

## Sección 6 – Inicios en GitHub

### ...or create a new repository on the command line

```
echo "# prueba" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/CrisRmrz/prueba.git
git push -u origin main
```

El “ git branch –M main es para cambiar el nombre de la rama Master a rama Main

El git remote es para que haga match a la hora de hacer push

Y el git push –u origin main es solo para que a la hora de querer hacer un push en visual studio solo se tenga que escribir “git push” por defecto

### Hacer push de los tags de nuestro repositorio

\*\*GitHub no sube los tags que ya tengamos, tenemos que hacerlo de forma manual

git push --tags	Subir los tags a GitHub ya que no lo hace de una vez cuando subimos el repositorio
-----------------	--

### Hacer pull

Si tenemos cambios en el repositorio de GitHub y quiero traer esos cambios a mi proyecto en visual studio, en visual studio tengo que poner:

git pull
----------

### CLONAR EL REPOSITORIO

Clonar el repositorio nos descarga TODO, o sea que le estamos dando seguimiento al repositorio con todos sus commits. Si se descarga con .zip esto no pasa, tiene que ser clonado

git clone <a href="https://github.com/CrisRmrz/liga-justicia.git">https://github.com/CrisRmrz/liga-justicia.git</a> por ejemplo	Hacer esto en la carpeta donde quiero que este el repositorio
---	---

## CONFLICTOS EN GIT Y GITHUB

Estos conflictos son por ejemplo que tengamos un commit en un archivo en github, y en mi repositorio de visual studio otro commit que choque con ese, saldria este error:

```
1 file changed, 1 insertion(+)  
PS C:\Users\cris_OneDrive\Escritorio\Git-Github\liga-justicia> git pull  
remote: Enumerating objects: 5, done.  
remote: Counting objects: 100% (5/5), done.  
remote: Compressing objects: 100% (3/3), done.  
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), 706 bytes | 21.00 KiB/s, done.  
From https://github.com/CrisRmrz/liga-justicia  
   3cf23a5..f4fc9f9  main       -> origin/main  
fatal: Not possible to fast-forward, aborting.  
PS C:\Users\cris_OneDrive\Escritorio\Git-Github\liga-justicia> █
```

Si tenemos un conflicto poner el siguiente comando:

git pull --rebase	para traer cambios de github a mi vs. Esto nos va a dar una serie de instrucciones, hay que hacer un git merge ya que cuando hacemos el comando nos cambia a una rama temporal para hacer los cambios
git push	ya despues si queremos subir esos cambios a github

## Git Fetch

Imaginemos que tenemos cambios en el repositorio de github pero en el de vs no

Para ver en vs toda la historia con lg solo hacemos:

git fetch	Podemos hacer lg despues para que veamos que solo estamos viendo la historia y no tenemos esos cambios en nuestro repositorio de vs. Ya despues podemos hacer un git pull si queremos para tener todo el vs repositorio al de github
-----------	--

Esto esta bueno hacerlo antes de un git pull, solo para que veamos que cambios se harian antes de hacer el git pull y traer todos los cambios

## **FORK**

Esto es hacer una copia o clon del repositorio de otra persona

Una vez tengo el clon como un repositorio mio en mi cuenta de GitHub, nos vamos a mi repositorio del clon y copiamos el enlace en el boton de "code"

Luego para ya traer ese repositorio a mi VS en el powershell poner el siguiente comando:

```
git clone https://github.com/CrisRmrz/fork-legion-del-mal.git por ejemplo
```

## **Branch de los compañeros en VS**

Las ramas que tenemos en GH para que se pueda mirar en VS solo se pueden ver con el siguiente comando:

```
git branch --all
```

## **Milestone**

<https://docs.github.com/es/issues/tracking-your-work-with-issues/about-issues>

Es un agrupador de issues

## **Colaboradores**

Entrar al repositorio, settings y ahi aparece para seleccionar colaboradores

Tienen que aceptar por medio del email

## **Transferir permisos**

En settings donde dice transfer ownership ahi se pasan los permisos totales del repositorio

## **Wikis**

Las wikis son como la documentacion de mis proyectos

Literal es como una wikipedia de mi proyecto

## **Projects**

Es como una pizarra en grande diciendo las cosas pendientes de la aplicacion por hacer

## **Gists**

Estos son repositorios para poner codigo