**Práctica 03 – Primeros comandos**

**Contenido**

[1](#_heading=h.26in1rg) Cambiar el nombre de la rama Master a Main 2

[1.1](#_heading=h.lnxbz9) Cambiar el nombre de una rama 2

[1.2](#_heading=h.35nkun2) Configurar que por defecto la rama principal de un nuevo proyecto se llame main 2

[2](#_heading=h.1ksv4uv) Archivo Readme.md y comando log 3

[2.1](#_heading=h.44sinio) Ver los commits realizados 4

[3](#_heading=h.2jxsxqh) Adds y commits con visual studio code 5

[4](#_heading=h.z337ya) Diferentes formas de agregar archivos al escenario. 7

[4.1](#_heading=h.3j2qqm3) Añadir archivos sueltos 7

[4.2](#_heading=h.1y810tw) Añadir archivos usando el comodín \* 7

[4.3](#_heading=h.4i7ojhp) Carpetas vacías 8

[4.4](#_heading=h.2xcytpi) Archivo .gitkeep 9

[4.5](#_heading=h.1ci93xb) 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:

****

* 1. **Cambiar el nombre de una rama**

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



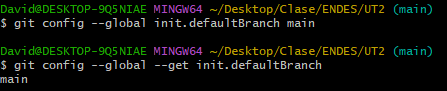
El -m indica que se va a cambiar de nombre a una rama.

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

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

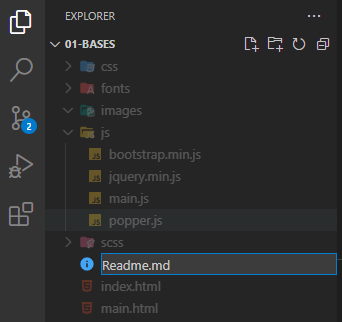
**(Pega aquí una captura de la ejecución del comando)**

* 1. **Configurar que por defecto la rama principal de un nuevo proyecto se llame main**

****

1. **Archivo Readme.md y comando log**

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

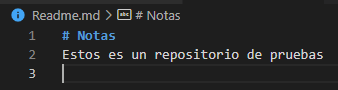


Pulsamos en el botón de nuevo archivo:

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.

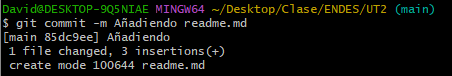
Escribimos algo de contenido en el archivo:



Le damos seguimiento al archivo:



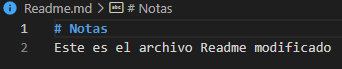
Hacemos commit:



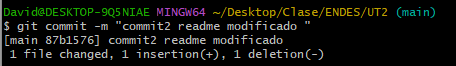
Borramos el archivo desde el visual estudio y como hemos hecho commit previamente podemos recuperarlo con:



Modificamos el archivo Readme:



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

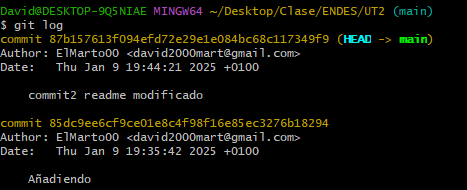


Este comando con –a sólo funciona si ya le estamos dando seguimiento al archivo, pero si estuviera marcado con la U de “untracked” es decir sin seguimiento no funcionaria.

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

* 1. **Ver los commits realizados**

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



Identifica los siguientes elementos en la captura y señálalos:

* Última versión del repositorio

Es la version que sale mas arriba en la pantalla

* Identificador hash del commit

los caracteres que siguen después de commit en este caso seria 87b157613f094efd72e29e1e084bc68c117349f9

* Autor del commit

lo marca con “author” en este caso ElMarto00

* Fecha y hora del commit

lo marca con “date” en este caso Thu Jan 9 19:35:42 2025 +0100

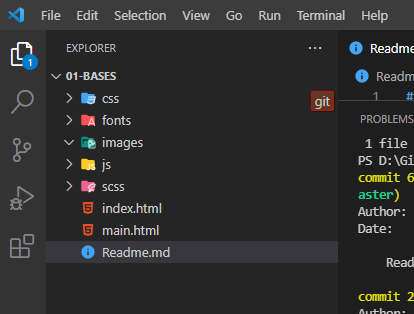
* Texto identificativo del commit

El nombre con el que lo guardas en este caso commit 2 readme modificado

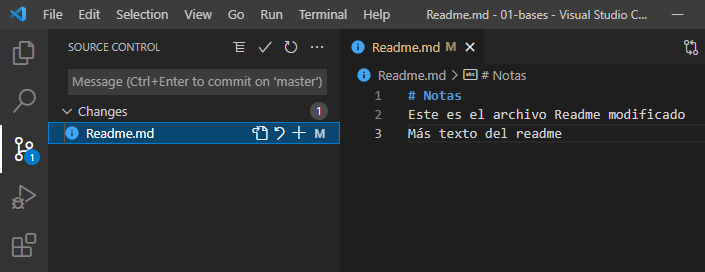
(**Pega aquí una captura de tus commits realizados**)

1. **Adds y commits con visual studio code**

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

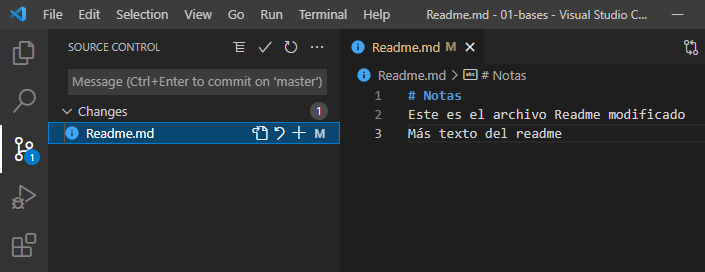


Con este icono abrimos el archivo:



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.

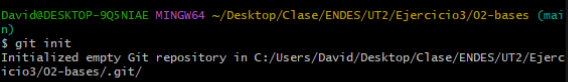
1. **Diferentes formas de agregar archivos al escenario.**

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.

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

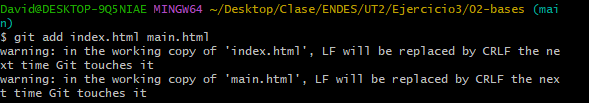
Arrastramos esta carpeta a visual studio code como hicimos con 01-bases

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



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

* 1. **Añadir archivos sueltos**



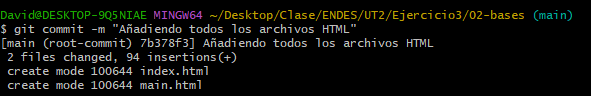
Podemos añadir archivos sueltos simplemente escribiendo sus nombres separados por espacios.

* 1. **Añadir archivos usando el comodín \***

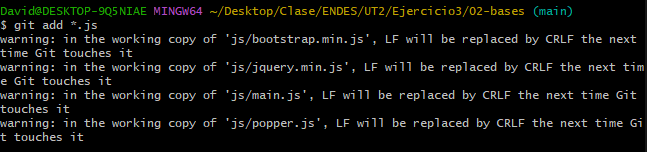
Podemos añadir todos los archivos html de la siguiente forma:



A continuación hacemos commit de estos archivos

****

Si ahora intentamos añadir todos los archivos .js



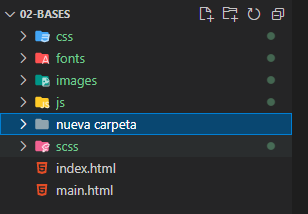
Esto no va a funcionar porque en el directorio actual no hay archivos .js , los archivos .js están en la carpeta js por lo que vamos a tener que indicarlo a la hora de ejecutar el comando:



(**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)

* 1. **Carpetas vacías**

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



Si hacemos ahora git status:

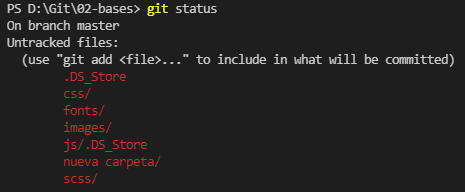
**(Pega aquí una captura de la ejecución del comando)**

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:



* 1. **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**

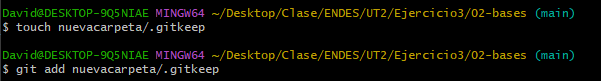
Este archivo está especialmente pensado para realizar esta función y ocupa un espacio muy pequeño.

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



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

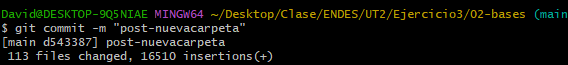
Añadimos este archivo con el siguiente comando:



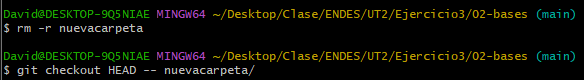
En algunas versiones este comando puesto así podría fallar y podríamos usar esta otra opción:



Hacemos commit:

****

Si borramos la carpeta y restauramos con:



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

* 1. **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:



Hacemos commit:git

