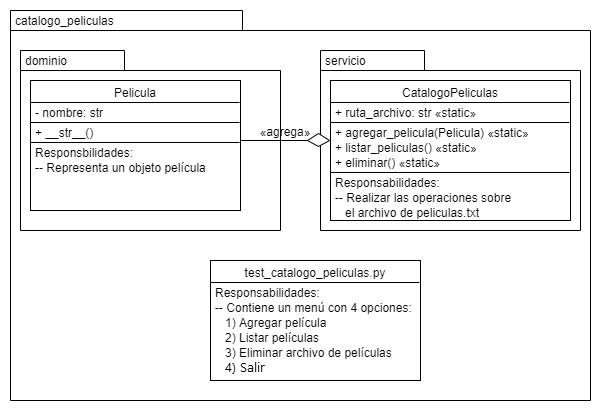
Para esta seccion se va a crear un proyecto de catálogo de películas:



El directorio se divide en 2 paquetes, dominio y servicio

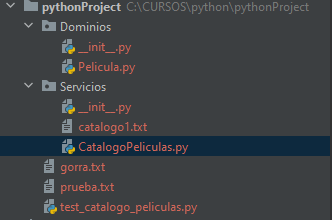
**USANDO PYCHARM**

Para esta sección decidí utilizar pycharm para poder crear paquetes de Python e importar módulos con archivos fuera del directorio local.

Este es el código de la clase CatalogoPeliculas:

from Dominios.Pelicula import Pelicula  
import os  
  
class CatalogoPeliculas:  
  
 ruta\_archivos = ''  
  
 def \_\_init\_\_(self, ruta):  
 self.ruta = ruta  
  
 def add\_pelicula(self, pelicula):  
 archivo = open(self.ruta, "a", encoding="utf8")  
 archivo.write(f'\n {pelicula}')  
  
 def listar\_peliculas(self):  
 archivo = open(self.ruta, "r", encoding="utf8")  
 for e in archivo.readlines():  
 print(e)  
  
 def eliminar\_catalogo(self):  
 os.remove(self.ruta)

como se ve se puede importar la clase Pelicula del modulo Pelicula.py a pesar de que la estructura del directorio sea la siguiente:



Ahora, en el archivo test contamos con el siguiente código:

print('Bienvenido al sistema de catalogo de peliculas'.center(100, '-'))  
print('Estas son las acciones que puedes realizar:')  
print(f'''  
1) Agregar pelicula  
2) listar películas  
3) Eliminar catalogo de peliculas  
4) salir  
''')  
  
n\_cata = input('Introduce el nombre para un nuevo catalogo de peliculas (es necesario el .txt)')  
  
accion = ''  
  
from Servicios.CatalogoPeliculas import CatalogoPeliculas  
  
cat = CatalogoPeliculas(n\_cata)  
  
with open(cat.ruta, "a+", encoding="utf8") as archivo:  
 while(accion != 4):  
 try:  
 accion = int(input('Escribe el numero de acción a realizar (1-4) '))  
 except Exception as e:  
 # print(f'error {e}')  
 print('por favor introduce algun valor válido')  
 continue  
  
 if (accion >= 1 and accion <= 3):  
 if(accion == 1):  
 new\_film = input('Introduce el nombre de la nueva película ')  
 cat.add\_pelicula(new\_film)  
 continue  
 elif(accion == 2):  
 cat.listar\_peliculas()  
 continue  
 elif(accion == 3):  
 archivo.close()  
 cat.eliminar\_catalogo()  
 print(f'Programa finalizado, gracias por usar el sistema!'.center(75, '-'))  
 break  
 elif(accion != 4):  
 print('el numero ingresado no corresponde a alguna accion')  
 continue  
 else:  
 print('saliendo del programa'.center(50, '-'))

las primeras 3 lineas de print son de instrucción para el usuario que utiliza el sistema

la siguiente linea input es para crear el archivo .txt para utilizar como catalogo

Luego se importa la clase CatalogoPeliculas y se inicializa un objeto con la ruta que el usuario introdujo anteriormente.

La siguiente ruta sirve para abrir el archivo en el “manager context”. El archivo se abre en modo apend+ para poder tanto leer como escribir datos

El bloque principal del programa es un while() que se ejecuta mientras acción sea distinto a 4, por tanto lo primero que se hace al “entrar” al bloque es pedir al usuario un valor para acción. En base a este, se seguirán distintos caminos. Las acciones en este caso son:

1. Agregar películas, para lo cual se llama al método add\_pelicula() de la instancia de la clase CatalogoPeliculas
2. Listar películas, para lo cual se llama al método listar\_peliculas() de la instancia de la clase CatalogoPeliculas
3. Eliminar el catalogo. En este caso, es necesario cerrar el archivo dentro de este bloque ya que no es posible eliminar un archivo mientras se esta utilizando. Luego de cerrar, se invoca al archivo eliminar\_catalogo()
4. Salir del programa. Las modificaciones en el archivo se conservan y se cierra el programa.