



CREA UNA CARPETA Y SUBE TUS SOP'S Y ARCHIVOS

Propósito del documento

Este documento tiene como propósito el ser una guía práctica y sencilla para que nuevos usuarios de GitHub sean capaces de crear una carpeta dentro del repositorio, más específicamente dentro de su rama, para que después puedan comenzar a subir sus trabajos en ese apartado.

Escenarios en donde se aplica

Este documento se aplica para aquellos usuarios relativamente nuevos en GitHub sean capaces de crear una carpeta para comenzar a subir sus trabajos dentro de su Branch en GitHub.

Definiciones y acrónimos

GitHub	Es una plataforma en línea que permite a los usuarios almacenar, compartir y trabajar en proyectos de software.
VScode	Es un editor de código para programadores gratuito, de código abierto y multiplataforma.
Git	Git es un sistema de control de versiones. Permite a los desarrolladores controlar y administrar los cambios en su código fuente a lo largo del tiempo.
Repositorio de GitHub	Es un espacio de almacenamiento virtual donde se puede guardar el código, los archivos y el historial de revisiones de un proyecto.
Rama (branch)	División del estado del código, que se encarga de desarrollar un nuevo mecanismo para su evolución.
Comando ls	Funciona para ver la lista de contenido de un directorio.
Comando cd	(change directory) permite moverse entre carpetas y regresar a la carpeta de usuario.
git commit	El comando git commit guardará todos los cambios hechos en la zona de

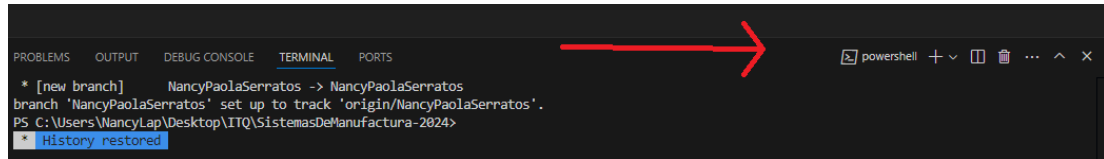


	montaje o área de preparación, junto con una breve descripción del usuario, en un "commit" al repositorio local.
git push	Te permite subir los commits desde tu rama (branch) local en tu repositorio git local al repositorio remoto.
Powershell	Es la herramienta de Windows que nos permite dar instrucciones a través de comandos.
Git bash	Es una aplicación para Windows que permite a los usuarios utilizar las funciones de Git a través de una línea de comandos
Formato PascalCase	Pascal case combina palabras poniéndolas todas con la primera letra en mayúscula: CosasParaHacer

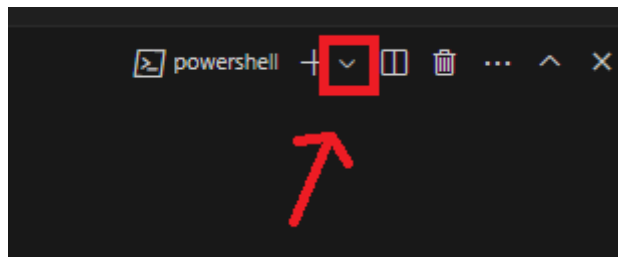


Nota antes de iniciar: Es posible realizar todo este procedimiento desde el apartado de git bash en VScode.

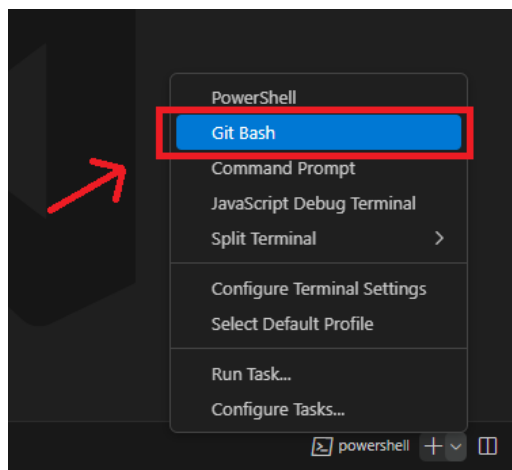
Para entrar al apartado de bash en VScode te diriges al lado derecho de la pantalla de inicio de VScode, cerca de los apartados de terminal y ports.



Das clic al siguiente botón.



Se desplegarán una serie de opciones y se selecciona el apartado de Git bash



Después de seleccionar git bash se abrirá el siguiente formato y puedes empezar a trabajar desde git bash, ya que trabajar desde ahí te permitirá tener más claridad desde que rama estas trabajando y no arriesgarse a que accidentalmente estes trabajando desde la rama de main.



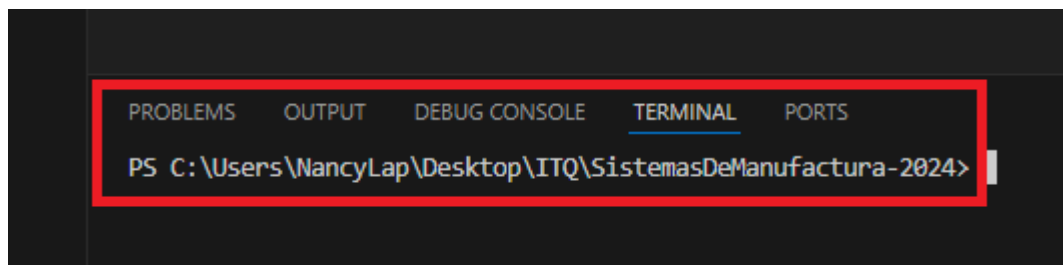


Nota: El procedimiento desde git bash y powershell es el mismo, solo cambia el formato que aparece en la terminal por lo que no hay problema desde donde realices los pasos.

Las capturas del proceso son desde powershell cuando las letras son blancas y cuando son de colores son desde git bash.

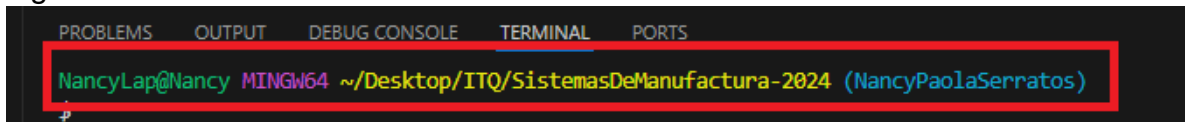
INICIO DEL PROCEDIMIENTO PARA LA CARPETA

1. Para comenzar a crear una rama, es necesario que la dirección de la terminal siga la siguiente estructura.



(Si no aparece el apartado de la **terminal** en VScode, presionar las teclas Ctrl+ñ, ambas al mismo tiempo para que aparezca el apartado).

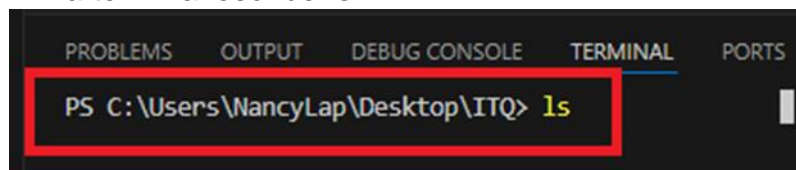
En caso de estar trabajando desde git bash la estructura tiene que ser la siguiente:



Nota: Al final tiene que estar escrito el nombre de tu Branch, ya que si aparece main quiere decir que estas trabajando en la rama incorrecta y corres el riesgo de que te bajen calificación por modificar archivos desde main.

Si no tiene esas alguna de las estructuras anteriores en la dirección de la terminal (despendiendo desde donde estas trabajando, powershell o git bash), sigue los siguientes pasos.

- 1.1 En la terminal escribe ls.





1.2 Después de presionar enter, el comando nos desplegará información, de la cual tenemos que copiar el nombre del repositorio.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\NancyLap\Desktop\ITQ> ls

Directorio: C:\Users\NancyLap\Desktop\ITQ

Mode                LastWriteTime         Length Name
----                -
d-----          04/09/2024  06:48 p. m.      SistemasDeManufactura-2024
```

1.3 Una vez copiado el nombre, en la terminal escribir el comando `cd` seguido del nombre del repositorio anteriormente copiado y presionar enter.

Ejemplo:

`cd SistemasDeManufactura-2024`

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Directorio: C:\Users\NancyLap\Desktop\ITQ

Mode                LastWriteTime         Length Name
----                -
d-----          04/09/2024  06:48 p. m.      SistemasDeManufactura-2024

PS C:\Users\NancyLap\Desktop\ITQ> cd SistemasDeManufactura-2024
```

1.4 Una vez presionado enter, podrás notar que la dirección ha cambiado, esto quiere decir que se ingresó a la carpeta del repositorio.

```
PS C:\Users\NancyLap\Desktop\ITQ> cd SistemasDeManufactura-2024
PS C:\Users\NancyLap\Desktop\ITQ\SistemasDeManufactura-2024>
```



1.5 Ahora en la terminal escribe git branch (esto te mostrara la rama principal main y tu rama).

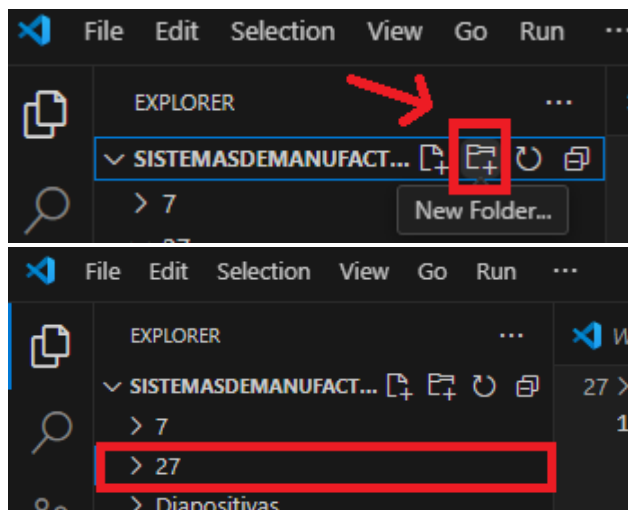
```
PS C:\Users\NancyLap\Desktop\ITQ> cd SistemasDeManufactura-2024
PS C:\Users\NancyLap\Desktop\ITQ\SistemasDeManufactura-2024> git checkout -b NancyPaolaSerratos
Switched to a new branch 'NancyPaolaSerratos'
PS C:\Users\NancyLap\Desktop\ITQ\SistemasDeManufactura-2024> git branch
* NancyPaolaSerratos
main
PS C:\Users\NancyLap\Desktop\ITQ\SistemasDeManufactura-2024>
```

NOTA: Recuerda que main y tu rama (que en la imagen aparece en verde) son 2 ramas distintas, **no puedes trabajar desde main**, siempre tienes que estar trabajando desde tu rama (el color verde te indica en que rama estas trabajando, es decir el nombre de la rama que se encuentra en verde es la rama en donde te encuentras en caso de estar trabajando desde powershell).

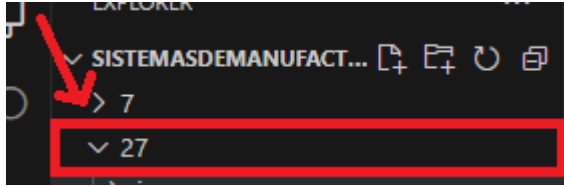
Si estas trabajando desde git bash, el nombre de tu rama aparecerá al final de la dirección de la terminal.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
NancyLap@Nancy MINGW64 ~/Desktop/ITQ/SistemasDeManufactura-2024 (NancyPaolaSerratos)
#
```

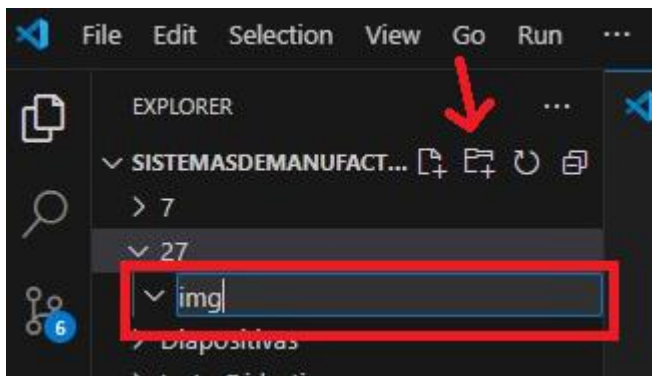
2. Una vez hecho esto, te tienes que dirigir a la parte superior de VScode, donde esta la carpeta de sistemas de manufactura. Y das clic en crear una nueva carpeta a la cual nombraras con tu número de lista y será tu carpeta principal.



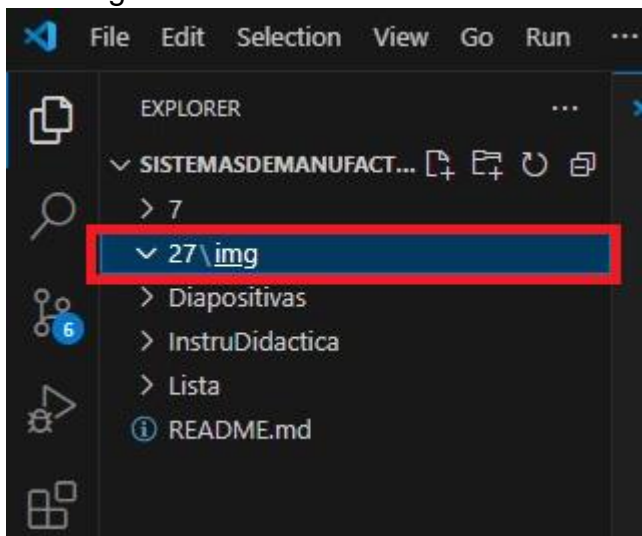
3. Una vez creada tu carpeta principal, dar clic en la carpeta para poder crear los otros archivos, pero ya dentro de tu carpeta (al dar clic en tu carpeta principal, la flecha que está a la izquierda tiene que quedar viendo hacia abajo).



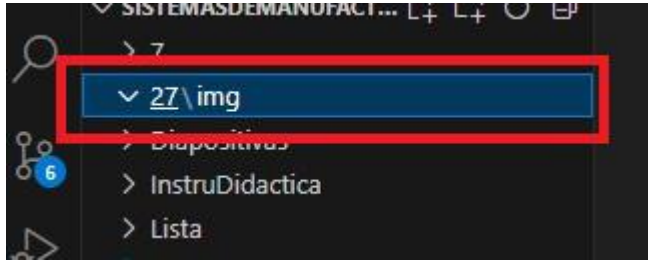
4. Cuando ya tengas desplegada tu carpeta, dar clic nuevamente en crear una nueva carpeta y nombrarla **img**, en ella subirás todos los SOP's, hojas de registro, imágenes y archivos que vamos a usar en el proyecto integrador.



Después de crear la carpeta img, las carpetas que has creado te aparecerán de la siguiente manera:



5. Ahora crearas 2 archivos nuevos, sin embargo, para hacerlo tienes que estar dentro de tu carpeta principal, para entrar a tu carpeta principal solo tienes que seleccionar el nombre de la carpeta:



Nota: te darás cuenta que estas en tu carpeta principal porque el nombre de la carpeta estará subrayado.

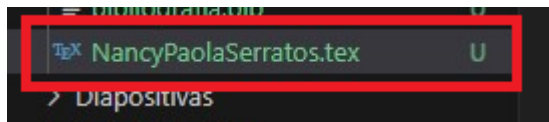
5.1 Para crear archivos solo tienes que dar clic en siguiente apartado:



5.2 El primer archivo llevará tu nombre en formato PascalCase y tendrá como terminación **.tex**

Ejemplo:

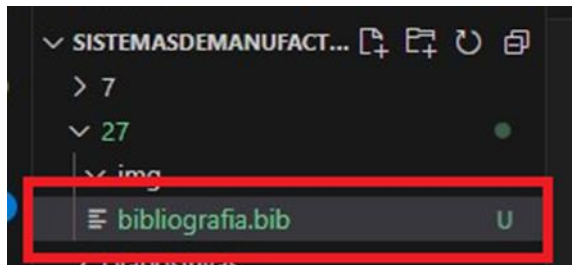
NancyPaolaSerratos.tex



5.3 El segundo archivo lo nombraras como bibliografía y tendrá una terminación(**.bib**), en este archivo vas a subir las referencias bibliográficas que vas a ocupar en el proyecto integrador.

Ejemplo:

bibliografía.tex



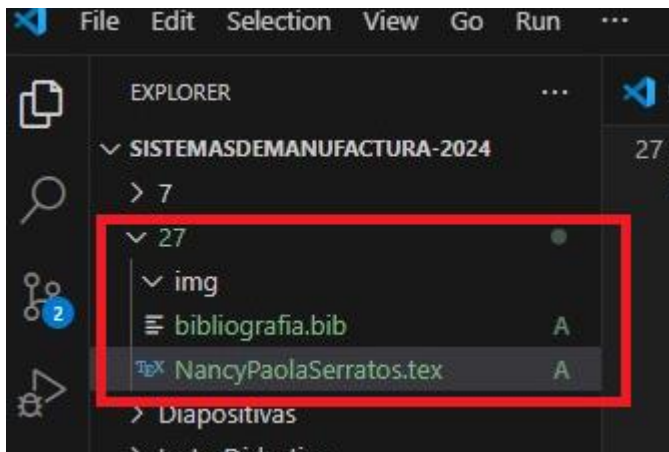
Nota: Después de crear ambos archivos notaremos que del lado de cada archivo hay una letra U y están de color verde, VScode nos esta diciendo que acabamos de agregar varios archivos, sin embargo, aún no están guardados.



6. Ahora en la terminal escribe **git add *** para agregar todas las carpetas y archivos nuevos, este procedimiento siempre se tiene que hacer cuando agreguemos archivos nuevos, para que se guarde lo que hicimos.

```
NancyLap@Nancy MINGW64 ~/Desktop/ITQ/SistemasDeManufatura-2024 (NancyPaolaSerratos)
$ git add *
```

7. Si escribiste el comando correctamente en la terminal, podrás notar que ahora en los nombres de los archivos aparece al lado una **"A"** lo que significa que para guardar los cambios tienes que hacer un commit.



8. Para guardar los cambios con el commit, dejando un mensaje claro que te recuerde porque agregaste los archivos, así podrás dar un seguimiento y trazabilidad de como estas avanzando en el proyecto integrador.

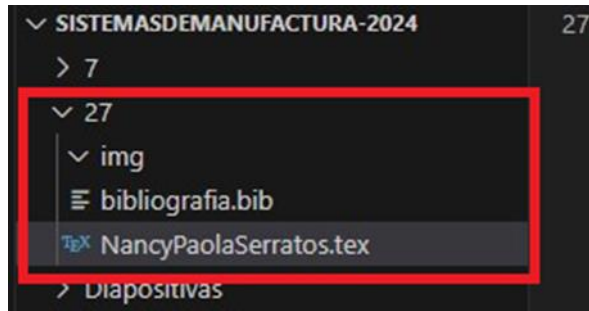
En la terminal escribirás: **git commit -am** "mensaje que vas a poner" y das enter en el teclado.

Ejemplo:

`git commit -am "Esta es mi carpeta"`

```
NancyLap@Nancy MINGW64 ~/Desktop/ITQ/SistemasDeManufatura-2024 (NancyPaolaSerratos)
$ git commit -am "Esta es mi carpeta"
```

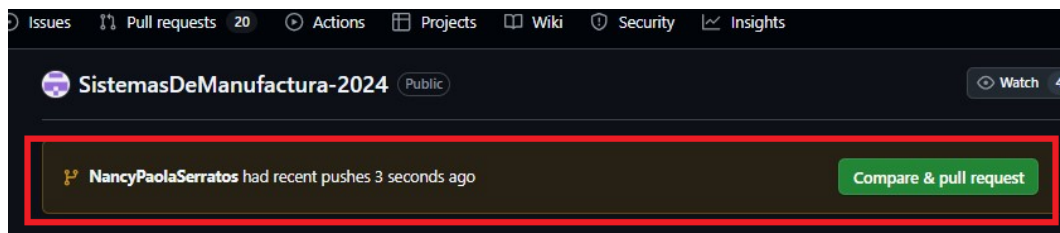
Notaras que el estado de tus archivos cambiara ha guardado y ya no aparecerá la letra **A**.



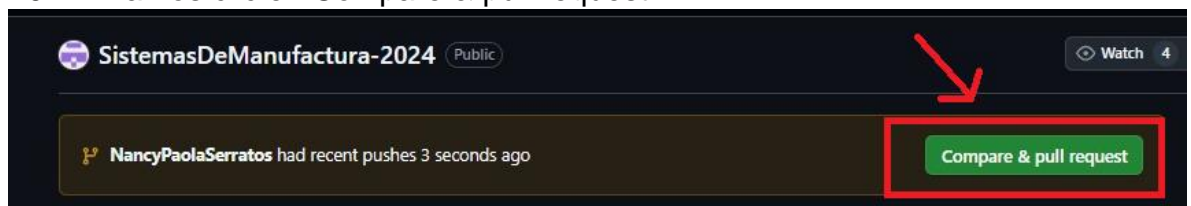
9. Para subir los archivos y que sean visibles en git hub, en la terminal escribe **git push**.



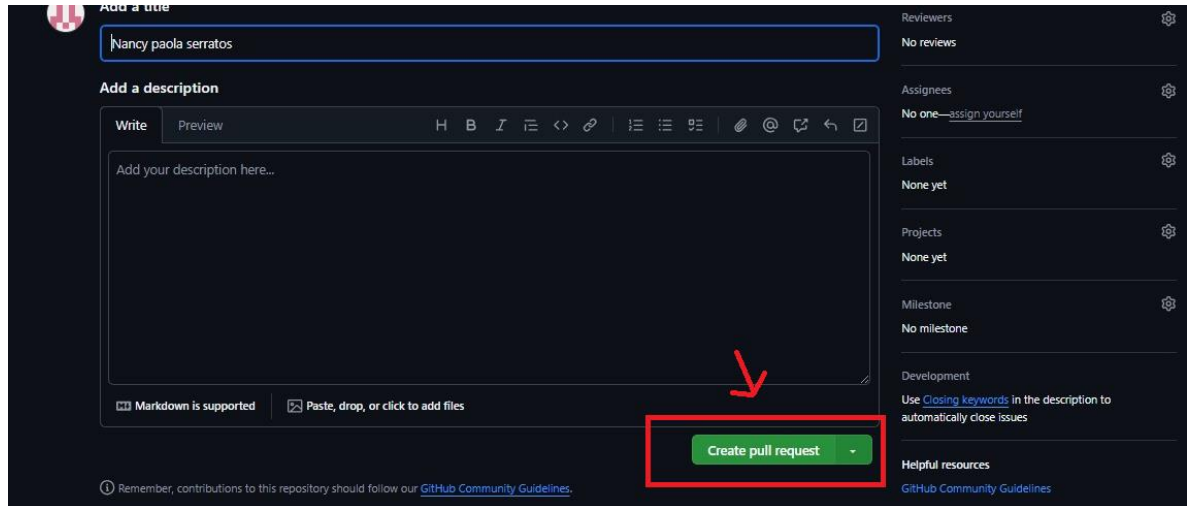
10. Ahora es necesario ir al repositorio y ver que nos aparece lo siguiente, que nos indica que comparemos y hagamos una petición para extraer los cambios.



10.1 Damos clic en Compare & pull request.

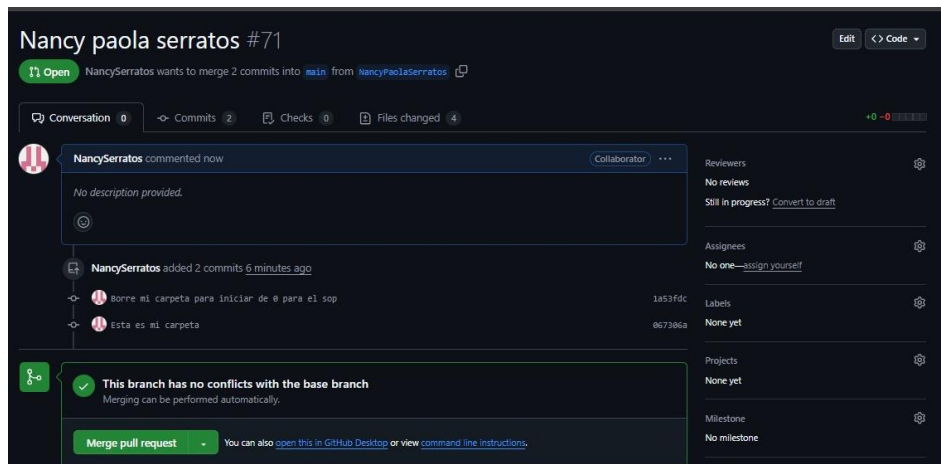


10.2 Ahora da clic en Create pull request.



Nota: En este paso puedes dejar algún mensaje o imagen, dependiendo de las dudas o errores que tengas en tu proyecto.

11. Cuando llegues a este punto podrás darte cuenta que has terminado con el procedimiento y solo tienes que esperar a que el profesor te de observaciones a tus documentos o cambios realizados.



NOTA IMPORTANTE: NO le des MERGE (deja las cosas así) o se te bajaran 10 puntos a tu calificación del tema que se este viendo.