

# Práctica 03 – Primeros comandos

## Contenido

|     |  |    |
|-----|--|----|
| 1   | Cambiar el nombre de la rama Master a Main.....                                      | 2  |
| 1.1 | Cambiar el nombre de una rama .....  | 2  |
| 1.2 | Configurar que por defecto la rama principal de un nuevo proyecto se llame main..... | 2  |
| 2   | Archivo Readme.md y comando log .....  | 3  |
| 2.1 | Ver los commits realizados.....  | 4  |
| 3   | Adds y commits con visual studio code .....  | 5  |
| 4   | Diferentes formas de agregar archivos al escenario. ....                             | 7  |
| 4.1 | Añadir archivos sueltos .....  | 7  |
| 4.2 | Añadir archivos usando el comodín * .....  | 7  |
| 4.3 | Carpetas vacías.....   | 8  |
| 4.4 | Archivo .gitkeep.....  | 9  |
| 4.5 | Añadir una carpeta y todo su contenido.....  | 10 |

# 1 Cambiar el nombre de la rama Master a Main

En general, una **rama de desarrollo** ("Git Branch") es una bifurcación del estado del código que crea un nuevo camino para la evolución del mismo.

El comando git branch nos indica en que rama estamos trabajando:

(Pega aquí una captura de la ejecución del comando)

## 1.1 Cambiar el nombre de una rama

Para cambiar el nombre de una rama usamos el siguiente comando:

El -m indica que se va a cambiar de nombre a una rama.

A continuación, ponemos el nombre de la rama a renombrar (en este caso master).

Finalmente indicamos el nuevo nombre (en este caso main).

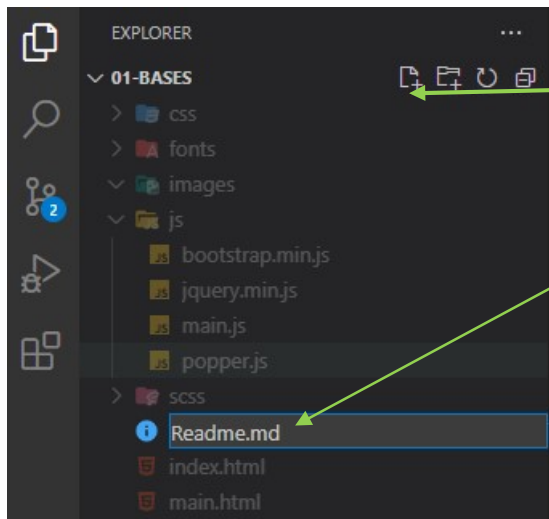
(Pega aquí una captura de la ejecución del comando)

## 1.2 Configurar que por defecto la rama principal de un nuevo proyecto se llame main

(Pega aquí una captura de la ejecución del comando)

## 2 Archivo Readme.md y comando log

Vamos a crear un archivo nuevo en nuestro proyecto, al que llamaremos Readme.md:

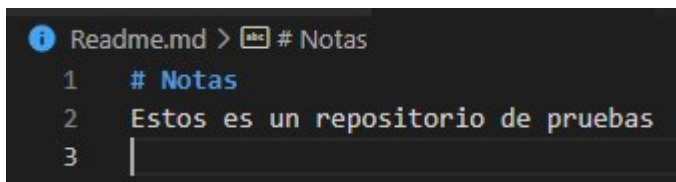


Pulsamos en el botón de nuevo archivo:

Y escribimos el nombre en el recuadro que aparece:

Un archivo README contiene información acerca de otros archivos en un directorio. Es una forma de documentación de software. En Github es habitual agregar un archivo README a un repositorio para comunicar información importante sobre el proyecto.

Escribimos algo de contenido en el archivo:



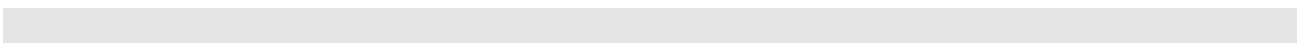
Le damos seguimiento al archivo:



Hacemos commit:



Borramos el archivo desde el visual estudio y como hemos hecho commit previamente podemos recuperarlo con:



(Pega aquí una captura de la ejecución de los comandos anteriores)

Modificamos el archivo Readme:

```
Readme.md > # Notas
1 # Notas
2 Este es el archivo Readme modificado
```

Hacemos un segundo commit con una versión nueva del comando checkout:

Este comando con `-a` sólo funciona si ya le estamos dando seguimiento al archivo, pero si estuviera marcado con la U de “untracked” es decir sin seguimiento no funcionaría.

(Pega aquí una captura de la ejecución del comando)

## 2.1 Ver los commits realizados

Vamos a ver los commits que tenemos hechos hasta el momento, para ello vamos a utilizar el siguiente comando:

(Pega aquí una captura de la ejecución del comando)

Vemos los tres commits que hemos hecho hasta ahora. La etiqueta head indica la última versión

```
PS D:\Git\01-bases> git log
commit 60280ed41e09e2bec6161e4c45d65e3bf2a9f129 (HEAD -> master)
Author: Alvaro <alvarosf@educastur.org>
Date: Wed Dec 1 16:50:21 2021 +0100
    Readme modificado

commit 206e3b4a753b794d7a21749817ba5338005111aa
Author: Alvaro <alvarosf@educastur.org>
Date: Wed Dec 1 16:49:35 2021 +0100
    Readme añadido

commit 6b6aaaeb184a2ba8c66c60cb80efd5021b4a0bd8
Author: Alvaro <alvarosf@educastur.org>
Date: Wed Dec 1 16:48:07 2021 +0100
    Primer commit
```

del repositorio:

Identificador (hash) del commit

Autor del commit

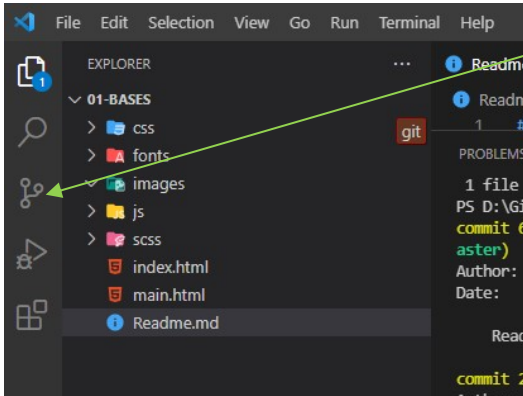
Fecha y hora del commit

Texto identificativo del commit

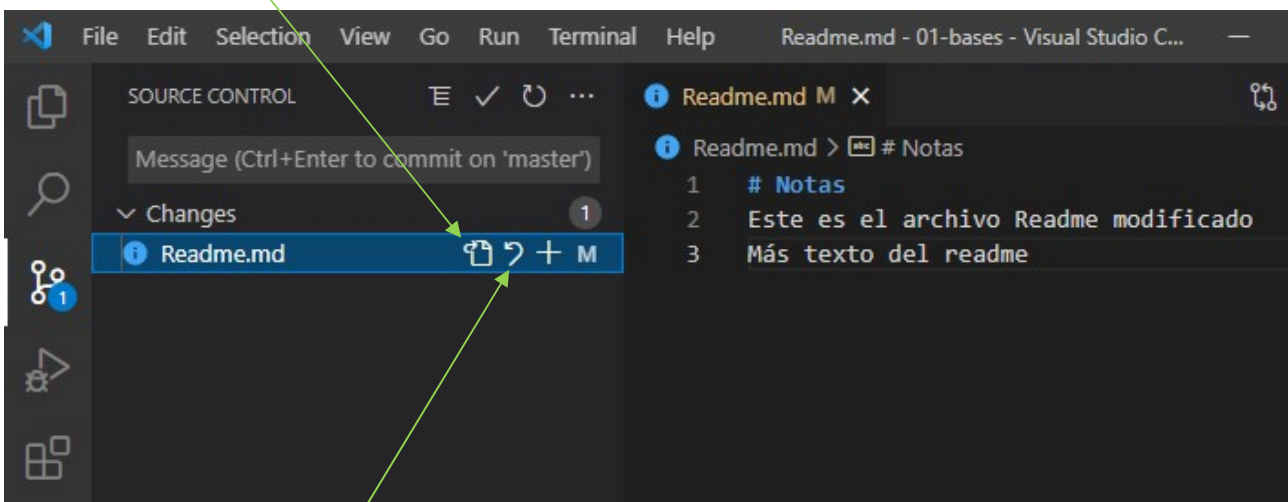
(Pega aquí una captura de tus commits realizados)

### 3 Adds y commits con visual studio code

En Visual studio code tenemos un apartado para realizar operaciones de git:

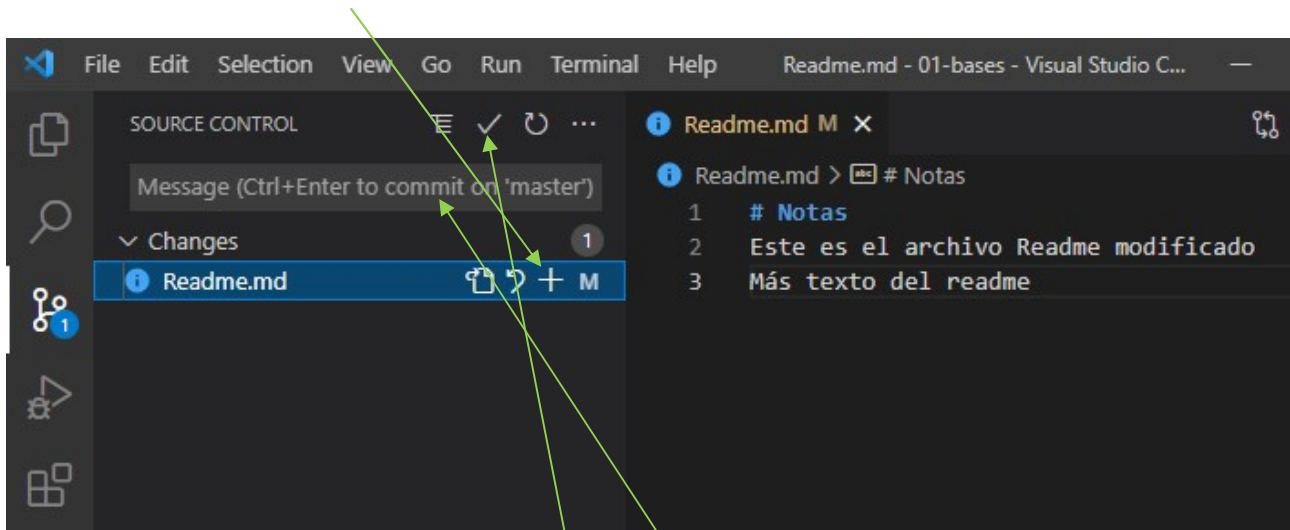


Con este icono abrimos el archivo:



Con este otro deshacemos los cambios realizados en el archivo:

Para añadir los cambios al escenario usamos este botón:



Y para hacer un commit escribimos el mensaje en este campo. Y a continuación utilizamos la combinación Ctrl+Enter o pulsamos en el botón de commit.

## 4 Diferentes formas de agregar archivos al escenario.

Cerramos en Visual Studio Code nuestro proyecto actual que era el de la carpeta 01-bases. Para ello vamos al menú File -> Close Folder o pulsamos la combinación Ctrl + K y luego F.

Descomprimos el archivo 02-bases.zip y copiamos la carpeta 02-bases a la carpeta donde habíamos copiado la carpeta 01-bases.

Arrastramos esta carpeta a visual studio code como hicimos con 01-bases

Nos situamos en la carpeta 02-bases e inicializamos el repositorio:

```
git init
```

Vamos a ver cómo hacer commits de grupos de archivos en vez de commit de todos los archivos como hicimos en apartados anteriores.

### 4.1 Añadir archivos sueltos

(Pega aquí una captura de la ejecución del comando)

Podemos añadir archivos sueltos simplemente escribiendo sus nombres separados por espacios.

### 4.2 Añadir archivos usando el comodín \*

Podemos añadir todos los archivos html de la siguiente forma:

A continuación hacemos commit de estos archivos

(Pega aquí una captura de la ejecución de los comandos anteriores)

Si ahora intentamos añadir todos los archivos .js

(Pega aquí una captura de la ejecución del comando)

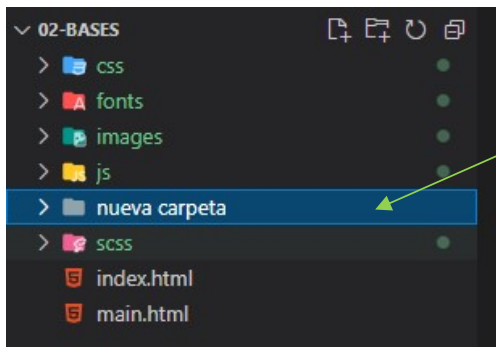
Esto no va a funcionar porque en el directorio actual no hay archivos .js , los archivos .js están en la carpeta js por lo que vamos a tener que indicarlo a la hora de ejecutar el comando:

**(Pega aquí una captura de la ejecución del comando)**

**(Nota:** Parece que en la versión actual si ponemos `add *.js` si que busca en las carpetas a partir de la carpeta actual y añade todos los archivos .js)

## 4.3 Carpetas vacías

Git no hace seguimiento a las carpetas vacías. Vamos a crear una carpeta nueva para verlo.

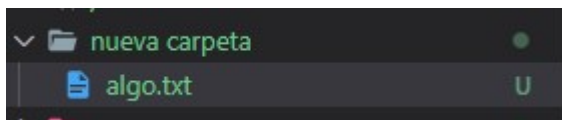


Si hacemos ahora `git status`:

**(Pega aquí una captura de la ejecución del comando)**

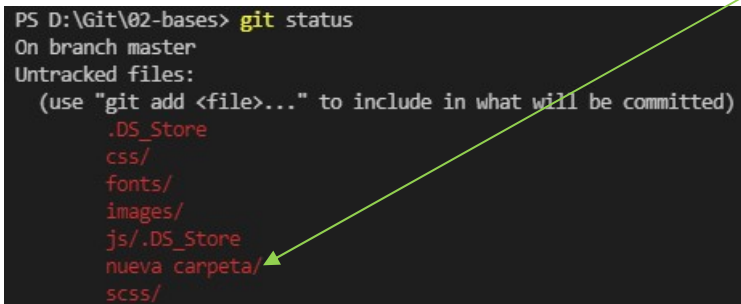
Vemos que git ni la muestra en la lista de archivos de los que no se hace seguimiento.

Si queremos que git la tenga en cuenta tenemos que crear un archivo dentro:





Y si ahora hacemos git status, vemos que ahora si aparece la nueva carpeta:



```
PS D:\Git\02-bases> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .DS_Store
        css/
        fonts/
        images/
        js/.DS_Store
        nueva carpeta/
        scss/
```

## 4.4 Archivo .gitkeep

Cuando queramos añadir al menos un archivo a una carpeta vacía para que git la tenga en cuenta, en vez de crear cualquier archivo existe un archivo con un nombre especial que podemos crear: el archivo **.gitkeep**

Este archivo está especialmente pensado para realizar esta función y ocupa un espacio muy pequeño.

(Nota: Renombramos “nueva carpeta” como “nuevacarpeta”)



Vemos además que visual studio code le pone un icono específico al archivo con este nombre.

Añadimos este archivo con el siguiente comando:

(Pega aquí una captura de la ejecución del comando)

En algunas versiones este comando puesto así podría fallar y podríamos usar esta otra opción:

(Pega aquí una captura de la ejecución del comando)

Hacemos commit:

(Pega aquí una captura de la ejecución del comando)

Si borramos la carpeta y restauramos con:

**(Pega aquí una captura de la ejecución del comando)**

Vemos que se restaura tanto el archivo .gitkeep como la carpeta que lo contenía.

## 4.5 Añadir una carpeta y todo su contenido

Si queremos añadir por ejemplo todo el contenido de la carpeta css (tanto archivos como directorios que contiene) podemos hacerlo con el siguiente comando:

**(Pega aquí una captura de la ejecución del comando)**

Hacemos commit:

**(Pega aquí una captura de la ejecución del comando)**