

Integración: Git, GitHub y VSC

Ing. Luis Guillermo Molero Suárez



Git y GitHub

¿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>

¿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.

La interfaz de GitHub es bastante fácil de usar para el desarrollador novato que quiera aprovechar las ventajas del Git. Sin GitHub, usar un Git generalmente requiere de un poco más de conocimientos de tecnología y uso de una línea de comando (Bash).

Sitio de descarga: <https://desktop.github.com/>

Como instalar Github: <https://www.youtube.com/watch?v=tn6tloweTUs>

Hechos

QUE

CONECTAN 

#MisiónTIC2022

¿Qué Es una Versión de Control?

Una Versión de Control 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.

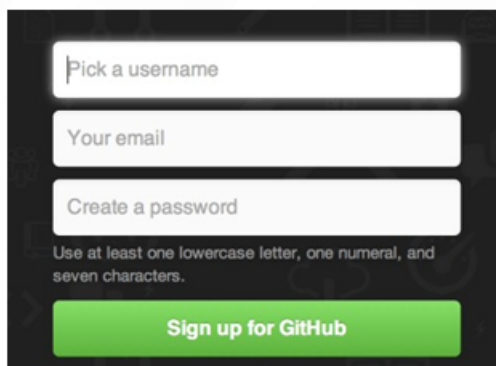
Luego, una vez que el desarrollador logre que su parte del código funcione de forma apropiada, esta persona podría fusionar este código al código fuente principal para hacerlo oficial. Todos estos cambios luego son registrados y pueden ser revertidos si es necesario.

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

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

Creación de una cuenta en GitHub

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”.



Formulario de registro de GitHub con los siguientes campos:

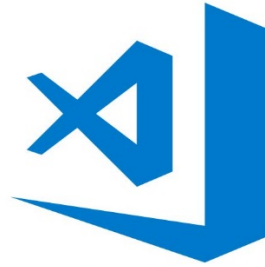
- Pick a username
- Your email
- Create a password

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

Botón: 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).

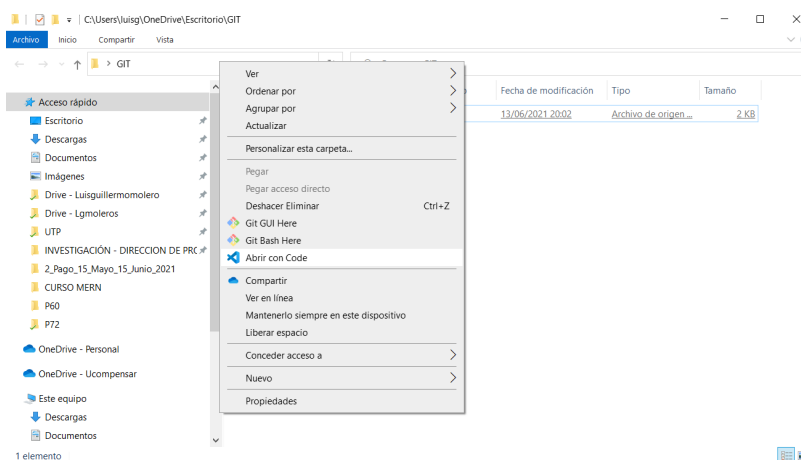
Para ampliar esta información: <https://n9.cl/nqu9>



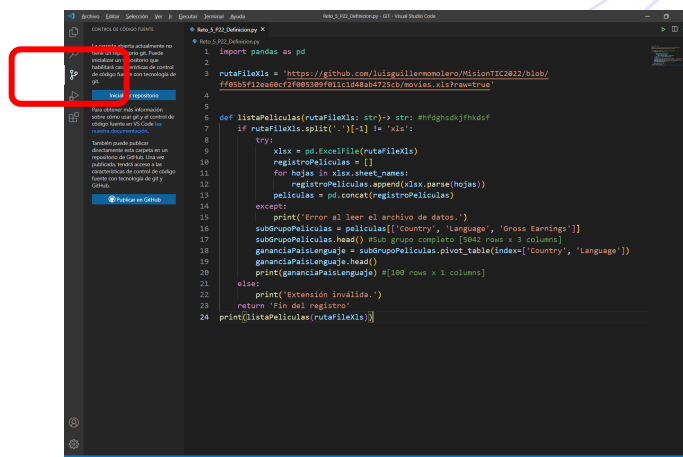
Visual Studio Code

Integración de GIT, GitHub y Visual Studio Code

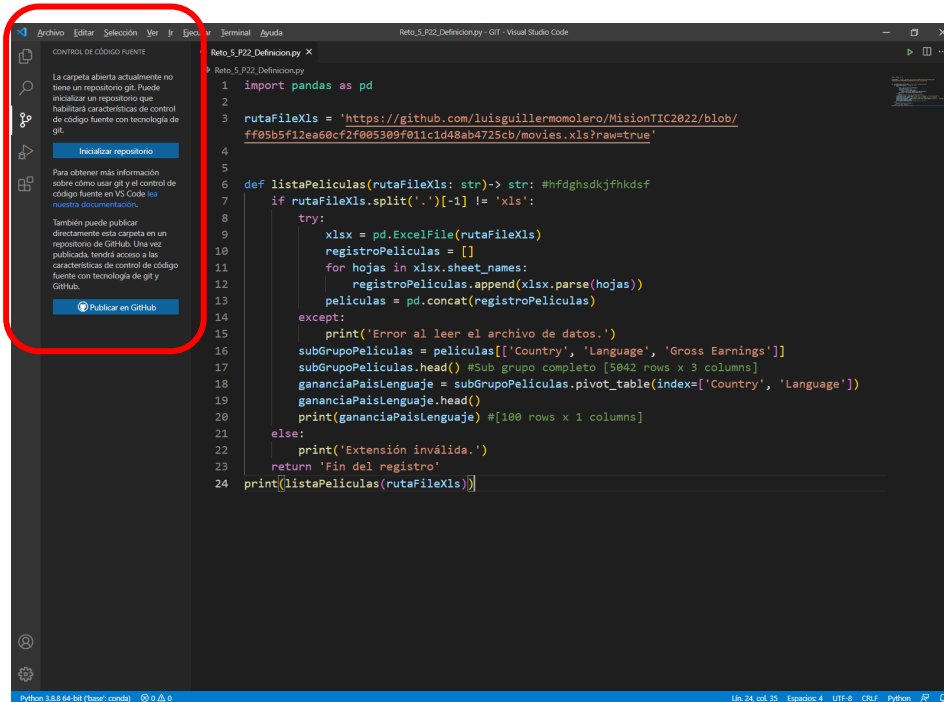
Paso 1: Abrimos nuestro proyecto en VSC, de la forma:



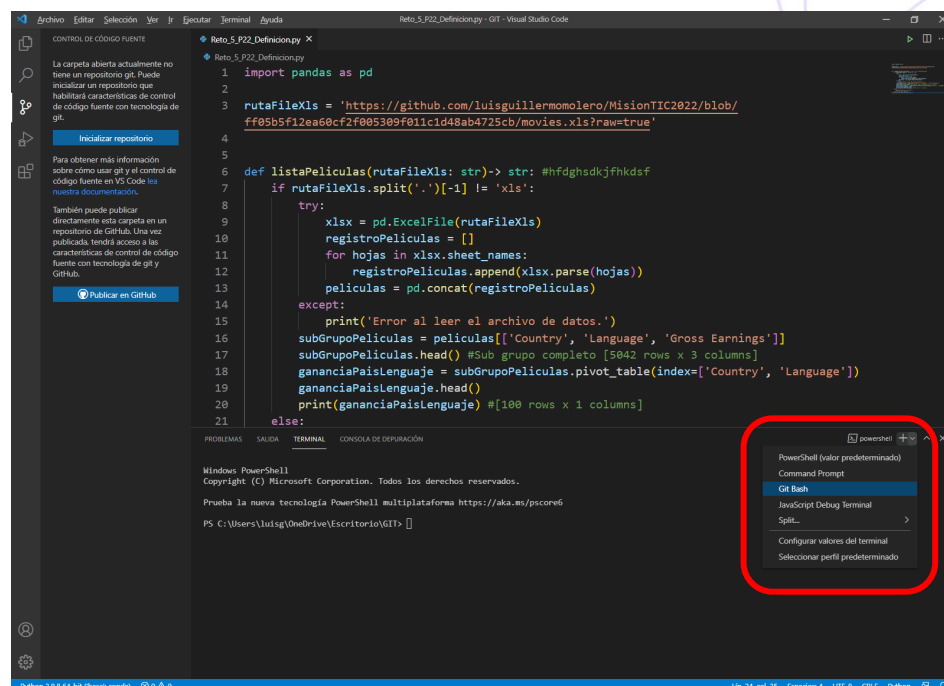
2.- Desde nuestra carpeta de código hacemos clic en el botón “Control de código fuente” de VSC .

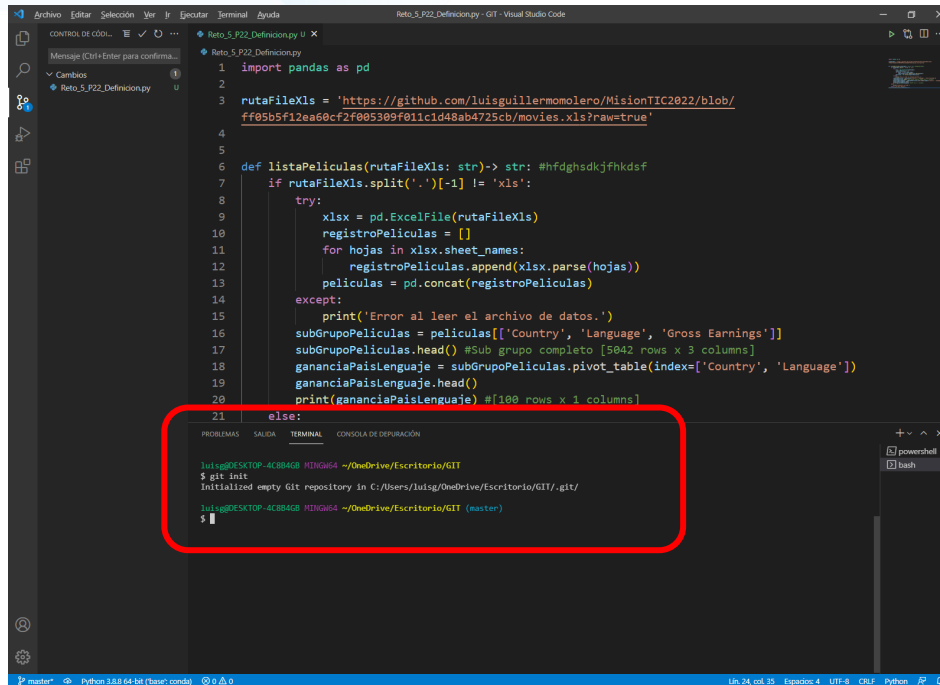


Nos aparece la siguiente interacción



3.- Abrimos la terminal de GiT en VSC y escribimos git init



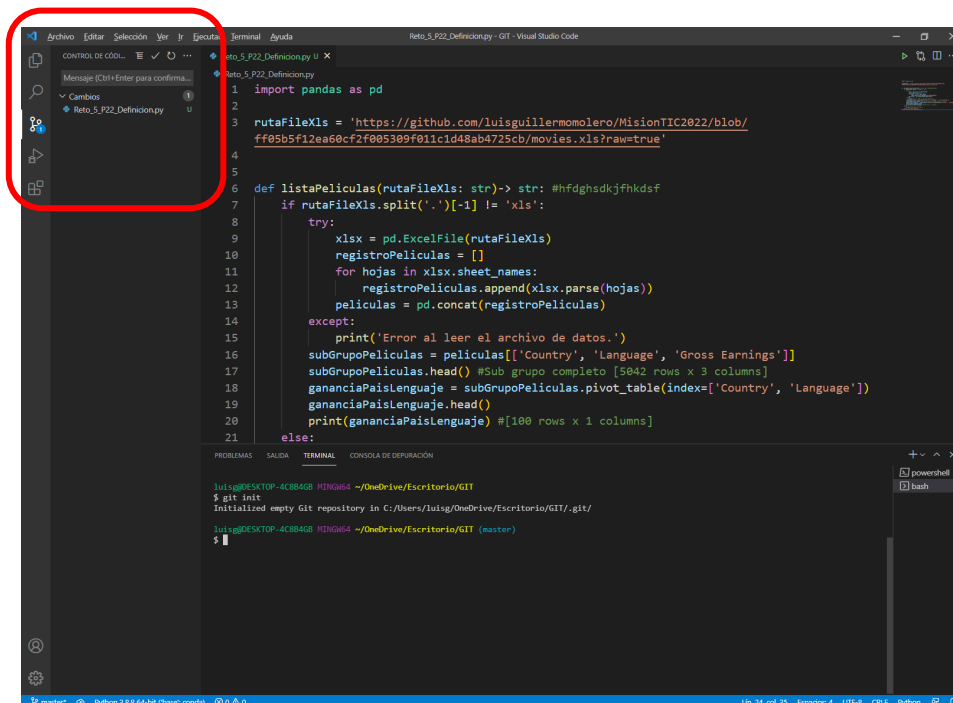


```

1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/luisguillermomolero/MisionTIC2022/blob/ff05b5f12ea60cf2f085309f011c1d48ab4725cb/movies.xls?raw=true'
4
5
6 def listaPelículas(rutaFileXls: str)-> str: #hfdghsdskjfhkdsf
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:
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

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

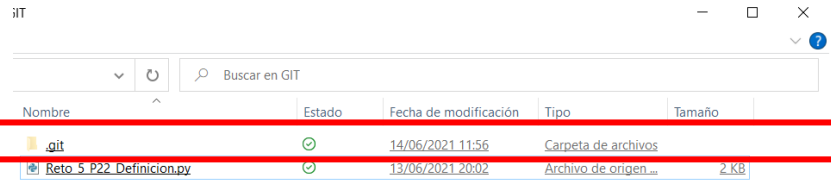


```

1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/luisguillermomolero/MisionTIC2022/blob/ff05b5f12ea60cf2f085309f011c1d48ab4725cb/movies.xls?raw=true'
4
5
6 def listaPelículas(rutaFileXls: str)-> str: #hfdghsdskjfhkdsf
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:
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

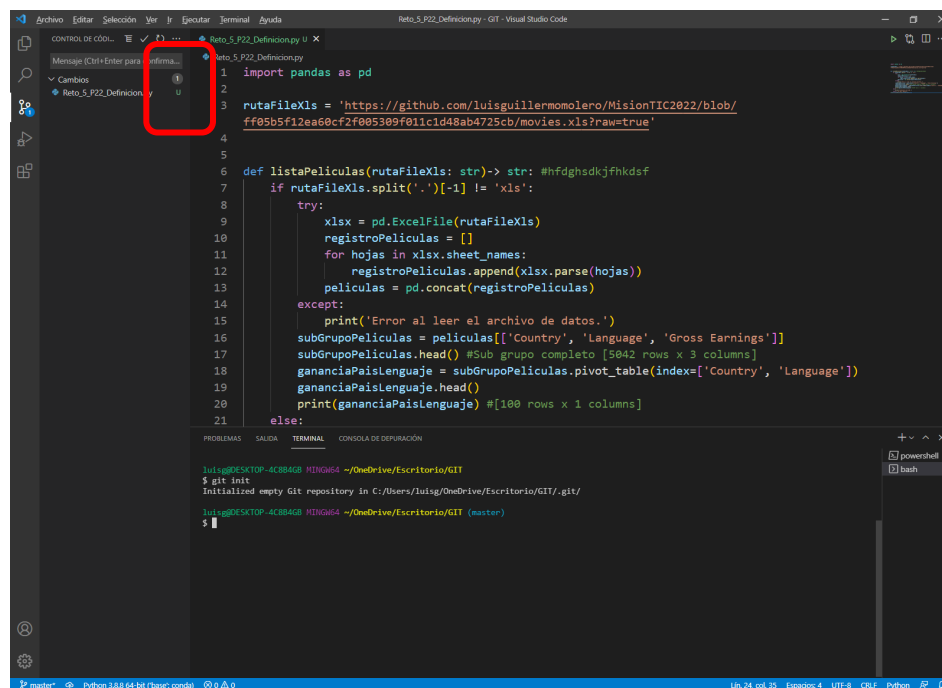

Podemos confirmar que ya está inicializado el repositorio de la siguiente manera:
Nos vamos a la carpeta donde se ubica el proyecto y observaremos una carpeta oculta .git
(Cada vez que queramos ocultar una carpeta colocamos el “.” antes del nombre).



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

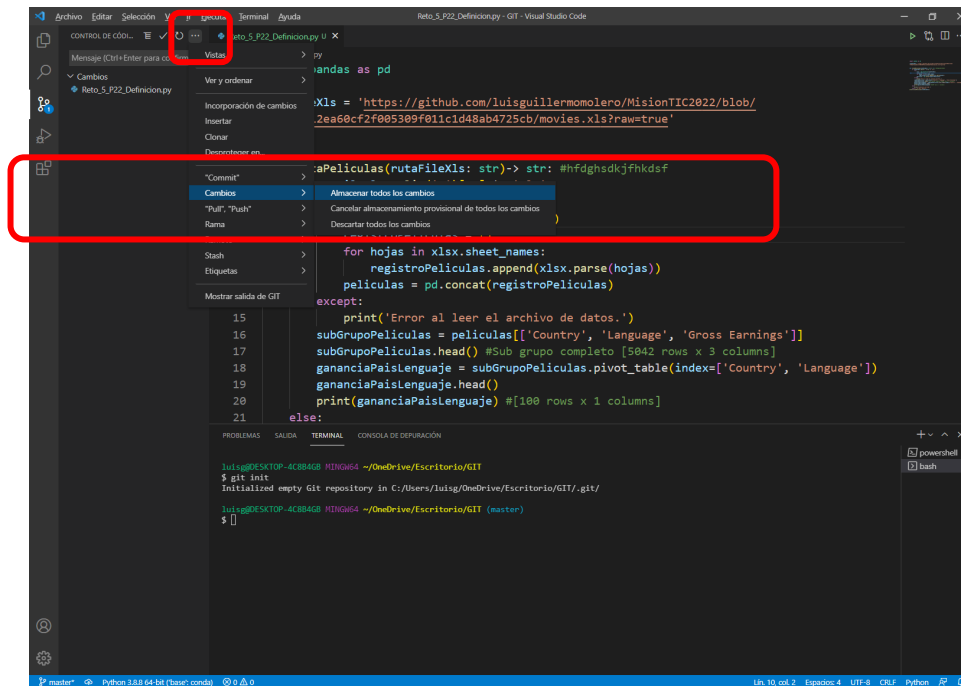


```

1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/luiguiulermomoleno/MisionTIC2022/blob/ff05b5f12ea6cf2f065309f011c1d48ab4725cb/movies.xls?raw=true'
4
5
6 def listaPelículas(rutaFileXls: str) -> str: #hfdghsdjkfhdksf
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            gananciaPaísLenguaje = subGrupoPelículas.pivot_table(index=['Country', 'Language'])
19            gananciaPaísLenguaje.head()
20            print(gananciaPaísLenguaje) #100 rows x 1 columns
21    else:

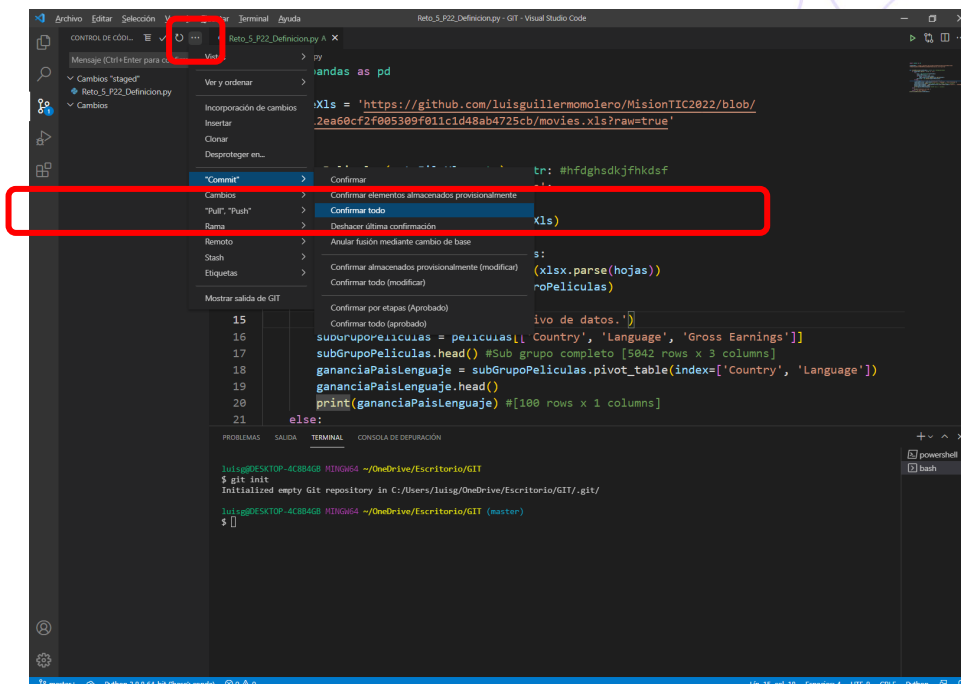
```

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

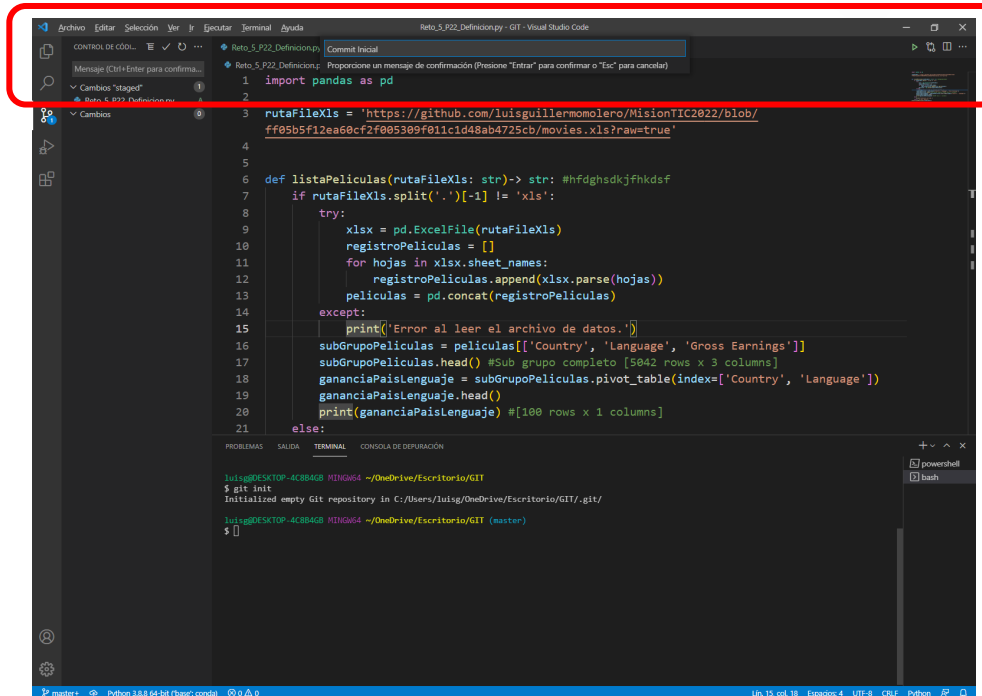


Una vez hecho esto, nos aparece el identificador “A” (Add).

4.- Como paso seguido, procedemos a “confirmar todo”



5.- 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/ff05b5f12ea6cf2f005309f011c1d48ab4725cb/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            gananciaPaísLenguaje = subGrupoPelículas.pivot_table(index=['Country', 'Language'])
19            gananciaPaísLenguaje.head()
20            print(gananciaPaísLenguaje) #[100 rows x 1 columns]
21        else:
```

```
luis@DESKTOP-AC8BAG8 MINGW64 ~/OneDrive/Escritorio/GIT
$ git init
Initialized empty Git repository in C:/Users/luisg/OneDrive/Escritorio/GIT/.git/
luis@DESKTOP-AC8BAG8 MINGW64 ~/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:

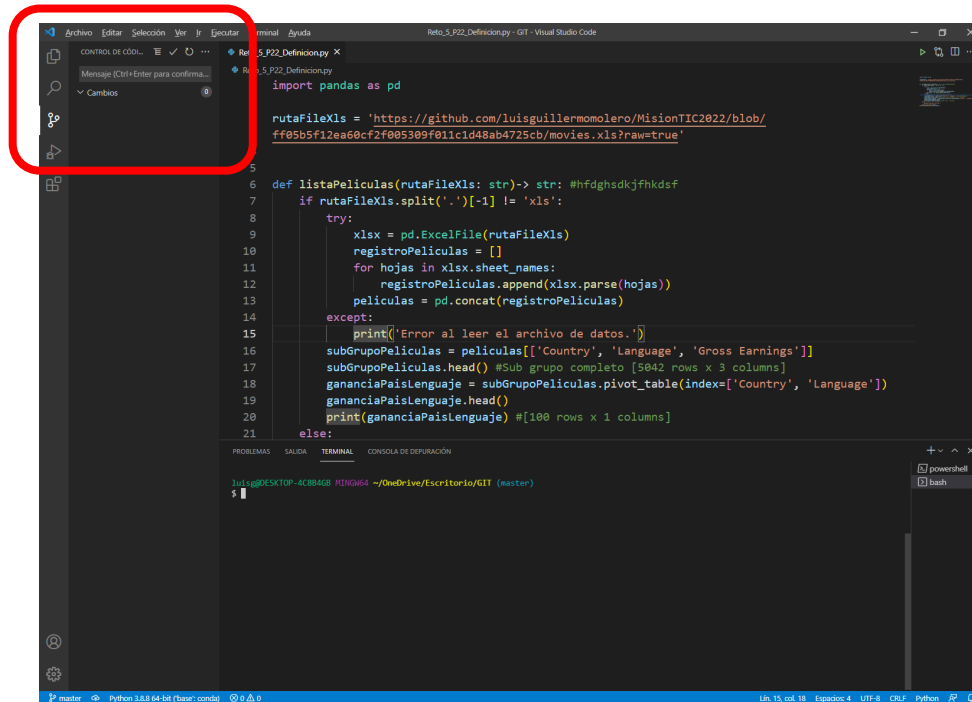
```
git config --global user.email "tu_correo_de_GITHUB"
git config --global user.name "tu_usuario_de_GITHUB"
```

RECUERDA:

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



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

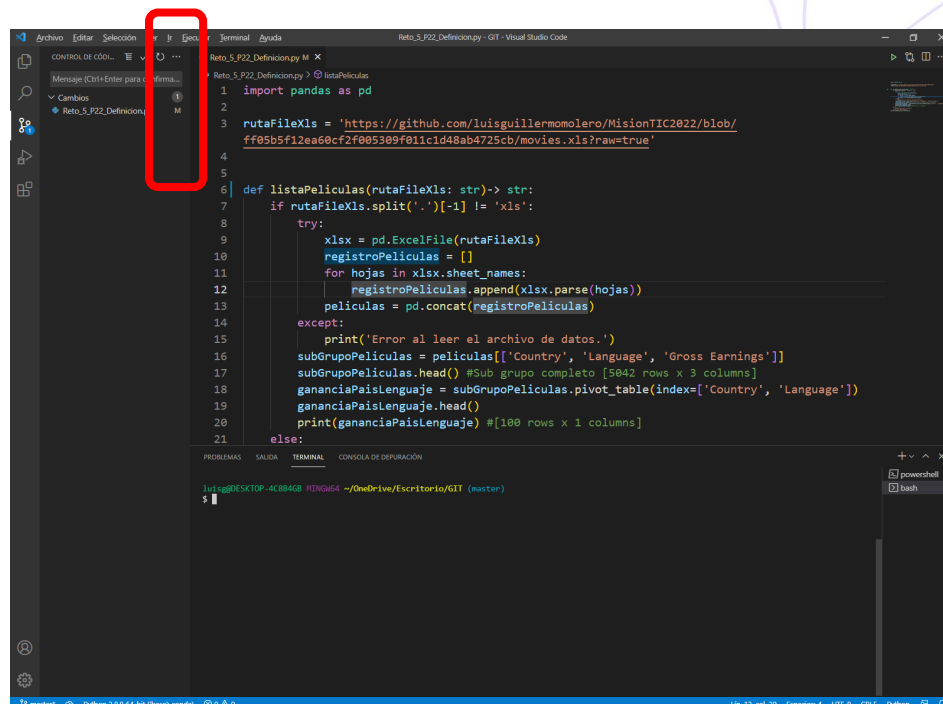


```

1  import pandas as pd
2
3  rutaFileXls = 'https://github.com/luisguillermomolero/MisionTIC2022/blob/ff05b5f1ea60cf2f085309f011c1d48ab4725cb/movies.xls?raw=true'
4
5
6  def listaPelículas(rutaFileXls: str) -> str: #hfdghsdskjfhkdsf
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             gananciaPaísLenguaje = subGrupoPelículas.pivot_table(index=['Country', 'Language'])
19             gananciaPaísLenguaje.head()
20             print(gananciaPaísLenguaje) #100 rows x 1 columns]
21     else:

```

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

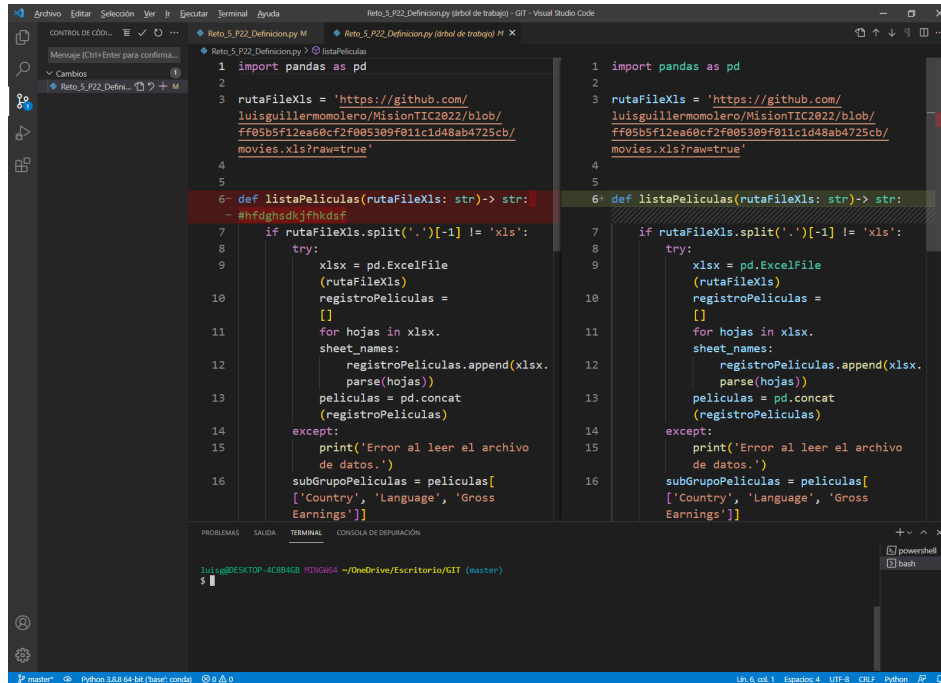


```

1  import pandas as pd
2
3  rutaFileXls = 'https://github.com/luisguillermomolero/MisionTIC2022/blob/ff05b5f1ea60cf2f085309f011c1d48ab4725cb/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 [5042 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:

```

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

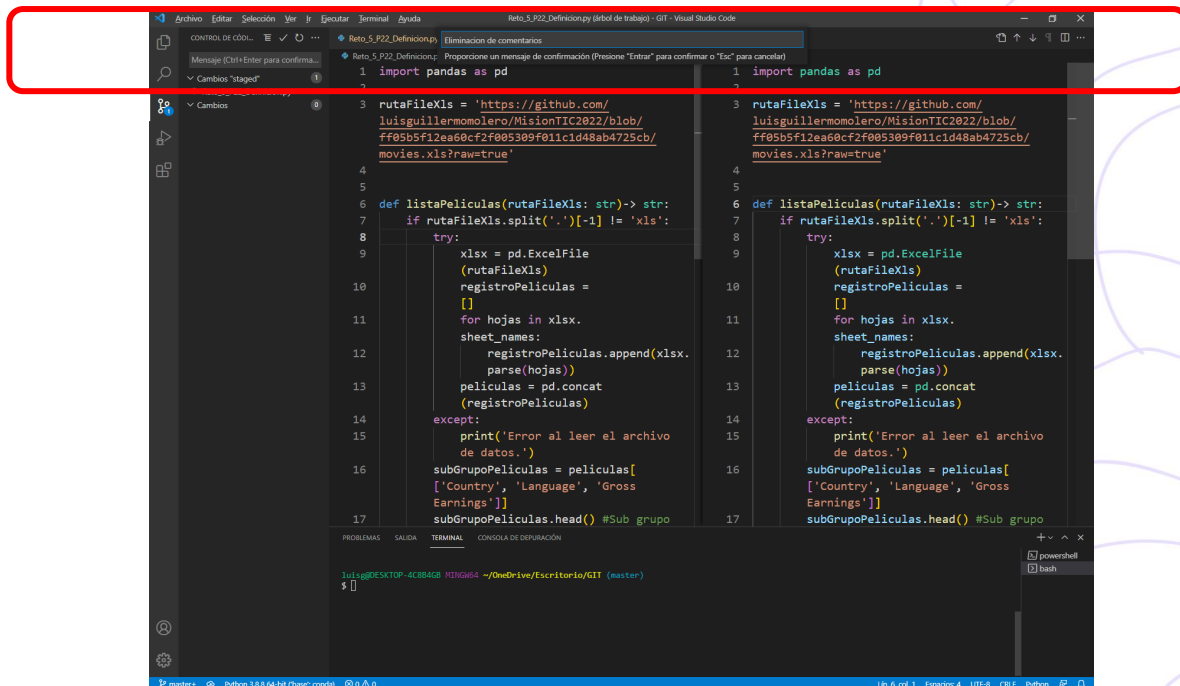


```

1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/
4 luisguillermomolero/MisionTIC2022/blob/
5 ff05b5f12ea60cf2f005309f011cd48ab4725cb/
6 movies.xls?raw=true'
7
8 def listaPelículas(rutaFileXls: str) -> str:
9     #hfdghsdqjfhkdsf
10    if rutaFileXls.split('.')[-1] != 'xls':
11        try:
12            xlsx = pd.ExcelFile
13            (rutaFileXls)
14            registroPelículas =
15            []
16            for hojas in xlsx.
17            sheet_names:
18                registroPelículas.append(xlsx.
19                parse(hojas))
20            películas = pd.concat
21            (registroPelículas)
22        except:
23            print('Error al leer el archivo
24            de datos.')
25            subGrupoPelículas = películas[
26            ['Country', 'Language', 'Gross
27            Earnings']]
28
29
30

```

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/
4 luisguillermomolero/MisionTIC2022/blob/
5 ff05b5f12ea60cf2f005309f011cd48ab4725cb/
6 movies.xls?raw=true'
7
8 def listaPelículas(rutaFileXls: str) -> str:
9     if rutaFileXls.split('.')[-1] != 'xls':
10        try:
11            xlsx = pd.ExcelFile
12            (rutaFileXls)
13            registroPelículas =
14            []
15            for hojas in xlsx.
16            sheet_names:
17                registroPelículas.append(xlsx.
18                parse(hojas))
19            películas = pd.concat
20            (registroPelículas)
21        except:
22            print('Error al leer el archivo
23            de datos.')
24            subGrupoPelículas = películas[
25            ['Country', 'Language', 'Gross
26            Earnings']]
27
28
29

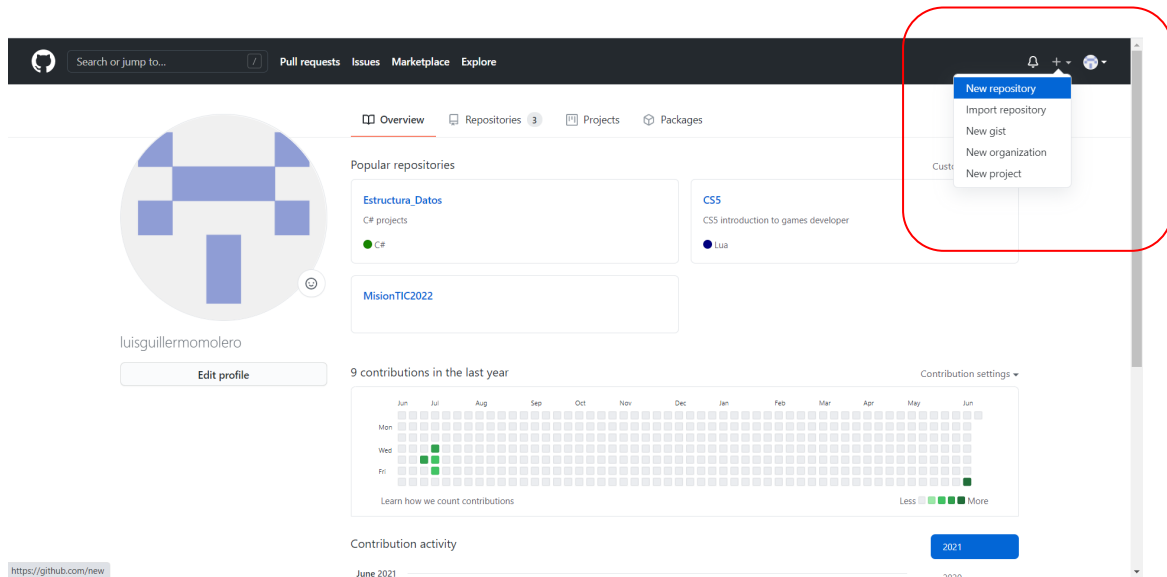
```

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

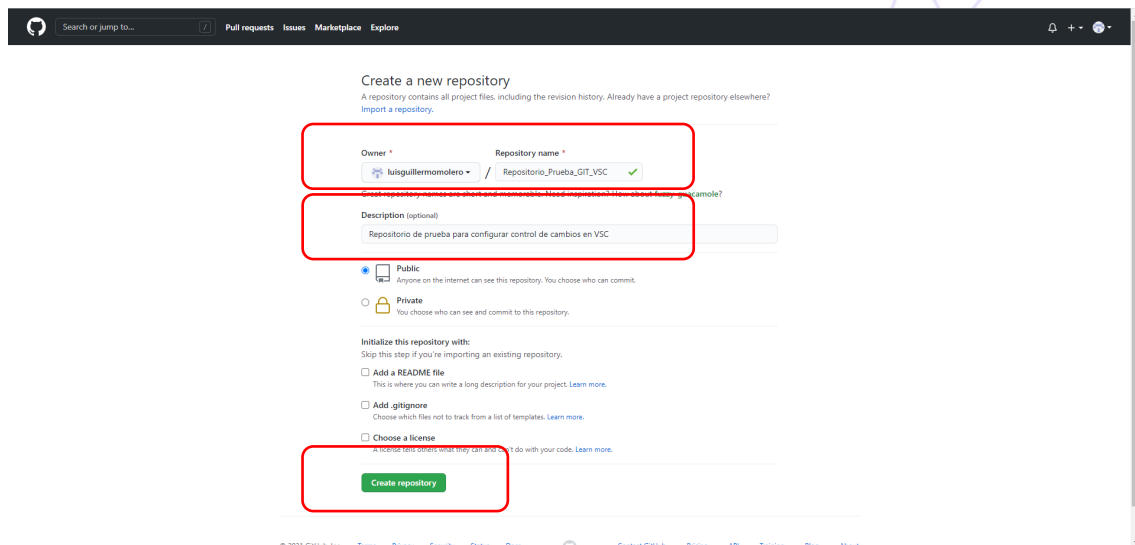
RECUERDA:

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

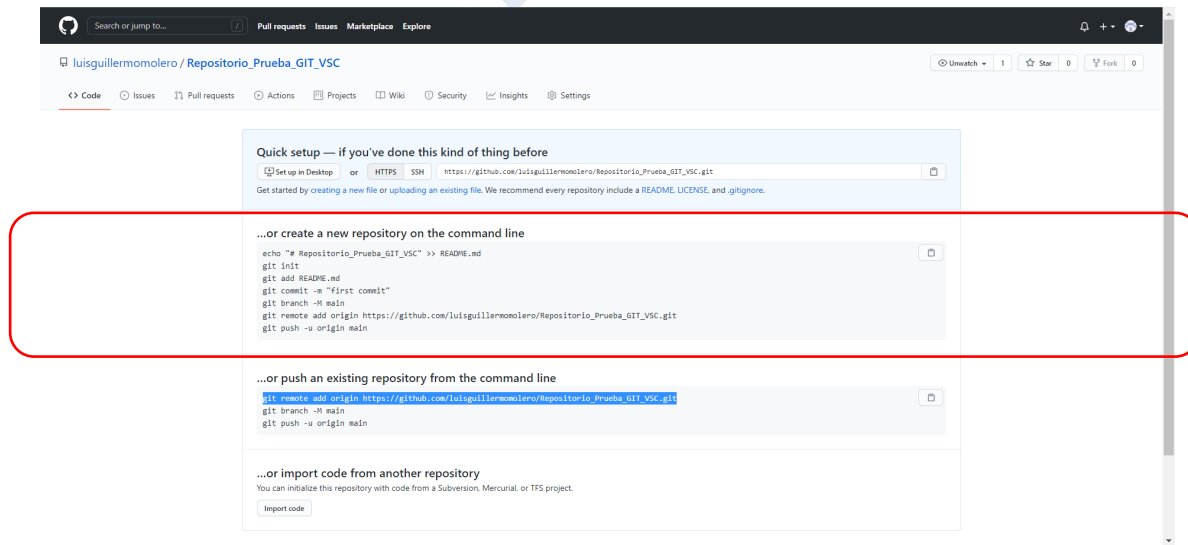
1.- Nos vamos a nuestra cuenta de GITHUB en la web y creamos un “Nuevo Repositorio”



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

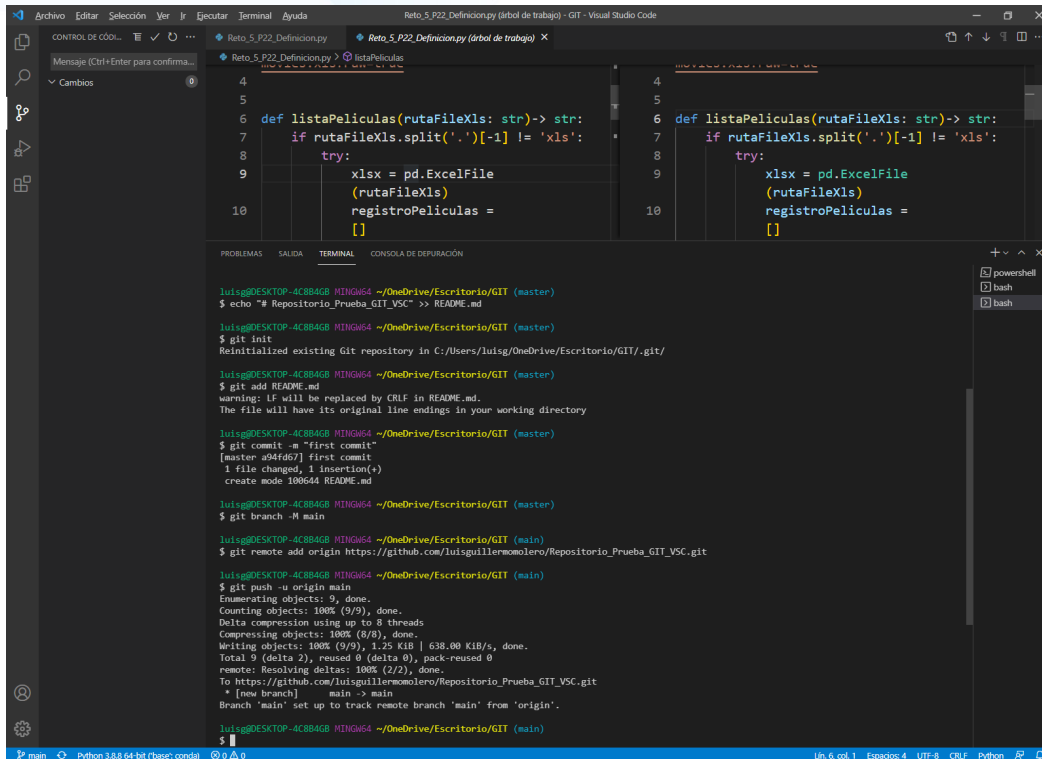


2.- Una vez creado el repositorio, copiamos las siguientes líneas de código que se encuentra en GitHub, en nuestra termina de GIT de VSC.



```
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
```





```

4
5
6 def listaPelículas(rutaFileXls: str)-> str:
7     if rutaFileXls.split('.')[1] != 'xls':
8         try:
9             xlsx = pd.ExcelFile
10            (rutaFileXls)
11            registroPelículas =
12
13
14
15
16
17
18
19
20
21
22
23
24

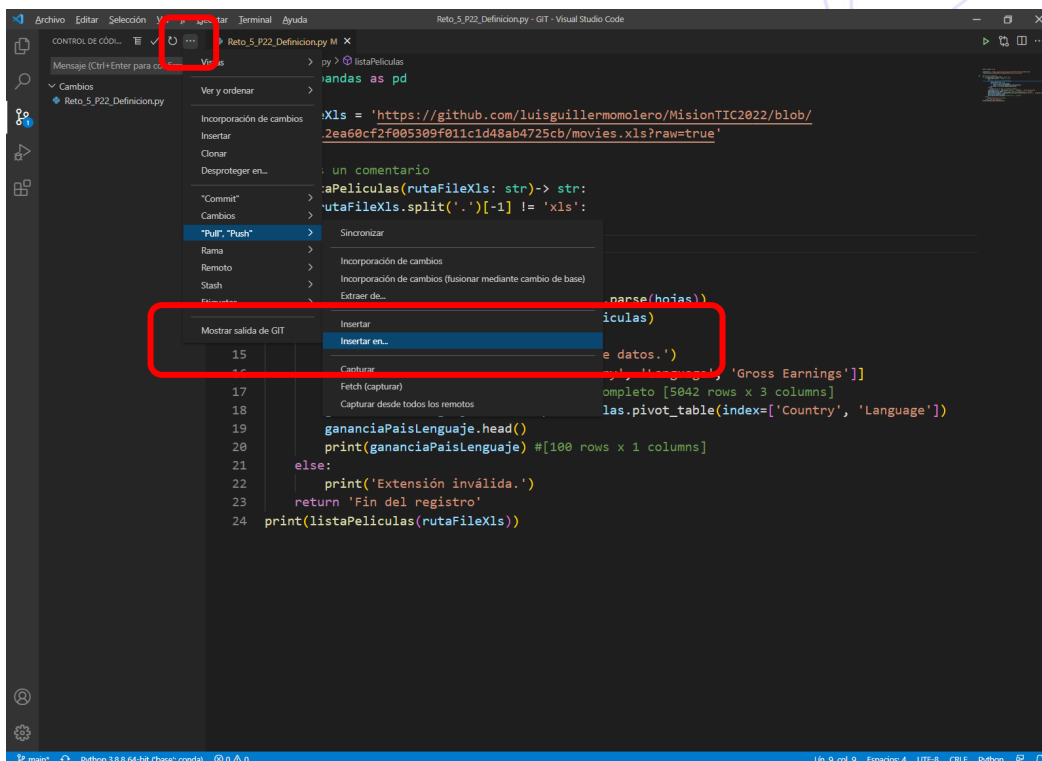
```

```

luig@DESKTOP-4C8B4GB MINGW64 ~/OneDrive/Escritorio/GIT (master)
$ echo "# Repositorio_Prueba_GIT_VSC" >> README.md
luig@DESKTOP-4C8B4GB MINGW64 ~/OneDrive/Escritorio/GIT (master)
$ git init
Reinitialized existing Git repository in C:/Users/luig/OneDrive/Escritorio/GIT/.git/
luig@DESKTOP-4C8B4GB MINGW64 ~/OneDrive/Escritorio/GIT (master)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
luig@DESKTOP-4C8B4GB MINGW64 ~/OneDrive/Escritorio/GIT (master)
$ git commit -m "first commit"
[master a04d6e7] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
luig@DESKTOP-4C8B4GB MINGW64 ~/OneDrive/Escritorio/GIT (master)
$ git branch -M main
luig@DESKTOP-4C8B4GB MINGW64 ~/OneDrive/Escritorio/GIT (main)
$ git remote add origin https://github.com/luiguillermomolero/Repositorio_Prueba_GIT_VSC.git
luig@DESKTOP-4C8B4GB MINGW64 ~/OneDrive/Escritorio/GIT (main)
$ git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 1.25 KiB | 638.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/luiguillermomolero/Repositorio_Prueba_GIT_VSC.git
  [new branch]   main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
luig@DESKTOP-4C8B4GB MINGW64 ~/OneDrive/Escritorio/GIT (main)
$

```

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



```

15
16
17
18
19
20
21
22
23
24

```

```

Xls = 'https://github.com/luiguillermomolero/MisionTIC2022/blob/
2ea60cf2f085309f011cd48ab4725cb/movies.xls?raw=true'

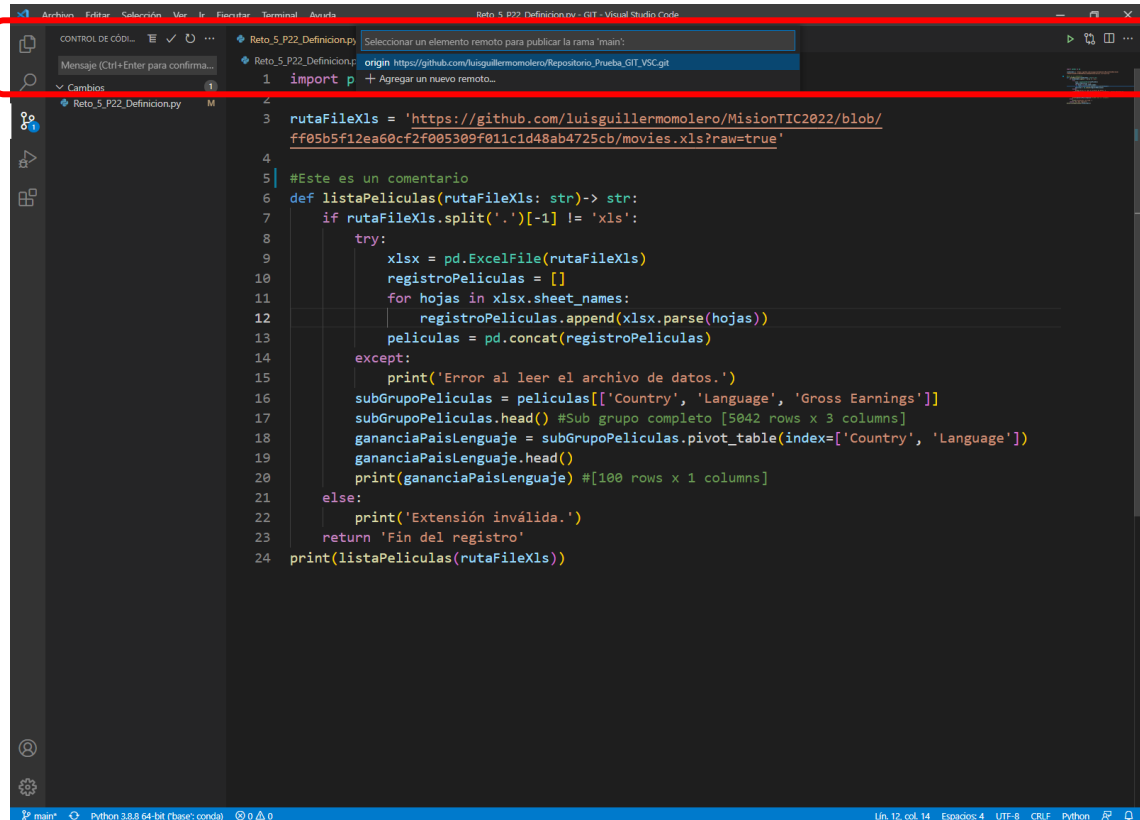
```

```

def listaPelículas(rutaFileXls: str)-> str:
    if rutaFileXls.split('.')[1] != 'xls':
        try:
            xlsx = pd.ExcelFile
            (rutaFileXls)
            registroPelículas =

```


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

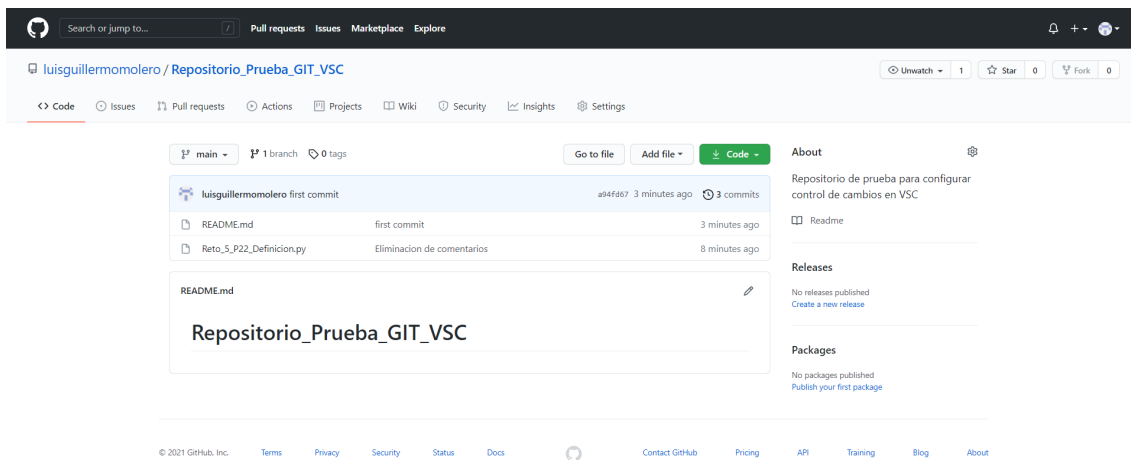


```

1 import pandas as pd
2
3 rutaFileXls = 'https://github.com/luiguillermomolero/MisionTIC2022/blob/ff05b5f12ea60cf2f005309f011c1d48ab4725cb/movies.xls?raw=true'
4
5 #Este es un comentario
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 [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))
  
```



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.