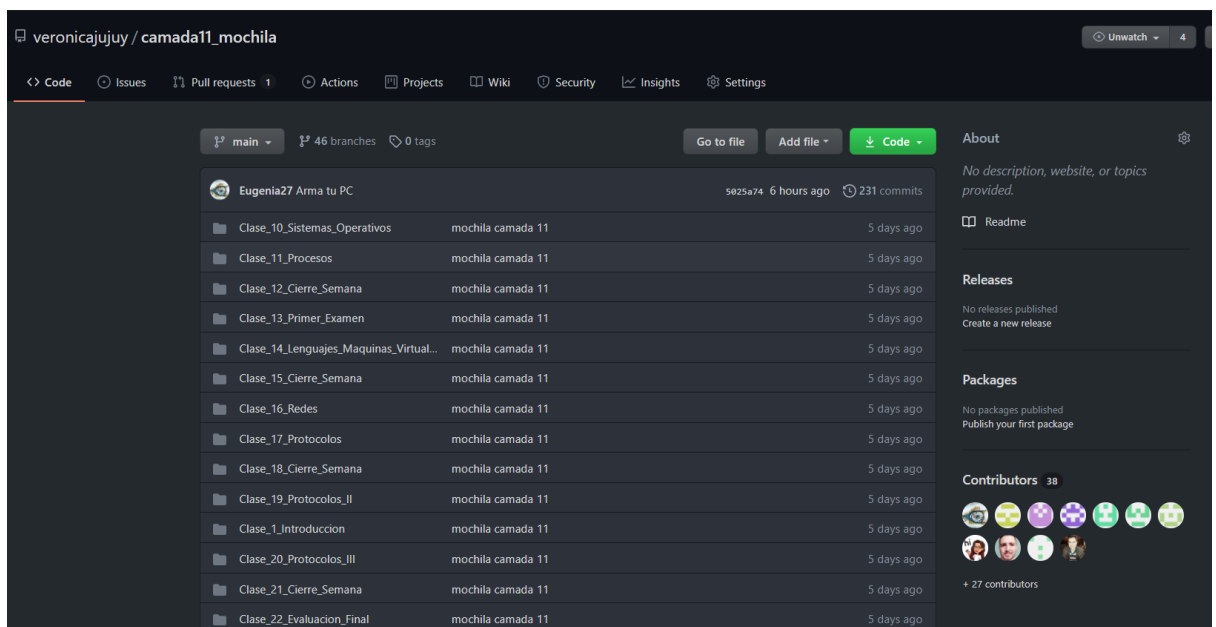


3. Al hacer clic en el botón "view invitation" te llevará a tu github para que puedas aceptar la invitación:



Es importante que antes de este paso ya te hayas logueado a github con tu usuario y contraseña.

4. Una vez que hayas aceptado la invitación te aparecerá el repositorio correspondiente a tu camada:



Antes de clonar el repositorio

5. Chequeá que esté configurado tu usuario y mail en el github con los comandos:

```
git config user.name
```

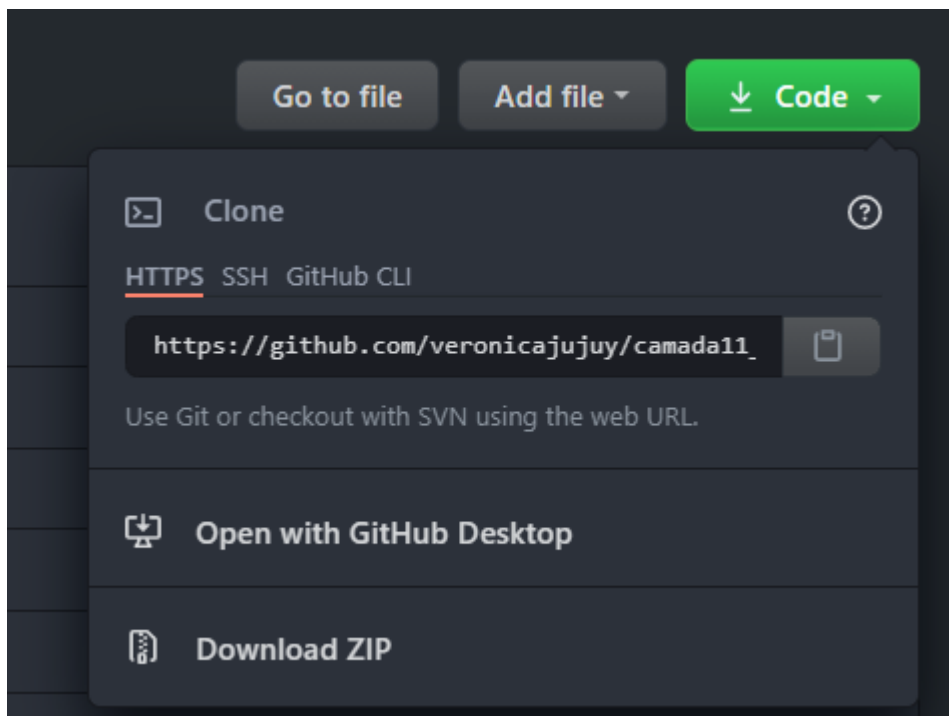
```
git config user.email
```


6. Si no estuviese configurado, es momento que lo hagas, y que el usuario sea el que le pasaste a tu profesor o profesora:

```
git config user.name "nombre_usuario"  
git config user.email "email@email.com"
```

Clonando el repositorio en mi computadora

7. Luego, para clonar el repositorio en tu computadora debés hacer clic en el botón **Code** ubicado en el margen superior. Allí se desplegará una ventana con la dirección URL correspondiente al repositorio. ¡Es muy importante que la opción **HTTPS** esté subrayada (Verificá que no esté subrayada la opción SSH).



8. Copia la dirección haciendo clic en en el icono de la carpeta: 
9. En tu computadora, elegí una carpeta para clonar el repositorio. Puede ser la carpeta de la materia, ya que una vez que se ejecute el comando de clonación se creará una carpeta dentro de ella que se llamará como el repositorio de la camada. No hace falta que inicialices esa carpeta con git (no hace falta que hagas git init en esa carpeta).

10. ¡Ahora podés clonar!

Ejecutá el comando `git clone "url_del_repositorio"` en la carpeta elegida:

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH
$ git clone https://github.com/veronicajujuy/camada11_mochila.git
```

11. Si es la primera vez que clonas un repositorio, github te pedirá que te loguees.

En ese caso deberás introducir tu usuario y contraseña donde te indique o te redireccionará a una página en donde deberás aceptar la conexión.

12. Una vez clonado te aparecerá en tu terminal algo como esto:

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH
$ git clone https://github.com/veronicajujuy/camada11_mochila.git
Cloning into 'camada11_mochila'...
remote: Enumerating objects: 1000, done.
remote: Counting objects: 100% (493/493), done.
remote: Compressing objects: 83% (200/240) eceiving objects: 0% (1/10)
remote: Compressing objects: 93% (224/240) eceiving objects: 1% (10/1)
remote: Compressing objects: 100% (240/240), done.
remote: Total 1000 (delta 325), reused 405 (delta 250), pack-reused 507R

Receiving objects: 100% (1000/1000), 11.83 MiB | 1.67 MiB/s, done.
Resolving deltas: 100% (490/490), done.

User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH
$
```

13. ¡Ahora ya puedes acceder a la mochila! Si ejecutas el comando `ls`, te darás cuenta que habrá una nueva carpeta con el nombre del repositorio clonado.

Para acceder a él, solo deberás cambiar de carpeta y posicionarte en la nueva:

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH
$ cd camada11_mochila/

User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila (main)
$
```

14. Te darás cuenta que estas en la carpeta correcta porque al lado del nombre de la mochila aparecerá en color celeste la rama principal "main".

Si listas las carpetas te aparecerán todas las carpetas de las clases donde deberás, a partir de ahora, subir la práctica realizada en las mismas:

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila (main)
$ ls
Clase_10_Sistemas_Operativos/
Clase_11_Procesos/
Clase_12_Cierre_Semana/
Clase_13_Primer_Examen/
Clase_14_Lenguajes_Maquinas_Virtuales/
Clase_15_Cierre_Semana/
Clase_16_Redes/
Clase_17_Protocolos/
Clase_18_Cierre_Semana/
Clase_19_Protocolos_II/
Clase_1_Introduccion/
Clase_20_Protocolos_III/
Clase_21_Cierre_Semana/
Clase_22_Evaluacion_Final/
Clase_23_Amenazas_Informaticas/
Clase_24_Cierre_Semana/
Clase_25_Etica_Informatica/
Clase_26_Cierre/
```

Creando mi rama y subiendo las ejercitaciones

15. Para crear una rama utilizamos el comando

```
git branch <nombre-rama>
```

Para la mochila crearemos una rama con nuestro apellido y nombre: Ejemplo:

```
git branch Potter_Harry
```

Luego nos cambiamos a esa rama utilizando el comando

```
git checkout <nombre-rama>
```

En nuestro caso:

```
git checkout Potter_Harry
```

16. Podremos visualizar que estamos en nuestra rama porque en vez de (main) nos figurará nuestro apellido y nombre:

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila (main)
$ git branch Potter-Harry

User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila (main)
$ git checkout Potter-Harry
Switched to branch 'Potter-Harry'

User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila (Potter-Harry)
$
```

17. Ingresamos a la clase correspondiente del día. Si no está especificado un nombre en particular crearemos la carpeta:

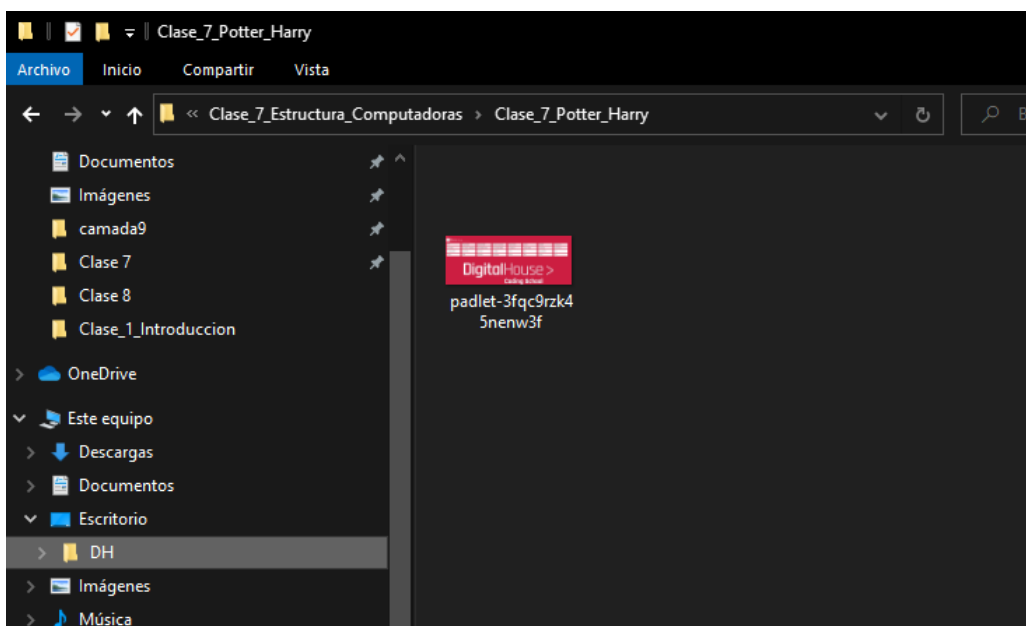
Clase_X_Apellido_y_Nombre

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila (Potter-Harry)
$ cd Clase_7_Estructura_Computadoras/

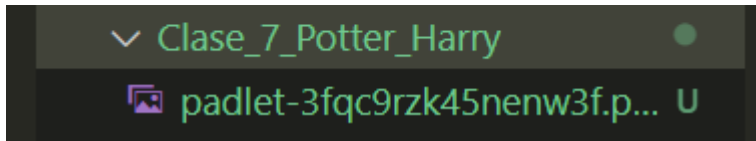
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Computadoras (Potter-Harry)
$ mkdir Clase_7_Potter_Harry

User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Computadoras (Potter-Harry)
$
```

18. Allí crearemos o subiremos el contenido trabajado en clases. Si se trata de un archivo .jpg .png o .pdf podés copiarlo directamente desde el explorador de archivos:



19. En el Visual Studio code, nos aparecerá que tenemos un nuevo archivo Untracked:



20. Si corremos el comando git status nos aparecerá la nueva carpeta creada lista para hacerle un add:

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Computadoras (Potter-Harry)
$ git status
On branch Potter-Harry
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Clase_7_Potter_Harry/

nothing added to commit but untracked files present (use "git add" to track)
```

21. Es hora de hacer el git add . y el commit

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Computadoras (Potter-Harry)
$ git add .

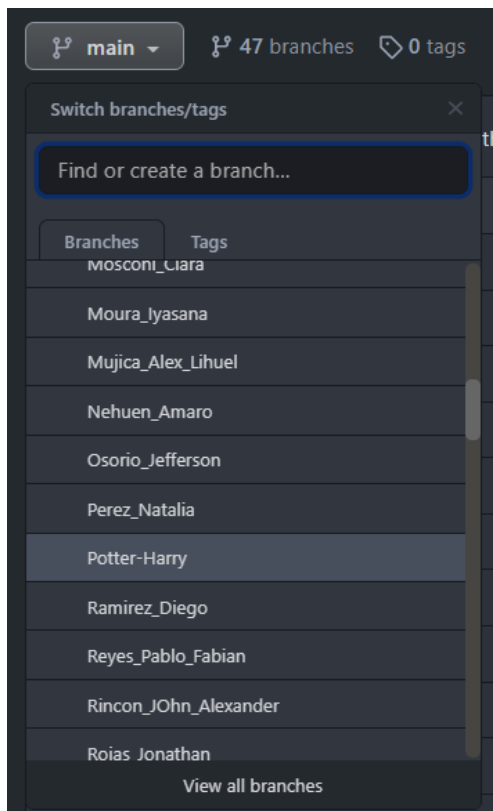
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Computadoras (Potter-Harry)
$ git commit -m "archivo clase 7 agregado"
[Potter-Harry c4e76a3] archivo clase 7 agregado
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Clase_7_Estructura_Computadoras/Clase_7_Potter_Harry/padlet-3fqc9rzk45nenw3f.png
```

22. Una vez que hicimos el commit podemos pushear a nuestra rama con el comando:

```
git push origin <nombre-rama>
```

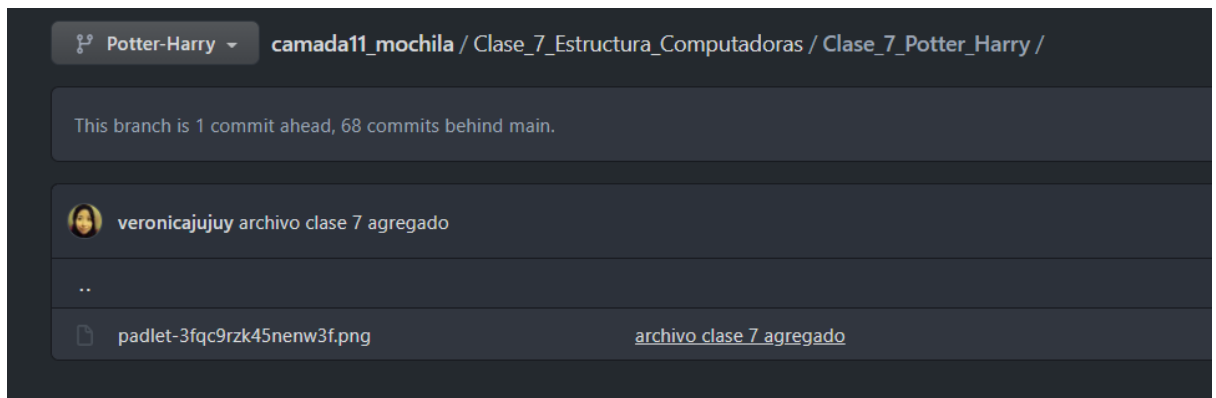
```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Compu
tadoras (Potter-Harry)
$ git push origin Potter-Harry
Enumerating objects: 90, done.
Counting objects: 100% (85/85), done.
Delta compression using up to 4 threads
Compressing objects: 100% (40/40), done.
Writing objects: 100% (67/67), 1004.04 KiB | 363.00 KiB/s, done.
Total 67 (delta 30), reused 60 (delta 26), pack-reused 0
remote: Resolving deltas: 100% (30/30), completed with 8 local objects.
remote:
remote: Create a pull request for 'Potter-Harry' on GitHub by visiting:
remote:   https://github.com/veronicajujuy/camada11_mochila/pull/new/Potter-Harr
y
remote:
To https://github.com/veronicajujuy/camada11_mochila.git
 * [new branch]      Potter-Harry -> Potter-Harry
```

23. Podemos ver en el repositorio remoto que nuestra rama fue creada y está listada junto a la rama de nuestros compañeros:



Si selecciono la rama, puedo ver el último commit que hice y el archivo en la

carpeta correspondiente a la clase:



Haciendo un merge local

24. Luego de hacer el push a mi rama, mergearemos la misma con la rama main, para eso nos cambiamos de rama con:

```
git checkout main
```

25. Será conveniente que traigamos los cambios del repositorio remoto primero, antes de hacer un push. Seguramente nuestros compañeros habrán subido muchos cambios desde que clonamos:

(Seguramente se cargarán muchas carpetas, no te preocupes que es normal)

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Compu
tadoras (main)
$ git pull origin main
remote: Enumerating objects: 307, done.
remote: Counting objects: 100% (297/297), done.
remote: Compressing objects: 100% (201/201), done.
remote: Total 277 (delta 115), reused 236 (delta 74), pack-reused 0
Receiving objects: 100% (277/277), 7.90 MiB | 1.88 MiB/s, done.
Resolving deltas: 100% (115/115), completed with 11 local objects.
From https://github.com/veronicajujuy/camada11_mochila
 * branch          main      -> FETCH_HEAD
    5025a74..6fae2ac  main      -> origin/main
Updating 5025a74..6fae2ac
Fast-forward
 .../Clase_7_Amaro_Nehuen/clase_7_amaro_nehuen.txt      | 3 +++
 .../Clase_7_Jimenez_jeifer/Clase_7_Jimenez_Jeifer.PNG | Bin 0 -> 323970 bytes
 .../Clase_7_Jimenez_Jeifer.docx                       | Bin 0 -> 14125 bytes
 Clase_7_Estructura_Computadoras/Clase_7_Ramirez_Diego | 1 +
 .../Clase_7_VilchesBrenda/Clase7_VilchesBrenda.PNG    | Bin 0 -> 439997 bytes
 .../Clase_7_VilchesBrenda/Clase7_VilchesBrenda.txt    | 18 +++++
 Clase 8 Memorias/Clase 8 Rivera Pablo/Taller 1.PNG    | Bin 0 -> 44443 bytes
```

26. Este comando solo me traerá los cambios en la rama main, si hago git status me tendria que salir todo limpio:

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Compu
tadoras (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

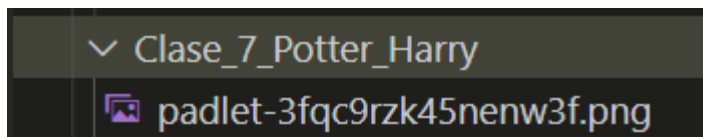
nothing to commit, working tree clean
```

27. ¡Es hora de mergear! Para ello utilizamos el comando:

```
git merge <nombre-rama>
```

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Compu
tadoras (main)
$ git merge Potter-Harry
Merge made by the 'recursive' strategy.
.../Clase_7_Potter_Harry/padlet-3fqc9rz45nenw3f.png | Bin 0 -> 427224 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Clase_7_Estructura_Computadoras/Clase_7_Potter_Harry/padlet-3fq
c9rz45nenw3f.png
```

28. Ahora mi carpeta y su contenido estarán en la rama main:



29. Si ejecuto un git status luego de hacer el merge, solo debería avisarme que la rama remota está desactualizada y debo hacer un git push con los cambios:

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Compu
tadoras (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
```

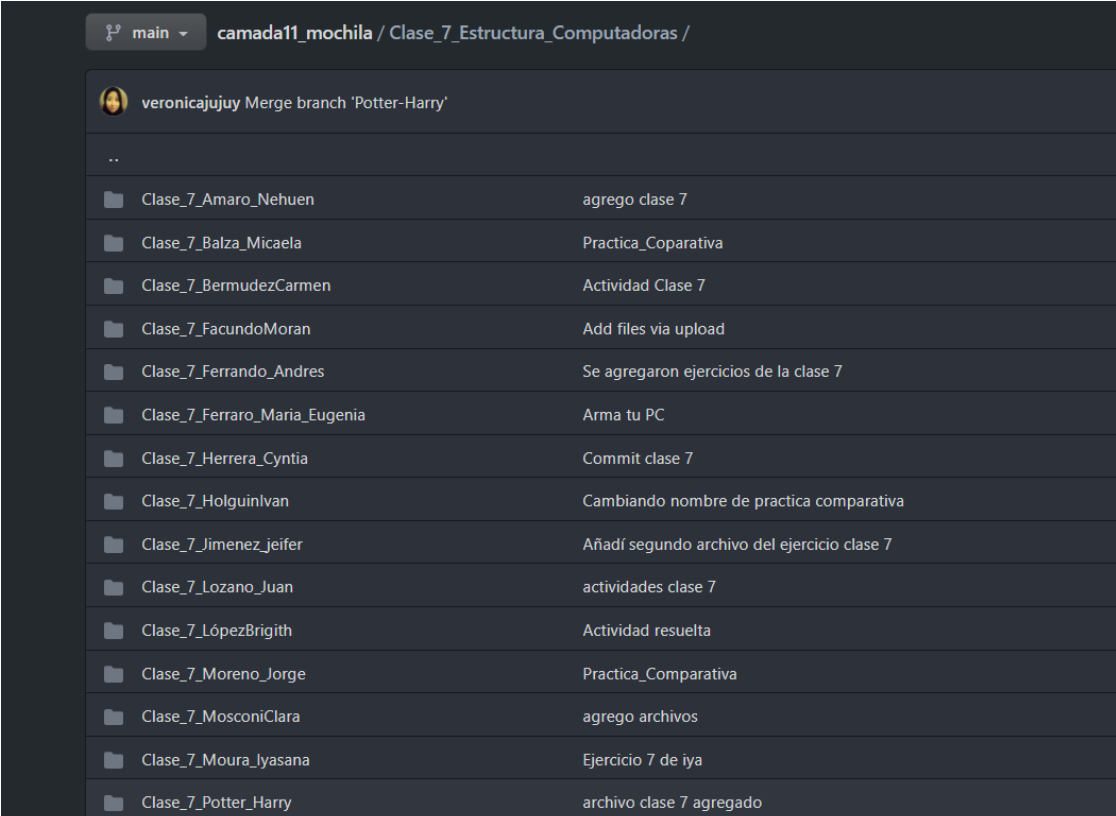
30. Si hago un git log --oneline veremos que nuestro merge generó un nuevo commit automáticamente:

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Compu
tadoras (main)
$ git log --oneline
06f78fb (HEAD -> main, origin/main, origin/HEAD) Merge branch 'Potter-Harry'
c4e76a3 (origin/Potter-Harry, Potter-Harry) archivo clase 7 agregado
6fae2ac Merge branch 'Rojas_Jonathan'
```

31. Ahora puedo actualizar el repositorio remoto con mis cambios ya en la rama main:

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Computadoras (main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 372 bytes | 372.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/veronicajujuy/camada11_mochila.git
6fae2ac..06f78fb main -> main
```

32. Posteriormente, podemos chequear el repositorio remoto y ya estará mi carpeta y su contenido en la rama main:



The screenshot shows the GitHub interface for the repository 'camada11_mochila / Clase_7_Estructura_Computadoras'. At the top, it indicates a merge of the 'Potter-Harry' branch into 'main' by user 'veronicajujuy'. Below this, a table lists the files and folders that were added or modified during the merge.

File/Folder	Description
Clase_7_Amaro_Nehuen	agrego clase 7
Clase_7_Balza_Micaela	Practica_Coparativa
Clase_7_BermudezCarmen	Actividad Clase 7
Clase_7_FacundoMoran	Add files via upload
Clase_7_Ferrando_Andres	Se agregaron ejercicios de la clase 7
Clase_7_Ferraro_Maria_Eugenia	Arma tu PC
Clase_7_Herrera_Cyntia	Commit clase 7
Clase_7_HolguinIvan	Cambiando nombre de practica comparativa
Clase_7_Jimenez_jeifer	Añadí segundo archivo del ejercicio clase 7
Clase_7_Lozano_Juan	actividades clase 7
Clase_7_LópezBrigith	Actividad resuelta
Clase_7_Moreno_Jorge	Practica_Comparativa
Clase_7_MosconiClara	agrego archivos
Clase_7_Moura_Iyasana	Ejercicio 7 de iya
Clase_7_Potter_Harry	archivo clase 7 agregado

Agregando más archivos a mi rama

Cuando tengamos que agregar más archivos porque vamos avanzando en las clases, vamos a seguir los siguientes pasos:

1. Nos dirigimos a la rama que creamos anteriormente con el comando:
`git checkout <nombre-rama>`
2. Si hacemos un `git log --oneline` veremos que el último commit que nos aparece en la rama main no está presente en mi rama:

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Computadoras (Potter-Harry)
$ git log --oneline
c4e76a3 (HEAD -> Potter-Harry, origin/Potter-Harry) archivo clase 7 agregado
5025a74 (origin/Ferraro_Maria_Eugenia) Arma tu PC
48e028e Update PracticaComparativa.md
81bfdbf Update PracticaComparativa.md
```

Eso quiere decir que mi rama está un commit por detrás de la rama main, está desactualizada. Para emparejar las ramas, tengo que mergear la rama master a mi rama. El proceso inverso que hice anteriormente:

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Computadoras (Potter-Harry)
$ git merge main
Updating c4e76a3..06f78fb
Fast-forward
 .../Clase_7_Amaro_Nehuen/clase_7_amaro_nehuen.txt      | 3 +++
 .../Clase_7_Jimenez_jeifer/Clase_7_Jimenez_Jeifer.PNG | Bin 0 -> 323970 bytes
 .../Clase_7_Jimenez_Jeifer.docx                       | Bin 0 -> 14125 bytes
 Clase_7_Estructura_Computadoras/Clase_7_Ramirez_Diego | 1 +
 .../Clase_7_VilchesBrenda/Clase7_VilchesBrenda.PNG   | Bin 0 -> 439997 bytes
 .../Clase_7_VilchesBrenda/Clase7_VilchesBrenda.txt   | 18 +++++
 Clase_8_Memorias/Clase_8_Rivera_Pablo/Taller 1.PNG    | Bin 0 -> 44443 bytes
```

3. Probablemente también me actualice con las carpetas de los compañeros que actualice en la rama main. ¡No te preocupes, esto es correcto!
4. Si hago un `git log`, ahora ya estoy nuevamente al día con la rama main:

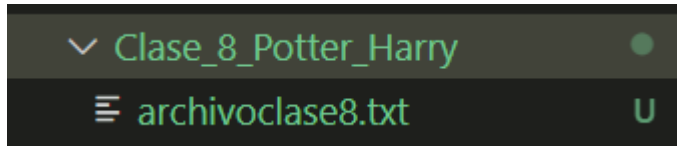
```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_7_Estructura_Computadoras (Potter-Harry)
$ git log --oneline
06f78fb (HEAD -> Potter-Harry, origin/main, origin/HEAD, main) Merge branch 'Potter-Harry'
c4e76a3 (origin/Potter-Harry) archivo clase 7 agregado
```

5. ahora puedo repetir el proceso desde el punto 17!!

Mergeando desde el repositorio Remoto

Hay otra opción para hacer un merge directamente desde github, para ello debemos seguir los siguientes pasos:

6. Nos cambiamos de rama, actualizamos la misma con un merge y agregamos un nuevo archivo a alguna carpeta.



7. Hacemos un git add, un git commit tal cual vinimos haciendo anteriormente:

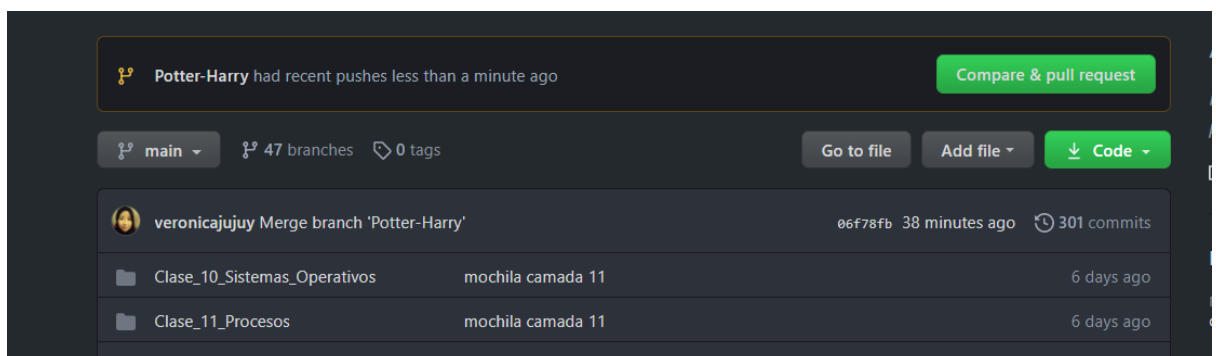
```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_8_Memorias (Potter-Harry)
$ git add .

User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_8_Memorias (Potter-Harry)
$ git commit -m "archivo clase 8 agregado"
[Potter-Harry d9f5f44] Âarchivo clase 8 agregado
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Clase_8_Memorias/Clase_8_Potter_Harry/archivoclase8.txt
```

8. Ahora pusheamos desde nuestra rama:

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_8_Memorias (Potter-Harry)
$ git push origin Potter-Harry
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 411 bytes | 205.00 KiB/s, done.
Total 5 (delta 2), reused 1 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/veronicajujuy/camada11_mochila.git
c4e76a3..d9f5f44 Potter-Harry -> Potter-Harry
```

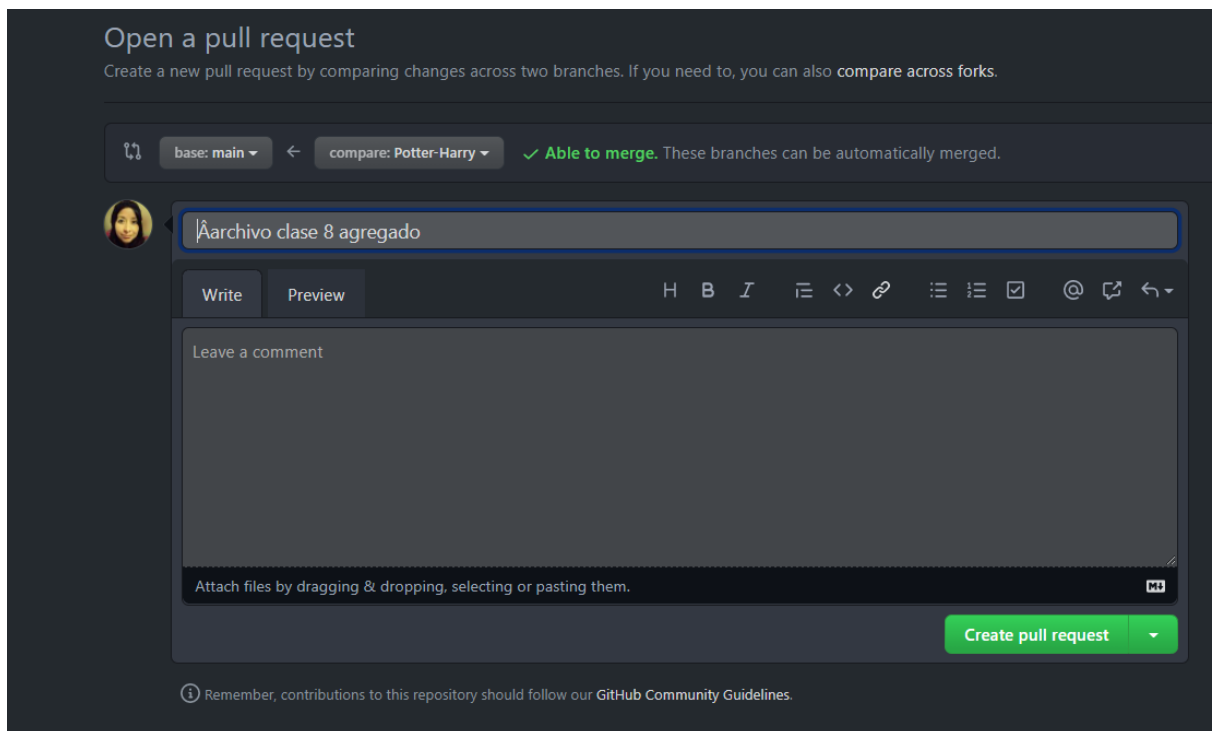
9. Si vamos a github, al repositorio de la camada me debería aparecer lo siguiente:



10. Eso quiere decir que mi cambio está subido a mi rama y puedo mergearlo.

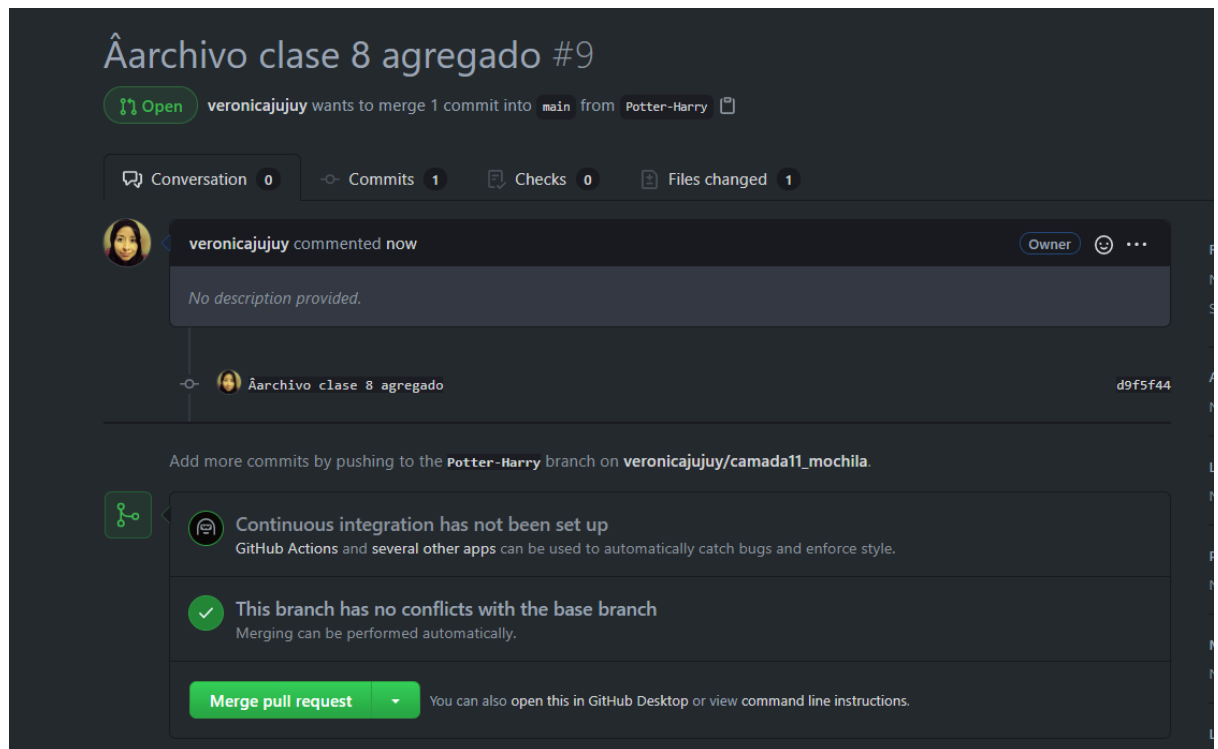
Anteriormente habíamos realizado ese merge localmente desde nuestra terminal en el visual studio code, ahora probaremos mergear desde el repositorio remoto.

Para ello, hacemos clic en el botón **Compare & Pull request**. Allí ingresamos a una nueva ventana donde nos indicará que haremos un pull request. *Pull request* es la acción de validar un código que se va a mergear de una rama a otra.

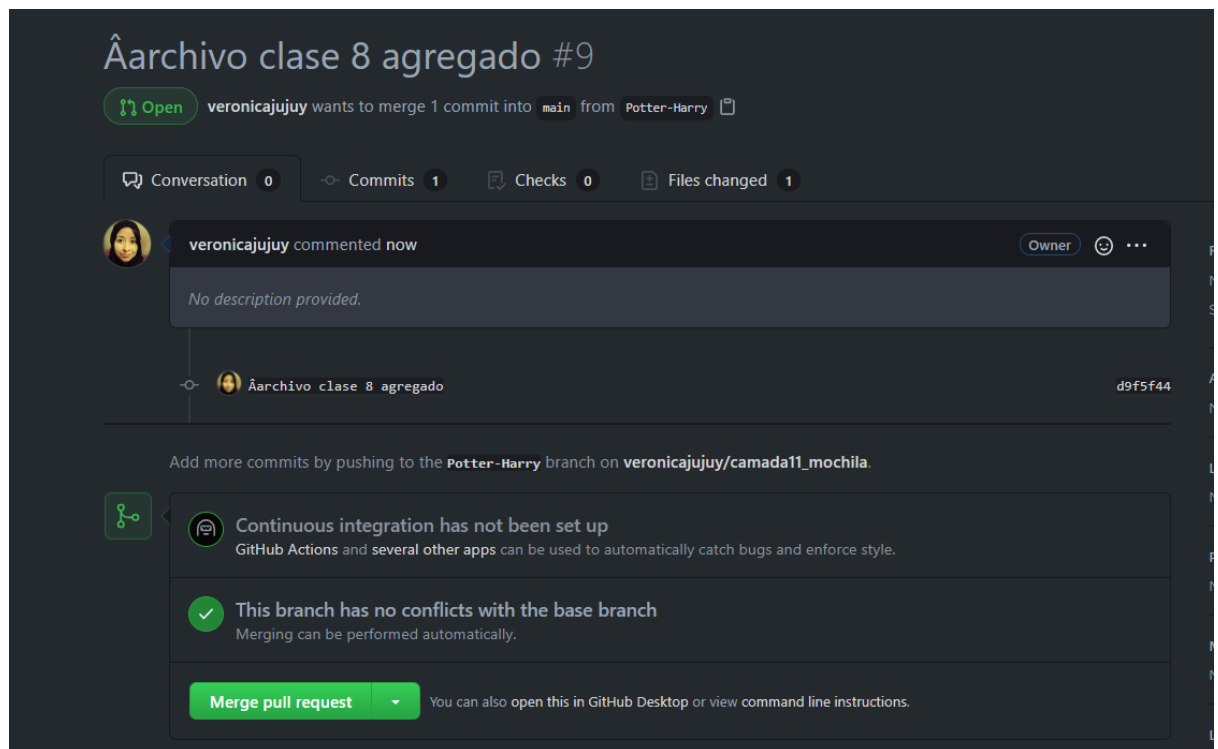


11. En la ventana vemos que nos da la opción de elegir desde qué rama a que rama voy a mergear, en este caso dejo tal cual están las ramas por defecto que son: base: main y compare: Potter-Harry. Luego presiono el botón Create pull Request

12. Nos aparecerá una nueva ventana que nos habilitará a mergear la rama:



Si existiesen conflictos van a aparecer y deberemos arreglarlos. En general no debería suceder ya que estamos trabajando en nuestras propias carpetas.



13. Luego, presiono el botón Merge pull request y el botón Merge pull request cambiará a “confirm merge”.



14. La ventana con una leyenda Merged en color morado, confirmará que está todo bien y ya mergeamos nuestra rama.

15. Queda un paso más y es bajar ese merge a mi repositorio local. En el Visual Studio code, y si todavía estoy en mi rama, hago un checkout al main:

```
git checkout main
```

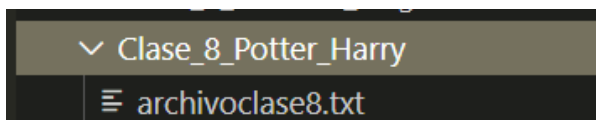
```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_8_Memorias (Potter-Harry)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_8_Memorias (main)
$
```


16. Posteriormente, hacemos un git pull de los cambios haciendo git pull origin main:

```
User@DESKTOP-8JN8B10 MINGW64 ~/Desktop/DH/camada11_mochila/Clase_8_Memorias (main)
$ git pull origin main
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 651 bytes | 162.00 KiB/s, done.
From https://github.com/veronicajujuy/camada11_mochila
* branch          main          -> FETCH_HEAD
   06f78fb..40a3995  main          -> origin/main
Updating 06f78fb..40a3995
Fast-forward
 Clase_8_Memorias/Clase_8_Potter_Harry/archivoclase8.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Clase_8_Memorias/Clase_8_Potter_Harry/archivoclase8.txt
```

17. Debe aparecer en tu rama main el mergeo que hice en el repositorio remoto
18. Podemos comprobar esto en la estructura de archivos de visual:



19. ¡Muy bien! Ahora podemos volver a nuestra rama y continuar agregando archivos siguiendo el paso 1 del apartado: **Agregando más archivos a mi rama.**

¡Esperamos que esta guía te ayude a trabajar correctamente en tu mochila!