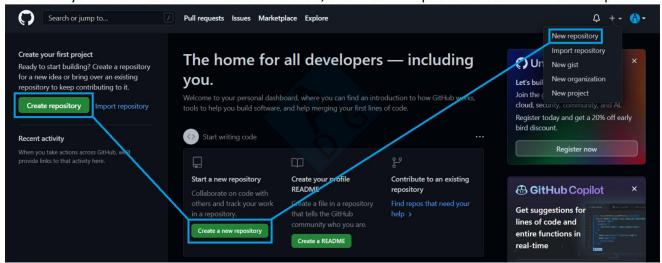


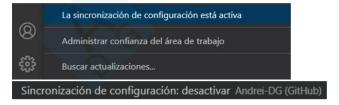
1. Hacer un repositorio en remoto.

Teniendo ya una cuenta recién creada de Github, creamos un repositorio desde una de las opciones:



Elegir el propietario, nombre del repositorio, descripción, y click en Crear repositorio.

Conectar el Microsoft Visual Studio Code con GitHub:

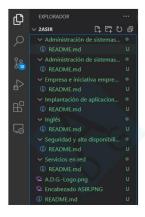


2. Clonar el repositorio en el Microsoft Visual Studio Code

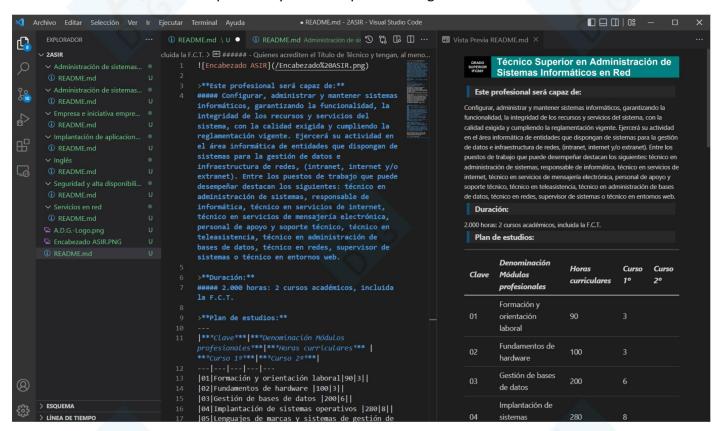
Copiar el enlace que da presionando F1, Git: clonar, Clonar desde Github, click en repositorio de Github, elegir carpeta del equipo local donde almacenarlo y Abrir.



3. Crear un README.md del repositorio, las carpetas de asignaturas y un README.md en cada una de ellas, y añadir los dos png's que usaré en el md del repositorio:

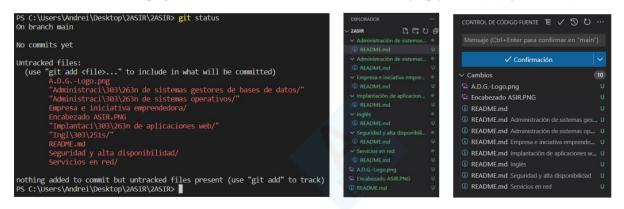


Modificar el README.md del repositorio y de las carpetas de asignaturas:

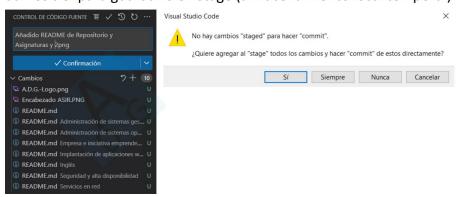


Podemos comprobar con *git status* los archivos o carpetas que no se siguen o no están guardados en local, al igual que en el *Explorador* por los signos U (untracked, sin seguimiento) ó el punto verde de las carpetas.

En el Control de código fuente se ven solamente los archivos png y readme, que es donde Confirmamos.



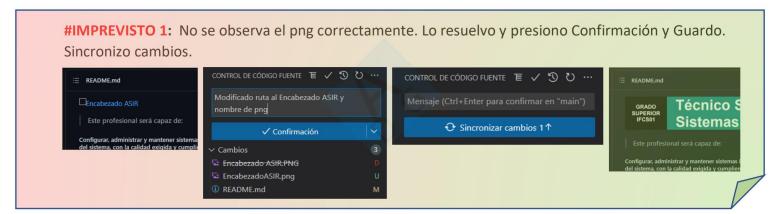
Escribimos en el recuadro un mensaje recordatorio de lo hecho para el commit, click en *Confirmación* y le damos a *Sí* para guardarlo en stage (almacenamiento local temporal).



4. Preparamos el terreno para poder crear un conflicto:

Enviamos todos nuestros cambios guardados en stage (local) al remoto (Github).

PS C:\Users\Andrei\Desktop\2ASIR\2ASIR> git push -u origin main Enumerating objects: 19, done.
Counting objects: 190% (19/19), done.
Delta compression using up to 8 threads
Compressing objects: 190% (12/12), done.
Writing objects: 190% (19/19), 32.03 KiB | 10.67 MiB/s, done.
Writing objects: 190% (19/19), 32.03 KiB | 10.67 MiB/s, done.
Total 19 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Andrei-DG/2ASIR.git
* [new branch] main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\Andrei\Desktop\2ASIR\2ASIR>



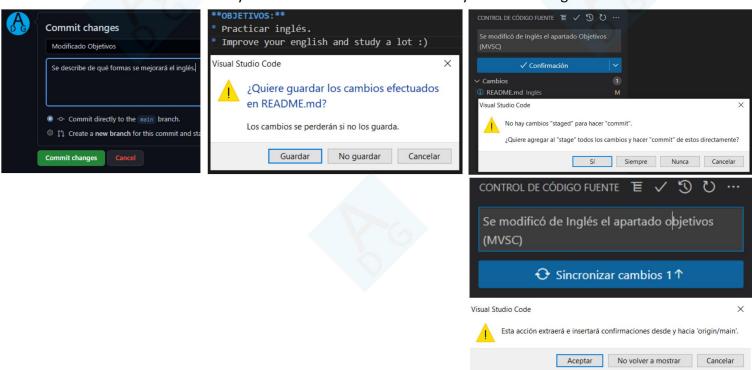
Como recién tenemos hecho el push del local al remoto, estará todo actualizado en Github. Teniendo todavía nuestro repositorio abierto y sincronizado en el MVSC, sin cerrarlo, entramos en Github:

1. Modificamos el README.md de la carpeta Inglés en Github (ya que esa la dejé muy cutre (3)):

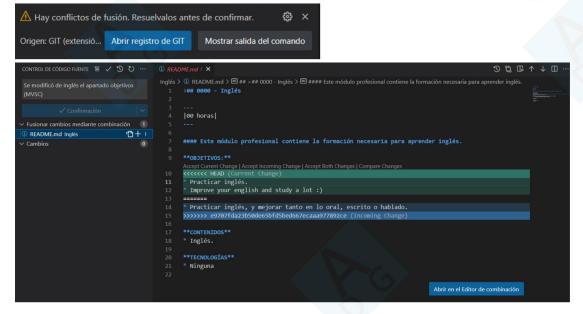


2. Modificamos el README.md de la carpeta Inglés en Local en MVSC:

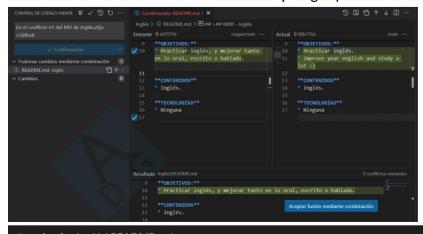
3. Hacemos commit en GitHub y en Local del mismo README.md y es cuando se genera el conflicto:



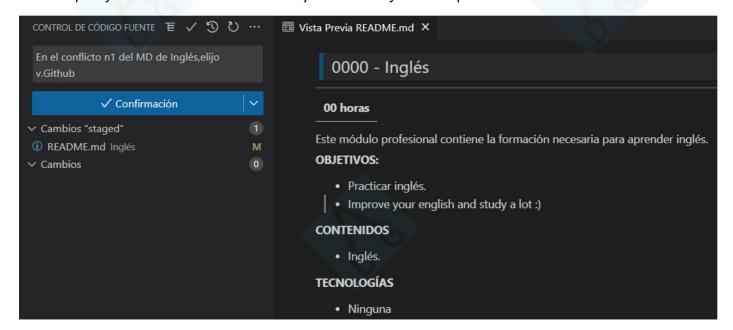
En cuanto hacemos los dos commit de remoto y local del mismo archivo a la vez, nos salta el aviso:



Click en Abrir en el Editor de combinación y elegir quedarnos con la versión que se subió a Github:



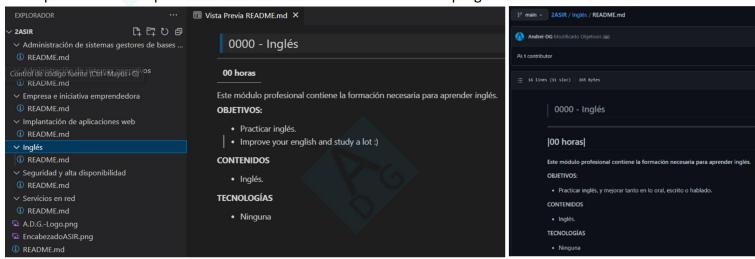
Click Aceptar fusión mediante combinación y Click en Confirmación para el commit de local a remoto.



Sincronizamos y Aceptamos:



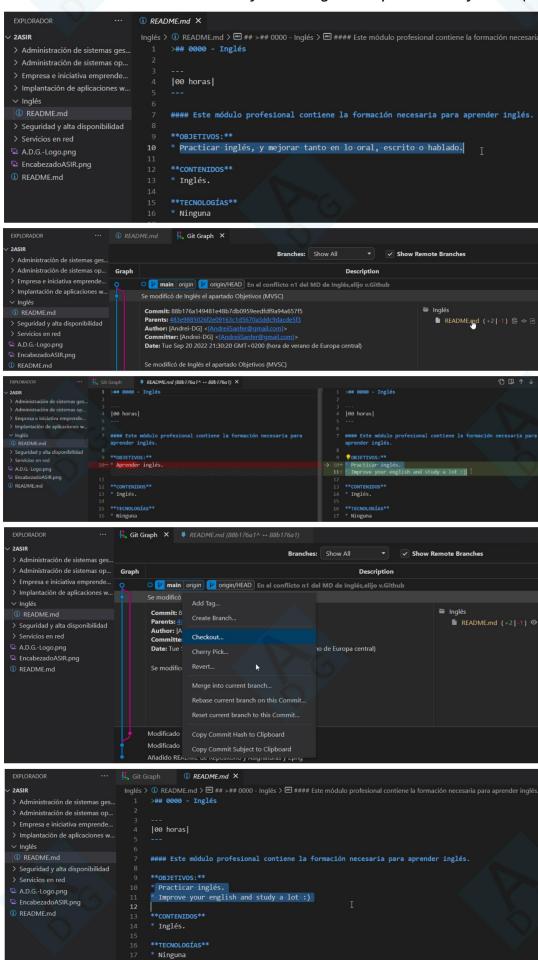
Se puede observar que el conflicto fue solucionado a elección del programador: MVSC – Github



#IMPREVISTO 2: Tuve un problema que no entendía muy bien los checkout, iba probando y luego al final decidí volver a clonarlo del remoto a local con los últimos cambios guardados (y correctos sin checkouts, para poder hacer las capturas).

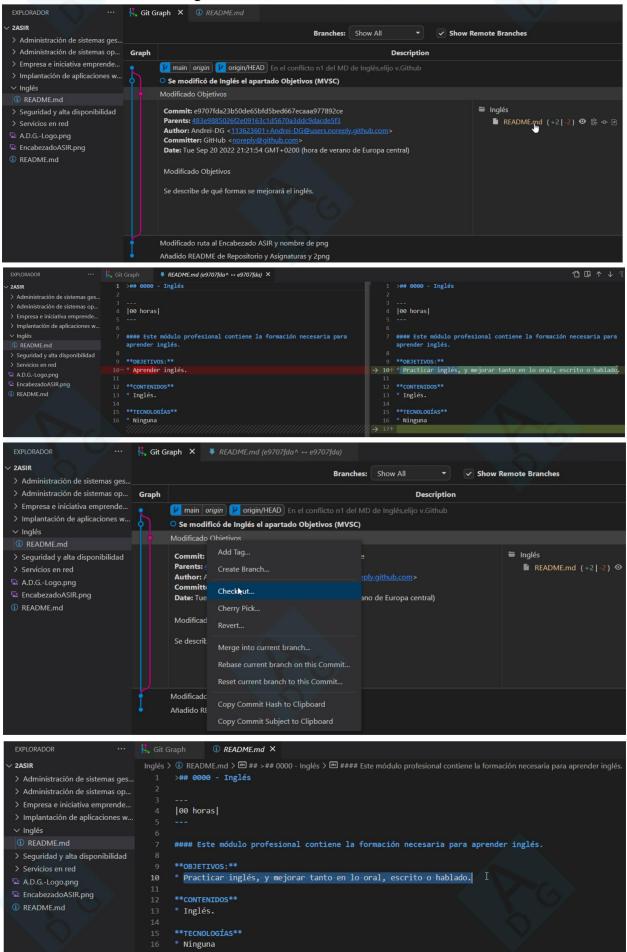
5. Hacer Checkout a varios commits.

1. Commit con nombre: Se modificó de Inglés el apartado Objetivos (MVSC)



2. Commit con nombre: Modificado Objetivos

Desde el estado anterior se sigue modificando el mismo archivo mediante checkouts de commits:



He dejado la versión de README.md de Inglés que tenía antes.

Captura de mi repositorio Github:

