Introducción a Git y Gitlab en VSCode

Realizado por Alejandro Ruiz Becerra

Crear y configurar un repositorio

Crear e inicializar un proyecto

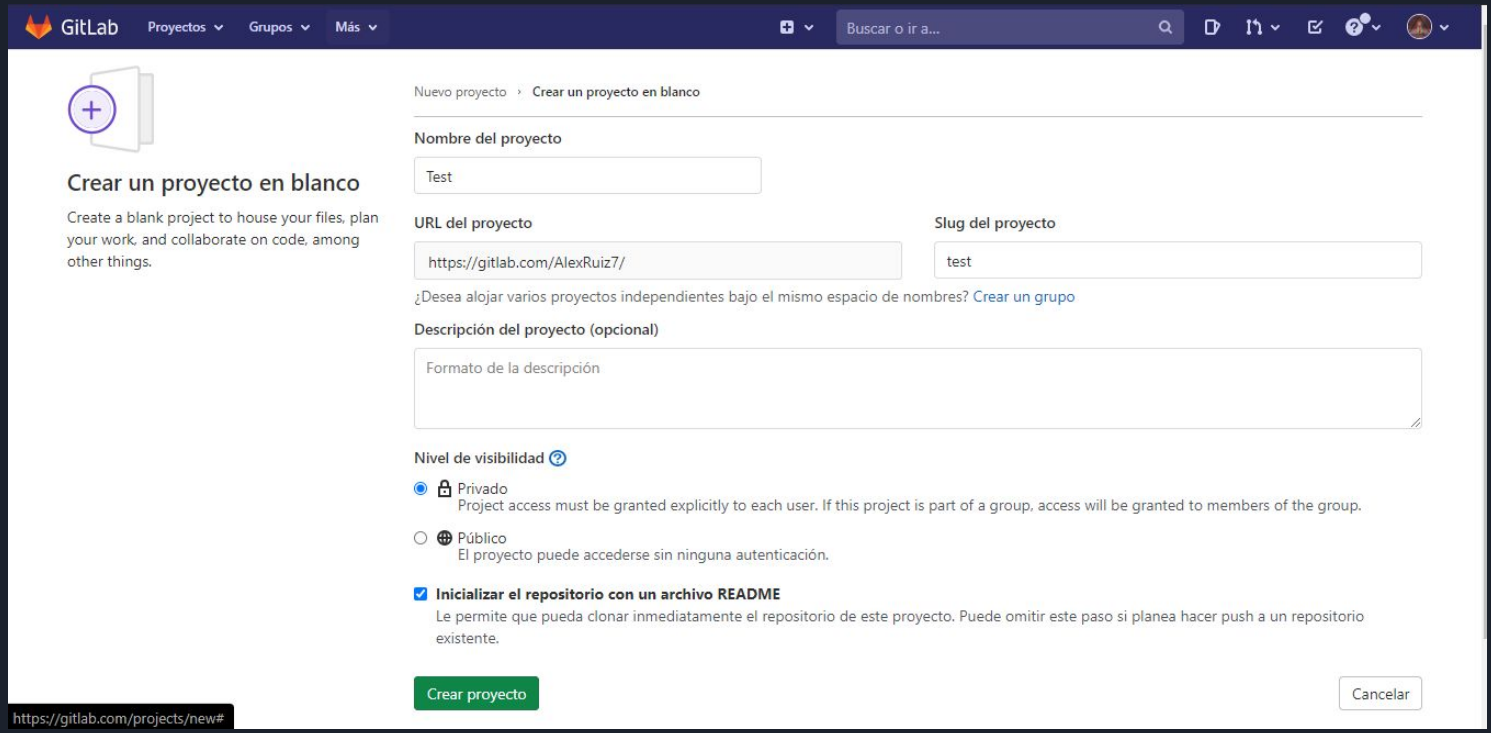
Esta parte está dedicada a los usuarios con permisos de tipo Maintainer y Owner.

Puedes saltar esta sección si no es tu caso.

Para crear e inicializar un repositorio correctamente seguimos los siguientes pasos:

1. Hacemos click en “Nuevo proyecto”
2. Seleccionamos “Crear un proyecto en blanco” en la nueva ventana que nos aparece.
3. Configuramos el nombre y la visibilidad del proyecto.
4. Activamos la casilla “Inicializar el repositorio con un archivo README”

Inicializa el repositorio con un README



Configuración del repositorio

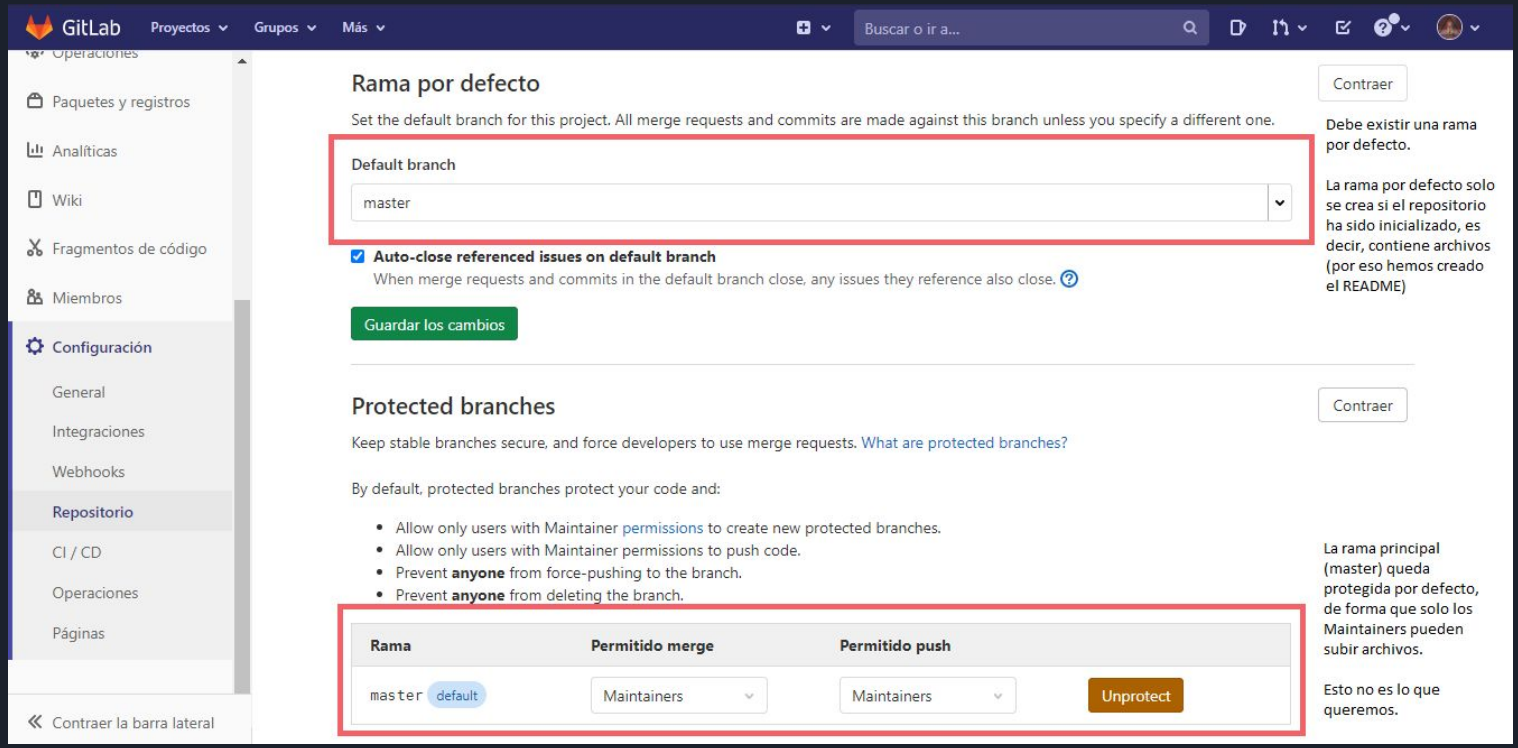
Una vez tenemos el proyecto creado e inicializado con un README es hora de subir código.

Gitlab por defecto solo permite a los usuarios con rol de Maintainer “tocar” el código, por lo que el rol de desarrollador (el rol por defecto cuando se añaden nuevos miembros al proyecto) resulta bastante inútil con esta configuración.

Para cambiar esta configuración, vamos al proyecto, y en la barra de navegación del lateral izquierdo, hacemos click en “Configuración” y luego en “Repositorio”.

Veremos la siguiente configuración:

Configuración del repositorio



Configuración del repositorio

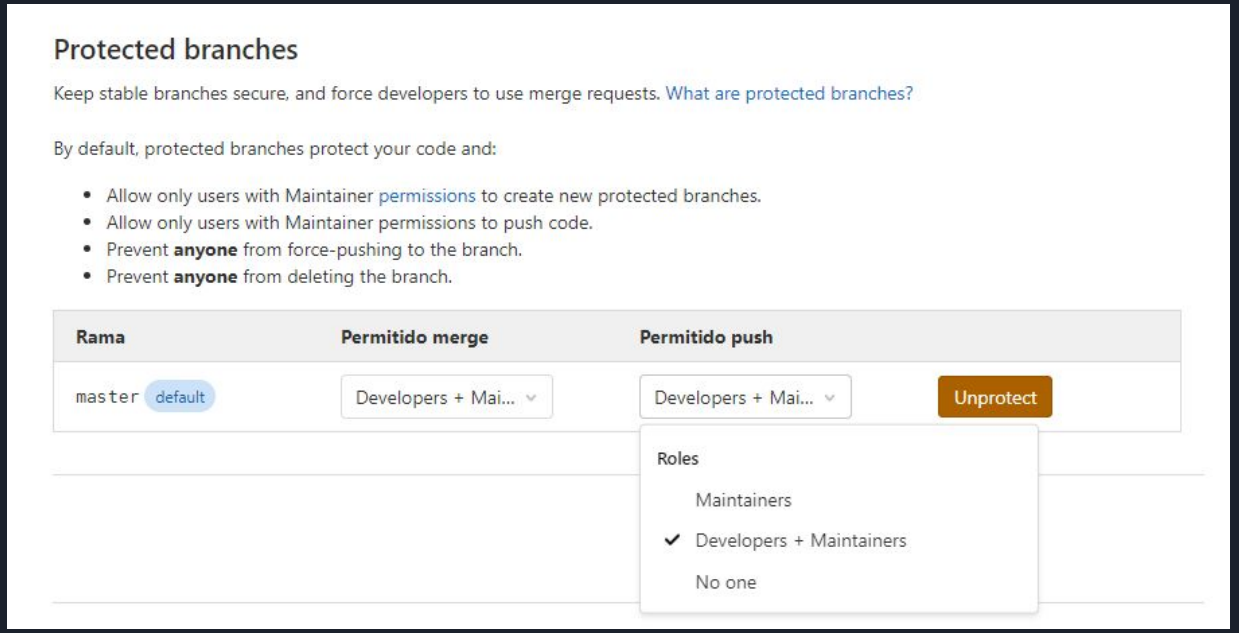
Como vemos la rama master queda protegida de forma que solo los Maintainers pueden realizar push y merge.

Tenemos 3 opciones para solventar este problema:

1. Dar permisos a los desarrolladores.
2. Desproteger la rama master haciendo click en el botón “Unprotect”.
3. Ascender a todos los miembros a Maintainers.

Configuración del repositorio

Opción 1: Dar permisos a los desarrolladores.



Configuración del repositorio

Opción 3: Ascender miembros a Maintainers

- En la barra lateral cambiamos a la sección “Members”
- Cambiamos los roles de los miembros a Maintainers

Configuración de SSH

ssh-keygen

La mejor forma de comunicarnos con *git* es usando *ssh*.

La herramient *ssh-keygen* nos genera un par de claves únicas y relacionadas.

- La clave **privada** (por defecto *id_rsa*) **no debe compartirse con nadie**.
- La clave pública (por defecto *id_rsa.pub*) es la que debemos proporcionar a Gitlab.

Cada vez que nos conectemos a Gitlab (pull, push...), se comprueba que la clave pública proporcionada es correcta, comparándola con la clave privada de nuestro ordenador.

Para generar el par de claves escribimos en una terminal *ssh-keygen* (*ssh-keygen.exe* en Windows) y pulsamos Intro hasta completar el proceso.

ssh-keygen

Al completar el proceso el par de claves se habrá generado en la ruta por defecto. *ssh-keygen* también nos dice dónde ha almacenado las claves.

La ruta por defecto es:

- Linux:
  - `~/.ssh/`
- Windows:
  - `X:\Users\<usuario>\.ssh`

Añadir la clave pública en Gitlab

- Abrimos la clave pública (***id_rsa.pub***) con cualquier editor de texto y copiamos todo su contenido.
- En Gitlab, vamos a nuestra configuración de usuario, haciendo click en nuestro icono en la esquina superior derecha.
- Vamos a la sección de “Claves SSH”
- Pegamos nuestra **clave pública** y la añadimos.

La primera vez que nos comuniquemos con Git mediante SSH nos pedirá si queremos añadirlo a la lista de hosts conocidos, indicamos que sí escribiendo YES.

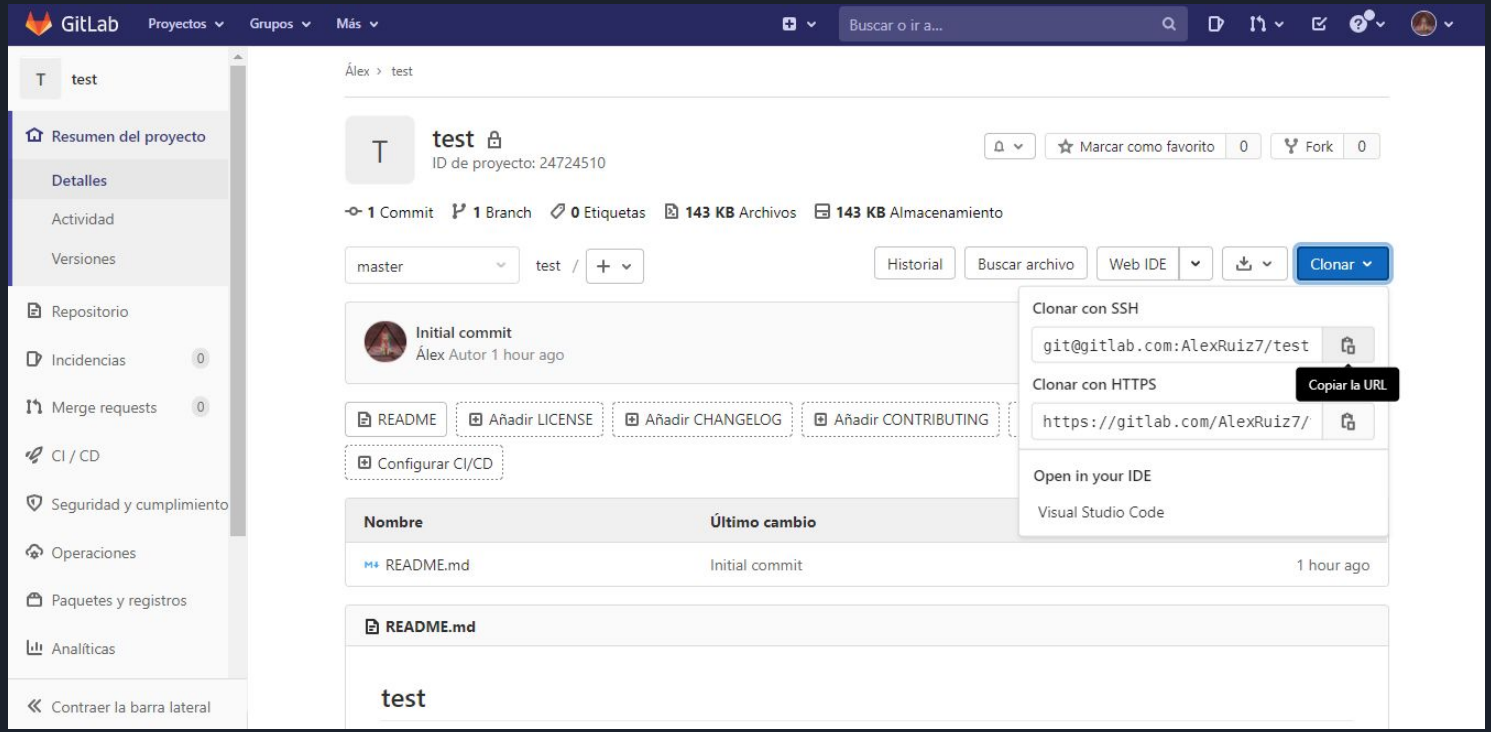
Visual Studio Code

VS Code: clonar un repositorio

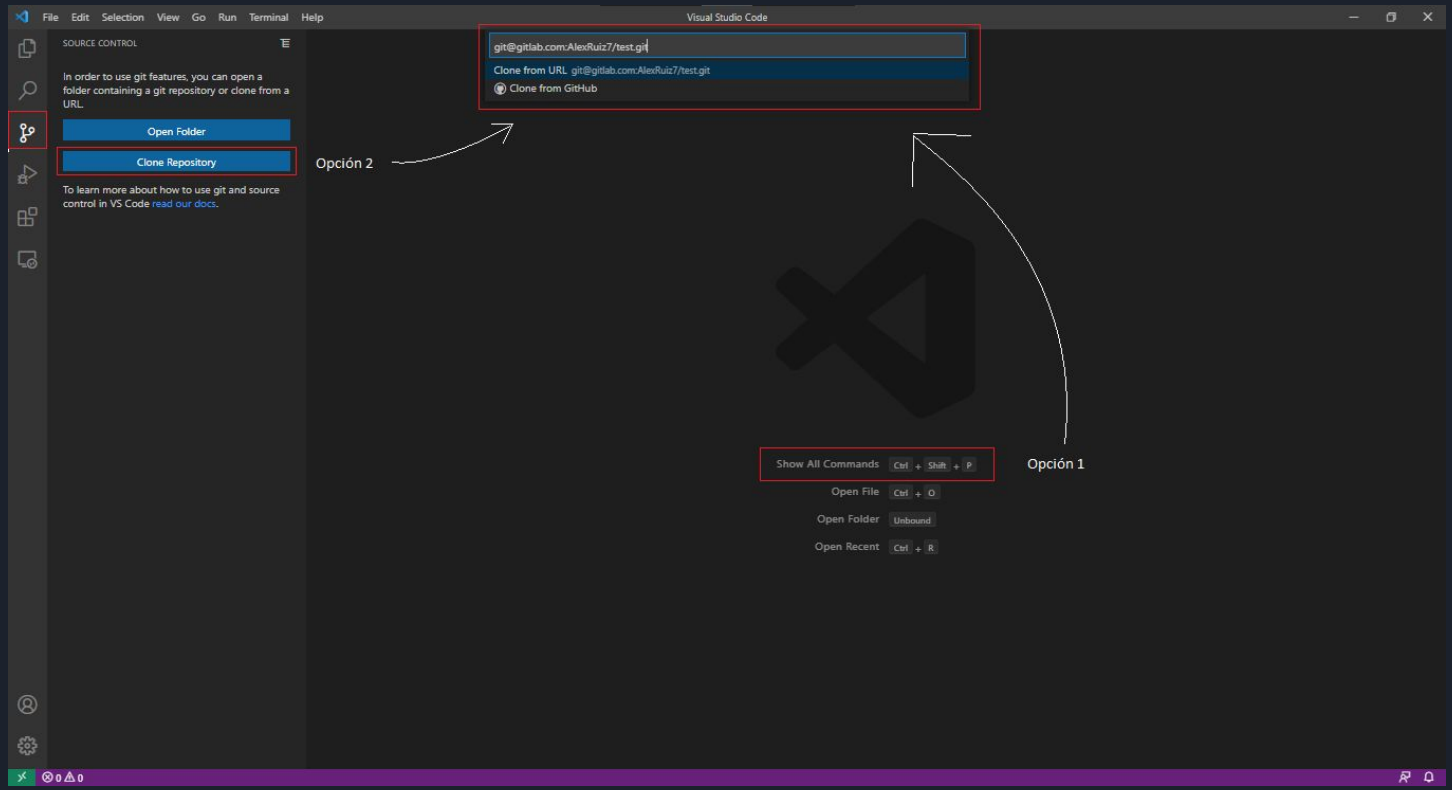
Para clonar un repositorio usando VS Code hacemos lo siguiente:

- En Gitlab:
  - Vamos a la página del repositorio en Gitlab y hacemos click en “Clonar”.
  - Clonamos mediante SSH, así que hacemos click en copiar la URL SSH.
- En VS Code:
  - Opción 1: ´Ctrl + Shift + P´ y escribimos “clone”, click en git:clone y pegamos la URL.
  - Opción 2: abrimos la sección de “Source Control” en la barra izquierda y hacemos click en “Clone Repository”. Pegamos la URL.

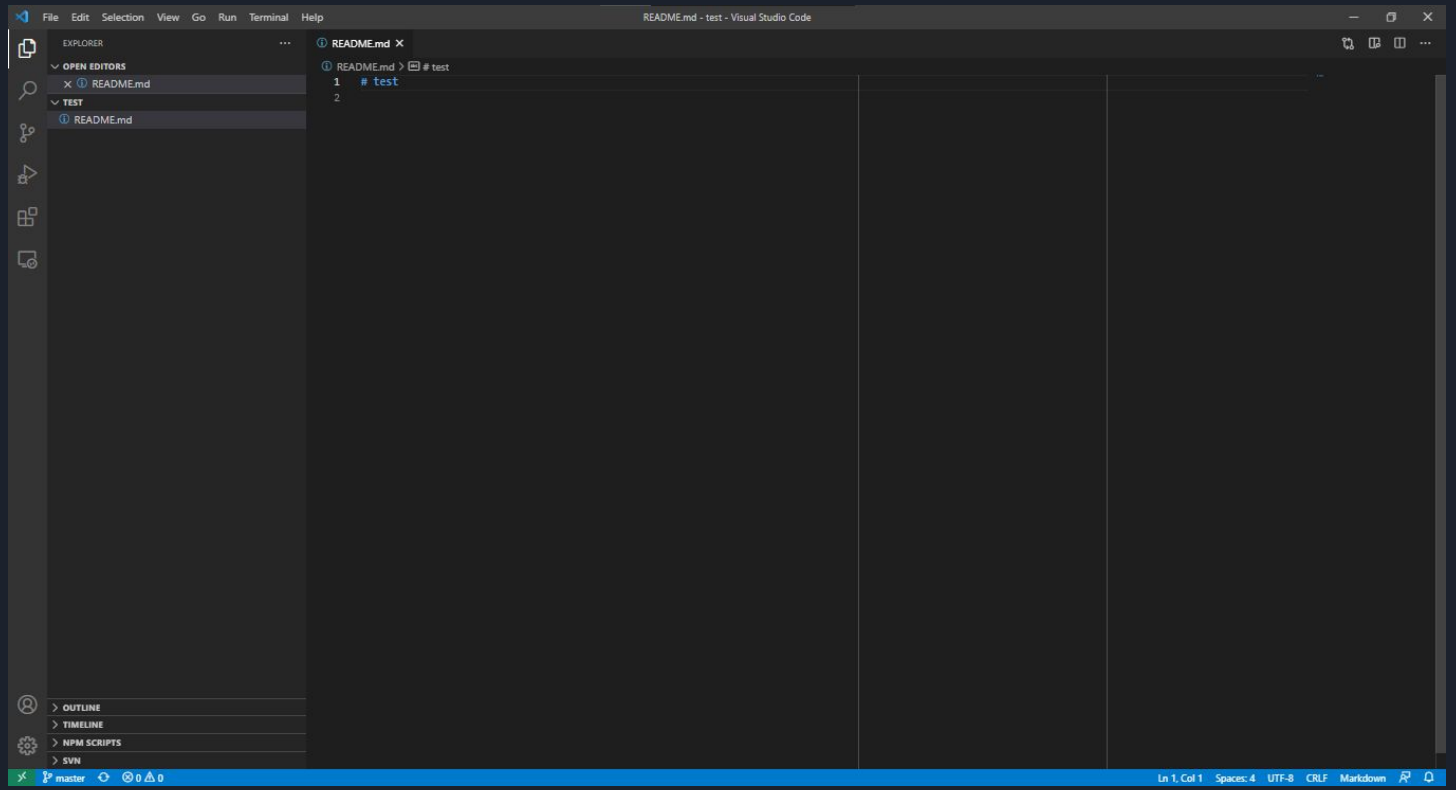
Copiar URL del repositorio (SSH)



Clonar repositorio usando VS Code



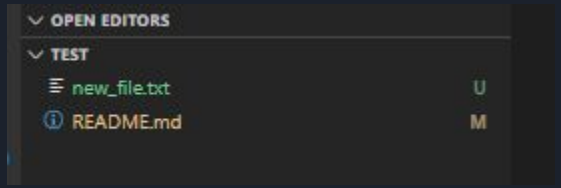
Resultado


File Edit Selection View Go Run Terminal Help
README.md - test - Visual Studio Code
EXPLORER
OPEN EDITORS
README.md
TEST
README.md
README.md
test
1 # test
2
OUTLINE
TIMELINE
NPM SCRIPTS
SVN
master
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Markdown

Subir cambios a Git usando VS Code
git commit & git push

Visual Studio Code nos irá coloreando los ficheros en distintos colores según su estado comprado con lo que hay subido en Git:

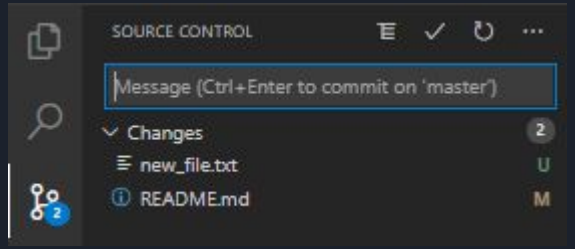
● Verde	Untracked	ficheros nuevos
● Naranja claro	Modified	ficheros modificados
● Rojo	Deleted	ficheros eliminados



Subir cambios a Git usando VS Code
git commit & git push

Para los subir los cambios a git hay que realizar 3 pasos:

1. Añadir los cambios (stash - git add)
2. Guardar los cambios (commit - git commit)
3. Subir los cambios (push - git push)



En VS Code vamos a la sección de “Source Control” y hacemos click en el tick para realizar un commit.

Nos preguntará si queremos añadir todos los cambios detectados al commit, le decimos que sí. Luego, nos pedirá un mensaje para el commit.

Alternativamente podemos acceder a estas y a más opciones haciendo click en los 3 puntos suspensivos.

Subir cambios a Git usando VS Code
git commit & git push

Una vez hemos realizado el *commit* podemos subirlo a Gitlab realizando un *push*.

Para ello hacemos click en los puntos suspensivos y luego en Push.

Listo, nuestros cambios ya se han subido a Gitlab.

Podemos hacer cuantos commits queramos, sin necesidad de subirlos uno a uno. Cuando hagamos push se subirán todos los commits que haya hasta ese momento.

Resultado en Gitlab