

```
scriptFork2.py X Script_Fork.py
scriptFork2.py > obtener_enlaces_validos
1 import os
2 import requests
3 from bs4 import BeautifulSoup
4 from urllib.parse import urljoin
5
6 # Crear carpetas si no existen
7 for folder in ["libros", "temporal"]:# 1. reduci las lineas de codigo en una sola en el cual se creen las dos carpetas si es que no existen
8     if not os.path.exists(folder):
9         os.makedirs(folder)
```

- Como primer cambio fue quitar la librería *shutil*.
- Además de también reducir líneas de código declarando las carpetas temporal y libros, en vez de hacerlo uno por uno declare las dos juntas.

```
11 def obtener_enlaces_validos(base_url, main_url):# 2. validamos los enlaces de la pagina en la cual realizara la descarga
12 # Realizar solicitud HTTP a la página principal
13     response = requests.get(main_url)
14     soup = BeautifulSoup(response.content, "html.parser")
15 # Encontrar y almacenar enlaces
16     enlaces_validos = []
17     for enlace in soup.find_all("a", href=True):
18         if "2023/" in enlace["href"] and enlace["href"].endswith(".htm"):# 3. reasignamos los valores que toman cada link de cada imagen
19             # Extraer el nombre de la carpeta de cada enlace y almacenarlo
20             enlaces_validos.append(enlace["href"].split("/")[1].split(".")[0][-5:])
21     return enlaces_validos # 4. hacemos retornar el valor del enlace valido
22
```

- Realice dentro de una función llamada *obtener_enlaces_validos*, la solicitud http a la página y a la misma función pedir como parámetros los url's de la página el *base_url* y el *main_url*.
- Añadi con ciertos parámetros los enlaces que han sido validados y los retornamos en la misma función.

```
24 def descargar_imagenes(image_base_url, enlaces_validos):# 5. creo un metodo que descargara las imagenes y tendra por parametros la base del url y si en en.
25     for folder_name in enlaces_validos:
26         subfolder_path = os.path.join("libros", folder_name)
27         if not os.path.exists(subfolder_path):
28             os.makedirs(subfolder_path)
29 # Descargar y guardar las imágenes en la carpeta "temporal"
30     for image_number in range(401):
31         image_download_url = image_base_url.format(folder_name, str(image_number).zfill(3))
32         image_response = requests.get(image_download_url)
33         if image_response.status_code == 200:
34             image_path = os.path.join("temporal", f"{folder_name}_{image_number:03d}.jpg")
35             with open(image_path, "wb") as f:
36                 f.write(image_response.content)
37 # Mover archivos de la carpeta temporal a las subcarpetas correspondientes en "libros"
38 def mover_archivos_a_carpetas():
39     temporal_folder = "temporal"
40     libros_folder = "libros"
41     # Listar los archivos en la carpeta temporal
42     for file_name in os.listdir(temporal_folder):
43         if file_name.endswith(".jpg"):
44             # Obtener la cadena de 5 letras del nombre de archivo
45             folder_name = file_name.split("_")[0]
46             # Obtener la ruta completa del archivo en temporal
47             temp_file_path = os.path.join(temporal_folder, file_name)
48             # Obtener la ruta completa de la subcarpeta en libros
49             subfolder_path = os.path.join(libros_folder, folder_name)
50             # Crear la subcarpeta si no existe
51             if not os.path.exists(subfolder_path):
52                 os.makedirs(subfolder_path)
53             # Mover el archivo a la subcarpeta en libros
54             new_file_path = os.path.join(subfolder_path, file_name)
55             os.rename(temp_file_path, new_file_path)
56
```

- Cree otro método llamado *Descargar_imagenes* el cual lleva como parámetros la *url base* y la extensión validada de la imagen que se descargarán de cada url.
- En la misma función coloque el procedimiento donde se crearán las subcarpetas donde contendrán las imágenes de los libros correspondientes al url del libro en cuestión, además de también todas las consultas y llamadas del url y carpetas del proyecto.

```
57 if __name__ == "__main__": # 5 declaramos dentro del if los metodos creado arriba repectivamente con sus variables url y sus parametros
58     base_url = "https://www.conaliteg.sep.gob.mx/"
59     main_url = urljoin(base_url, "primaria.html")
60     image_base_url = urljoin(base_url, "2023/c/{}/{}.jpg")
61
62     enlaces_validos = obtener_enlaces_validos(base_url, main_url)
63     descargar_imagenes(image_base_url, enlaces_validos)
64     mover_archivos_a_carpetas() # 6. elimine la libreria shutil, y mandamos los temporales a la carpeta libros por otro metodo
```

- Dentro de un if ingrese los parámetros de *base_url*, *main_url*, y un *join* del url que será la respectiva para cada libro de la pagina
- Al igual de la variable aquí mismo declare las 3 funciones de arriba.

Resultado



