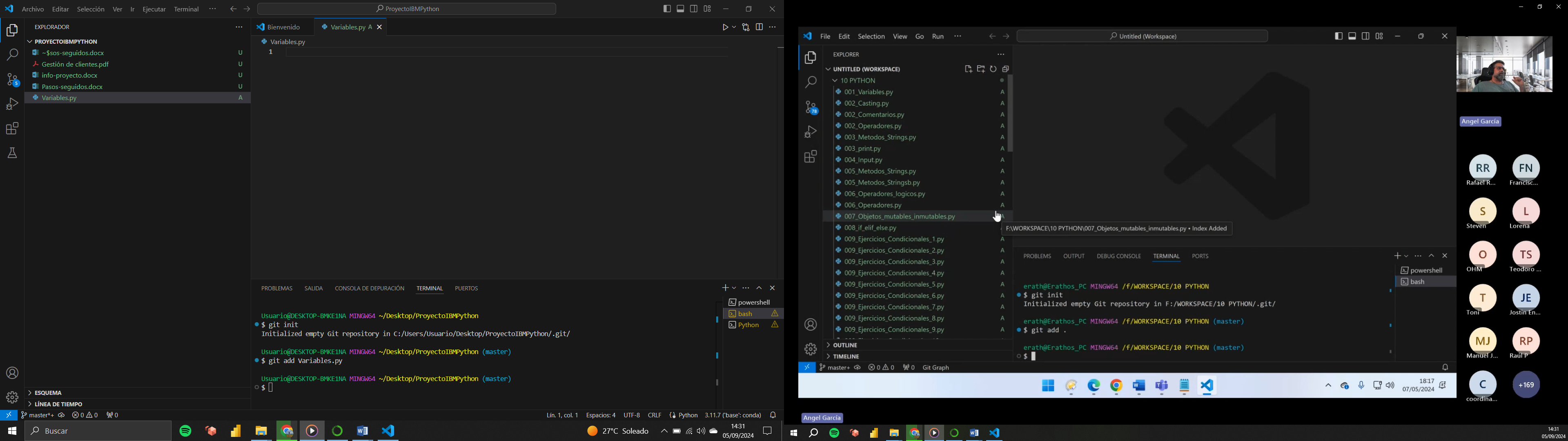
creado el repositorio en GitHub – IBMProject1

Abierto VS studio – Carpeta de interés. Abrir integrated terminal. Una vez ahí, click en el lado derecho y abrir git terminal (o bash terminal).

$ git init #comienza a trabajar en el proyecto

$ git add . # los introduce en la stagging área (sala de espera) (todos en general) o git add nombrearchivo.txt

En mi caso voy a realizar el seguimiento solo a un archivo .py que voy a crear para el video online de variables.



La A significa que está monitorizando los archivos, y que en el momento que yo haga un cambio puedo hacerle un commit y lo implantará, los que tienen la U no tienen seguimiento.

A los commit siempre se les pone un comentario. –m “Primera modificación”

Despues de modificar el archivo, he hecho un commit pero desde la interfaz de VS code.  
He creado un segundo archivo. Condicionales.py y le he realizado un seguimiento con

$ git add Condicionales.py

A continuación, voy a añadirlo a mi repositorio online de Github con el siguiente código (que lo obtuve cuando cree el repositorio)

git remote add origin https://github.com/Niketelocreas/Proyecto-IBM1.git

Ahora hay que sincronizar la rama con la que se está trabajando, que por defecto en VS code es la rama master, pero github te da el código para llamarla main, asi que hay que cambiar el código para no provocar conflictos.

git push -u origin main

Al final he terminado añadiendo todos los archivos en la carpeta con un

$ git add .

Y después un

$ git commit –m “Segunda Version”

Finalmente, en la barra lateral le he dado a sincronizar cambios, para que salgan en github

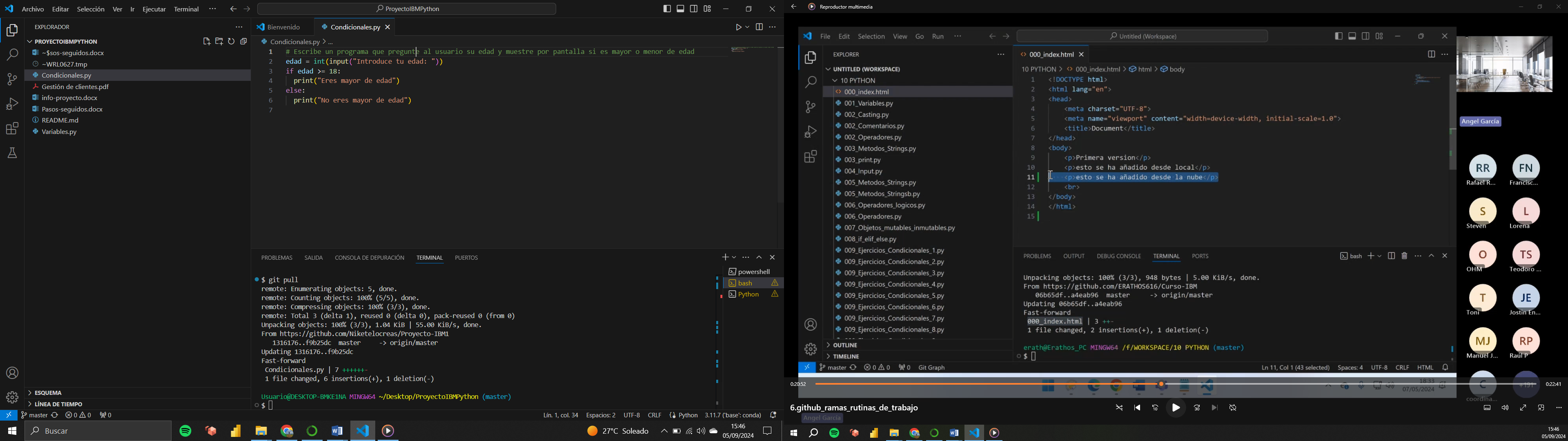
Cuando lo que necesite sea recuperar un archivo de la nube o descargar el archivo que nos haya pasado alguna compañera, se usa git pull!!

Para demostrar la diferencia realizaré el cambio en la nube y me descargaré la versión en local en el documento condicionales

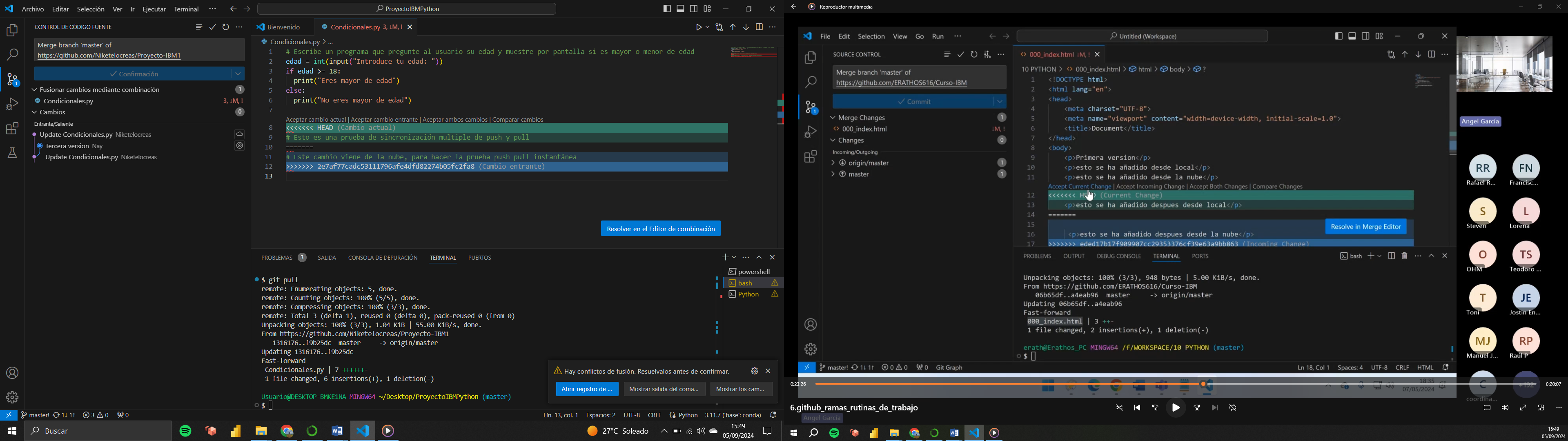
Solo hay que realizar:

$ git pull

Y aquí se observan los resultados:



A continuación realizo un cambio en local, añadiendo un comentario, y haciendo un commit, y hago lo mismo en la nube.  
Le doy al botoncito de recircular que hay al lado del nombre de la rama master, y eso genera un pull y push automáticos desde local y nube, de manera que el documento adquiere ambos cambios:



Si le doy a resolver conflicto, me muestra lo siguiente:



Estoy trabajando sobre la rama master porque he estado practicando, pero en un proyecto nunca se trabaja sobre la rama master. Hasta que el código no es funcional, no se mezcla la rama master con las accesorias

Con: “poner nombre orientativo” se abre una rama alternativa

$ git branch A001

A continuación, tengo que posicionarme en esa rama. Para ello, se usa el git checkout “nombredelarama”

No hace falta publicar los branches hasta que no nos los pidan o no esté medianamente terminado. Una vez terminas de trabajar en tu rama, hay que hacer un merge.

Primero hay que moverse a la rama master con git checkout master y una vez allí hacer un git merge y el nombre de la rama que quieres fusionar:

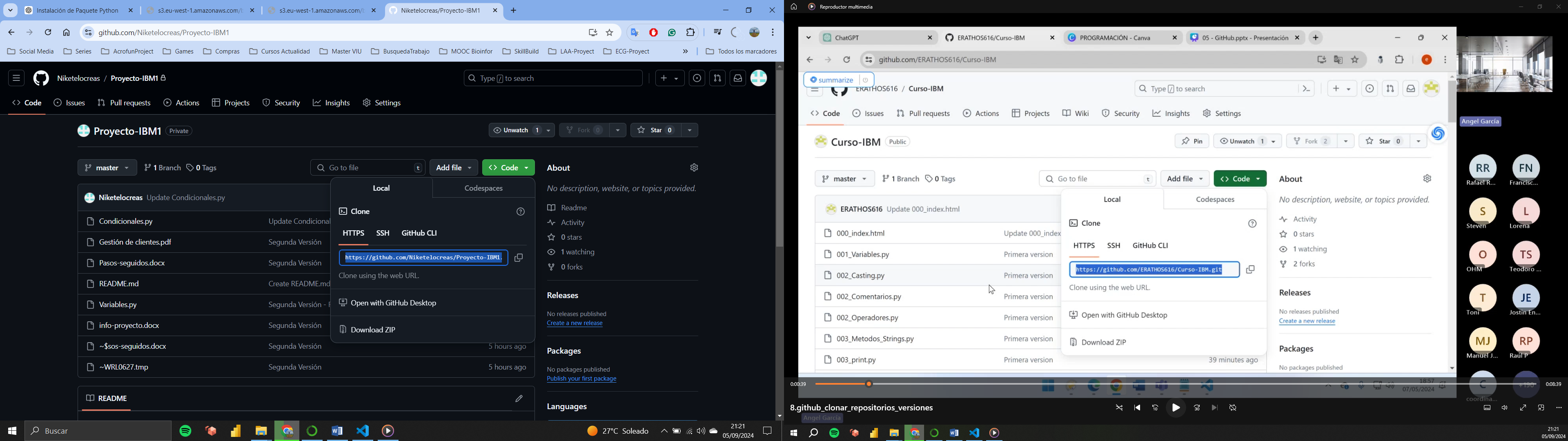
$ git merge A001

Para descargarme cambios que se hayan producido desde la nube, tengo que poner:

$ git pull

# Esto “descarga” los cambios de la nube automáticamente.

Se puede clonar automáticamente todo el contenido de un repositorio a local usando el link que se consigue bajo la pestaña code:



<https://github.com/Niketelocreas/Proyecto-IBM1.git>

Para ello solo hay que hacer:

$ git clone url

Los tags son las versiones definitivas del código:

Con el comando $ git log --oneline

Lo que consigo es ver la ID de los commits y las descripciones (o comentarios)

Los tags al ser las versiones definitivas, por lo que es importante añadir V1.0.1 o V2… Cuando se cambian cosas pequeñas, se cambia el último punto V1.0.2, si se han añadido nuevas funciones, una pagina nueva, etc, se cambia el segundo V1.1.2, y si se hace un cambio muy significativo, que casi son programas diferentes o con funciones bastante diferentes, se cambia el primero. V2.

Para crear un tag solo hay que:

$ git tag nombre –m “descripción”

El nombre es V1.0.1 o lo que sea.

Una vez se hace eso, si se quiere subir a la nube, hay que utilizar el git push - - tags