

```

#encoding:utf-8

from bs4 import BeautifulSoup
import urllib.request
from tkinter import *
from tkinter import messagebox
import sqlite3
import lxml
from datetime import datetime
# lineas para evitar error SSL
import os, ssl
if (not os.environ.get('PYTHONHTTPSVERIFY', '') and
    getattr(ssl, '_create_unverified_context', None)):
    ssl._create_default_https_context = ssl._create_unverified_context

def cargar():
    respuesta = messagebox.askyesno(title="Confirmar",message="Esta seguro que quiere recargar los datos. \nEsta operaci3n puede ser lenta")
    if respuesta:
        almacenar_bd()

def almacenar_bd():

    conn = sqlite3.connect('peliculas.db')
    conn.text_factory = str
    conn.execute("DROP TABLE IF EXISTS PELICULA")
    conn.execute('''CREATE TABLE PELICULA
        (TITULO          TEXT NOT NULL,
        TITULO_ORIGINAL  TEXT          ,
        PAIS            TEXT,
        FECHA           DATE,
        DIRECTOR        TEXT,
        GENEROS         TEXT);''')

    f = urllib.request.urlopen("https://www.elseptimoarte.net/estrenos/")
    s = BeautifulSoup(f, "lxml")
    lista_link_peliculas = s.find("ul", class_="elements").find_all("li")
    for link_pelicula in lista_link_peliculas:
        f =
        urllib.request.urlopen("https://www.elseptimoarte.net/"+link_pelicula.a['href'])
        s = BeautifulSoup(f, "lxml")
        datos = s.find("main",
            class_="informativo").find("section",class_="highlight").div.dl
            titulo_original = datos.find("dt",string="TÃ-tulo
original").find_next_sibling("dd").string.strip()
            #si no tiene tÃ-tulo se pone el tÃ-tulo original
            if (datos.find("dt",string="TÃ-tulo")):
                titulo =
datos.find("dt",string="TÃ-tulo").find_next_sibling("dd").string.strip()
            else:
                titulo = titulo_original

```

```

        pais =
"".join(datos.find("dt",string="PaÃ-s").find_next_sibling("dd").stripped_strings
)
        fecha = datetime.strptime(datos.find("dt",string="Estreno en
EspaÃa").find_next_sibling("dd").string.strip(), '%d/%m/%Y')

        generos_director = s.find("div",id="datos_pelicula")
        generos =
"".join(generos_director.find("p",class_="categorias").stripped_strings)
        director =
"".join(generos_director.find("p",class_="director").stripped_strings)

        conn.execute("""INSERT INTO PELICULA (TITULO, TITULO_ORIGINAL, PAIS,
FECHA, DIRECTOR, GENEROS) VALUES (?, ?, ?, ?, ?, ?)""",
                    (titulo,titulo_original,pais,fecha,director,generos))
        conn.commit()

        cursor = conn.execute("SELECT COUNT(*) FROM PELICULA")
        messagebox.showinfo("Base Datos",
                            "Base de datos creada correctamente \nHay " +
str(cursor.fetchone()[0]) + " registros")
        conn.close()

def buscar_por_titulo():
    def listar(event):
        conn = sqlite3.connect('peliculas.db')
        conn.text_factory = str
        cursor = conn.execute("SELECT TITULO, PAIS, DIRECTOR FROM PELICULA
WHERE TITULO LIKE '%" + str(entry.get()) + "%'")
        conn.close
        listar_peliculas(cursor)
    ventana = Toplevel()
    label = Label(ventana, text="Introduzca cadena a buscar ")
    label.pack(side=LEFT)
    entry = Entry(ventana)
    entry.bind("<Return>", listar)
    entry.pack(side=LEFT)

def buscar_por_fecha():
    def listar(event):
        conn = sqlite3.connect('peliculas.db')
        conn.text_factory = str
        try:
            fecha = datetime.strptime(str(entry.get()),"%d-%m-%Y")
            cursor = conn.execute("SELECT TITULO, FECHA FROM PELICULA WHERE
FECHA > ?", (fecha,))
            conn.close
            listar_peliculas_1(cursor)
        except:
            conn.close
            messagebox.showerror(title="Error",message="Error en la

```

```

fecha\nFormato dd-mm-aaaa")
    v = Toplevel()
    label = Label(v, text="Introduzca la fecha (dd-mm-aaaa) ")
    label.pack(side=LEFT)
    entry = Entry(v)
    entry.bind("<Return>", listar)
    entry.pack(side=LEFT)

def buscar_por_genero():
    def listar(Event):
        conn = sqlite3.connect('peliculas.db')
        conn.text_factory = str
        cursor = conn.execute("SELECT TITULO, FECHA FROM PELICULA where
GENEROS LIKE '%" + str(entry.get())+"%'")
        conn.close
        listar_peliculas_1(cursor)

    conn = sqlite3.connect('peliculas.db')
    conn.text_factory = str
    cursor = conn.execute("SELECT GENEROS FROM PELICULA")

    generos=set()
    for i in cursor:
        generos_pelicula = i[0].split(",")
        for genero in generos_pelicula:
            generos.add(genero.strip())

    v = Toplevel()
    label = Label(v, text="Seleccione un g nero ")
    label.pack(side=LEFT)
    entry = Spinbox(v, values=list(generos, state='readonly'))
    entry.bind("<Return>", listar)
    entry.pack(side=LEFT)

    conn.close()

def listar_peliculas_1(cursor):
    v = Toplevel()
    sc = Scrollbar(v)
    sc.pack(side=RIGHT, fill=Y)
    lb = Listbox(v, width=150, yscrollcommand=sc.set)
    for row in cursor:
        s = 'T TULO: ' + row[0]
        lb.insert(END, s)
        lb.insert(END, "-----")
        fecha = datetime.strptime(row[1], "%Y-%m-%d %H:%M:%S") #sqlite almacena
las fechas como str
        s = "        FECHA DE ESTRENO: " + datetime.strftime(fecha,"%d/%m/%Y")
        lb.insert(END, s)
        lb.insert(END, "\n\n")

```

```
lb.pack(side=LEFT, fill=BOTH)
sc.config(command=lb.yview)
```

```
def listar_peliculas(cursor):
    v = Toplevel()
    sc = Scrollbar(v)
    sc.pack(side=RIGHT, fill=Y)
    lb = Listbox(v, width=150, yscrollcommand=sc.set)
    for row in cursor:
        s = 'TÍTULO: ' + row[0]
        lb.insert(END, s)
        lb.insert(END,
"-----")
        s = "      PAÍS: " + str(row[1]) + ' | DIRECTOR: ' + row[2]
        lb.insert(END, s)
        lb.insert(END, "\n\n")
    lb.pack(side=LEFT, fill=BOTH)
    sc.config(command=lb.yview)
```

```
def ventana_principal():
    def listar():
        conn = sqlite3.connect('peliculas.db')
        conn.text_factory = str
        cursor = conn.execute("SELECT TITULO, PAIS, DIRECTOR FROM PELICULA")
        conn.close
        listar_peliculas(cursor)

    raiz = Tk()

    menu = Menu(raiz)

    #DATOS
    menudatos = Menu(menu, tearoff=0)
    menudatos.add_command(label="Cargar", command=cargar)
    menudatos.add_command(label="Listar", command=listar)
    menudatos.add_command(label="Salir", command=raiz.quit)
    menu.add_cascade(label="Datos", menu=menudatos)

    #BUSCAR
    menubuscar = Menu(menu, tearoff=0)
    menubuscar.add_command(label="Título", command=buscar_por_titulo)
    menubuscar.add_command(label="Fecha", command=buscar_por_fecha)
    menubuscar.add_command(label="Géneros", command=buscar_por_genero)
    menu.add_cascade(label="Buscar", menu=menubuscar)

    raiz.config(menu=menu)

    raiz.mainloop()
```

```
if __name__ == "__main__":  
    ventana_principal()
```