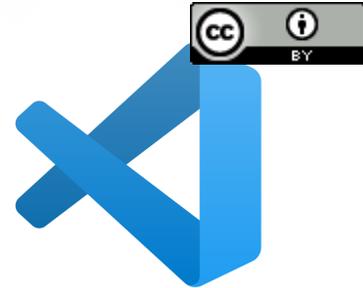




Integración: Git, GitHub y VSCode

Ing. Luis Guillermo Molero Suárez



¿Qué es Git?

Es un sistema de control de versiones, es distribuido, es decir que múltiples personas pueden trabajar en equipo, es open source y también se adapta a todo tipo de proyectos desde pequeños hasta grandes, además, se pueden fusionar archivos, guarda una línea de tiempo a lo largo de todo el proyecto. Maneja una interfaz tipo Bash. **GIT, es el software de control de versiones en el que se basa GitHub.**

Sitio de descarga: <https://git-scm.com/downloads>

Como instalar Git: <https://www.youtube.com/watch?v=ExdLS6lZaAY>

Documentación Git: <https://git-scm.com/book/es/v2>

¿Qué es GitHub?

A diferencia de Git, GitHub es un sitio web y un servicio en la nube que ayuda a los desarrolladores a almacenar y administrar su código, al igual que llevar un registro y control de cualquier cambio sobre este código. En otras palabras, es una plataforma de desarrollo colaborativo, o también llamada la red social de los desarrolladores donde se alojan los repositorios, el código se almacena de forma pública pero se puede hacer privado con una cuenta de pago.

¿Qué es Control de Versiones?

El control de versiones ayuda a los desarrolladores a llevar un registro y administrar cualquier cambio en el código del proyecto de software. A medida que crece este proyecto, la versión de control se vuelve esencial.

Con la bifurcación, un desarrollador duplica parte del código fuente (llamado repositorio). Este desarrollador, luego puede, de forma segura, hacer cambios a esa parte del código, sin afectar al resto del proyecto.



Luego, una vez que el desarrollador logre que su parte del código funcione de forma apropiada, esta persona podría fusionar este código al código fuente principal para hacerlo oficial. Todos estos cambios luego son registrados y pueden ser revertidos si es necesario.

Documentación de GitHub: <https://docs.github.com/es/github>

Documentación Git: <https://git-scm.com/book/es/v2>

Creación de una cuenta

Lo primero que necesitas es una cuenta de usuario gratuita. Simplemente visita <https://github.com>, elige un nombre de usuario que no esté ya en uso, proporciona un correo y una contraseña, y pulsa el botón verde grande “Sign up for GitHub”.

Pick a username

Your email

Create a password

Use at least one lowercase letter, one numeral, and seven characters.

Sign up for GitHub

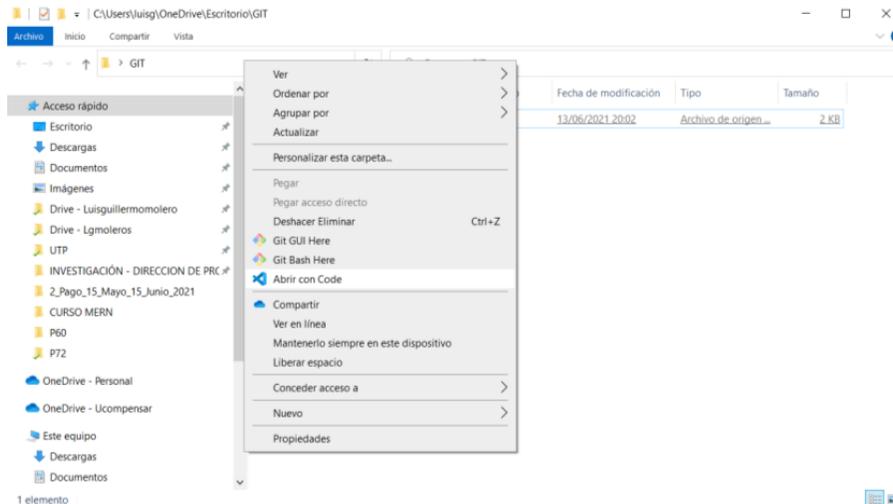
Lo siguiente que verás es la página de precios para planes mejores, pero lo puedes ignorar por el momento. GitHub te enviará un correo para verificar la dirección que les has dado. Confirmar la dirección ahora, es bastante importante (como veremos después).

Para ampliar esta información: <https://n9.cl/nqu9>

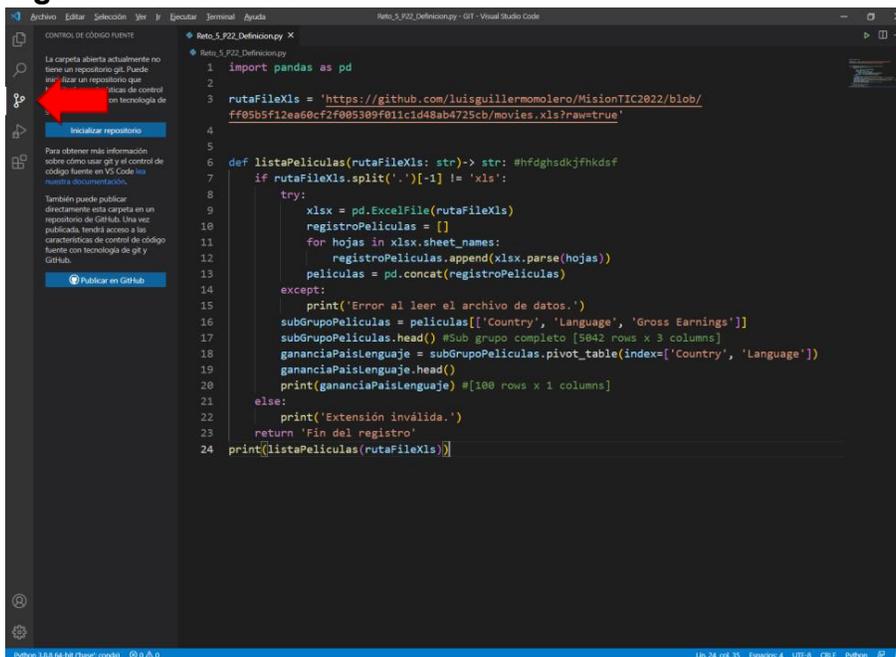


Integración con Visual Studio Code (VSC)

1. Abrimos nuestro proyecto en VSC, de la forma:



2. Desde nuestra carpeta de código hacemos clic en el botón “Control de código fuente” de VSC .



Nos aparece la siguiente interacción



```
1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/luisguillemomolero/MisionTIC2022/blob/
4 ff05b5f12ea60cf2f085309f011c1d48ab4725cb/movies.xls?raw=true'
5
6 def listaPelículas(rutaFileXls: str)-> str: #hfdghsdkjfhkdsf
7     if rutaFileXls.split('.')[-1] != 'xls':
8         try:
9             xls = pd.ExcelFile(rutaFileXls)
10            registroPelículas = []
11            for hojas in xls.sheet_names:
12                registroPelículas.append(xlsx.parse(hojas))
13            películas = pd.concat(registroPelículas)
14        except:
15            print('Error al leer el archivo de datos.')
16            subGrupoPelículas = películas[['Country', 'Language', 'Gross Earnings']]
17            subGrupoPelículas.head() #Sub grupo completo [5042 rows x 3 columns]
18            gananciaPaisLenguaje = subGrupoPelículas.pivot_table(index='Country', 'Language'])
19            gananciaPaisLenguaje.head()
20            print(gananciaPaisLenguaje) #[100 rows x 1 columns]
21        else:
22            print('Extensión inválida.')
23            return 'Fin del registro'
24    print(listaPelículas(rutaFileXls))
```

3. Abrimos la terminal de Git en VSC y escribimos `git init`

```
1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/luisguillemomolero/MisionTIC2022/blob/
4 ff05b5f12ea60cf2f085309f011c1d48ab4725cb/movies.xls?raw=true'
5
6 def listaPelículas(rutaFileXls: str)-> str: #hfdghsdkjfhkdsf
7     if rutaFileXls.split('.')[-1] != 'xls':
8         try:
9             xls = pd.ExcelFile(rutaFileXls)
10            registroPelículas = []
11            for hojas in xls.sheet_names:
12                registroPelículas.append(xlsx.parse(hojas))
13            películas = pd.concat(registroPelículas)
14        except:
15            print('Error al leer el archivo de datos.')
16            subGrupoPelículas = películas[['Country', 'Language', 'Gross Earnings']]
17            subGrupoPelículas.head() #Sub grupo completo [5042 rows x 3 columns]
18            gananciaPaisLenguaje = subGrupoPelículas.pivot_table(index='Country', 'Language'])
19            gananciaPaisLenguaje.head()
20            print(gananciaPaisLenguaje) #[100 rows x 1 columns]
21        else:
22            print('Extensión inválida.')
23            return 'Fin del registro'
24    print(listaPelículas(rutaFileXls))
```



```
1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/luisguillemomolero/MisionTIC2022/blob/ff05b5f12ea60cf2f005309f011c1d48ab4725cb/movies.xls?raw=true'
4
5
6 def listaPelículas(rutaFileXls: str)-> str: #hfdghsdkjfhkdsf
7     if rutaFileXls.split('.')[-1] != 'xls':
8         try:
9             xlsx = pd.ExcelFile(rutaFileXls)
10            registroPelículas = []
11            for hojas in xlsx.sheet_names:
12                registroPelículas.append(xlsx.parse(hojas))
13            películas = pd.concat(registroPelículas)
14        except:
15            print("Error al leer el archivo de datos.")
16            subGrupoPelículas = películas[['Country', 'Language', 'Gross Earnings']]
17            subGrupoPelículas.head() #Sub grupo completo [5042 rows x 3 columns]
18            gananciaPaisLenguaje = subGrupoPelículas.pivot_table(index='Country', 'Language'])
19            gananciaPaisLenguaje.head()
20            print(gananciaPaisLenguaje) #[100 rows x 1 columns]
21        else:
```

```
luis@DESKTOP-ACB40B: ~/OneDrive/Escritorio/GIT
$ git init
Initialized empty Git repository in C:/Users/luisg/OneDrive/Escritorio/GIT/.git/
luis@DESKTOP-ACB40B: ~/OneDrive/Escritorio/GIT (master)
$
```

Una vez ejecutado el comando, desaparece el área de interacción y ya podemos trabajar con el código fuente, ya que se inicializa un repositorio en local.

```
1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/luisguillemomolero/MisionTIC2022/blob/ff05b5f12ea60cf2f005309f011c1d48ab4725cb/movies.xls?raw=true'
4
5
6 def listaPelículas(rutaFileXls: str)-> str: #hfdghsdkjfhkdsf
7     if rutaFileXls.split('.')[-1] != 'xls':
8         try:
9             xlsx = pd.ExcelFile(rutaFileXls)
10            registroPelículas = []
11            for hojas in xlsx.sheet_names:
12                registroPelículas.append(xlsx.parse(hojas))
13            películas = pd.concat(registroPelículas)
14        except:
15            print("Error al leer el archivo de datos.")
16            subGrupoPelículas = películas[['Country', 'Language', 'Gross Earnings']]
17            subGrupoPelículas.head() #Sub grupo completo [5042 rows x 3 columns]
18            gananciaPaisLenguaje = subGrupoPelículas.pivot_table(index='Country', 'Language'])
19            gananciaPaisLenguaje.head()
20            print(gananciaPaisLenguaje) #[100 rows x 1 columns]
21        else:
```

```
luis@DESKTOP-ACB40B: ~/OneDrive/Escritorio/GIT
$ git init
Initialized empty Git repository in C:/Users/luisg/OneDrive/Escritorio/GIT/.git/
luis@DESKTOP-ACB40B: ~/OneDrive/Escritorio/GIT (master)
$
```

Podemos confirmar que ya está inicializado el repositorio de la siguiente manera:

Nos vamos a la carpeta donde se ubica el proyecto y observaremos una carpeta oculta `.git` (Cada vez que queramos ocultar una carpeta colocamos el `."` antes del nombre).



Nombre	Estado	Fecha de modificación	Tipo	Tamaño
git	✓	14/06/2021 11:56	Carpeta de archivos	
Reto 5 P22 Definicion.py	✓	13/06/2021 20:02	Archivo de origen ...	2 KB

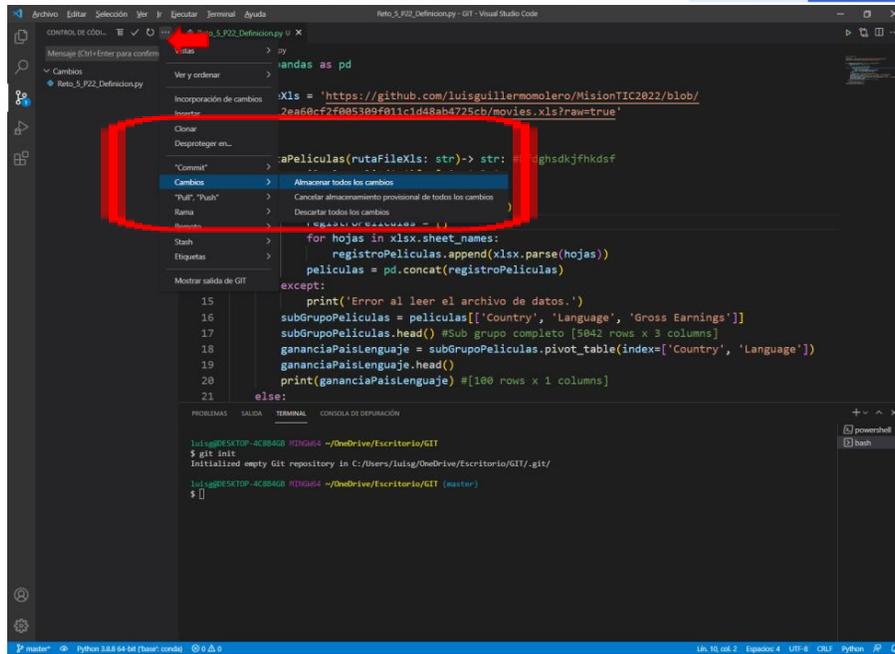
Luego de verificar que tenemos control de cambios en el proyecto, revisemos los siguiente.

En nuestro proyecto, el archivo abierto aparece con el identificador “U” que significa sin seguimiento

```
1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/luisguillermomolero/MisionTIC2022/blob/
  ffo5b5f12ea6c2f085309f011c1d48ab4725cb/movies.xls?raw=true'
4
5
6 def listaPelículas(rutaFileXls: str)-> str: #hfdghsdkjfhkdsf
7     if rutaFileXls.split('.')[-1] != 'xls':
8         try:
9             xls = pd.ExcelFile(rutaFileXls)
10            registroPelículas = []
11            for hojas in xls.sheet_names():
12                registroPelículas.append(xlsx.parse(hojas))
13            películas = pd.concat(registroPelículas)
14        except:
15            print('Error al leer el archivo de datos.')
16            subGrupoPelículas = películas[['Country', 'Language', 'Gross Earnings']]
17            subGrupoPelículas.head() #Sub grupo completo [5842 rows x 3 columns]
18            gananciaPaisLenguaje = subGrupoPelículas.pivot_table(index='Country', 'Language')
19            print(gananciaPaisLenguaje.head())
20        else:
21            print(gananciaPaisLenguaje) #100 rows x 1 columns
```

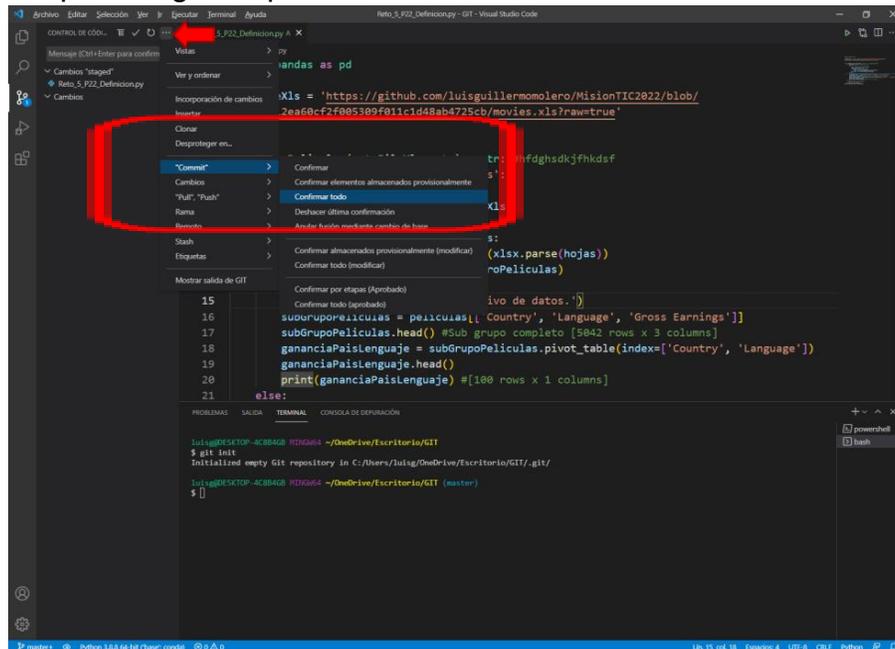
```
Julisa@DESKTOP-ACB4GE: ~/OneDrive/Escritorio/GIT
$ git init
Initialized empty Git repository in C:/Users/Julisa/OneDrive/Escritorio/GIT/.git/
Julisa@DESKTOP-ACB4GE: ~/OneDrive/Escritorio/GIT (master)
```

4. Entonces, hacemos clic en “almacenar todos los cambios”



Una vez hecho esto, nos aparece el identificador “A” (Add).

5. Como paso seguido, procedemos a “confirmar todo”



6. Nos abrirá la “paleta de comandos” y le colocamos un mensaje para el primer commit (Este mensaje debe hacerse tantas veces hagamos una actualización del proyecto para llevar el control de cambios), que para el caso de este ejemplo es: “**Commit Inicial**”.



```
1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/luisguillemomolero/MisionTIC2022/blob/
4 ff95b5f12ea6cf2f985309f011c1d48ab4725cb/movies.xls?raw=true'
5
6 def listaPelículas(rutaFileXls: str)-> str: #hfdghsdkjfhkdsf
7     if rutaFileXls.split('.')[-1] != 'xls':
8         try:
9             xlsx = pd.ExcelFile(rutaFileXls)
10            registroPelículas = []
11            for hojas in xlsx.sheet_names:
12                registroPelículas.append(xlsx.parse(hojas))
13            películas = pd.concat(registroPelículas)
14        except:
15            print("Error al leer el archivo de datos.")
16        subGrupoPelículas = películas[['Country', 'Language', 'Gross Earnings']]
17        subGrupoPelículas.head() #Sub grupo completo [5842 rows x 3 columns]
18        gananciaPaisLenguaje = subGrupoPelículas.pivot_table(index='Country', 'Language'])
19        gananciaPaisLenguaje.head()
20        print(gananciaPaisLenguaje) #100 rows x 1 columns]
21    else:
```

```
luis@RESKTOP-ACBAG8 MISION24 ~/OneDrive/Escritorio/GIT
$ git init
Initialized empty Git repository in C:/Users/luisg/OneDrive/Escritorio/GIT/.git/
luis@RESKTOP-ACBAG8 MISION24 ~/OneDrive/Escritorio/GIT (master)
$
```

IMPORTANTE: Si al momento de teclear <ENTER> para confirmar el mensaje del “commit” hecho genera un error de autenticación, introduzca en la terminal GIT las siguientes líneas de comandos:

```
git config --global user.email "tu_correo_de_GITHUB"
git config --global user.name "tu_usuario_de_GITHUB"
```

Recuerda que esta configuración de GIT, GitHub en VSC debe hacerse luego de haber instalado GIT en tu Pc y creado tu cuenta en GITHUB.

7. Paso seguido, veremos que no hay cambios que guardar en nuestro repositorio



```
1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/luisguillemomolero/MisionTIC2022/blob/
ff05b5f12ea68cf2f085309f011c1d48ab4725cb/movies.xls?raw=true'
4
5
6 def listaPelículas(rutaFileXls: str)-> str: #hfgdskjfhkdsf
7     if rutaFileXls.split('.')[-1] != 'xls':
8         try:
9             xlsx = pd.ExcelFile(rutaFileXls)
10            registroPelículas = []
11            for hojas in xlsx.sheet_names:
12                registroPelículas.append(xlsx.parse(hojas))
13            películas = pd.concat(registroPelículas)
14            except:
15                print('Error al leer el archivo de datos.')
16            subGrupoPelículas = películas[['Country', 'Language', 'Gross Earnings']]
17            subGrupoPelículas.head() #Sub grupo completo [5842 rows x 3 columns]
18            gananciaPaísLenguaje = subGrupoPelículas.pivot_table(index='Country', 'Language'])
19            gananciaPaísLenguaje.head()
20            print(gananciaPaísLenguaje) #[100 rows x 1 columns]
21        else:
```

8. Una vez que hayamos modificado nuestro archivo del proyecto, aparecerá el identificador “M” en la ventana de “Control de cambios”

```
1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/luisguillemomolero/MisionTIC2022/blob/
ff05b5f12ea68cf2f085309f011c1d48ab4725cb/movies.xls?raw=true'
4
5
6 def listaPelículas(rutaFileXls: str)-> str:
7     if rutaFileXls.split('.')[-1] != 'xls':
8         try:
9             xlsx = pd.ExcelFile(rutaFileXls)
10            registroPelículas = []
11            for hojas in xlsx.sheet_names:
12                registroPelículas.append(xlsx.parse(hojas))
13            películas = pd.concat(registroPelículas)
14            except:
15                print('Error al leer el archivo de datos.')
16            subGrupoPelículas = películas[['Country', 'Language', 'Gross Earnings']]
17            subGrupoPelículas.head() #Sub grupo completo [5842 rows x 3 columns]
18            gananciaPaísLenguaje = subGrupoPelículas.pivot_table(index='Country', 'Language'])
19            gananciaPaísLenguaje.head()
20            print(gananciaPaísLenguaje) #[100 rows x 1 columns]
21        else:
```

9. En ese sentido, al hacer clic sobre el archivo modificado, aparecerá otra pantalla donde se observan los cambios.



```
1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/
luisguillermomolero/MisionTIC2022/blob/
ff05b5f12ea60cf2f005309f011c1d48ab4725cb/
movies.xls?raw=true'
4
5
6 def listaPelículas(rutaFileXls: str)-> str:
  ~ #hfdghsdkjfhkdsf
  7 if rutaFileXls.split('.')[-1] != 'xls':
  8     try:
  9         xlsx = pd.ExcelFile
             (rutaFileXls)
             registroPelículas =
             []
             for hojas in xlsx.
             sheet_names:
             registroPelículas.append(xlsx.
             parse(hojas))
             películas = pd.concat
             (registroPelículas)
             except:
             print('Error al leer el archivo
             de datos.')
             subGrupoPelículas = películas[
             ['Country', 'Language', 'Gross
             Earnings']]
             16
             17
```

10. Una vez observado estos cambios y estar conforme con ellos, nos vamos de nuevo a “almacenar todos los cambios” y luego procedemos a “confirmar todo “de nuevo (pasos anteriores). Para este ejemplo el mensaje será “Eliminación de comentarios”

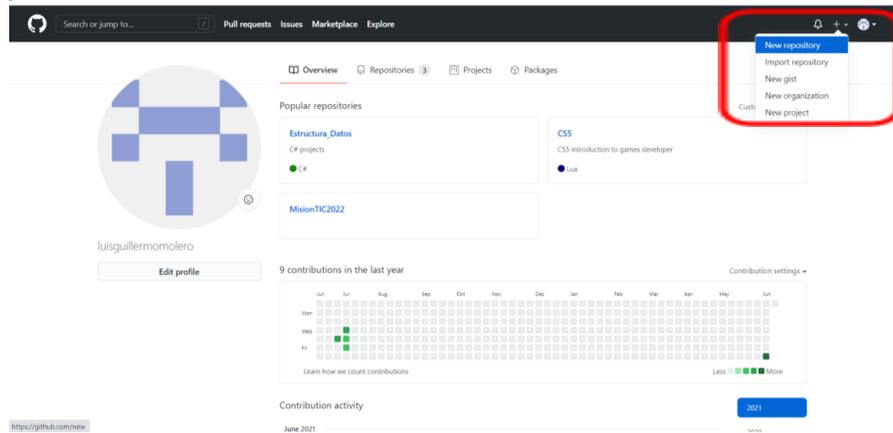
```
1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/
luisguillermomolero/MisionTIC2022/blob/
ff05b5f12ea60cf2f005309f011c1d48ab4725cb/
movies.xls?raw=true'
4
5
6 def listaPelículas(rutaFileXls: str)-> str:
  7 if rutaFileXls.split('.')[-1] != 'xls':
  8     try:
  9         xlsx = pd.ExcelFile
             (rutaFileXls)
             registroPelículas =
             []
             for hojas in xlsx.
             sheet_names:
             registroPelículas.append(xlsx.
             parse(hojas))
             películas = pd.concat
             (registroPelículas)
             except:
             print('Error al leer el archivo
             de datos.')
             subGrupoPelículas = películas[
             ['Country', 'Language', 'Gross
             Earnings']]
             subGrupoPelículas.head() #Sub grupo
             17
```

Finalmente, confirmamos que no se vea cambios en nuestro “Control de cambios”.

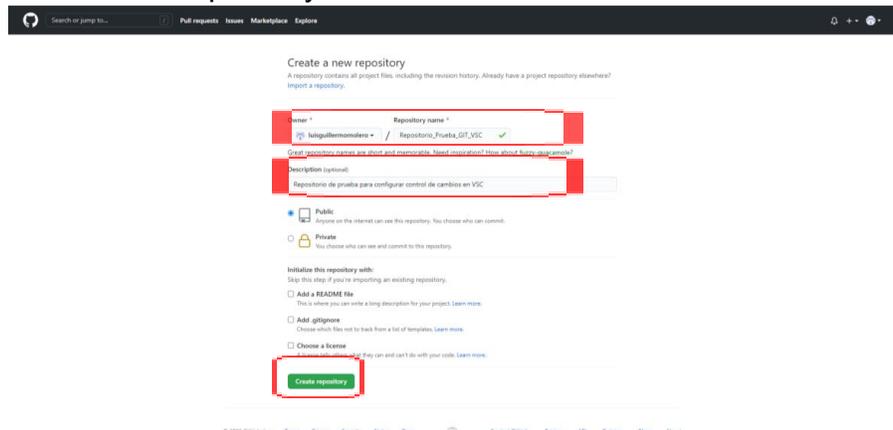


Recuerda que todos los cambios se están haciendo de manera local, ahora vamos a configurar nuestro GITHUB para poder hospedar estos cambios en la nube.

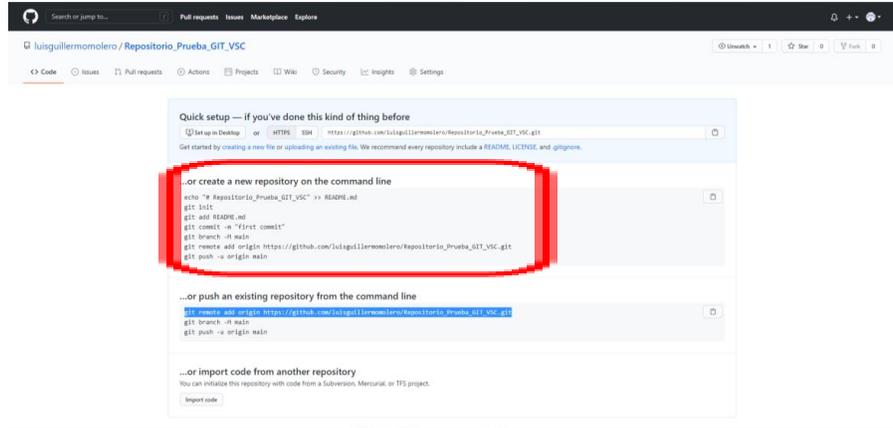
11. Nos vamos a nuestra cuenta de GITHUB en la web y creamos un “Nuevo Repositorio”



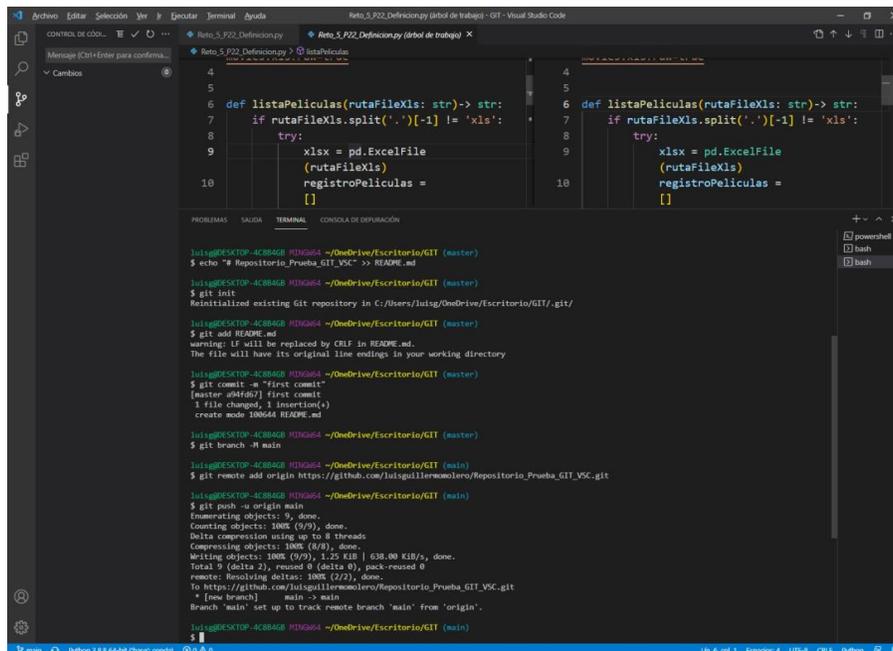
12. Le colocamos un nombre, una descripción a nuestro repositorio y hacemos clic en “Create repository”



13. Una vez creado el repositorio, copiamos las siguientes líneas de código que se encuentra en GitHub, en nuestra terminal de GIT de VSC.



```
echo "# Repositorio_Prueba_GIT_VSC" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin
https://github.com/luisguillermomolero/Repositorio_Prueba_GIT_VSC.git
git push -u origin main
```



14. Luego de que eventualmente se realice algún cambio en nuestro código nos vamos a “Insertar en”



```
def listaPelículas(rutaFileXls: str) -> str:
    if rutaFileXls.split('.')[-1] != 'xls':
        return 'Extensión inválida.'
    try:
        xls = pd.ExcelFile(rutaFileXls)
        registroPelículas = []
        for hojas in xls.sheet_names():
            registroPelículas.append(xlsx.parse(hojas))
        películas = pd.concat(registroPelículas)
        subGrupoPelículas = películas[['Country', 'Language', 'Gross Earnings']]
        subGrupoPelículas.head() #Sub grupo completo [5042 rows x 3 columns]
        gananciaPaísLenguaje = subGrupoPelículas.pivot_table(index='Country', 'Language')
        gananciaPaísLenguaje.head()
        print(gananciaPaísLenguaje) #100 rows x 1 columns
    except:
        print('Error al leer el archivo de datos.')
    return 'Fin del registro'

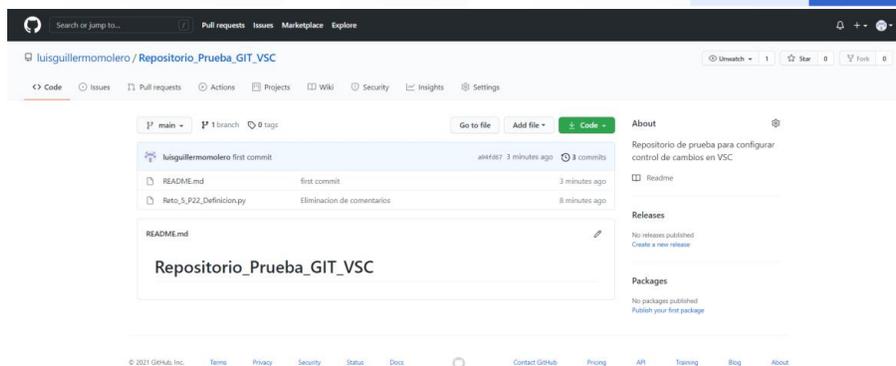
print(listaPelículas(rutaFileXls))
```

15. Y en la “paleta de comandos” nos listara los repositorios donde podremos actualizar nuestro proyecto.

```
def listaPelículas(rutaFileXls: str) -> str:
    if rutaFileXls.split('.')[-1] != 'xls':
        return 'Extensión inválida.'
    try:
        xls = pd.ExcelFile(rutaFileXls)
        registroPelículas = []
        for hojas in xls.sheet_names():
            registroPelículas.append(xlsx.parse(hojas))
        películas = pd.concat(registroPelículas)
        subGrupoPelículas = películas[['Country', 'Language', 'Gross Earnings']]
        subGrupoPelículas.head() #Sub grupo completo [5042 rows x 3 columns]
        gananciaPaísLenguaje = subGrupoPelículas.pivot_table(index='Country', 'Language')
        gananciaPaísLenguaje.head()
        print(gananciaPaísLenguaje) #100 rows x 1 columns
    except:
        print('Error al leer el archivo de datos.')
    return 'Fin del registro'

print(listaPelículas(rutaFileXls))
```

16. Seleccionamos nuestro repositorio y listo, nos aparecerá en nuestro repositorio GitHub



IMPORTANTE: Los pasos de “Almacenar todos los cambios”, “Confirmar todo” e “Insertar en” deben llevarse a cabo luego de una modificación en el código.