

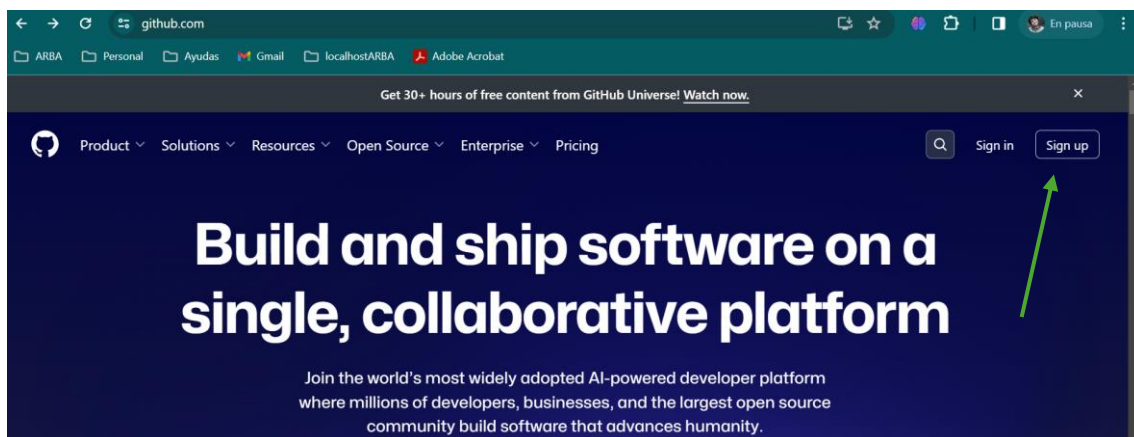
Guía Práctica

Descarga e instalación de componentes:

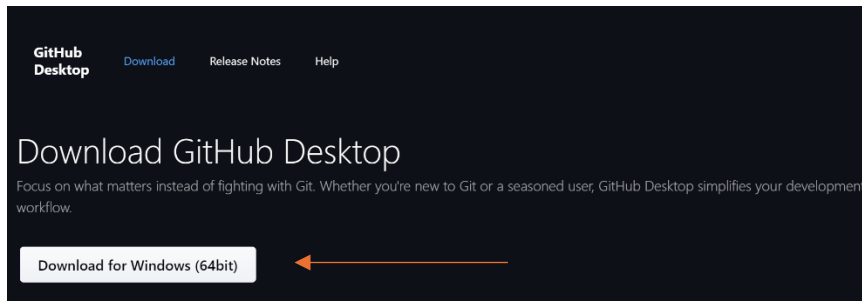
GitHub.

La misma es una herramienta de versionado que nos va a permitir trabajar ordenadamente y con una trazabilidad en la evolución de nuestro proyecto, para ello vamos a modular la tarea en dos actividades; la creación de nuestro perfil en la plataforma y la descarga de su versión de escritorio.

Para el primero es fundamental ir al [sitio web](#) de la herramienta y crear nuestra cuenta en el icono superior derecho *sign up*.

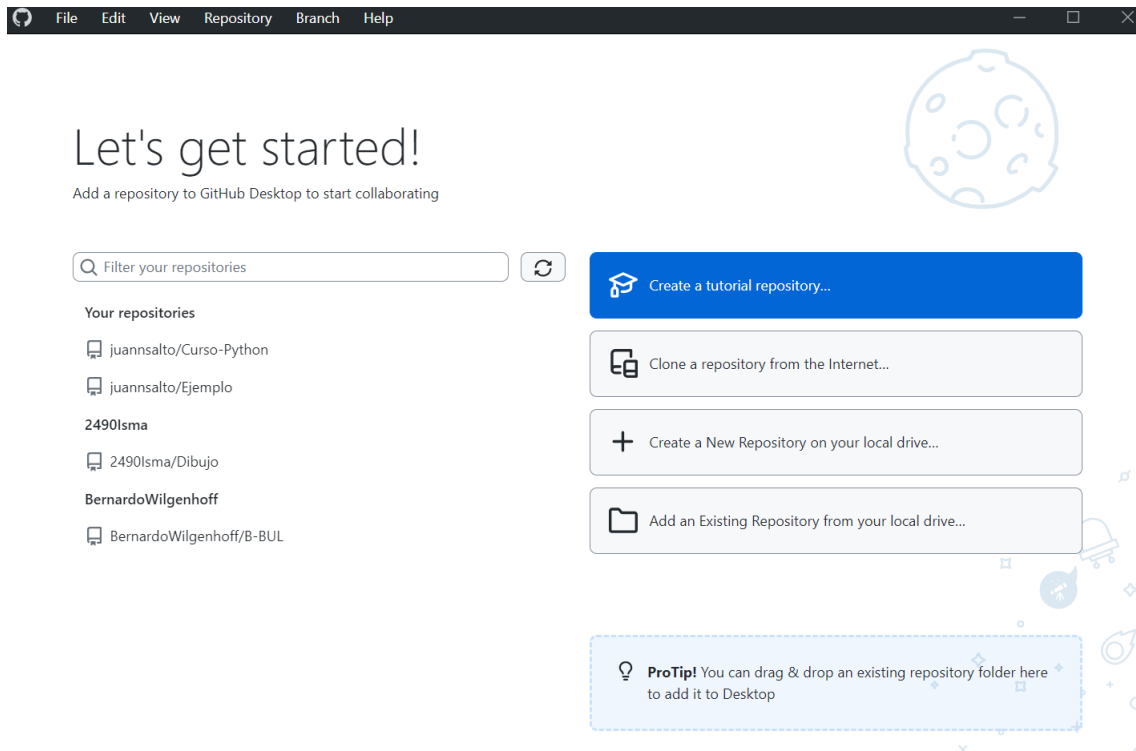


Una vez registrados (puesto nuestro email y validada la cuenta) tenemos que instalar la versión de escritorio. La misma se puede descargar del siguiente enlace: <https://desktop.github.com/download/> o buscando en el navegador *GitHub Desktop Download*. Es importante tener en cuenta la versión de la computadora que utilizamos.



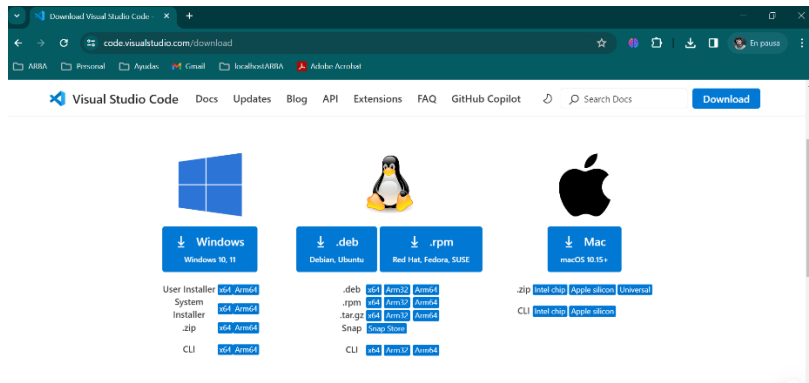
Una vez descargado, seguir los pasos de instalación del ejecutable descargado para la instalación de la herramienta. Se aloja en `C:\Users\Usuario\Downloads` y se puede abrir directamente de la pestaña descarga de cualquier navegador.

Llegando así al inicio



Visual Studio.

El siguiente complemento es el editor y entorno de desarrollo (IDE), el mismo es donde vamos a visualizar nuestros códigos y repositorios. Utilizamos este por estándar pero existen otros. El mismo se descarga siguiendo el [enlace](#) o buscando en el navegador *Visual Studio Download*. La primera opción del resultado de búsqueda suele ser la original, de todos modos, siempre verificar. Así mismo al entrar en la página oficial deben corroborar que la versión que van a descargar sea acorde a su sistema operativo.



Existen tres tipos de instalaciones y lo fundamental es que sepan que equipo tienen en un principio, para ello realicen los siguientes pasos:

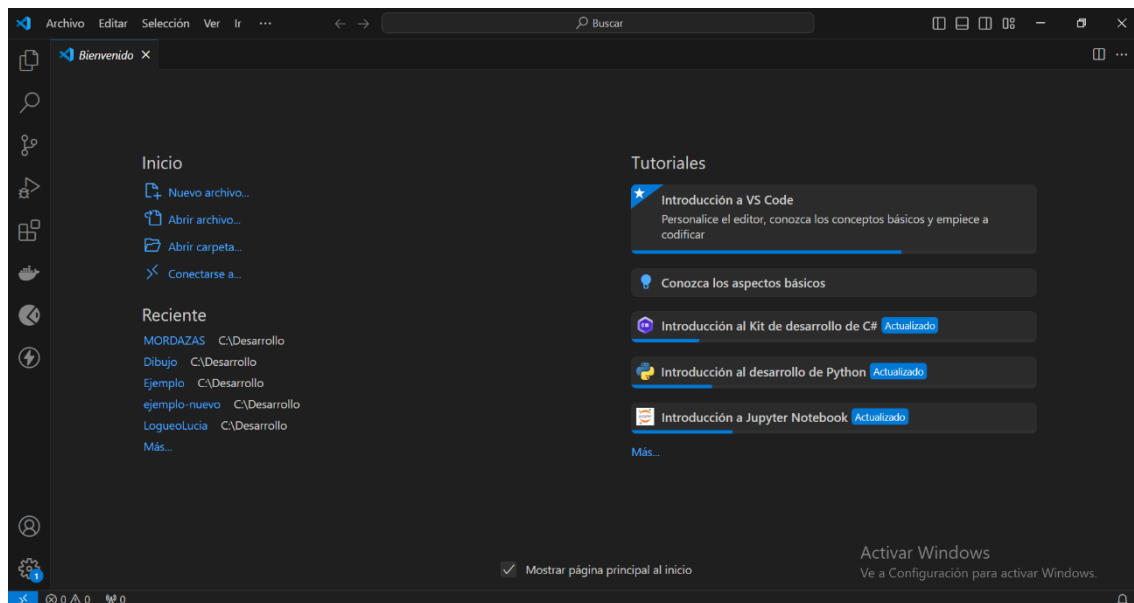
WINDOWS.

- Haz clic en el menú de inicio (el icono de Windows en la esquina inferior izquierda).
- Escribe "Configuración" y selecciona la opción que aparece.
- En la ventana de Configuración, selecciona "Sistema".
- Luego, haz clic en "Acerca de" (generalmente está al final de las opciones del menú).
- En la sección "Especificaciones de Windows", podrás ver el nombre y la edición de tu sistema operativo, así como la versión y el tipo de arquitectura (por ejemplo, 64 bits).

IOS.

- Haz clic en el icono de la manzana (ð) en la esquina superior izquierda de la pantalla.
- Selecciona "Acerca de este Mac" (About This Mac).
- Se abrirá una ventana que muestra información sobre tu Mac, incluyendo la versión de macOS que tienes instalada (por ejemplo, "macOS Ventura 13.0").

Una vez instalado debería verse tal que así:

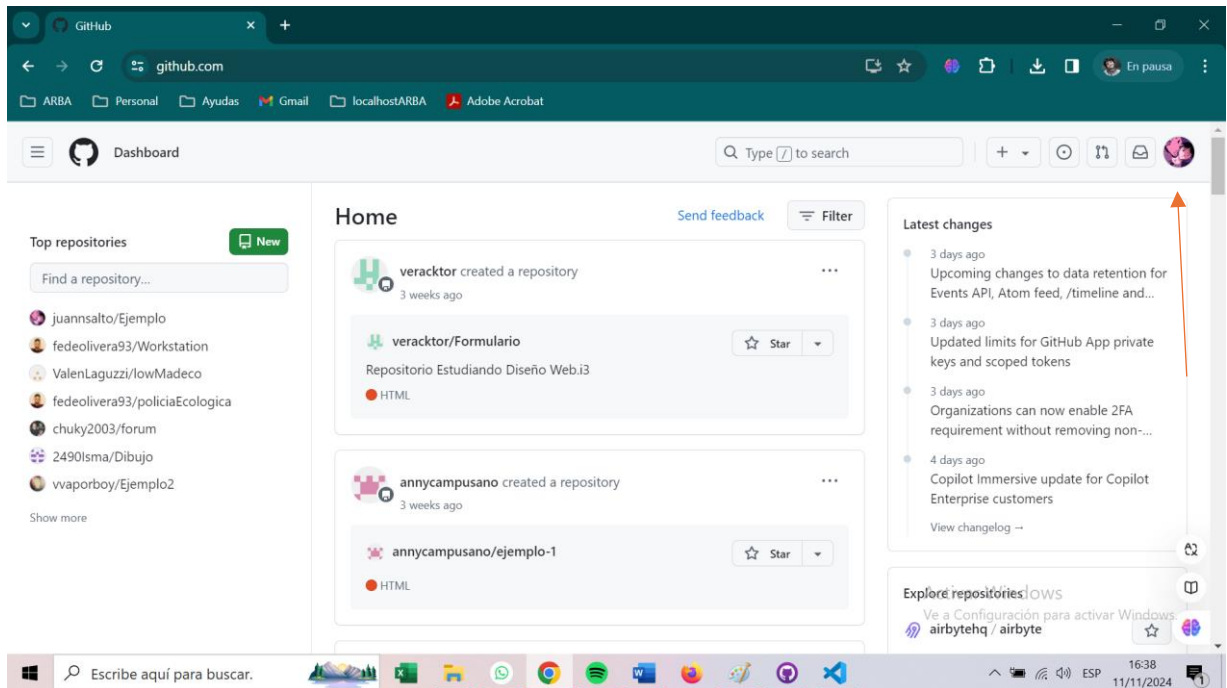


Tener en cuenta que esta versión ya se encuentra en español. Las formas de cambiarlo son **Ctrl+Shift+P** y buscar en la barra que despliega *configure display languages*. Si este método no funciona se puede buscar una extensión, para ello se va a una de las opciones que aparecen en el marcador a la izquierda y se busca la extensión que actualiza el idioma con *Spanish Language Pack for Visual Studio Code*.

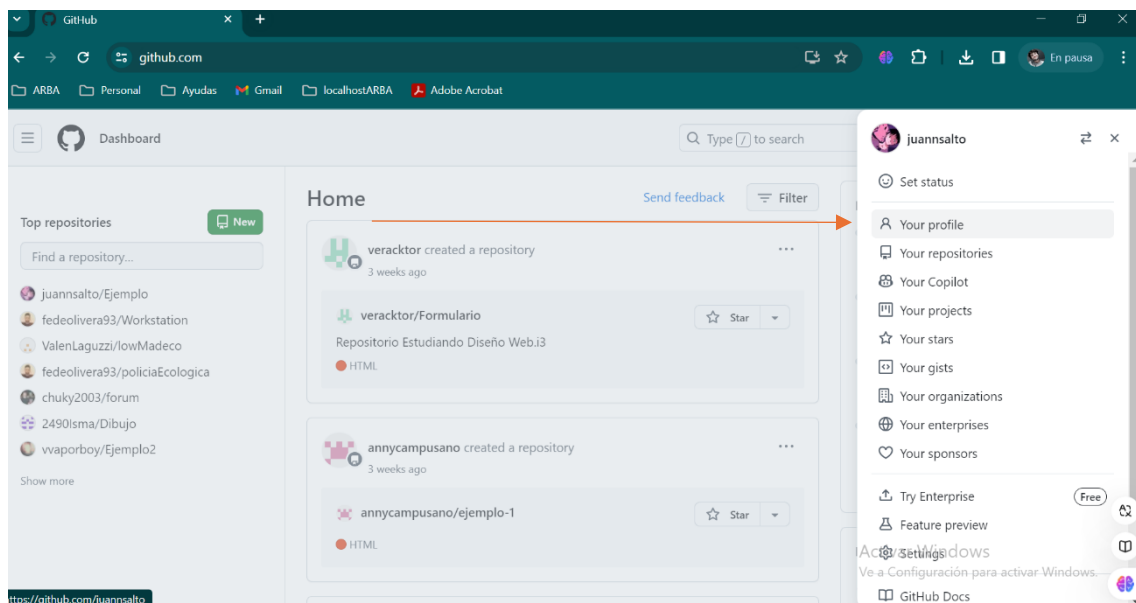


Creación del repositorio

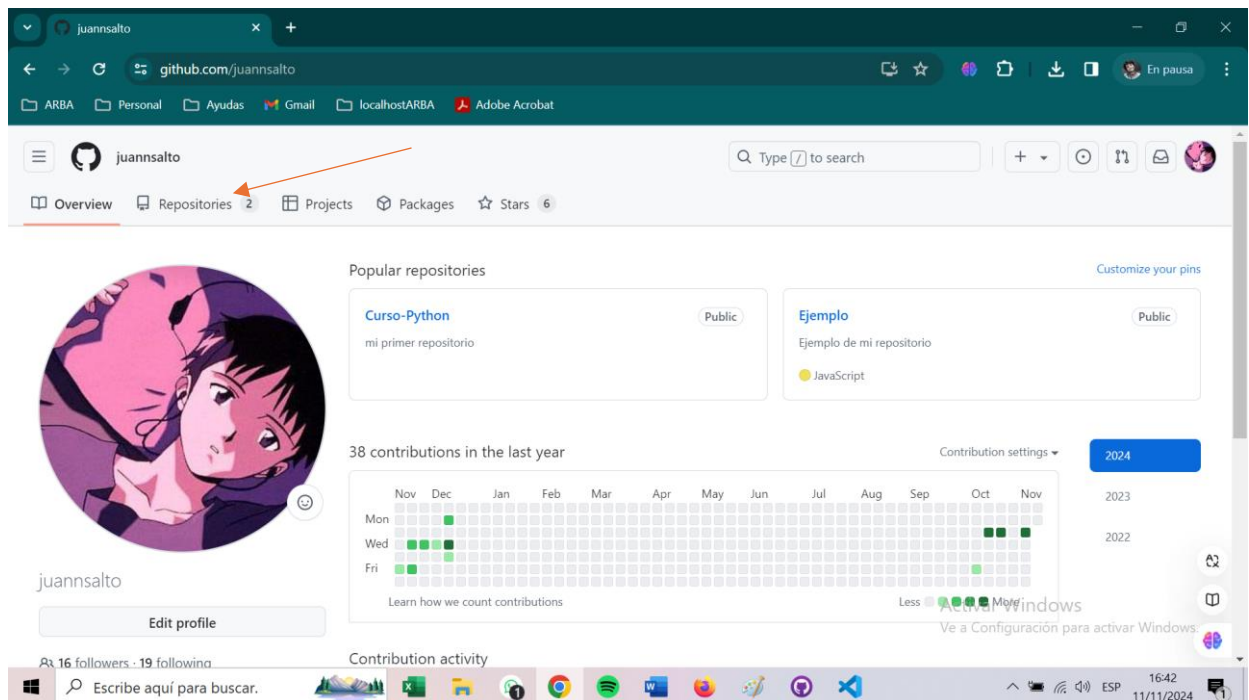
Primero vamos a abrir nuestra versión web de GitHub con el usuario y contraseña que hayamos creado.



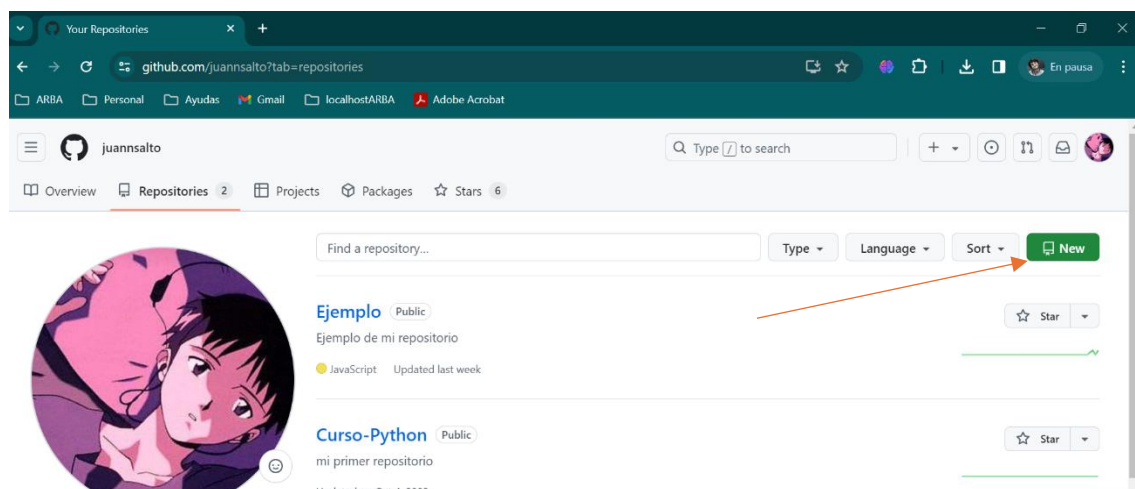
Una vez hecho, nos dirigimos a nuestro perfil, haciendo click en el icono de nuestro usuario y yendo a la opción *Your profile*



Una vez posicionados podemos ver nuestro perfil, seleccionar en *repositorios*



Desde allí tenemos la potestad de trabajar con todos los repositorios, como nuestro objetivo es crear el nuevo vamos a poner donde dice *New*.




Una vez ahí es necesario crear un repositorio publico con el nombre del proyecto que vamos a utilizar en el curso. Es importante no crear y borrar repositorios todo el tiempo porque necesitamos tener una trazabilidad de la evolución del curso, la misma se pierde si todo el tiempo se están creando proyectos nuevos. Se debe completar la descripción (opcional) y dar donde dice *Create Repository*

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 juansalto

Repository name *

primer repositorio

✓ Your new repository will be created as primer-repositorio.

The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about [super-duper-waddle](#) ?

Description (optional)

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

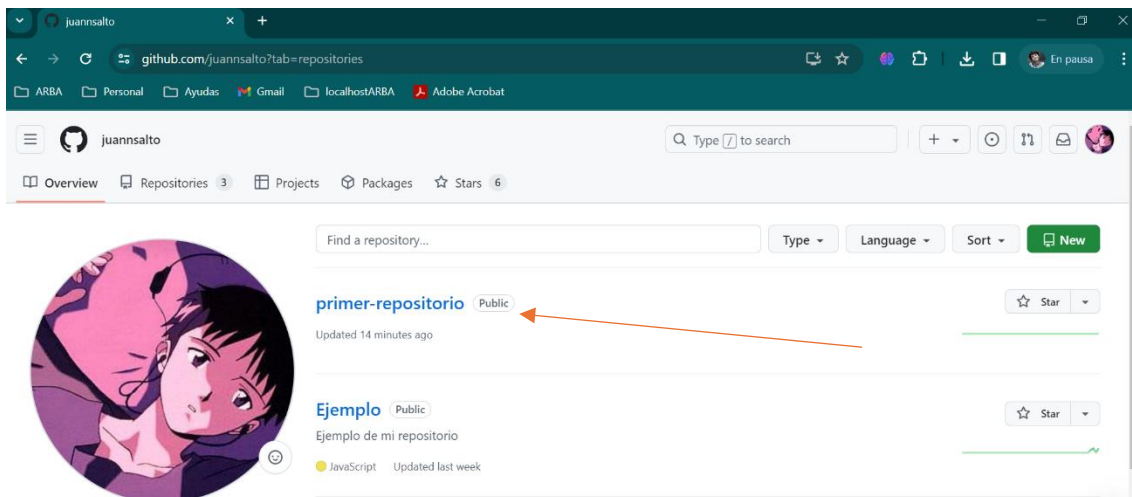
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

① You are creating a public repository in your personal account.

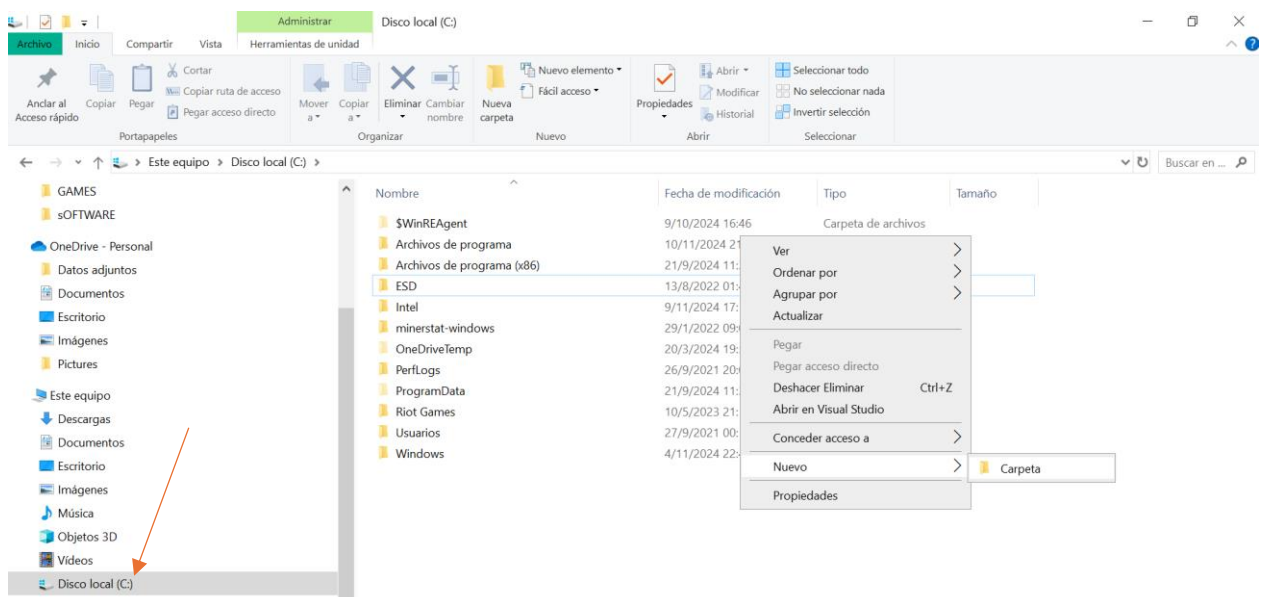
Create repository

Una vez configurado esto, tenemos nuestro repositorio creado. Yendo a *mi perfil* --> *repositorios* podemos visualizarlo.



Importar repositorio

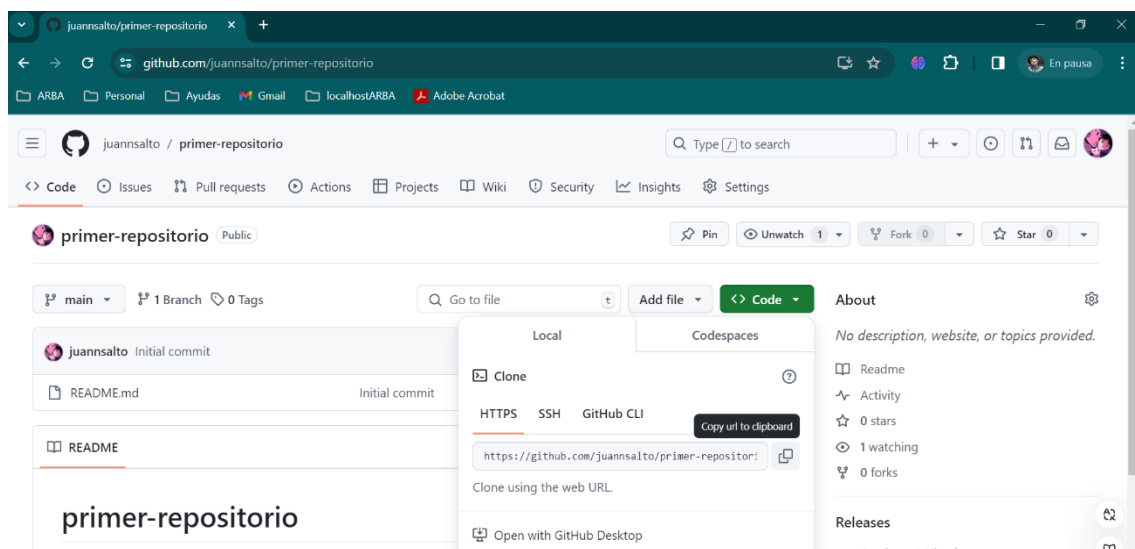
Es importante tener sincronizado el repositorio web que acabamos de crear con una instancia que nos permita modificarlo. El repositorio en su instancia actual no tiene ningún trabajo agregado, por ello el paso pertinente es traerlo a nuestra computadora y poder trabajar en nuestra maquina con el. Para ello vamos a ingresar al repositorio, bajarlo por protocolo HTTPS y bajarlo a nuestra computadora. Para ello debemos primero crear una ruta de carpeta especifica donde vamos a guardar nuestros trabajos, la ruta es `C:\Desarrollo`, para ello vamos a dirigirnos al disco C de nuestra computadora y ahí crear la carpeta desarrolla con *click derecho -> crear nueva carpeta*.



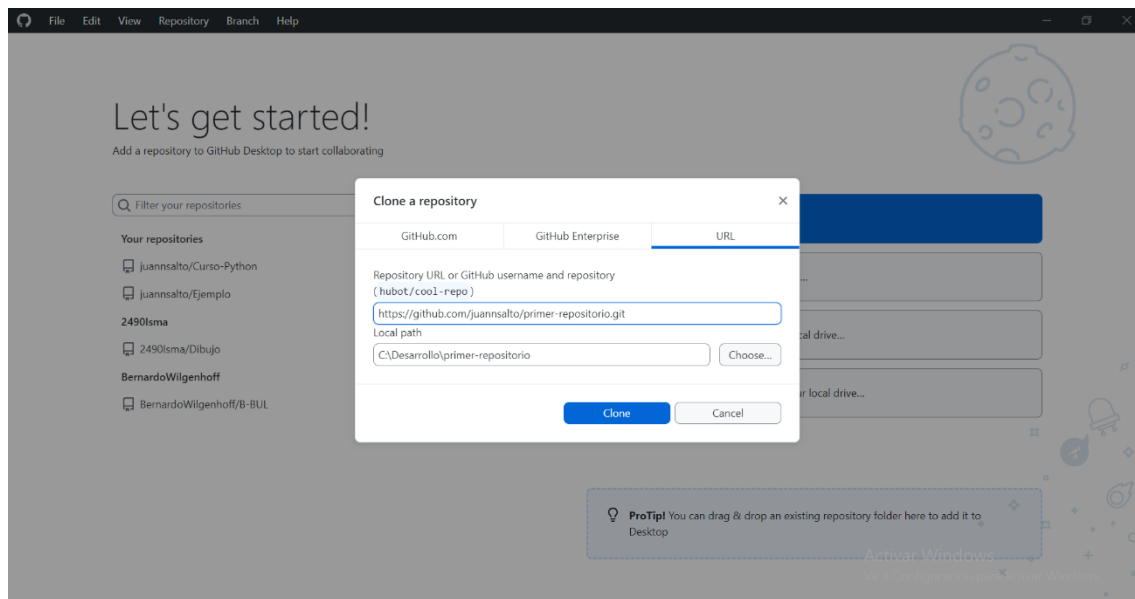
De esta manera nos queda, nos queda la carpeta con la ruta dicha anteriormente donde vamos a importar nuestro proyecto recientemente creado.

Nombre	Fecha de modificación	Tipo	Tamaño
\$WinREAgent	9/10/2024 16:46	Carpeta de archivos	
Archivos de programa	10/11/2024 21:06	Carpeta de archivos	
Archivos de programa (x86)	21/9/2024 11:36	Carpeta de archivos	
Desarrollo	11/11/2024 16:58	Carpeta de archivos	
ESD	13/8/2022 01:48	Carpeta de archivos	
Intel	9/11/2024 17:12	Carpeta de archivos	
minerstat-windows	29/1/2022 09:09	Carpeta de archivos	
OneDriveTemp	20/3/2024 19:56	Carpeta de archivos	
PerfLogs	26/9/2021 20:04	Carpeta de archivos	
ProgramData	21/9/2024 11:35	Carpeta de archivos	
Riot Games	10/5/2023 21:18	Carpeta de archivos	
Usuarios	27/9/2021 00:19	Carpeta de archivos	
Windows	4/11/2024 22:46	Carpeta de archivos	

Volvemos a nuestro navegador y en nuestro repositorio vamos a la ruta del botón verde `<>code->https` y copiamos el link

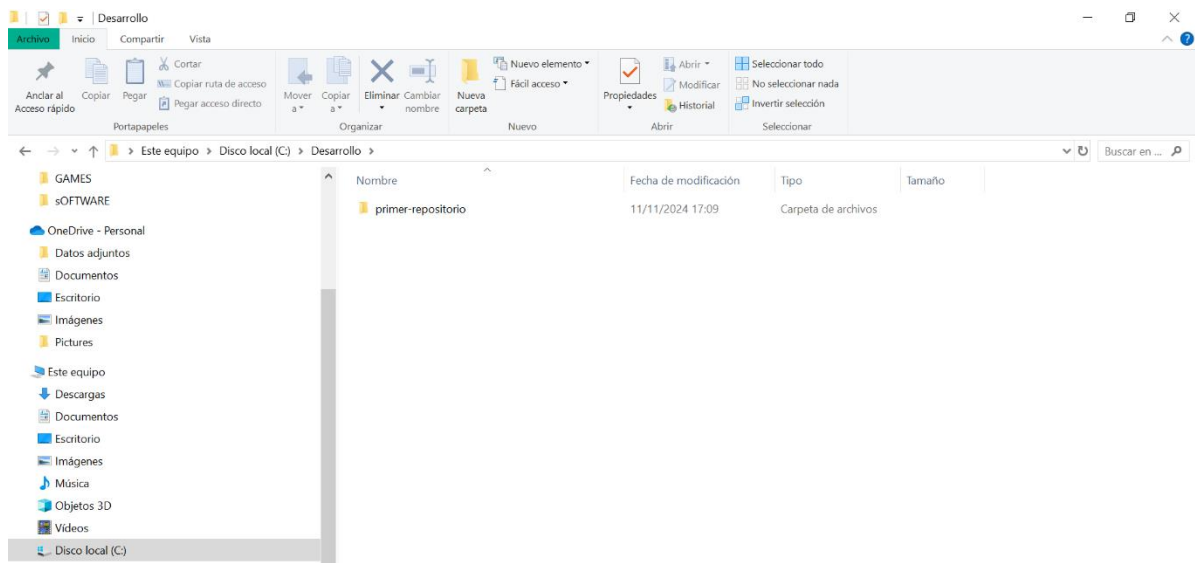


Abrimos la versión de escritorio, donde nos dirigimos a *file->clone repository->URL* (CTRL+SHIFT+O)



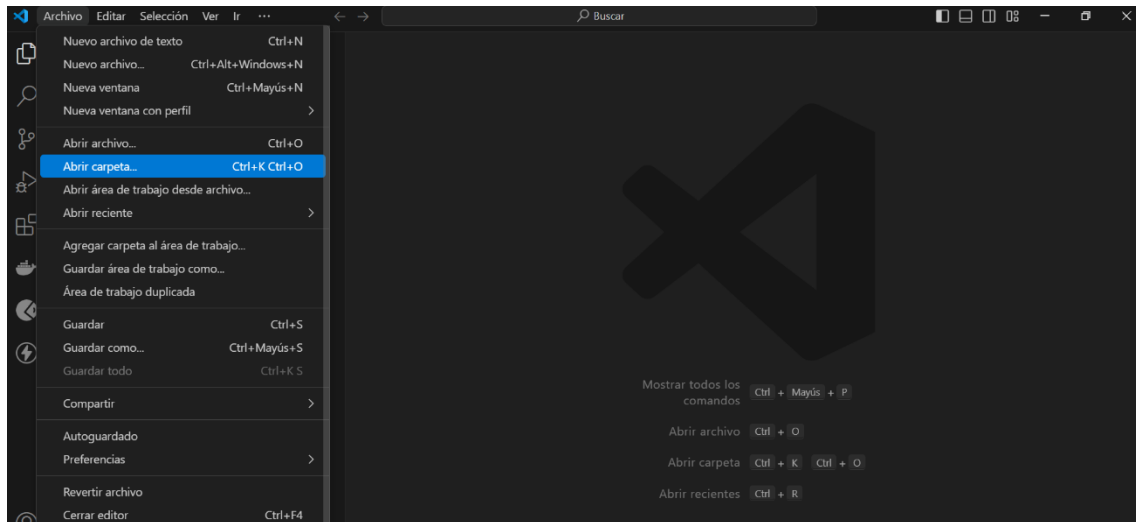
Tómese nota que en la opción *Local path* busca una ruta donde guardar el nuevo repositorio, la cual es la creada anteriormente en nuestra carpeta Desarrollo. Si aparece otra opción por defecto, modificarla para que apunte a la ruta creada. También es importante tener la URL correctamente copiada del repositorio sacada del GitHub en el navegador. Hecho esto procedemos a apretar en *Clone*

Si ahora nos dirigimos a la carpeta Desarrollo que creamos debería verse tal que así

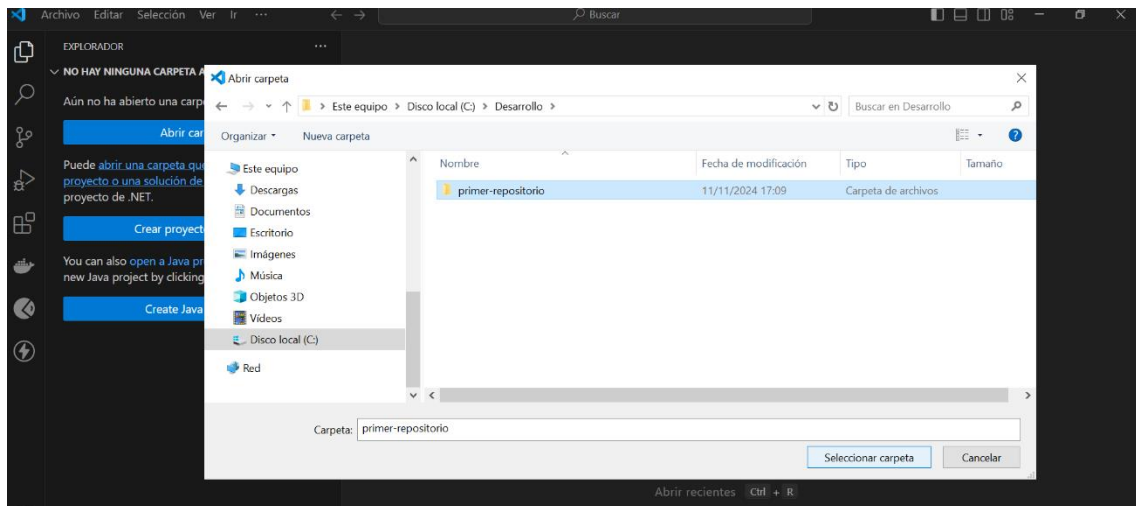


Primeros pasos VScode

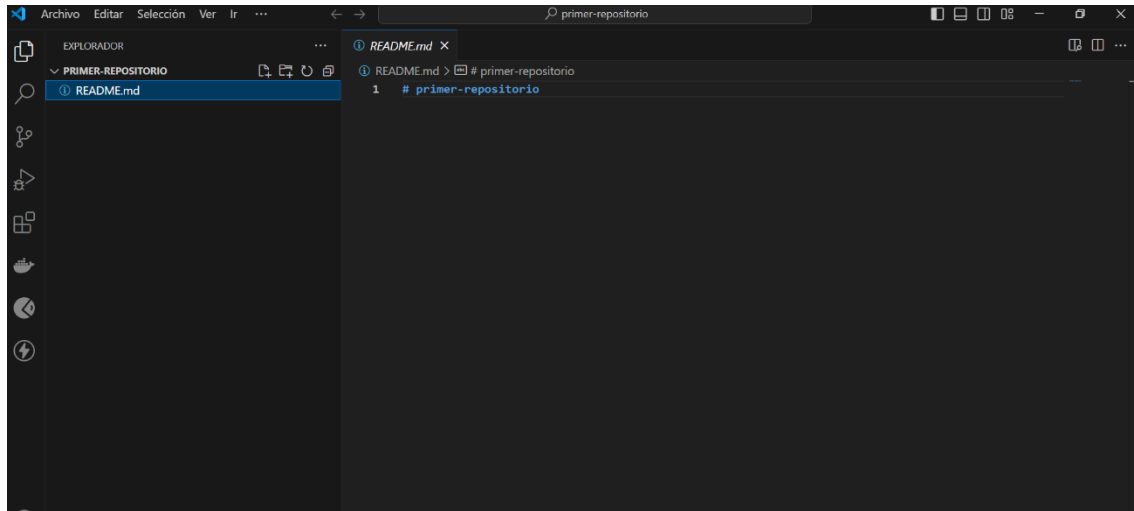
Lo primero que debemos hacer es abrir esta carpeta en Visual Studio, para ello nos dirigimos a la aplicación instalada y hacemos click en *Archivo -> Abrir Carpeta*



Una vez allí seleccionamos *C:\Desarrollo\primer-repositorio*



Una vez hecho esto ya tenemos nuestro repositorio listo para trabajar



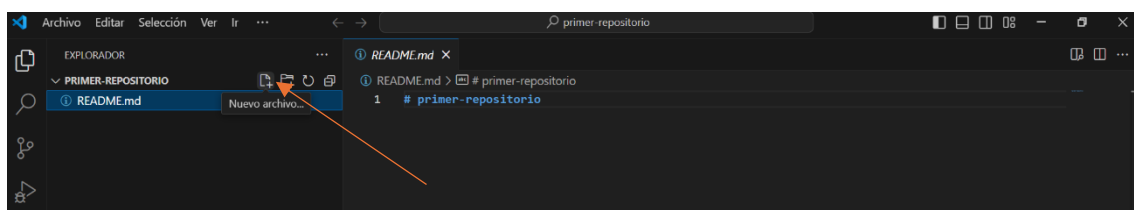
CREACION DE ARCHIVOS

Una vez que tenemos todo configurado tenemos que adaptar nuestro repositorio a las tecnologías necesarias para la confección de una página web, un modelo del mismo se encuentra subido a un repositorio de ejemplo (<https://github.com/juannsalto/Ejemplo>).

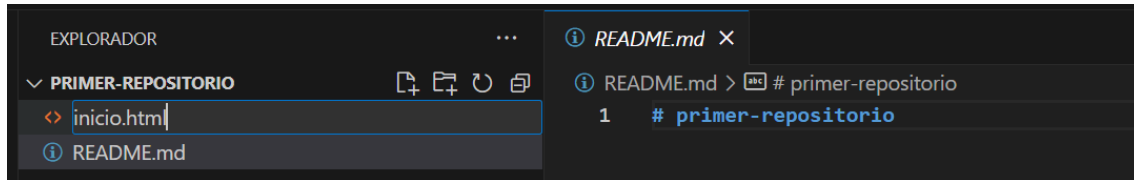
Lo pueden importar de igual forma que el ejemplo de este documento, siempre teniendo en cuenta que no importen uno dentro del otro. Es decir, que ambos queden dentro de la carpeta *Desarrollo* como pares.

Por fuera de esto se extienden las sintaxis de las guías HTML, CSS y Javascript correspondientes para las modificaciones pertinentes.

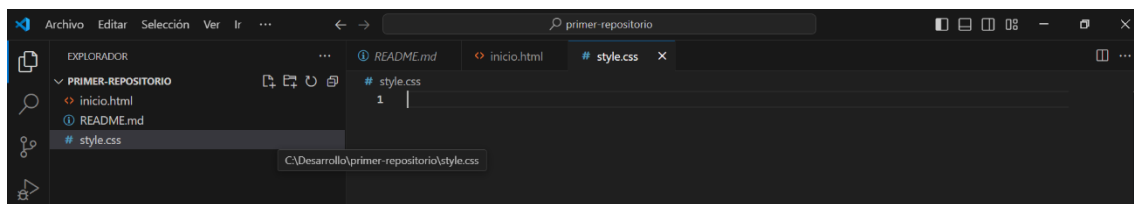
De todos modos, es fundamental poder crear estos archivos con su propio formato. Para ellos nos paramos en VS Code y creamos el HTML, para ello es tan sencillo como crear un nuevo archivo dentro de la carpeta, nombrarlo con un criterio lógico y darle la extensión *.html* (*css/js*) según corresponda.



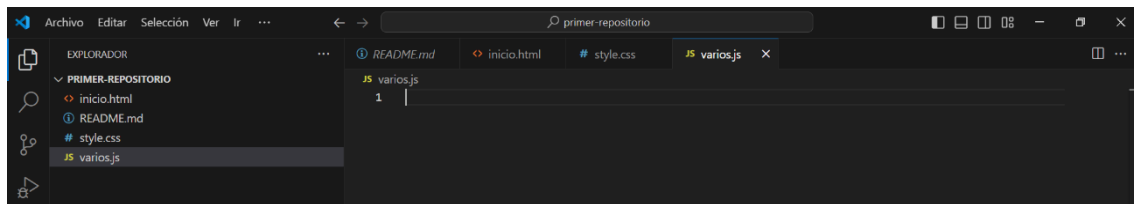
Agrego la extensión deseada. Para HTML:



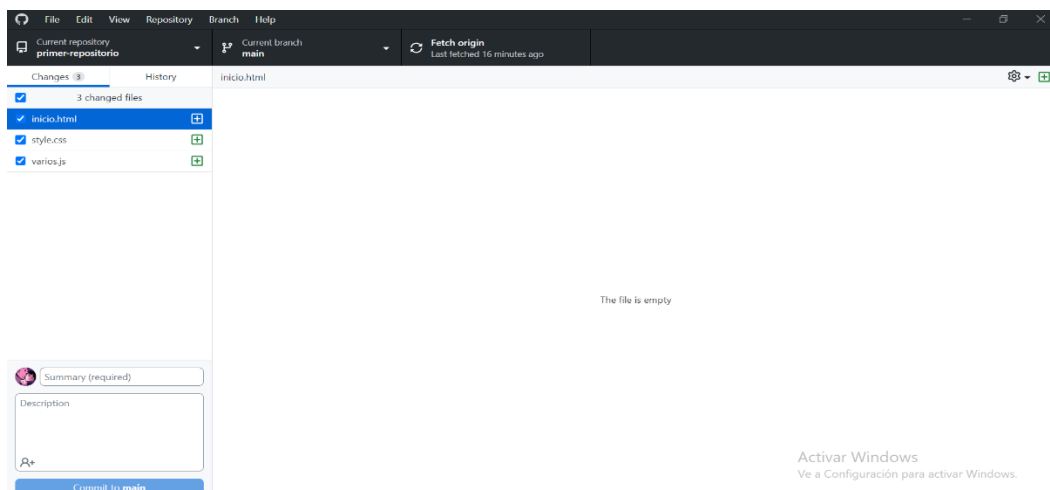
CSS:



Javascript:



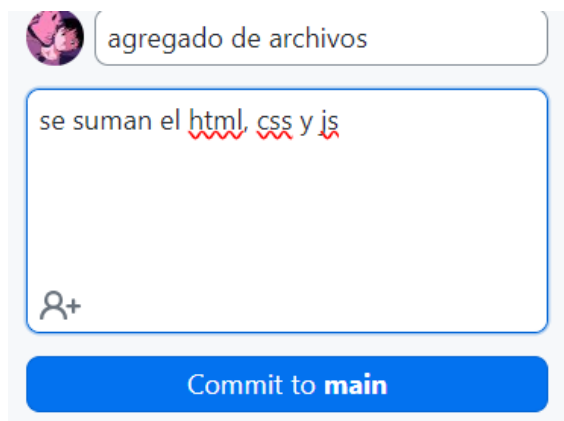
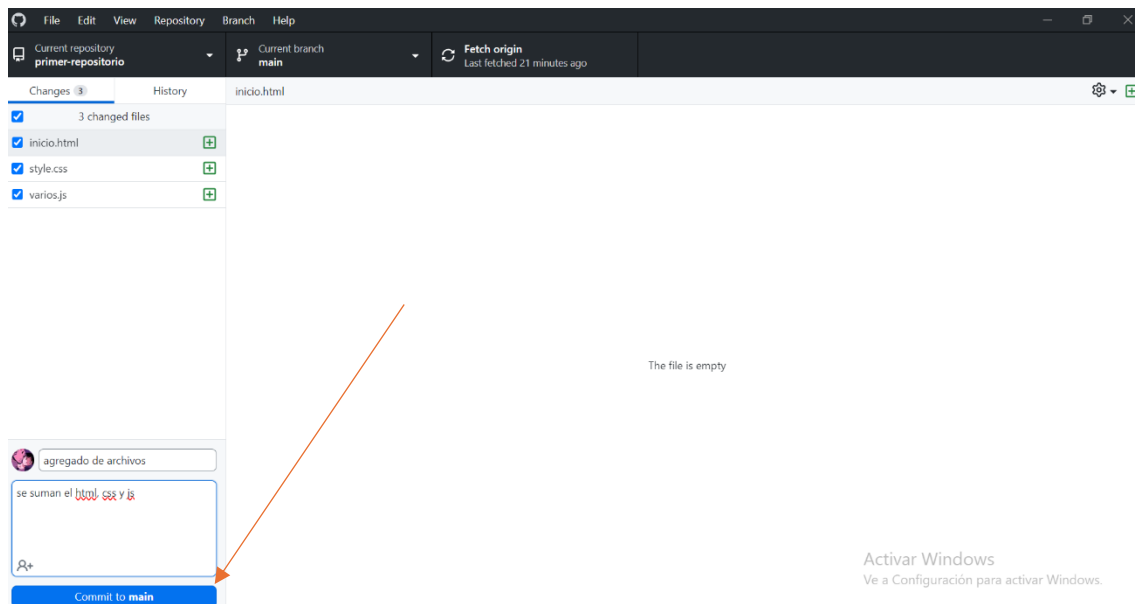
Si una vez creados estos archivos nos dirigimos a la versión Desktop (de escritorio) de GitHub, deberíamos notar diferencias en nuestro proyecto. Esto sucede porque agregamos nuevos archivos, debería verse tal que así:



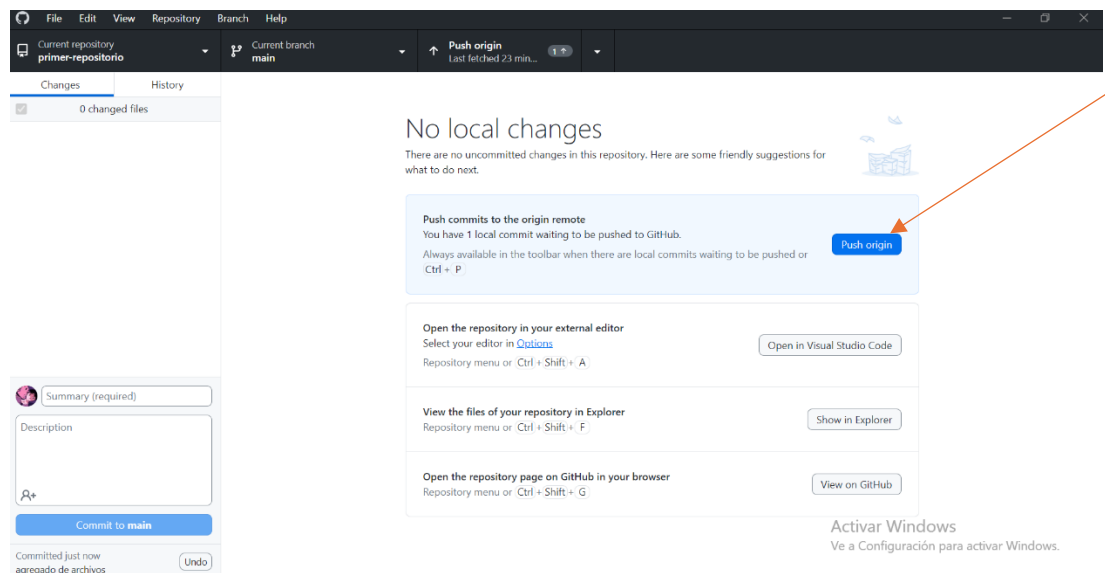
Por el momento son archivos vacíos porque no hemos agregado ningún código, cuando lo hagamos (siguiendo el repositorio de ejemplo y las guías de sintaxis) el mismo va a ser levantado y mostrado en la parte donde por el momento nos aparece *The file is empty*.

Subida a GitHub, repositorio local-remoto.

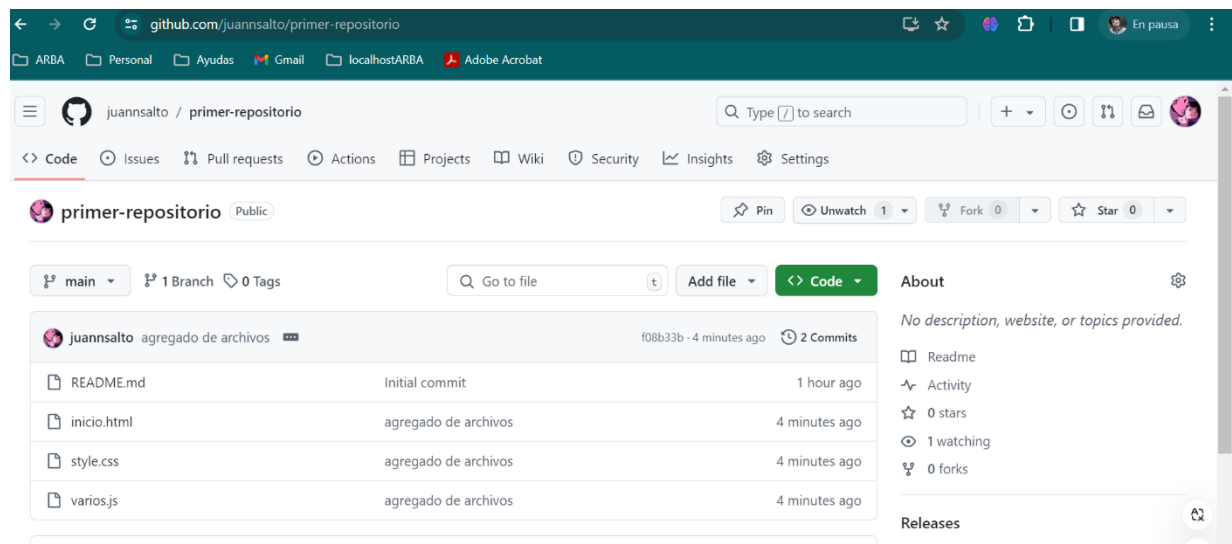
Por los pasos seguidos hasta ahora, tenemos que impactar nuestro trabajo hecho en nuestra PC (local) en el repositorio creado en la página de GitHub, para ellos se completa el cuadro en la esquina inferior izquierda y se apreta el botón de commit, se sube una etiqueta del cambio y una descripción. Por los que no quedaría así:



Al apretar el botón de Commit to main, lo que hacemos es fijar un compromiso con lo modificado en el repositorio. Esto genera que hasta ese momento y con esos cambios podemos impactar el repositorio, por lo cual hay que hacer este paso al estar seguro de que el código hecho es lo que realmente deseamos. De todos modos esto todavía no se encuentra subido al repositorio de la pagina de GitHub, para ello se debe ejecutar el comando *Push*



Una vez ejecutado podemos verificar que efectivamente se subió. Para ello nos dirigimos al navegador, refrescamos la página de GitHub y nuestros cambios tienen que verse impactados.



Efectivamente podemos constatar que los archivos que creamos en el proyecto se encuentran subidos. Recuerden que siempre que tienen un cambio en algún archivo dentro de la carpeta (C:\Desarrollo\primer-repositorio) GitHub lo va a detectar, mostrándolo en GitHub Desktop.