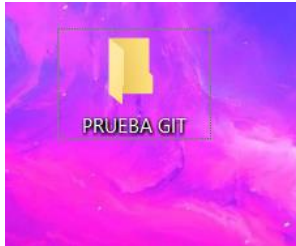


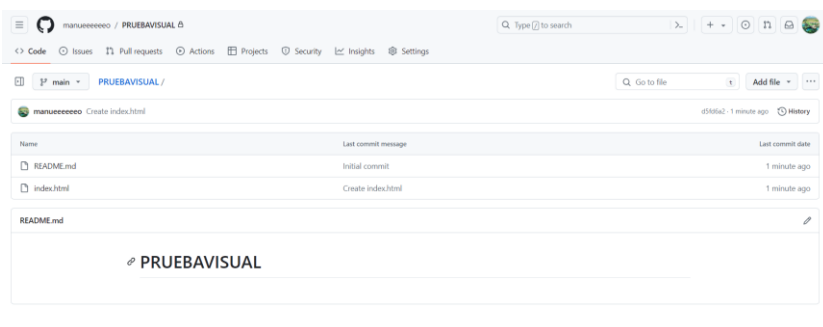
TRABAJAR CON GITHUB EN VSCODE

¡¡TIENES QUE TENER GIT INSTALADO EN TU ORDENADOR!!

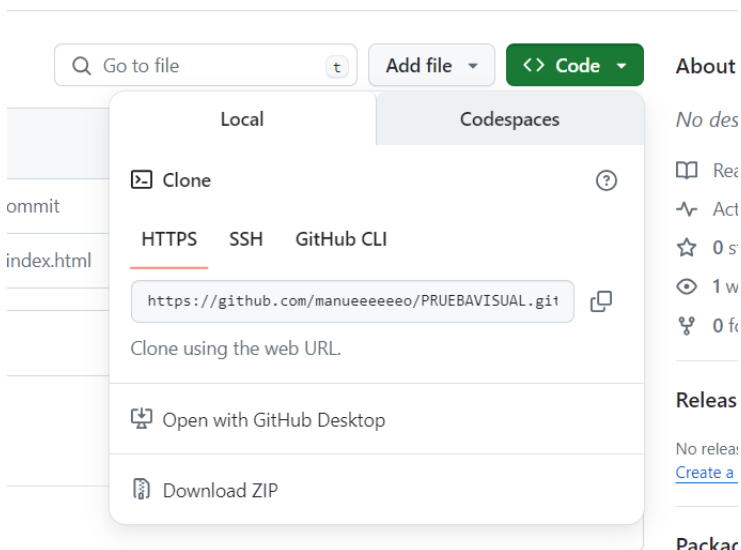
Primero que todo creé una carpeta en su escritorio con el nombre que usted desee.



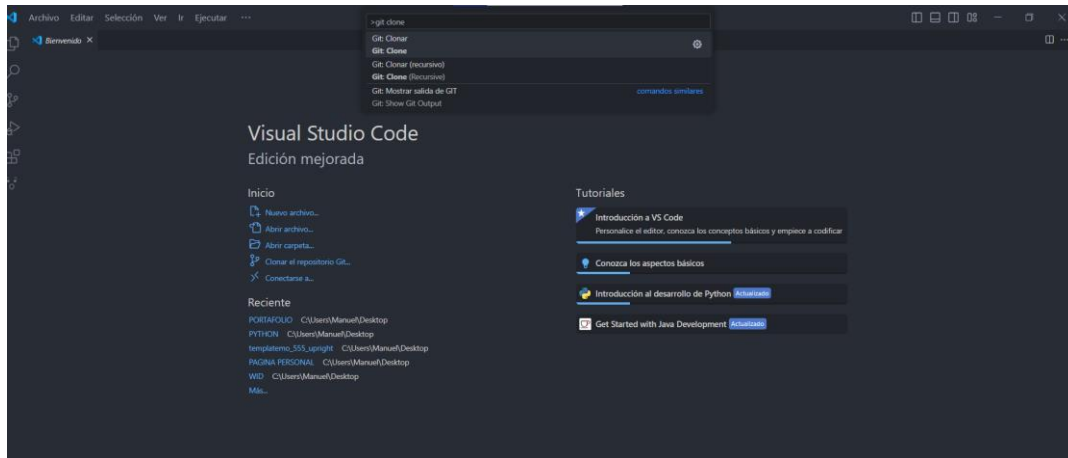
Váyase a su GitHub, creé un repositorio o diríjase al repositorio con el que va a trabajar. Puede crear archivos, para comprobar que todo está correcto.



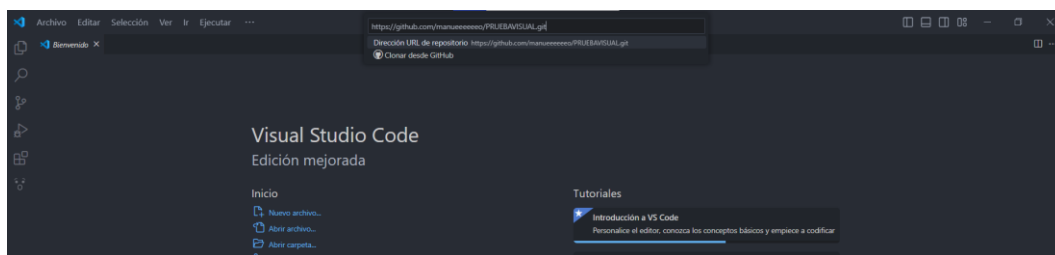
Para copiar el enlace del repositorio que necesitaremos más adelante, toque sobre donde dice “Code” y copie el enlace que ve:



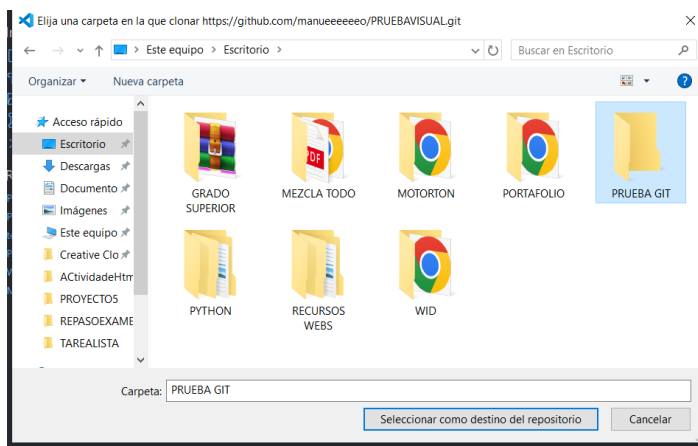
Abra VSCODE y déjelo completamente vacío.



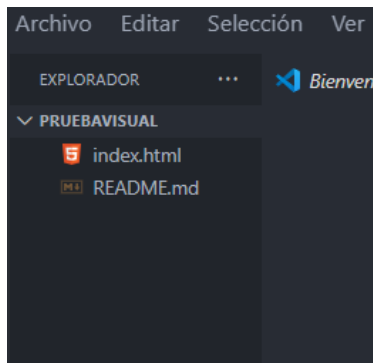
En la parte de buscar escriba “>git clone”, este comando lo que nos permite es copiarnos de forma local un repositorio que hay en la nube de GitHub a través de Git.



Nos solicitará seleccionar la carpeta en la cual vamos a guardar todo el repositorio, en este caso deberemos de señalar la carpeta que creamos al principio del tutorial.



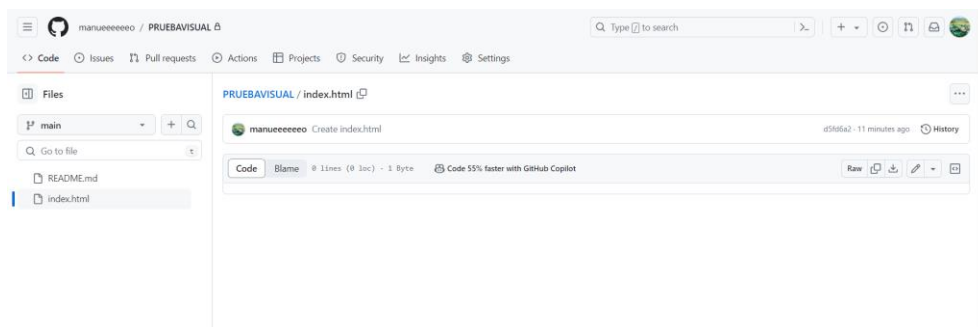
Una vez seleccionada y abierta y clonada, vemos que nos sale en el explorador de archivos del VSCODE los dos archivos que tenemos en el repositorio de GitHub



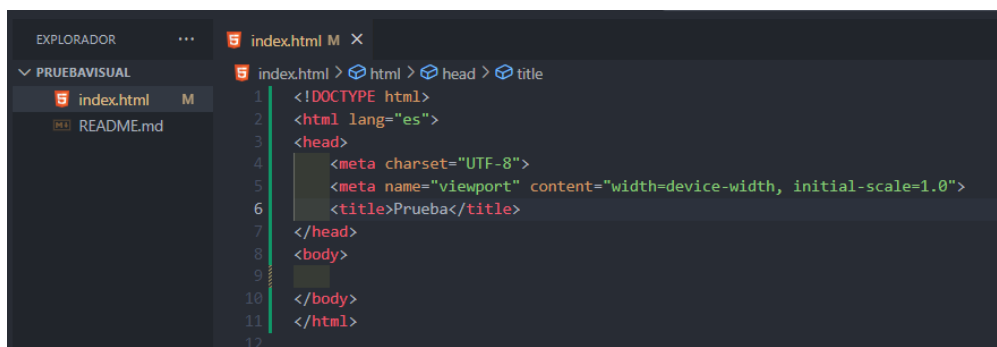
Antes de hacer ningún cambio si puede ser, deberemos de configurar el archivo de configuración de Git para que a la hora de subir los archivos, salga quien lo ha actualizado, borrado...

Usaremos los comandos en la consola del VSCODE de “git config user.name “TU NOMBRE”” y “git config user.email tu_email” el email preferiblemente que seá el mismo que tienes en la cuenta de GitHub.

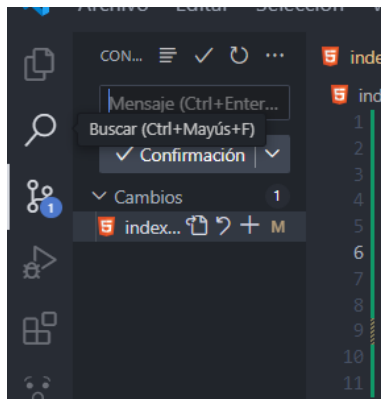
Como podemos ver en la imagen de abajo, el archivo index.html que tenemos en GitHub está vacío, vamos a cambiarle.



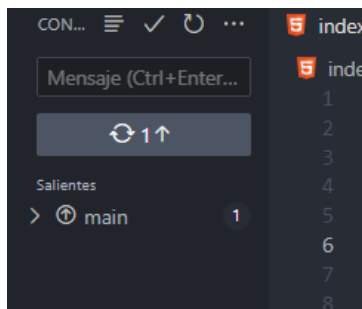
Hacemos un HTML normal.



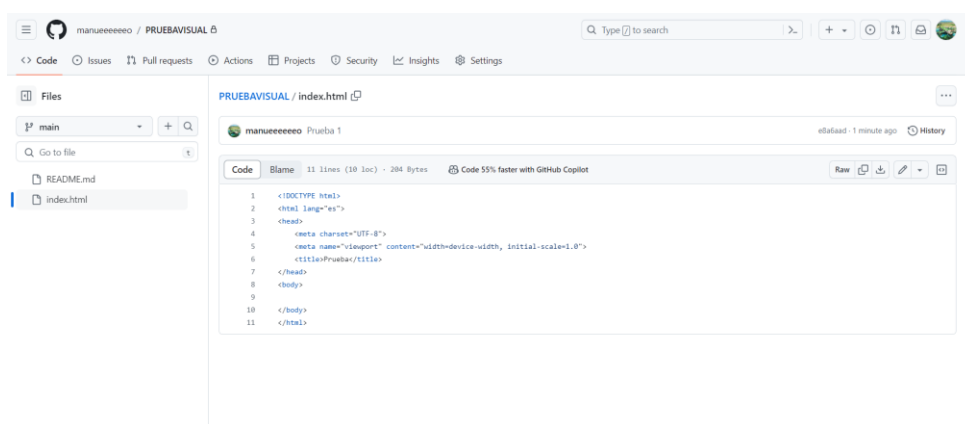
Vamos al apartado de Control de Versiones, que vamos a ver que tenemos como una notificación, ya que hemos hecho un cambio en local y no lo hemos subido. En donde pone Mensaje, escribimos un breve comentario para el commit y damos Ctrl+Enter, una vez dado:



No nos aparecerá nada como en la imagen de abajo, le debemos de dar a donde hay un símbolo de cargar y otro como de subir algo. Nos dirá que vamos a hacer un commit y se publicará le damos a que sí y esperamos a que se cargue.



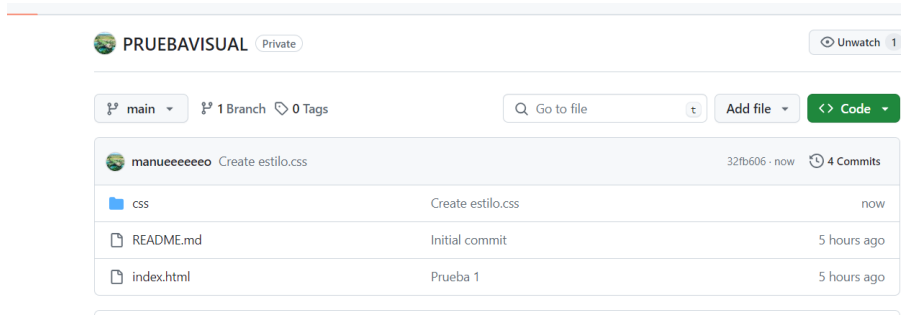
Ahora nos vamos a GitHub y como vemos sí que se ha actualizado el paquete.



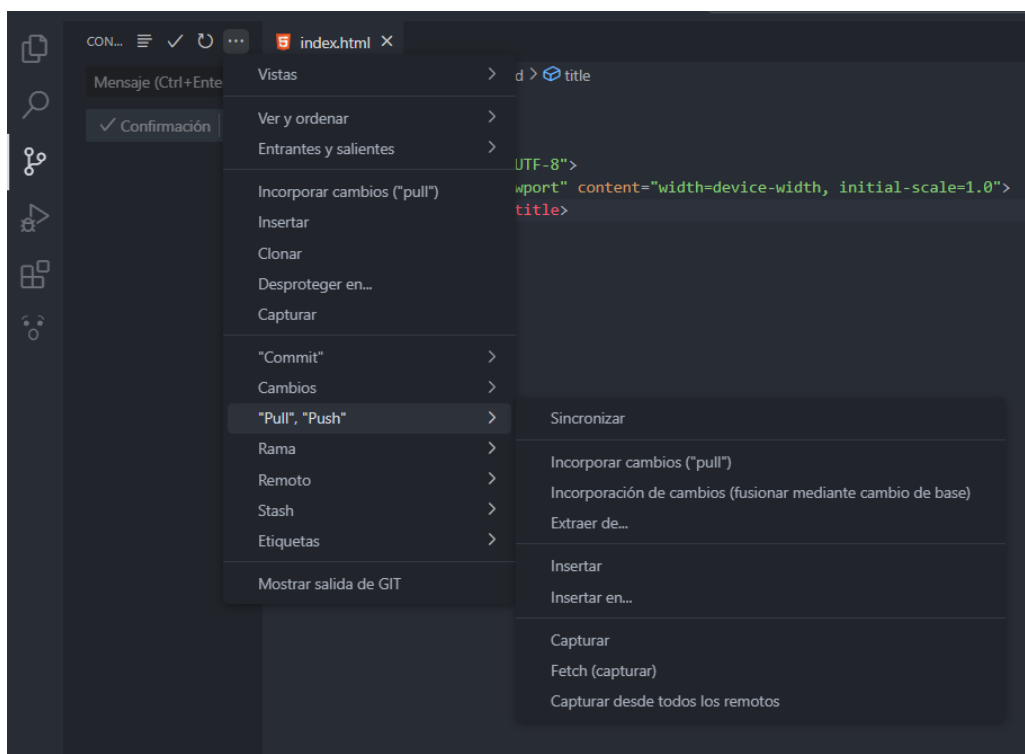
PARA OBTENER LOS CAMBIOS REALIZADOS EN EL REPOSITORIO

Primero que todo, saber que cuando nosotros hemos hecho un cambio en local y lo hemos subido a GitHub, aunque el compañero se hubiera descargado el repositorio y trabaje con él, está trabajando con él en local, por eso, cuando se actualice deberemos de actualizar nuestro repositorio local.

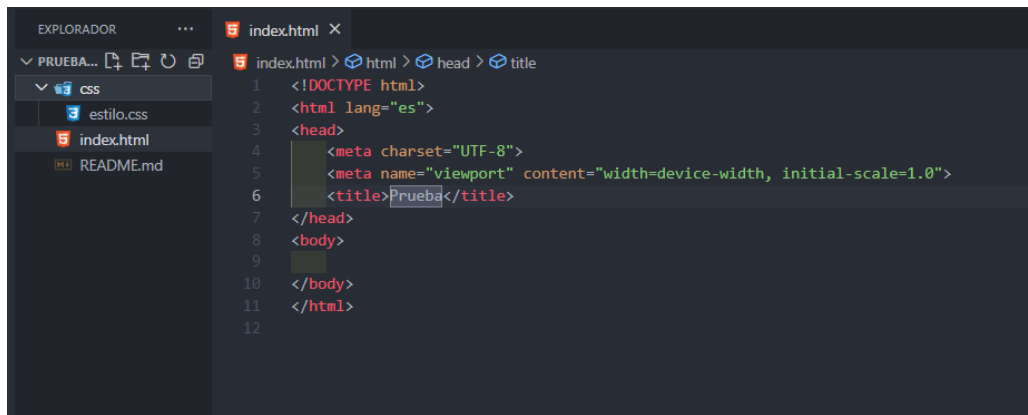
Hemos añadido en el repositorio una carpeta css, con un archivo de estilos.css vacío.



Nos vamos a VSCODE a la parte del Controlador de Versiones, damos click en los 3 puntitos, damos donde pone Pull y Push y ahora en la opción que pone “Incorporar cambios(“pull”)”.



Esperamos que deje de cargar la barrita que sale, volvemos al explorador de archivos y vemos que ahora tenemos el repositorio actualizado.



TODOS LOS COMANDOS EXISTENTES BÁSICOS

Crear un repositorio local

Git init -> Iniciar un repositorio

Git config user.name "Tu_nombre" -> Configurar el nombre de usuario para los cambios

Git config user.email tu_email -> Configurar el email de usuario para los cambios

Git --global config user.name "Tu_nombre"

Git --global config user.email tu_email

La denominación --global nos sirve para asignar esos valores para todos los proyectos que vamos a realizar, es mejor no usarlo.

Integrar un repositorio remoto

Git remote add origin "URL REPOSITORIO" -> Conectar la carpeta en la que estamos con el repositorio de GitHub

Git clone "URL REPOSITORIO" -> Clonar un repositorio de GitHub en una carpeta en la que nos situamos

Comandos para confirmar cambios

Git add <nombre_fichero> -> Añadir un fichero en particular

Git add . -> Añadir todo

Git commit -m "Mensaje del commit" -> Hacer un commit de los cambios y otorgarle un título

Git push origin <nombre_rama_local> -> Enviarlos al repositorio a la rama que nosotros deseemos

Git branch -> Comprobar la rama en la que estamos

DIAGRAMA DE SECUENCIAS DE GIT

