

MANUAL DE GIT

Para poder usar Git primero debemos tener un proyecto para poder crear un repositorio y poder subir los archivos.

Como en mi caso ya tengo un editor instalado (netbeans) vamos a crear directamente un proyecto para poder subirlo. En mi caso lo voy a subir desde esta dirección local:

/home/vmnueva/Clase/DESP APLI WEB/Ejercicios/Trabajo sobre git/Pagina Web

Ahora vamos a pasar a Git. Antes de usar git primero hay que abrir la consola que desde donde vamos a usarlo.

Los primeros comandos que recomiendo son los siguientes:

git config --global user.email correo@tucorreo.com → para poder asociar los cambios a un correo.

git config --global user.name tuusuario → para poder asociar los cambios a un usuario.

Ahora vamos a iniciar el proyecto y así poder encontrarnos ya en el Working Directory y para ello usamos el siguiente comando:

git init

Para poder ver los archivos en los que estamos trabajando usamos el siguiente comando:

git status

Se pueden ver las carpetas que estan listas para realizar los add y los commits

```

vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo sobre git/Pagina Web$ git status
En la rama master

No hay commits todavía

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)

    nbproject/
    public_html/

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo sobre git/Pagina Web$ ls
nbproject  public_html  test
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo sobre git/Pagina Web$ git status
En la rama master

No hay commits todavía

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)

    nbproject/
    public_html/
    test/

vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo sobre git/Pagina Web$

```

En la anterior imagen se puede ver que hay archivos en espera para realizar el add. Aun así estos archivos siguen si estar en mi Staging Area o área de trabajo. Ahora vamos a usar los comandos necesarios para añadirlos:

-. Para añadir archivos directamente al área de trabajo usaremos el siguiente comando:

git add tuarchivo.txt

```

vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo sobre git/Pagina Web$ git add hola.txt
En la rama master

No hay commits todavía

Cambios a ser confirmados:
  (usa "git rm --cached <archivo>..." para borrarlos del área de trabajo)

    nuevo archivo: hola.txt

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlos)

    ../nbproject/
    ../public_html/

vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo sobre git/Pagina Web$

```

Aquí podemos ver que después de hacer git add hola.txt después de realizar un git status aparecen los últimos archivos añadidos a mi área de trabajo. Una vez añadidos todos los archivos y carpetas podemos pasar a hacer el commit.

-. Ahora vamos a realizar un commit para hacer nuestra primera snapshot o punto de partida:

git commit

```
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo sobre git/Pagina Web$ git commit
[master (commit-raiz) 78042a3] Primer commit de Victor Manuel Nieva
13 files changed, 664 insertions(+)
create mode 100755 nbproject/private/private.properties
create mode 100755 nbproject/private/private.xml
create mode 100755 nbproject/project.properties
create mode 100755 nbproject/project.xml
create mode 100755 public_html/estilo1.css
create mode 100644 public_html/estilo2.css
create mode 100644 public_html/estilo3.css
create mode 100644 public_html/estilo4.css
create mode 100755 public_html/index.html
create mode 100644 public_html/portables.html
create mode 100644 public_html/xboxjuegos.html
create mode 100644 public_html/xboxultimas.html
create mode 100644 test/hola.txt
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo sobre git/Pagina Web$ git status
En la rama master
nada para hacer commit, el árbol de trabajo esta limpio
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo sobre git/Pagina Web$
```

Como se puede ver el área de trabajo esta limpia porque se han subido todos los archivos. Si actualizamos otro archivo tendremos que volver a repetir los pasos anteriores de add y commit.

Para que se vea que hay varias versiones de mis archivos, primero, por supuesto tienen que haber sufrido cambios. Para ver las versiones usamos **git log**.

```
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo sobre git/Pagina Web$ git log
commit aab8cd1e4ed5e06411612e3e0702116328f0ef7f (HEAD -> master)
Author: vmnieva <vmanuelnieva@gmail.com>
Date: Mon Sep 24 10:15:12 2018 +0200

    Version 2 de la página

commit 78042a383a8db58364826ed65ec39bcbca240793
Author: vmnieva <vmanuelnieva@gmail.com>
Date: Mon Sep 24 10:06:09 2018 +0200

    Primer commit de Victor Manuel Nieva
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo sobre git/Pagina Web$
```

Si por ejemplo hemos cometido un error o queremos volver al estado anterior y para revertir esos cambios utilizaremos el comando:

git checkout - - tu archivo

```

vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo s
obre git/Pagina Web$ git status
En la rama master
Cambios no rastreados para el commit:
  (usa "git add <archivo>..." para actualizar lo que será confirmado)
  (usa "git checkout -- <archivo>..." para descartar los cambios en el directorio d
e trabajo)

        modificado:      nbproject/private/private.xml
        modificado:      public_html/index.html

```

```

vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo s
obre git/Pagina Web$ git checkout -- nbproject/

```

```

vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo s
obre git/Pagina Web$ git status
En la rama master
Cambios no rastreados para el commit:
  (usa "git add <archivo>..." para actualizar lo que será confirmado)
  (usa "git checkout -- <archivo>..." para descartar los cambios en el directorio d
e trabajo)

        modificado:      public_html/index.html

sin cambios agregados al commit (usa "git add" y/o "git commit -a")

```

Y de esta manera revertimos los cambios de los archivos.

Para crear una rama distinta, de esta manera la crea y cambia a ella directamente:

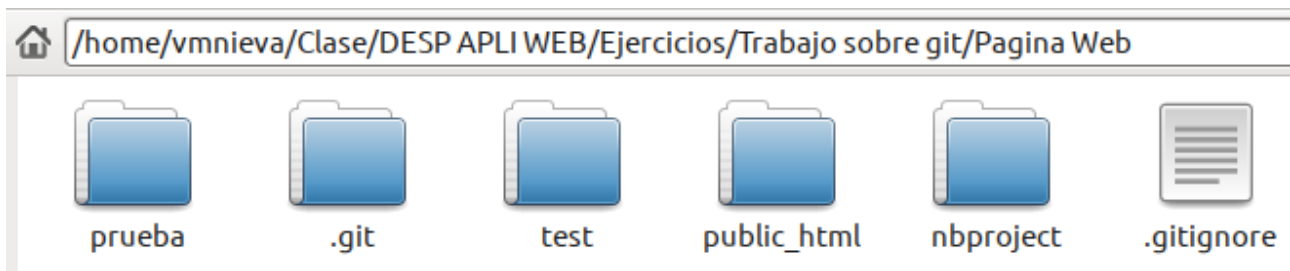
git checkout -b ramanueva

Y para cambiar a otra rama directamente **git checkout master**

Para fusionar las ramas hay que utilizar **git merge**

git merge master, en el caso de que estemos en otra rama.

Para hacer un git ignore es necesario crear en la carpeta un archivo llamado **.gitignore** y dentro podemos poner un listado de los archivos o carpetas que quieres que git ignore.



Aquí vemos la carpeta prueba creada y el **.gitignore** creados, ahora hay que introducir el nombre de la carpeta y ahora vamos a hacer un add del archivos gitignore.

He creado la carpeta prueba y al añadirla al git ignore no aparece para hacer el add de la carpeta.

```
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo s
obre git/Pagina Web$ git status
En la rama master
Cambios no rastreados para el commit:
  (usa "git add <archivo>..." para actualizar lo que será confirmado)
  (usa "git checkout -- <archivo>..." para descartar los cambios en el directorio d
e trabajo)

      modificado:   test/hola.txt

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)

      .gitignore

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
```

Ahora solo queda hacer el add del gitignore y hacer un commit

Y ahora toca subir todo a Github. Para ellos tenemos que tener una cuenta de github. Una vez tengas tu cuenta creada tienes que crear un nuevo repositorio desde <https://github.com/new> o hacer clic en “New repository”, ahora aparecerá la pagina de github para añadir el repositorio y lo primero que hay que poner es el nombre del repositorio, el nombre tendrá que ser único ya que no puede estar repetido en tu cuenta.

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 vmnieva ▾

Repository name

/ pagina-diw ✓

Great repository names are short and memorable. Need inspiration? How about **probable-fiesta**.

Description (optional)

Pagina Web que realicé para la asignatura DIW



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.



Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

En mi caso quedaría así, he añadido una descripción, yo lo quiero dejar publico porque en privado abría que pagar una cuota, no he marcado “Initialize this repository with a README” ya que es esta creado, y lo mismo con el .gitignore y no voy a añadir ninguna licencia y pulsamos Create Repository.

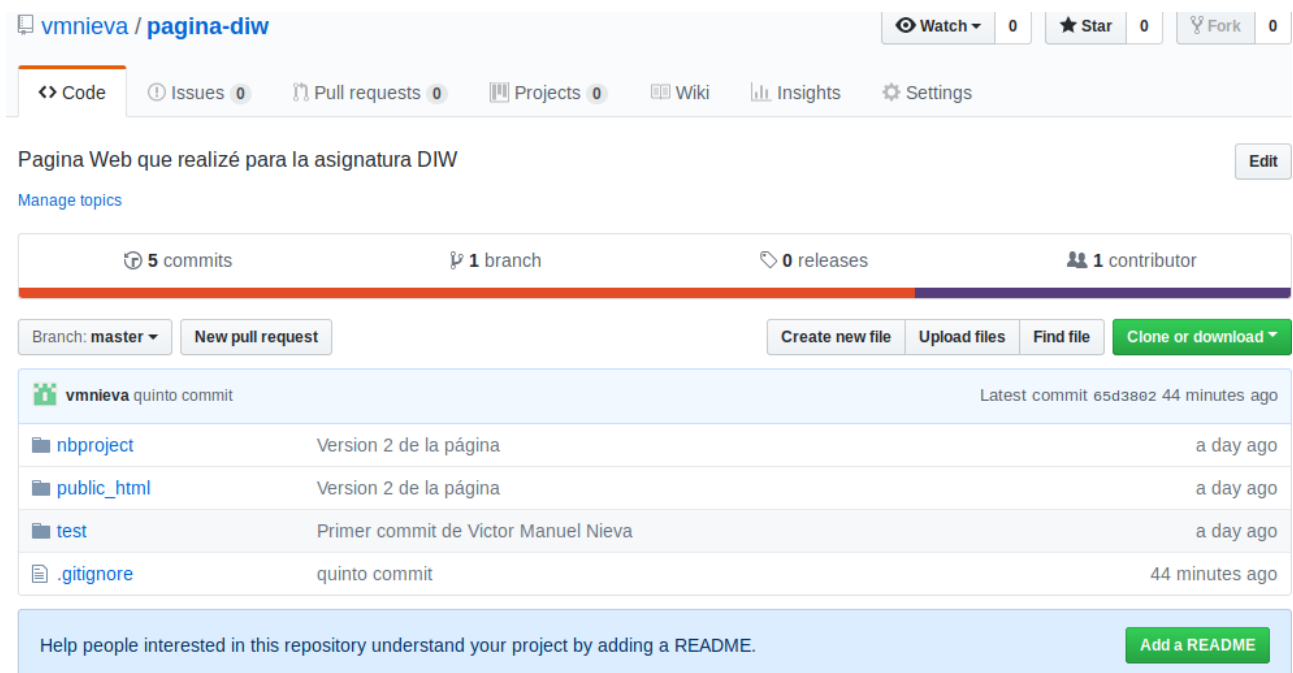
Una vez creado el repositorio ya solo queda subir el nuestro que esta en local a github, nos aparecen varios comandos para hacerlo desde terminal, y vamos a usarlos, el primero es:

```
git remote add origin https://github.com/vmniewa/pagina-diw.git
```

este sirve para crear el enlace de git al repositorio de github, y el siguiente seria:

```
git push -u origin master
```

Nos pedirá nuestro usuario y nuestra contraseña de github y una vez realizado ese paso ya estaria y con esto ya tendríamos listo nuestro repositorio en la nube.



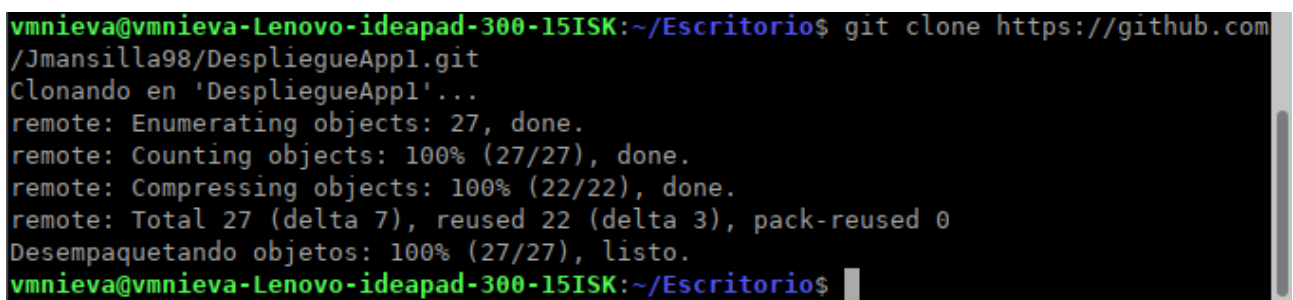
The screenshot shows the GitHub interface for a repository named 'pagina-diw' by user 'vmnieva'. At the top, there are buttons for 'Watch', 'Star', and 'Fork', each with a count of 0. Below these are tabs for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Insights', and 'Settings'. The repository description is 'Pagina Web que realizé para la asignatura DIW'. Below this, it shows '5 commits', '1 branch', '0 releases', and '1 contributor'. There are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A table lists the commits:

Commit	Message	Time
nbproject	Version 2 de la página	a day ago
public_html	Version 2 de la página	a day ago
test	Primer commit de Víctor Manuel Nieva	a day ago
.gitignore	quinto commit	44 minutes ago

At the bottom, there is a button 'Add a README'.

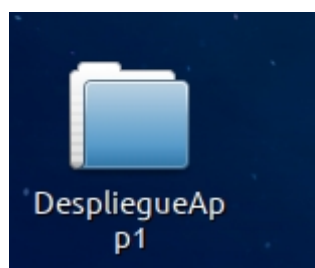
Ahora vamos a clonar el repositorio de un compañero, para hacer una clonacion se realiza de la siguiente manera:

git clone https://github.com/Jmansilla98/DespliegueApp1.git

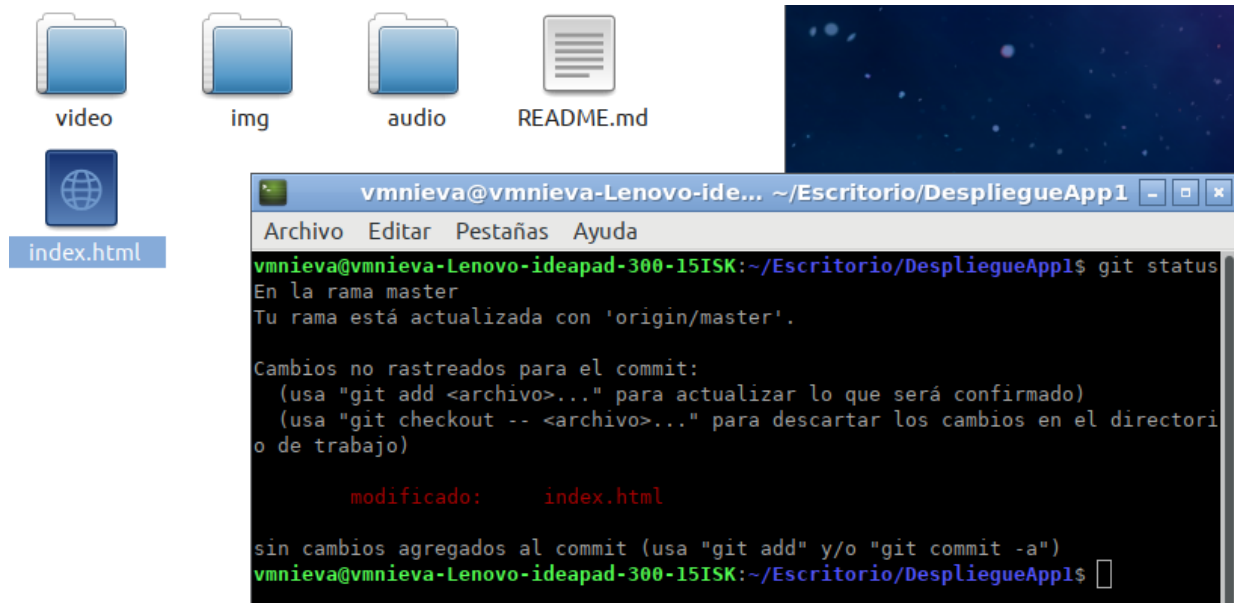


```
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Escritorio$ git clone https://github.com/Jmansilla98/DespliegueApp1.git
Clonando en 'DespliegueApp1'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 27 (delta 7), reused 22 (delta 3), pack-reused 0
Desempaquetando objetos: 100% (27/27), listo.
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Escritorio$
```

Y aquí ya tenemos la carpeta de mi compañero en el escritorio.



He realizado un cambio en el index.html de mi compañero, y seguidamente he realizado un git status y se puede ver el archivo que he modificado:



Ahora sería realizar un git add archivo, un git commit y finalmente un git push, para todo ello necesitamos que el compañero nos invite para colaborar en el proyecto. Cabe destacar que para poder realizar cambios en la nube es necesario que el compañero te invite a su repositorio.

```
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Escritorio/DespliegueApp1$ git add index.html
```

```
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Escritorio/DespliegueApp1$ git commit -m "Commit de Victor"
[master 1c5cd2a] Commit de Victor
1 file changed, 1 insertion(+)
```

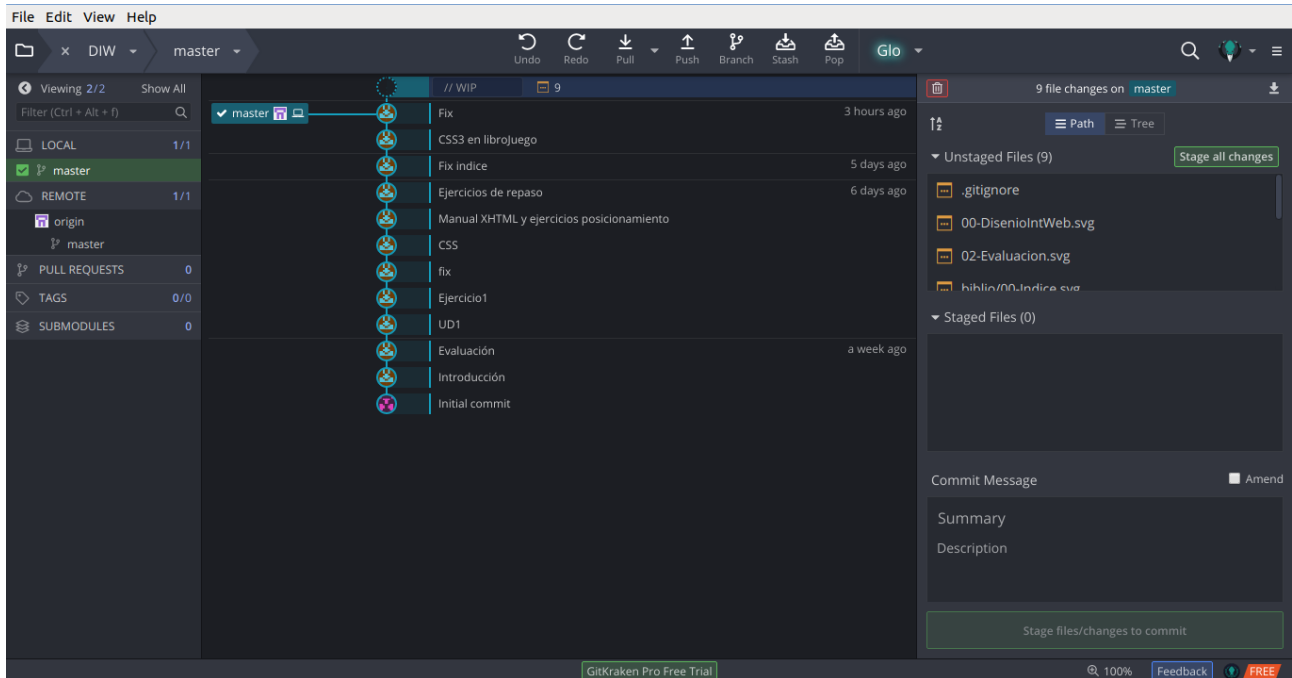
```
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Escritorio/DespliegueApp1$ git push
Username for 'https://github.com': vmanuelnieva@gmail.com
Password for 'https://vmanuelnieva@gmail.com@github.com':
Contando objetos: 3, listo.
Delta compression using up to 4 threads.
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 307 bytes | 307.00 KiB/s, listo.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Jmansilla98/DespliegueApp1.git
6bd9deb..1c5cd2a master -> master
```

Y para eliminar el repositorio habría que eliminar la carpeta oculta .git

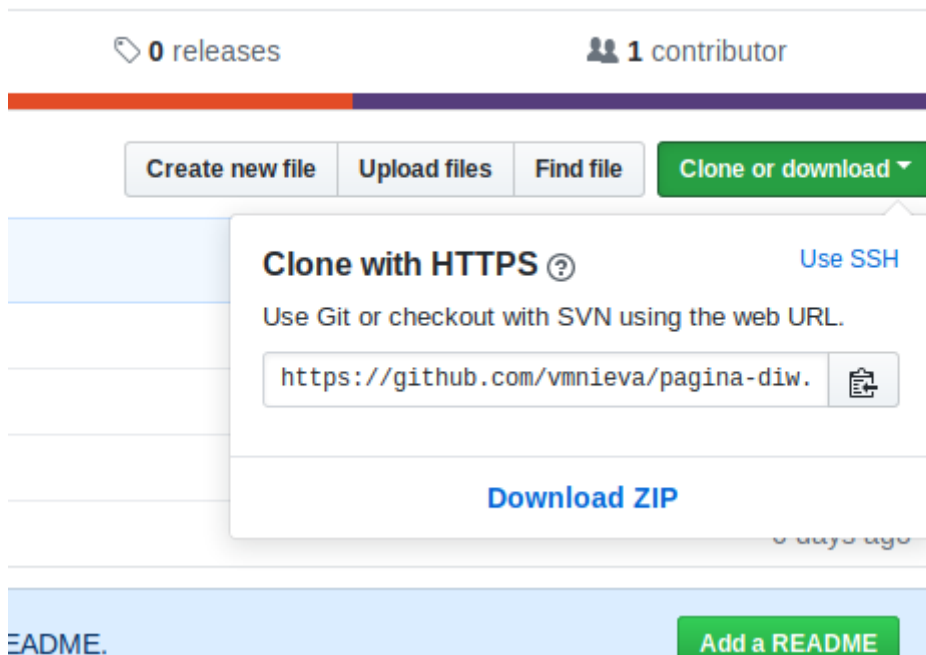
```
vmnieva@vmnieva-Lenovo-ideapad-300-15ISK:~/Clase/DESP APLI WEB/Ejercicios/Trabajo sobre git/Pagina Web$ rm -r .git
```


Git Kraken

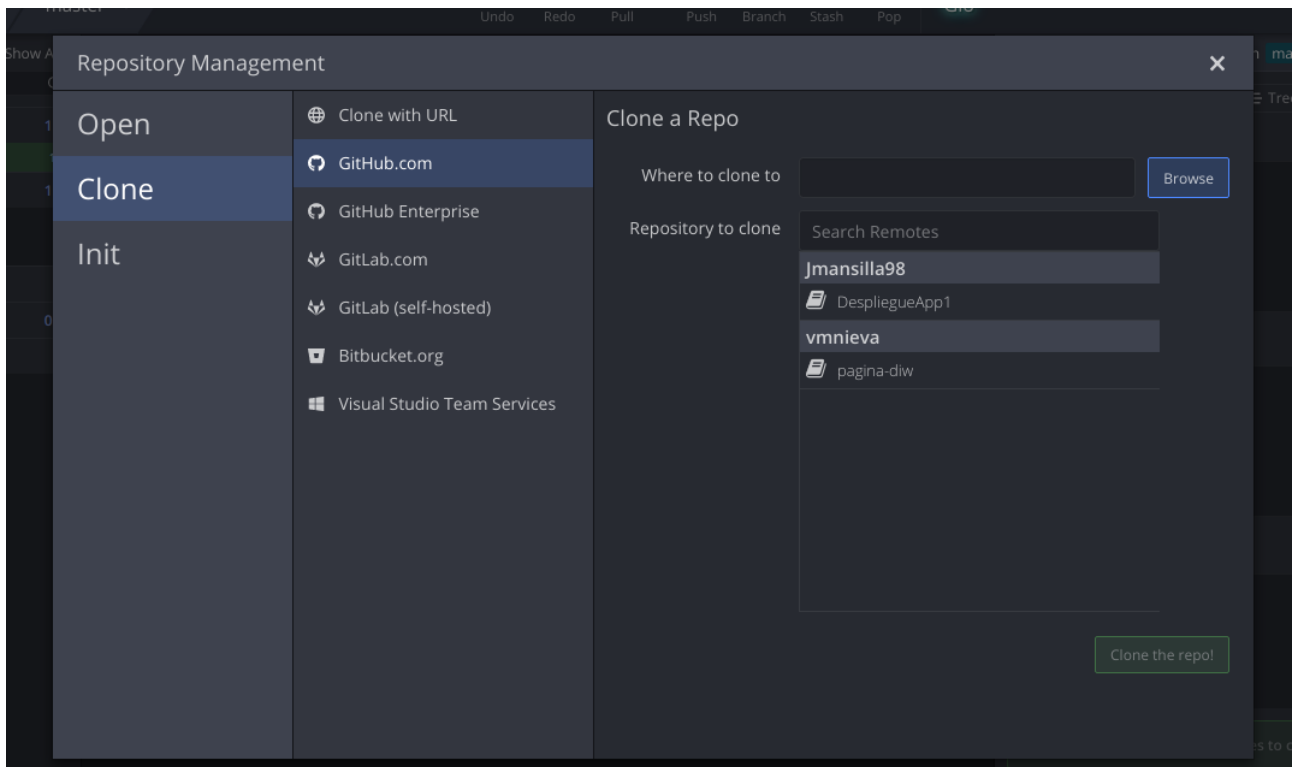
Git Kraken es una herramienta de GUI de Git, vamos a ver como se realiza el manejo de un repositorio.



Primero tenemos que ir a github y obtener el enlace del repositorio:

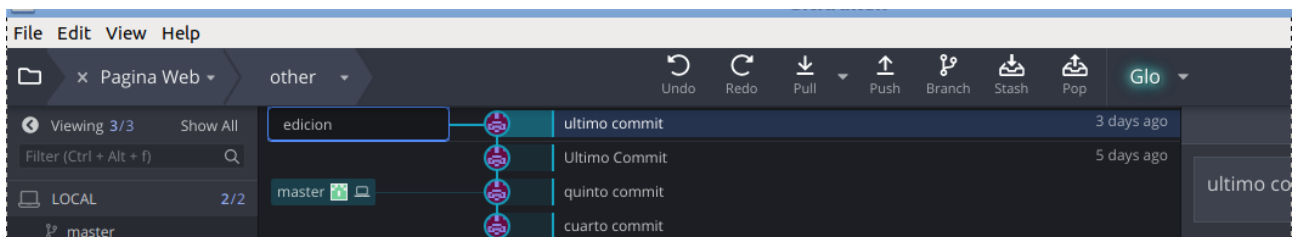


Seguidamente vamos al GitKraken para clonar el repositorio, tambien podemos clonarlo directamente si tenemos la cuenta vinculada a GitKraken



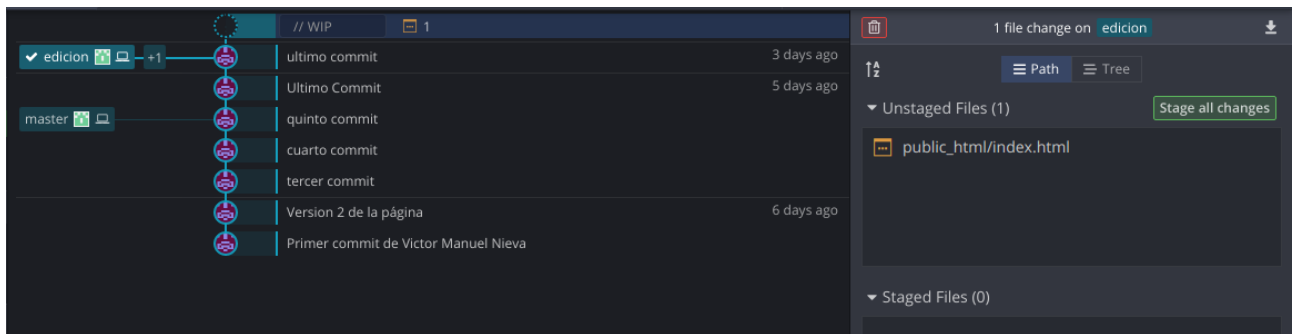
O si ya tenemos el repositorio en nuestro pc lo podemos abrir directamente desde Open como es mi caso. Tendriamos que abrir la carpeta .git de nuestro repositorio y ya apareceria en nuestro GitKraken.

Vamos a crear una nueva rama para editar unos archivos. Pulsamos branch e introducimos el nombre de otra nueva rama.

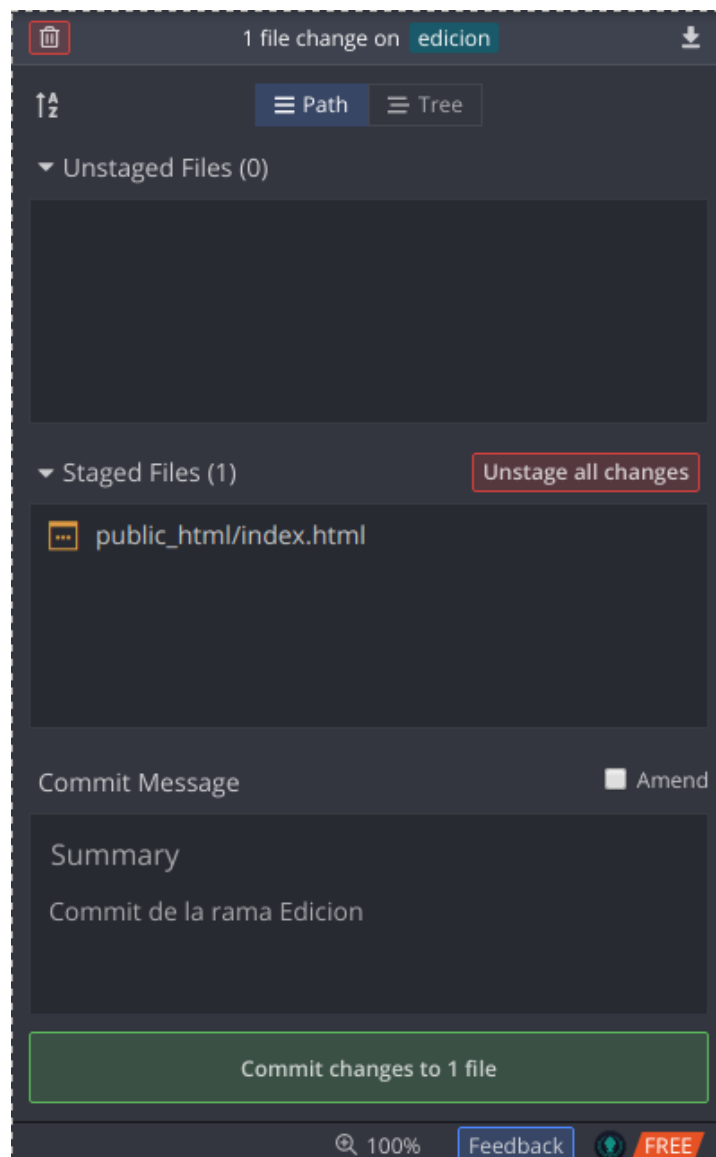


Al crear la nueva rama y realizar un push debemos introducir el nombre de la nueva rama a la hora de realizarlo si no está ya puesto.

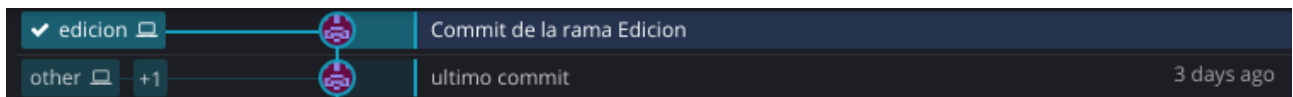
He realizado un cambio en el index.html y se puede observar a la derecha de la imagen.



Ahora vamos a relizar un commit, para realizarlo tenemos que cambiar el archivo de Unstaged Files a Staged Files, poner un comentario antes de realizar el commit y pulsar “Commit changes to n file”



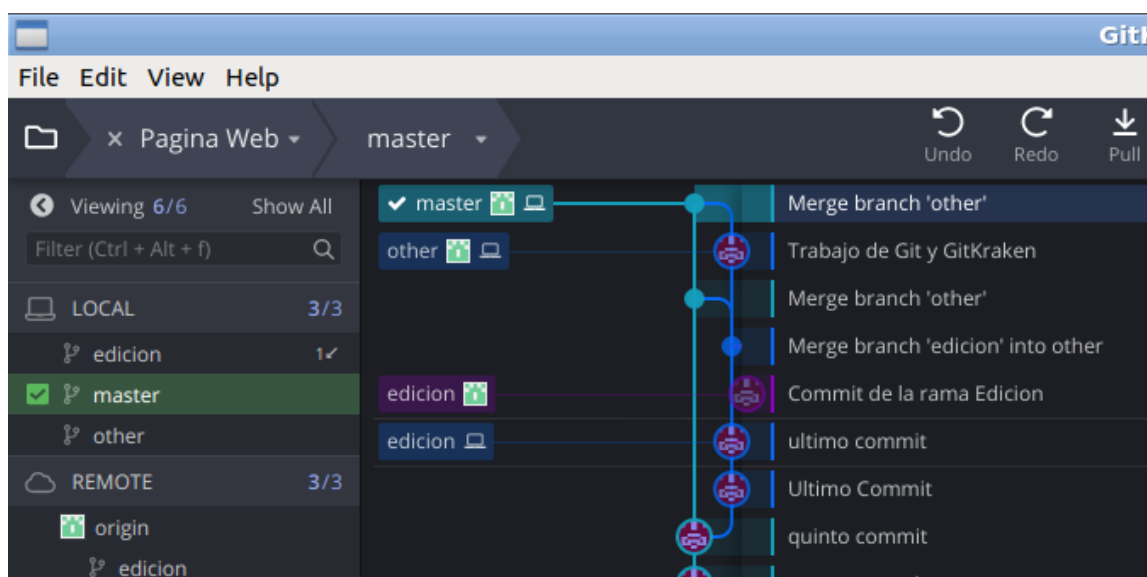
Y realizamos el commit:



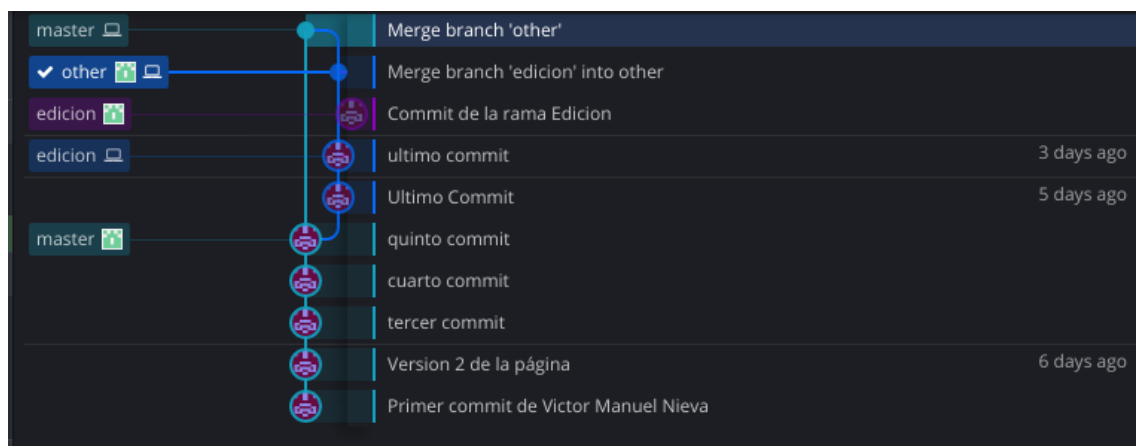
Seguidamente pulsamos push para subirlo.

Para realizar un pull simplemente pulsamos sobre Pull y bajara los archivos que se han cambiado recientemente.

Para cambiar de rama por ejemplo a la rama master, tenemos que hacer doble clic sobre ella a la izquierda.



Ahora vamos a terminar haciendo un Merge de las ramas, para ello solo hay que arrastrar las ramas para fusionarlas.



Con esto estaria el uso de GitKranken.

Bibliografia:

Git: https://www.youtube.com/watch?time_continue=2646&v=HiXLkL42tMU

GitKraken: <https://support.gitkraken.com/>