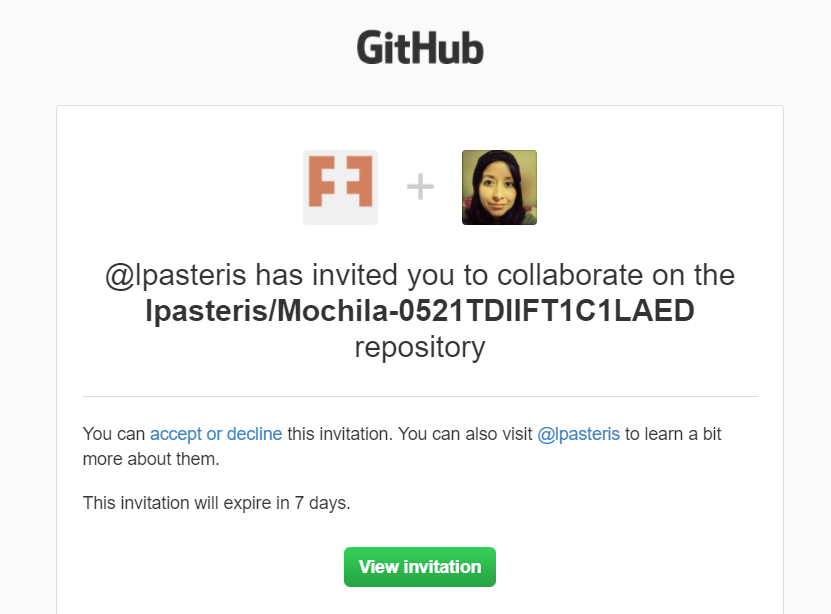
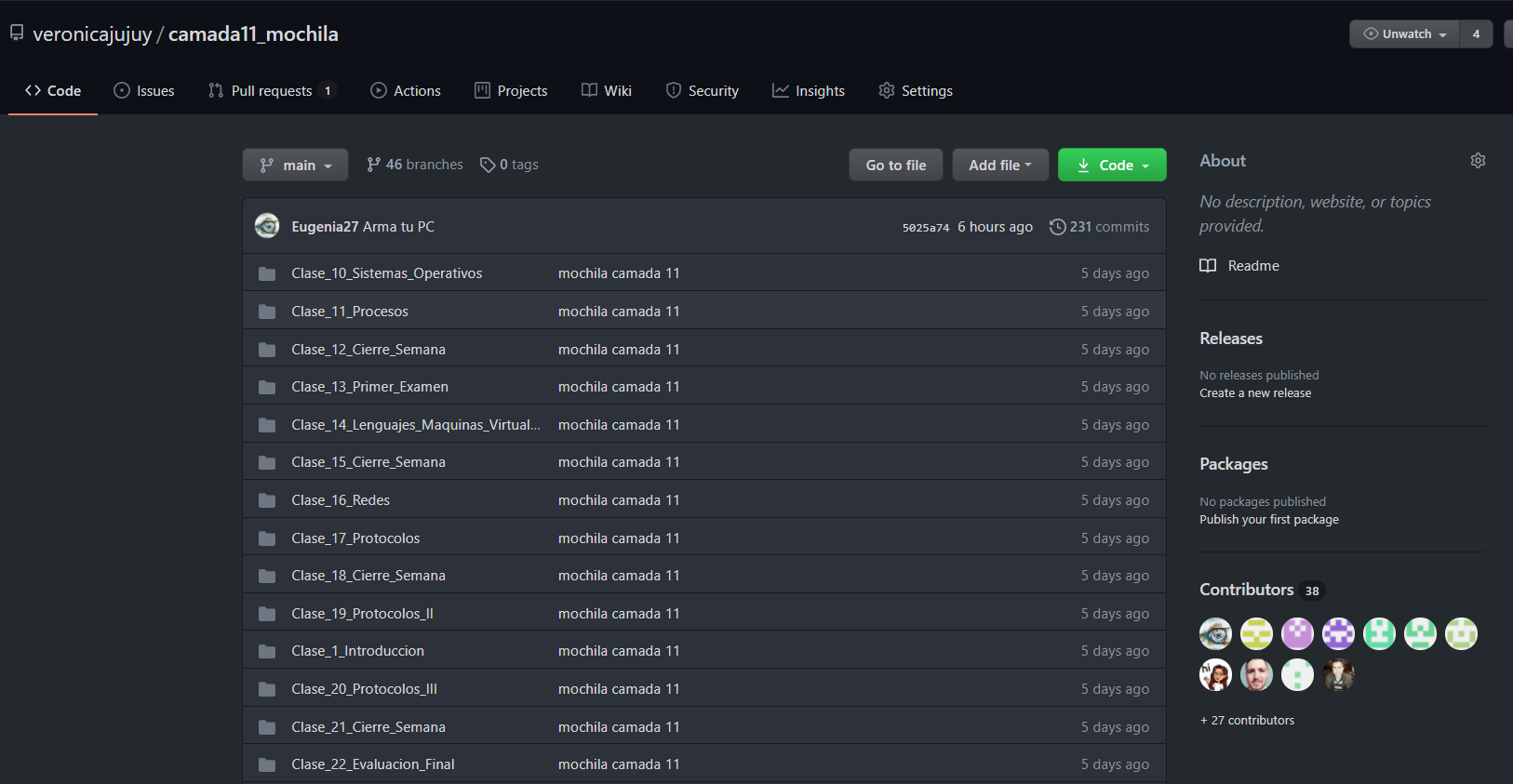
Introducción a la informática

** Git + GitHub**

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

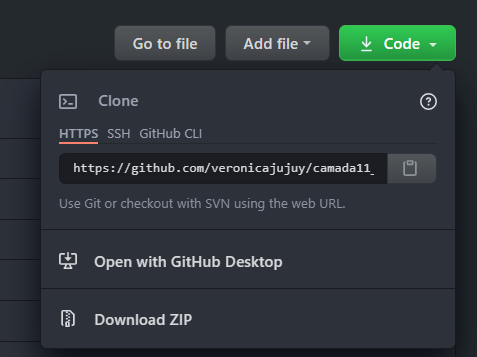
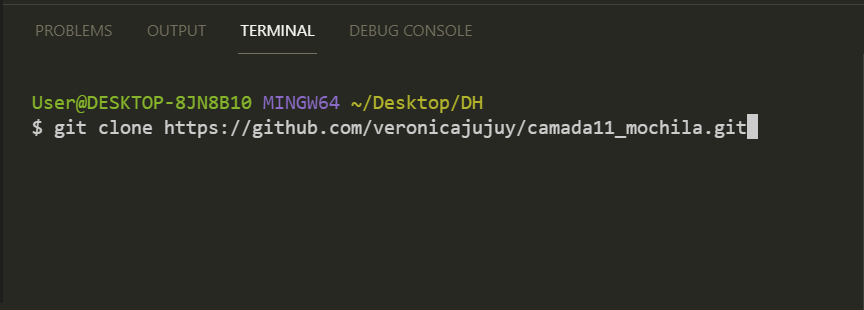
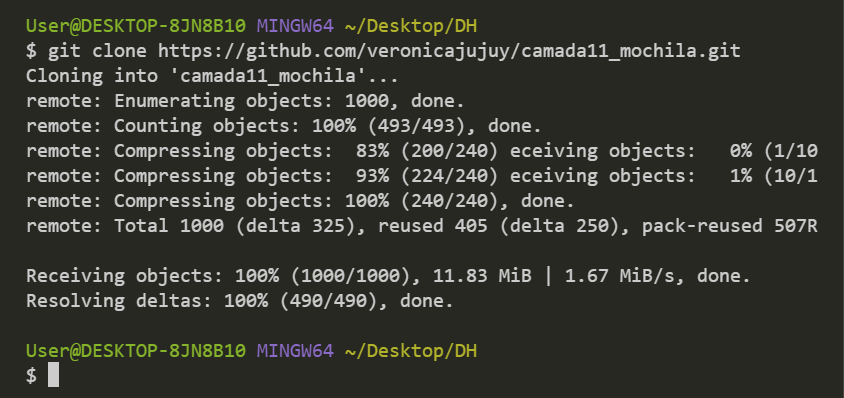
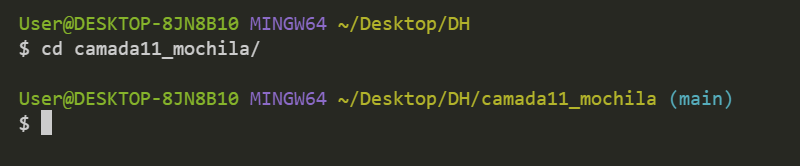
# Primeros pasos

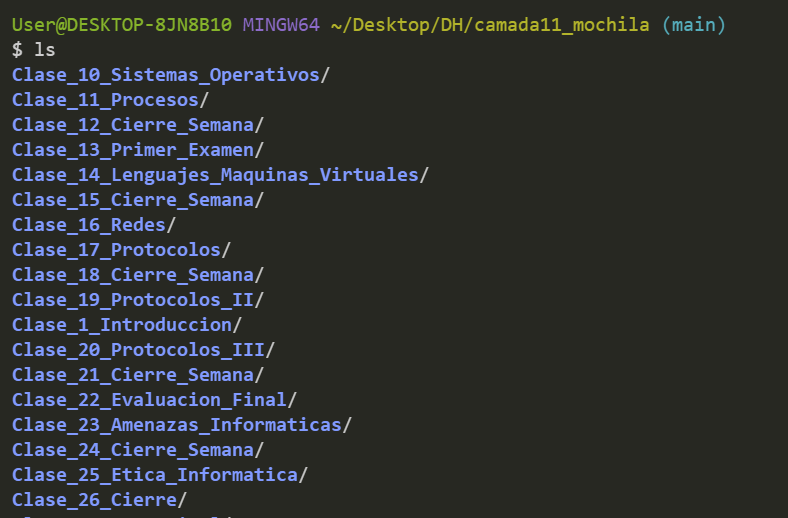
1. Lo primero que tenés que hacer es proporcionarle tu nombre de usuario a tu profesor o profesora para que te agregue como colaborador al repositorio.
2. Una vez que te agregó como colaborador, te llegará un correo electrónico con la invitación:
3. Al hacer clic en el botón ‘’view invitation’’ te llevará 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

1. Chequeá que esté configurado tu usuario y mail en el github con los comandos:  
   git config user.name  
   git config user.email
2. 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

1. 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).  
   
2. Copia la dirección haciendo clic en en el icono de la carpeta: 
3. 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).
4. ¡Ahora podés clonar!  
   Ejecutá el comando git clone “url\_del\_repositorio” en la carpeta elegida:  
   
5. 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.
6. Una vez clonado te aparecerá en tu terminal algo como esto:  
   
7. ¡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:  
   
8. 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:   


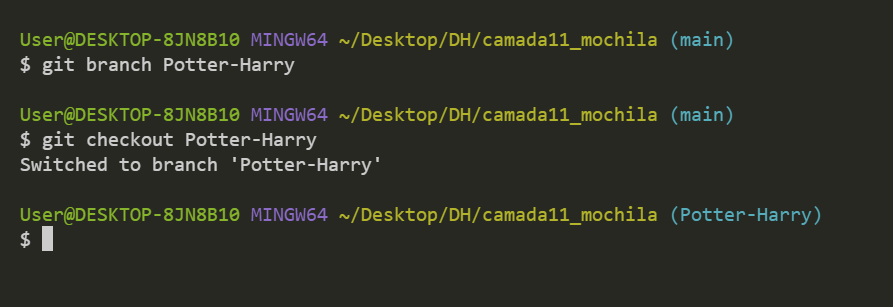
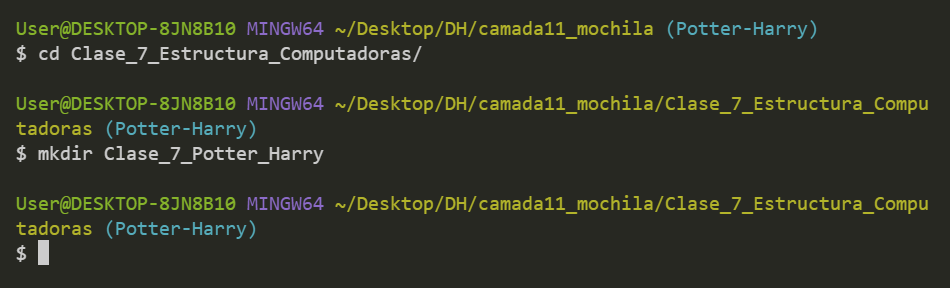
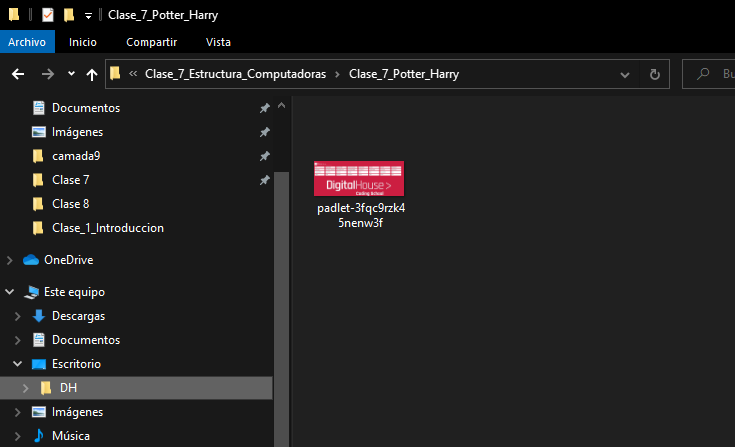
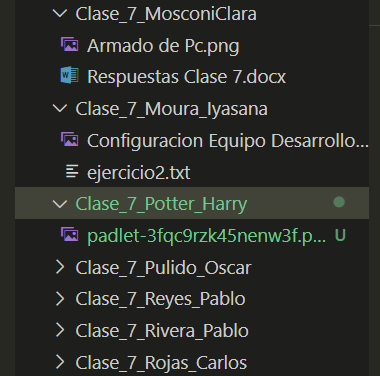
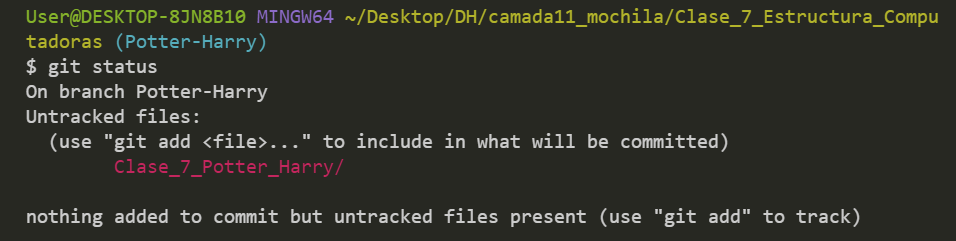
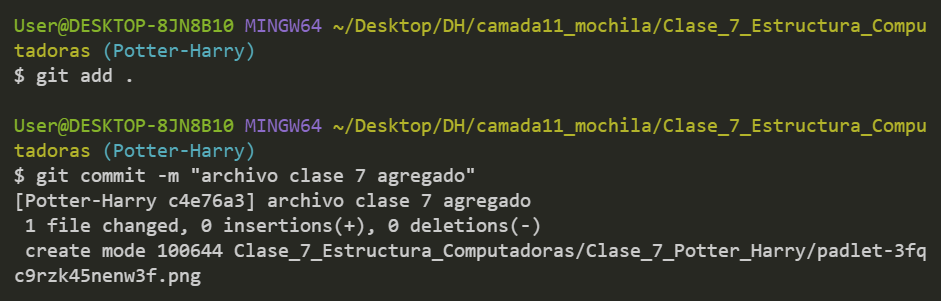
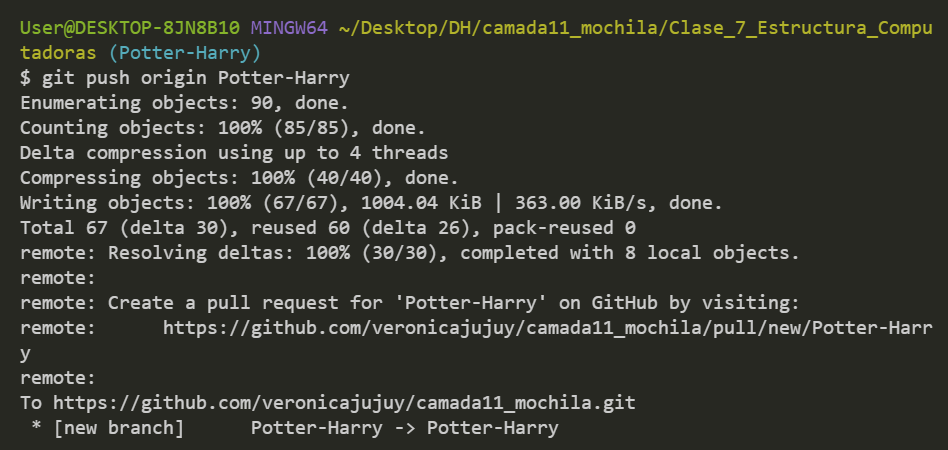
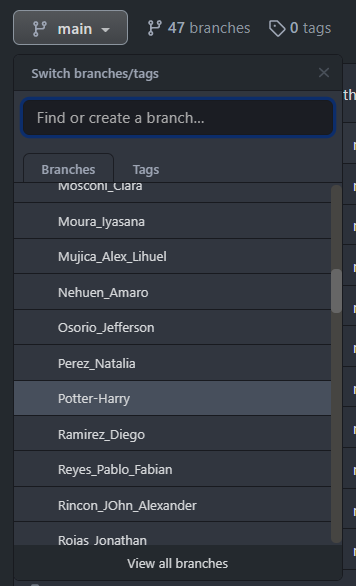
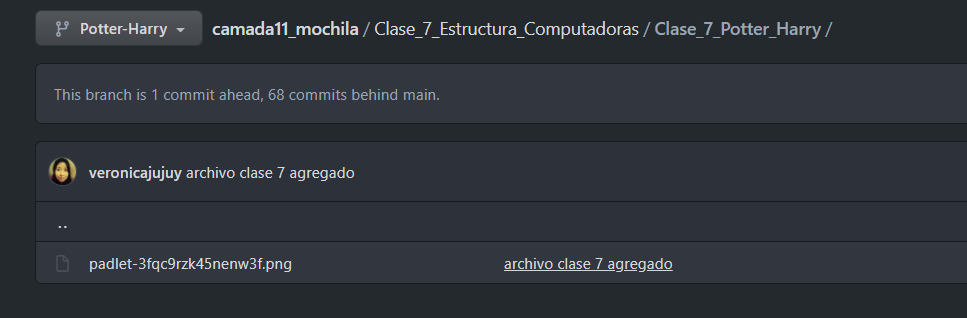
# Creando mi rama y subiendo las ejercitaciones

1. 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

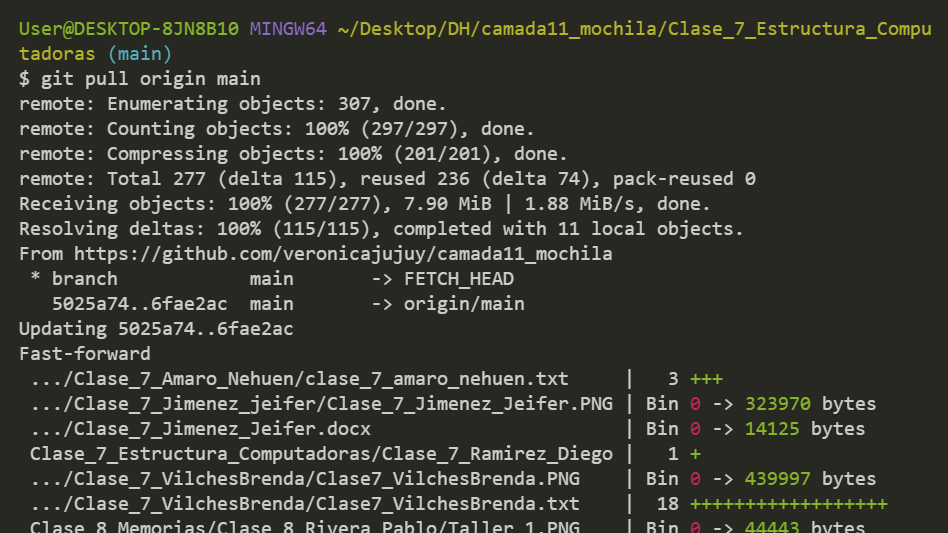
Luego nos cambiamos a esa rama utilizando el comando   
git checkout <nombre-rama>

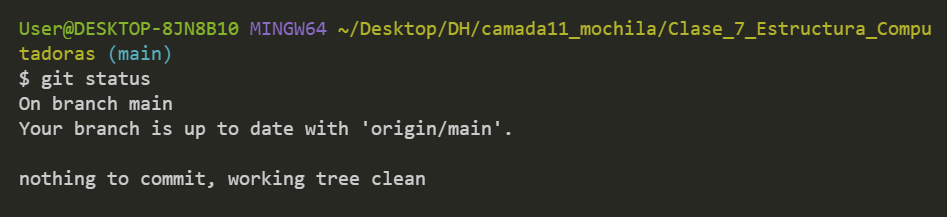
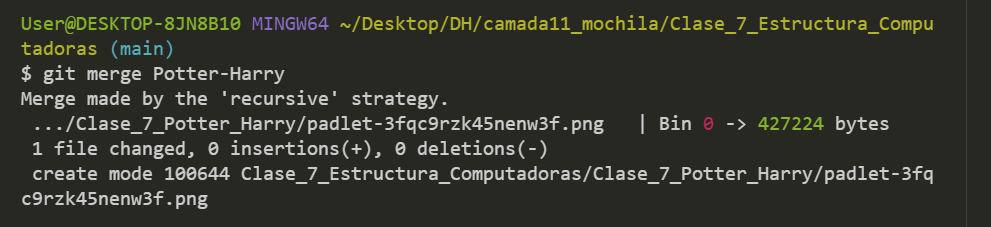
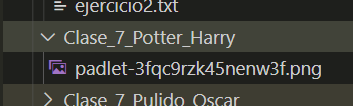
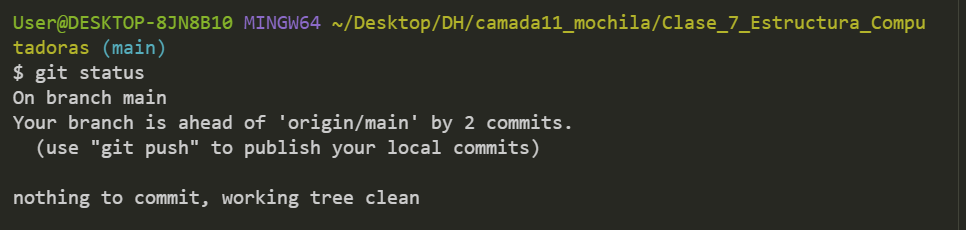
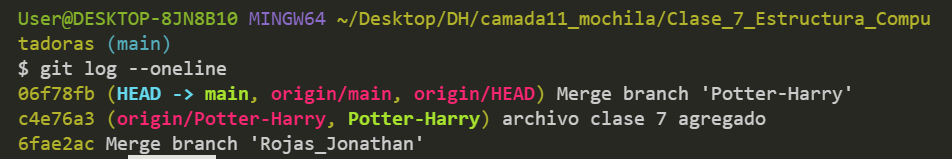
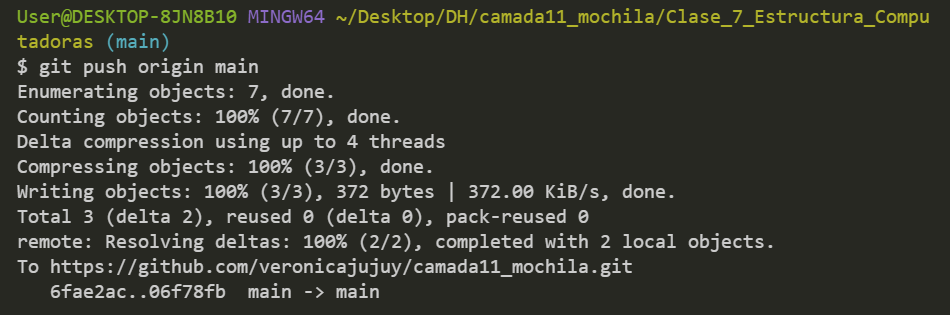
En nuestro caso:  
git checkout Potter\_Harry

1. Podremos visualizar que estamos en nuestra rama porque en vez de (main) nos figurará nuestro apellido y nombre:  
   
2. Ingresamos a la clase correspondiente del día. Si no está especificado un nombre en particular crearemos la carpeta:  
   **Clase\_X\_Apellido\_y\_Nombre**  
   
3. 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:  
   
4. En el Visual Studio code, nos aparecerá que tenemos un nuevo archivo Untracked:  
   
5. Si corremos el comando git status nos aparecerá la nueva carpeta creada lista para hacerle un add:  
   
6. Es hora de hacer el git add . y el commit  
   
7. Una vez que hicimos el commit podemos pushear a nuestra rama con el comando:  
   git push origin <nombre-rama>  
   
8. 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

1. 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
2. Será conveniente que traigamos los cambios del repositorio remoto primero, antes de hacer un push. Seguramente nuestros compañeros habran subido muchos cambios desde que clonamos:

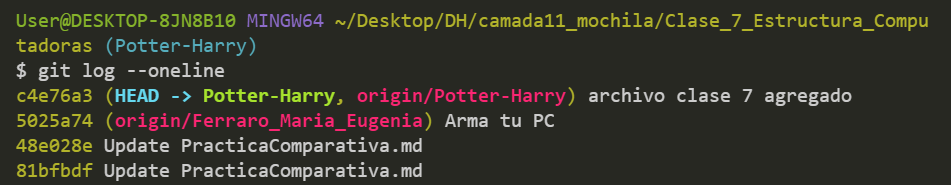
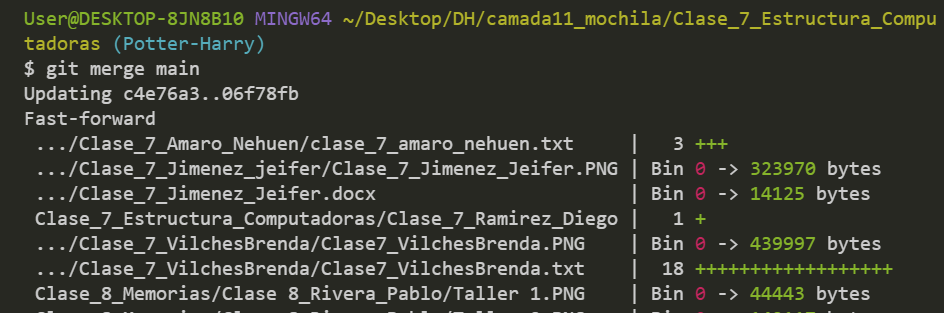
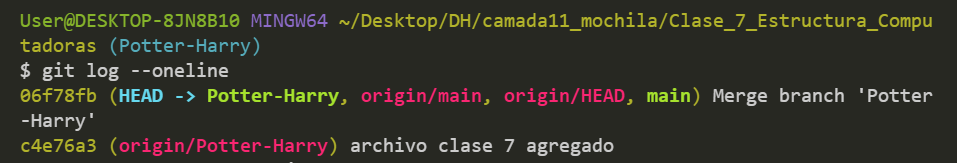
(Seguramente se cargarán muchas carpetas, no te preocupes que es normal)  


1. Este comando solo me traerá los cambios en la rama main, si hago git status me tendria que salir todo limpio:  
   
2. ¡Es hora de mergear! Para ello utilizamos el comando:  
   git merge <nombre-rama>  
   
3. Ahora mi carpeta y su contenido estarán en la rama main:  
   
4. 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:  
   
5. Si hago un git log --oneline veremos que nuestro merge generó un nuevo commit automáticamente:  
   
6. Ahora puedo actualizar el repositorio remoto con mis cambios ya en la rama main:  
   

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

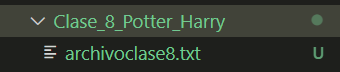
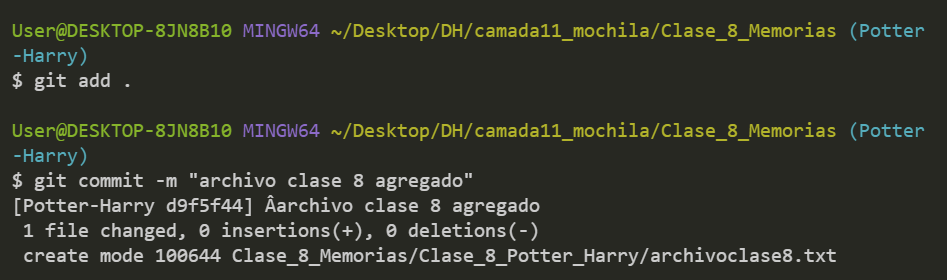
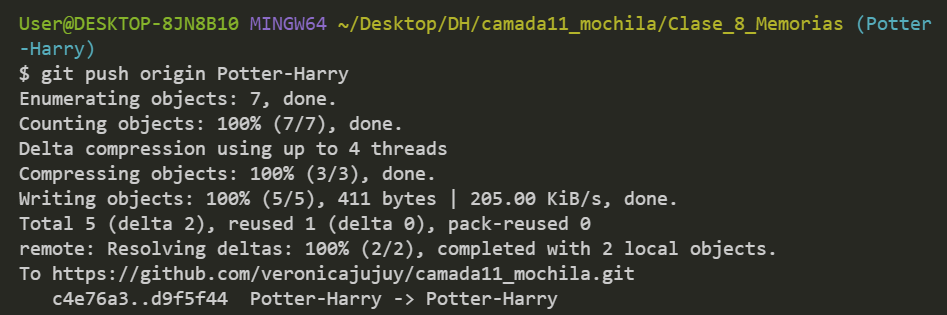
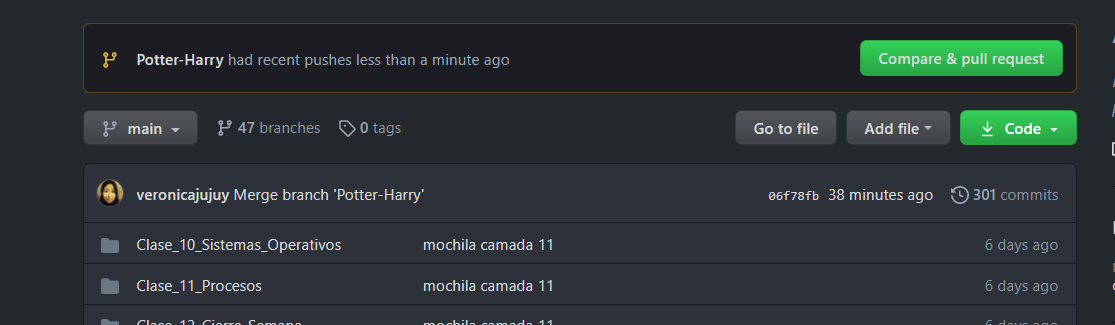
Agregando más archivos a mi rama

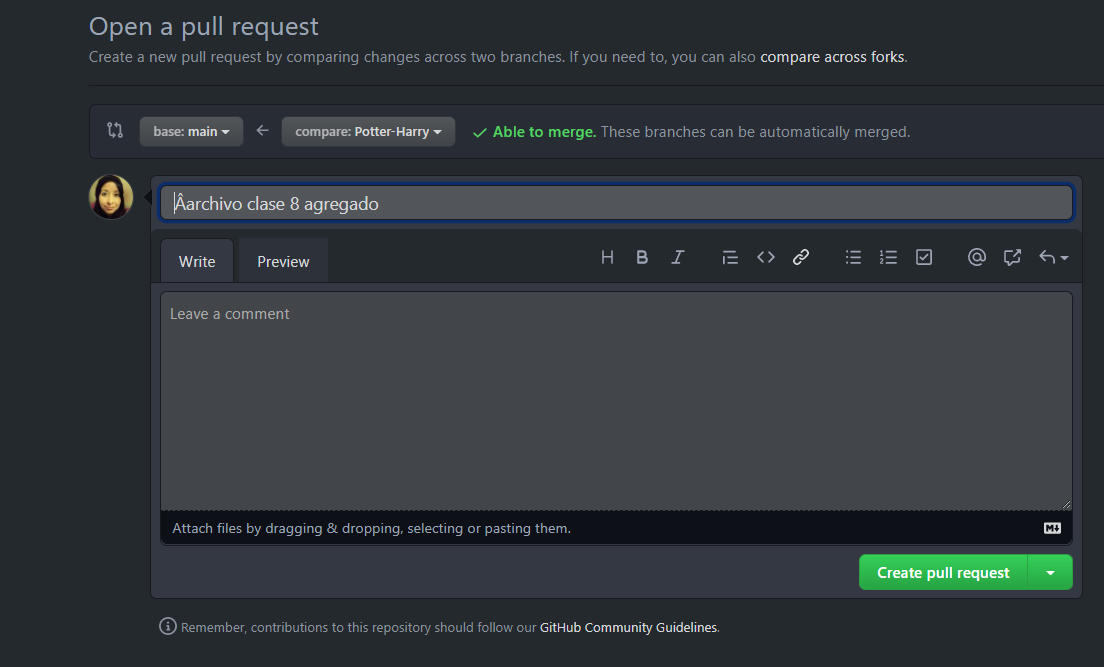
Cuando tengamos que agregar más archivos porque vamos avanzando en las clases, vamos a seguir los siguientes pasos:

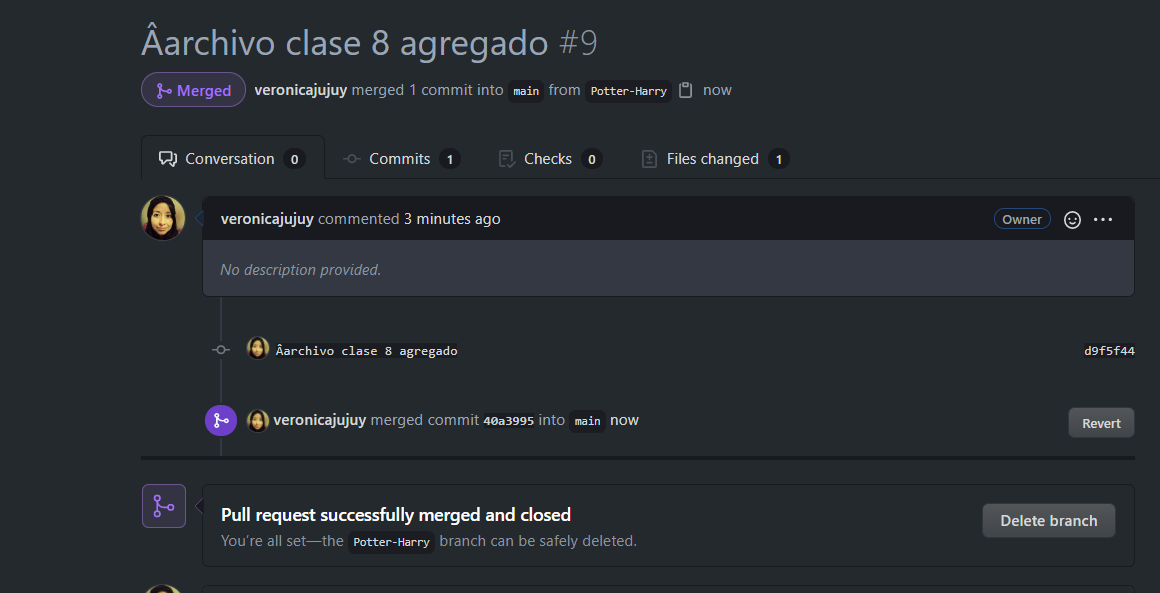
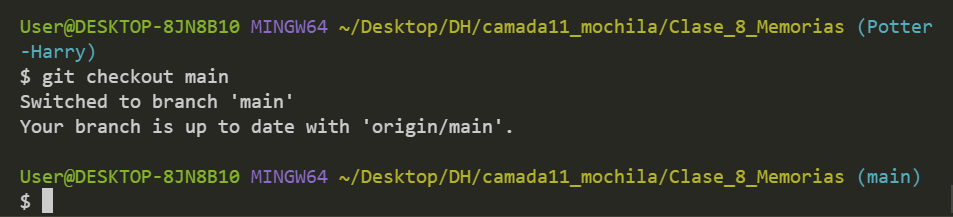
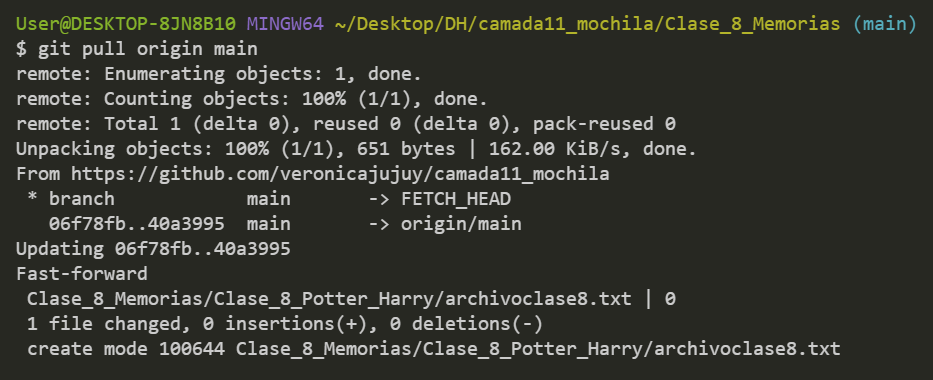
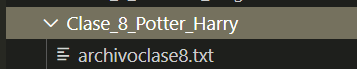
1. Nos dirigimos a la rama que creamos anteriormente con el comando:   
   git checkout <nombre-rama>
2. Si hacemos un git log --oneline veremos que el último commit que nos aparece en la rama main no está presente en mi rama:  
     
   Eso quiere decir que mi rama está un commit por detrás de la rama main, está desactualizada. Para emparejar las ramas, tengo que mergear la rama master a mi rama. El proceso inverso que hice anteriormente:  
   
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 dia con la rama main:  
   
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:

1. Nos cambiamos de rama, actualizamos la misma con un merge y agregamos un nuevo archivo a alguna carpeta.  
   
2. Hacemos un git add, un git commit tal cual vinimos haciendo anteriormente:  
   
3. Ahora pusheamos desde nuestra rama:  
   
4. Si vamos a github, al repositorio de la camada me debería aparecer lo siguiente:  
   
5. 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.   


1. 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
2. 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.  
   
3. Luego, presiono el botón Merge pull request y el botón Merge pull request cambiará a “confirm merge”. 
4. La ventana con una leyenda Merged en color morado, confirmará que está todo bien y ya mergeamos nuestra rama.
5. Queda un paso más y es bajar ese merge a mi repositorio local. En el Visual Studio code, y si todavia estoy en mi rama, hago un checkout al main:  
   git checkout main  
   
6. Posteriormente, hacemos un git pull de los cambios haciendo git pull origin main:  
   
7. Debe aparecer en tu rama main el mergeo que hice en el repositorio remoto
8. Podemos comprobar esto en la estructura de archivos de visual:  
   
9. ¡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!**