

CATÁLOGO DE PELÍCULAS CON ARCHIVOS Y PYTHON

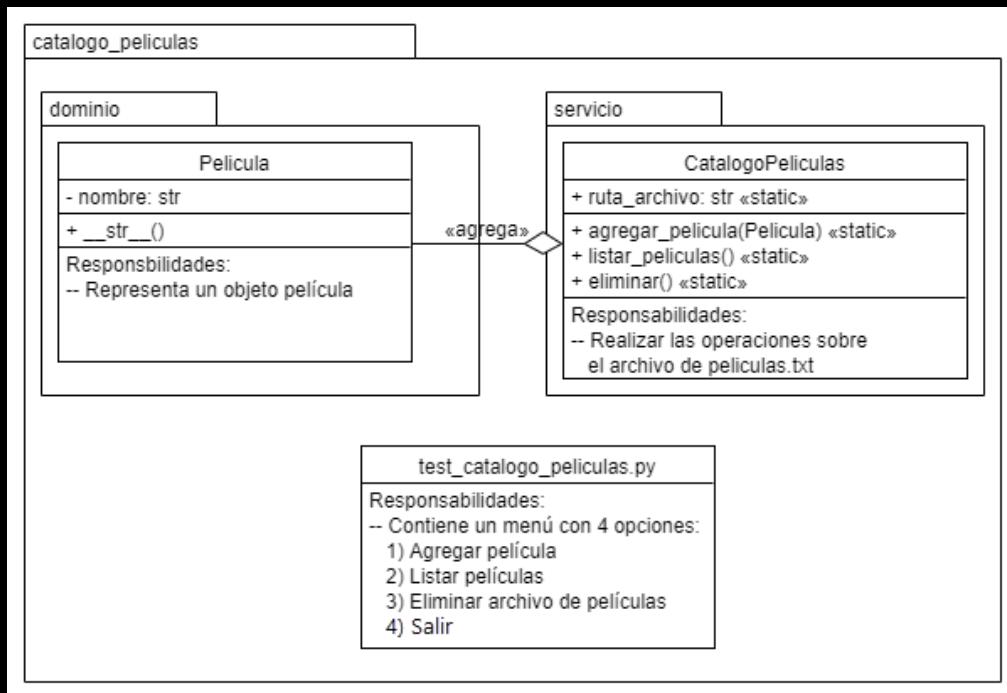


🎬 Laboratorio: Creación de una aplicación de Catálogo de Películas en Python

⭐ Introducción

En este laboratorio final, les dejamos crear una aplicación para la **administración de un catálogo de películas**. El proyecto estará organizado en una estructura de carpetas y archivos, incluyendo clases específicas para manejar la lógica de dominio y de servicio, así como un archivo de prueba que permitirá interactuar mediante un menú de opciones.

Esta aplicación permitirá **agregar películas, listar las películas existentes y eliminar el archivo de películas**, con una relación estructurada entre clases y su funcionamiento a través de la consola.



📝 Paso 1: Estructura de carpetas y archivos

Para comenzar, organizamos nuestro proyecto con la siguiente estructura:

📁 `catalogo_peliculas`

- 📁 `dominio`
 - `Pelicula.py`
- 📁 `servicio`
 - `CatalogoPeliculas.py`
- `test_catalogo_peliculas.py`

Esta organización facilita la separación de responsabilidades de cada clase y archivo dentro del proyecto.

📝 Paso 2: Crear la clase Pelicula

📄 **Archivo:** `Pelicula.py`

📁 **Ruta:** `/catalogo_peliculas/dominio/Pelicula.py`

 **Descripción breve:**

Esta clase representa una película y contendrá un único atributo privado llamado `nombre`. Además, sobreescribirá el método `__str__` para mostrar su estado al imprimir el objeto.

 **Explicación:**

La clase `Pelicula` es una **clase de dominio**, ya que representa directamente una película dentro de la lógica del problema. Su objetivo es almacenar y mostrar el nombre de la película cuando sea necesario.

 **Paso 3: Crear la clase CatalogoPeliculas** **Archivo:** `catalogoPeliculas.py` **Ruta:** `/catalogo_peliculas/servicio/CatalogoPeliculas.py` **Descripción breve:**

Esta clase actuará como **clase de servicio**, conteniendo los métodos para manipular el archivo de películas.

 **Explicación:**

La clase incluirá:

- Una variable estática `ruta_archivo` que contendrá el nombre del archivo (ejemplo: `peliculas.txt`).
- El método `agregar_pelicula`, que abrirá el archivo en modo append y añadirá una nueva película.
- El método `listar_peliculas`, que mostrará todas las películas guardadas en el archivo.
- El método `eliminar`, que eliminará el archivo utilizando el módulo `os` y su función `remove`.

 Para eliminar el archivo se utiliza:

```
import os
os.remove(ruta_archivo)
```

 Si el archivo está en la misma carpeta, basta con pasar solo su nombre. **Paso 4: Crear el archivo de prueba TestCatalogoPeliculas.py** **Archivo:** `test_catalogo_peliculas.py` **Ruta:** `/catalogo_peliculas/test_catalogo_peliculas.py`

Descripción breve:

Este archivo permitirá interactuar con el catálogo mediante un **menú en consola** con las siguientes opciones:

1. Agregar película
2. Listar películas
3. Eliminar el archivo de películas
4. Salir

Explicación:

El menú se implementará con un ciclo `while` que:

- Mostrará las opciones de manera repetida.
- Ejecutará la acción elegida por el usuario.
- Terminará la aplicación solo si el usuario elige la opción 4.

- Cada opción ejecutará su método correspondiente de la clase `CatalogoPeliculas`.
-

Paso 5: Ejecución esperada de la aplicación

Al ejecutar el archivo `test_catalogo_peliculas.py`, la aplicación mostrará el siguiente menú:

Opciones:

1. Agregar película
2. Listar películas
3. Eliminar catálogo de películas
4. Salir

 Si se selecciona la opción 1, pedirá ingresar el nombre de la película (por ejemplo, "Batman") y la agregará al archivo `peliculas.txt`.

 Al volver a elegir la opción 1, permitirá agregar otra película (por ejemplo, "Superman").

 Al elegir la opción 2, mostrará todas las películas guardadas.

 La opción 3 eliminará el archivo de películas.

 La opción 4 cerrará la aplicación.

- Cada acción se ejecuta de acuerdo con la opción seleccionada, manteniendo la aplicación activa hasta que se elija salir.
-

Conclusión

En esta guía les hemos agregado los pasos que deben realizar para construir la aplicación de **Catálogo de Películas** siguiendo un diseño basado en **clases de dominio y servicio**, aplicando conceptos como encapsulamiento, modularidad y uso de archivos de texto para persistencia de datos.

- Este proyecto les permitirá practicar la creación de clases organizadas en carpetas, interactuar con archivos e implementar un menú interactivo en consola.

👏 ¡Mucho éxito, vamos a ver la solución en los siguientes videos!. 🚀

Sigue adelante con tu aprendizaje 🚀, ¡el esfuerzo vale la pena!

¡Saludos! 🙋

Ing. Marcela Gamiño e Ing. Ubaldo Acosta

Fundadores de [GlobalMentoring.com.mx](https://www.globalmentoring.com.mx)