

SEGUNDO PARCIAL PROGRAMACIÓN

Proyecto: Sistema de Gestión de Libros con API de Google Books

Alumnos – Grupo:

Liendo Mateo, Avila Lucas y Pagano Amanda

Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional.

Programación

Docente Titular

Hualpa Ramiro

05 de Noviembre de 2025

Tabla de contenido

Introducción	3
Desarrollo	
Consigna	6
Consigna	12
Conclusión	13
Referencias	14

Segundo parcial programación

Introducción:

El presente proyecto consiste en el desarrollo de un sistema de gestión de libros que combina el manejo de archivos locales con la integración de una API externa (Google Books).

El programa fue diseñado aplicando una **estructura jerárquica de datos de tres niveles** —género, autor y año— que se mapea directamente al sistema de archivos, creando automáticamente las carpetas correspondientes según los datos de cada libro.

Los ítems se manejan internamente como **diccionarios de Python**, lo que permite una manipulación clara y flexible de la información.

Se implementó también una función **recursiva**, llamada `consolidar_libros_recursivamente()`, que recorre toda la jerarquía de carpetas y recopila los datos dispersos en un único conjunto de información global.

Gracias a esto, el sistema puede mostrar listados completos, generar estadísticas y aplicar filtros sin necesidad de recorrer manualmente cada archivo CSV.

Además, se aplicó un enfoque **modular**, separando las distintas responsabilidades del programa en archivos específicos: manejo de almacenamiento, validaciones, conexión con la API, estadísticas, ordenamiento... etc.

Esto permitió obtener un código más limpio, mantenible y escalable, cumpliendo con los principios de organización y reutilización propios de la programación estructurada.

Objetivo general:

- Desarrollar un sistema de gestión de libros que permita agregar, consultar, modificar, eliminar y analizar datos de manera local, con la posibilidad de integrar información desde una API externa.

Objetivos específicos:

1. Implementar una estructura de carpetas jerárquica dinámica que organice los libros por género, autor y año de publicación.
2. Desarrollar funciones CRUD (crear, leer, actualizar y eliminar) que interactúen con archivos CSV.
3. Aplicar una función recursiva que recorra la jerarquía de carpetas y consolide la información existente.
4. Incorporar un modo de trabajo alternativo conectado a la **Google Books API**, para realizar búsquedas de libros y almacenar resultados de forma estructurada.
5. Añadir validaciones estrictas para garantizar la coherencia de los datos y evitar errores en los archivos.

6. Integrar funcionalidades complementarias, como estadísticas, ordenamiento y filtrado avanzado (con búsqueda aproximada y sin acentos).
7. Promover el uso de **modularización**, separando las distintas responsabilidades del programa en archivos independientes para mejorar la organización y mantenimiento del código.

Desarrollo

El desarrollo del sistema lo realizamos en el lenguaje **Python**, utilizando los módulos os, csv, requests y unicodedata para la creación de carpetas, manejo de archivos y conexión con la API de Google Books, respectivamente.

El programa se estructuró siguiendo una **arquitectura modular**, separando las distintas funcionalidades en archivos específicos dentro de la carpeta src/, lo que facilita la organización, lectura y mantenimiento del código.

La aplicación implementa una **estructura jerárquica de tres niveles**:
género / autor / año / items.csv

Cada libro se guarda dentro de la carpeta correspondiente a su género, autor y año de publicación, y los datos se almacenan en un archivo CSV final llamado items.csv.

Esta jerarquía se genera de forma automática mediante la función ensure_path_for_book(), que crea las carpetas necesarias según la información ingresada por el usuario u obtenida desde la API.

Cada libro se representa internamente como un **diccionario**, conteniendo las claves género, autor, anio, titulo y páginas.

Este enfoque facilita la validación, actualización y eliminación de registros, además de permitir que las distintas funciones del programa manipulen los datos de manera consistente y clara.

Una de las partes centrales del proyecto es la función **recursiva** `consolidar_libros_recursivamente()`.

Esta función recorre toda la jerarquía de carpetas en busca de archivos `items.csv`, y a medida que los encuentra, lee su contenido y lo combina en una sola lista global de diccionarios.

De esta forma, el sistema puede acceder a todos los libros disponibles sin importar en qué carpeta estén ubicados, lo que resulta fundamental para generar listados, aplicar filtros, calcular estadísticas y realizar ordenamientos.

La modularización fue otro aspecto clave del proyecto.

Las funcionalidades se dividieron de la siguiente manera:

- ◆ **main.py**: gestiona el menú principal y permite cambiar entre el modo local (CSV) y el modo API (Google Books).
- ◆ **almacenamiento.py**: maneja la recursividad contiene la función recursiva que recorre toda la jerarquía de carpetas y consolida los datos.
- ◆ **api_libros.py**: conecta con la **Google Books API**, realiza búsquedas por aproximación, filtra los resultados válidos y permite guardarlos en la estructura local.
- ◆ `consolidar_libros_recursivamente()` para recorrer y consolidar la jerarquía.

- ◆ **ordenamiento.py** y **estadisticas.py**: implementan funciones para ordenar los libros y generar estadísticas globales del sistema.
- ◆ **Modificar.py**: contiene las funciones que permiten eliminar y editar datos del CSV
- ◆ **Guardado.py**: están las funciones relacionadas al CSV creación, guardado y validaciones.
- ◆ **Mostrar.py**: contiene las funciones de mostrar libros y las funciones auxiliares de estabilización de texto y búsqueda por aproximación, la función `_seleccionar_item_por_id()` que se usa para:
 - Mostrar todos los libros disponibles (leyendo desde la jerarquía con `consolidar_libros_recursivamente()`).
 - Permitir al usuario elegir uno por número o ID.
 - Devolver el libro seleccionado para **modificarlo o eliminarlo**.

En conjunto, estos módulos conforman un sistema robusto, flexible y extensible, capaz de funcionar tanto de manera local como conectada, aprovechando la API para ampliar el catálogo de libros sin perder la estructura jerárquica definida en el almacenamiento.

Opción 2 modo API:

```
hp@DESKTOP-53SCNNN MINGW64 ~/OneDrive/Escritorio/trabajos_practicos/Integrador-Programacion-Parte-2 (main)
$ C:/Users/hp/AppData/Local/Programs/Python/Python313/python.exe c:/Users/hp/OneDrive/Escritorio/trabajos_practicos/Integrador-Programacion-Parte-2/src/main.py
Elegí una opción: 2

● Ingresá un género, autor o palabra clave para buscar libros: harry

🔍 Resultados encontrados para: 'harry'

[1] El problema con Harry Hay
    Autor: Timmons, Stuart
    Género: Biography & Autobiography
    Año: 2025 | Páginas: 482
---
[2] Petirojo (Harry Hole 3)
    Autor: Jo Nesbo
    Género: Fiction
    Año: 2016 | Páginas: 628
---
[3] Eclipse (Harry Hole 13)
    Autor: Jo Nesbo
    Género: Fiction
    Año: 2023 | Páginas: 628
---
[4] Cucarachas (Harry Hole 2)
    Autor: Jo Nesbo
    Género: Fiction
    Año: 2015 | Páginas: 437
---
[5] Cuchillo (Harry Hole 12)
    Autor: Jo Nesbo
    Género: Fiction
    Año: 2019 | Páginas: 656
---
```

Consulta la API de Google Books y mostrar en pantalla una lista de libros que coincidan con una búsqueda o con un género, **obtenidos directamente desde internet, sin guardarlos** todavía en la estructura local (data/...).

Opción 1 modo API:

SEGUNDO PARCIAL PROGRAMACIÓN

10

```
hp@DESKTOP-53SCNNN MINGW64 ~\OneDrive\Escritorio\trabajos_practicos\Integrador-Programacion-Parte-2 (main)
$ C:/Users/hp/AppData/Local/Programs/Python/Python313/python.exe c:/Users/hp/OneDrive/Escritorio/trabajos_practicos/Integrador-Programacion-Parte-2/src/main.py

== MENÚ PRINCIPAL (Modo API) ==
1. Buscar y guardar libro (API)
2. Mostrar libro desde la API
3. Cambiar a modo local
4. Salir

Elegí una opción: 1
Ingresa el libro de la API que quieras guardar: ana with an e
⚠️ No se encontraron coincidencias exactas, mostrando resultados válidos de respaldo.

💡 Se encontraron 10 libros similares:

[1] Ana, la de Tejas Verdes 3. Ana, la de la isla - Lucy Maud Montgomery (2021) [196 páginas]
[2] Diario cinematográfico de Márbara Averbach - Márbara Averbach (2025) [786 páginas]
[3] Family Values - Wendy Cope (2011) [83 páginas]
[4] The L.M. Montgomery Reader - Benjamin Lefebvre (2013) [464 páginas]
[5] Foundations of Commutative Rings and Their Modules - Fanggui Wang, Hwankoo Kim (2024) [862 páginas]
[6] Educación para la ciudadanía activa: innovación y prácticas emergentes en los contextos educativos - María Teresa Bejarano Franco (2024) [436 páginas]
[7] Ethics and Time in the Philosophy of History - Natan Elgabisi, Bennett Gilbert (2023) [241 páginas]
[8] L.M. Montgomery and the Matter of Nature(s) - Rita Bode, Jean Mitchell (2018) [305 páginas]
[9] Call Me - P-P Hartnett (2025) [217 páginas]
[10] Connecting Children with Classics - Meagan Lacy, Pauline Dewan (2018) [417 páginas]

Seleccioná el número del libro que querés guardar (0 para cancelar): 10
📁 Carpeta verificada o creada: c:\Users\hp\OneDrive\Escritorio\trabajos_practicos\Integrador-Programacion-Parte-2\data\Language Arts & Disciplines\Meagan Lacy, Pauline Dewan\2018

Libro 'Connecting Children with Classics' guardado jerárquicamente en c:\Users\hp\OneDrive\Escritorio\trabajos_practicos\Integrador-Programacion-Parte-2\data\Language Arts & Disciplines\Meagan Lacy, Pauline Dewan\2018\items.csv

✅ Libro 'Connecting Children with Classics' guardado correctamente desde la API.
```

La opción **1** busca un libro y **lo guarda** en nuestro sistema jerárquico local (creando las carpetas género/autor/año y el items.csv correspondiente).

Opción 2 modo local:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
hp@DESKTOP-53SCNNI MINGW64 ~/OneDrive/Escritorio/trabajos_practicos/Integrador-Programacion-Parte-2 (main)
$ C:/Users/hp/AppData/Local/Programs/Python/Python313/python.exe c:/Users/hp/OneDrive/Escritorio/trabajos_practicos/Integrador-Programacion-Parte-2/src/main.py
Elegí una opción: 2

LISTADO GLOBAL DE LIBROS

¿Desea filtrar la lista? (titulo/autor/genero/no): no
---
[1] ID: Biography & Autobiography/Ana Frank/El diario de Ana Frank
    Título: El diario de Ana Frank (245 pgs)
    Ubicación: Biography & Autobiography / Ana Frank / 2025
---
[2] ID: Body, Mind & Spirit/Bernabé Tierno/El amor que es vida
    Título: El amor que es vida (223 pgs)
    Ubicación: Body, Mind & Spirit / Bernabé Tierno / 2015
---
[3] ID: ciencia ficción/Isaac asimov/Fundación
    Título: Fundación (256 pgs)
    Ubicación: ciencia ficción / Isaac asimov / 1951
---
[4] ID: Juvenile Fiction/J.K. Rowling/Harry Potter y la piedra filosofal
    Título: Harry Potter y la piedra filosofal (318 pgs)
    Ubicación: Juvenile Fiction / J.K. Rowling / 2015
---
[5] ID: Language Arts & Disciplines/Meagan Lacy, Pauline Dewan/Connecting Children with Classics
    Título: Connecting Children with Classics (417 pgs)
    Ubicación: Language Arts & Disciplines / Meagan Lacy, Pauline Dewan / 2018
---
[6] ID: terror/stephen king/Christine
    Título: Christine (450 pgs)
    Ubicación: terror / stephen king / 1987
---
[7] ID: Young Adult Fiction/Erin Doom/Fabricante de lágrimas
    Título: Fabricante de lágrimas (852 pgs)
    Ubicación: Young Adult Fiction / Erin Doom / 2023
---
[8] ID: Young Adult Fiction/Mercedes Ron/Culpa mía (Culpables 1)
    Título: Culpa mía (Culpables 1) (600 pgs)
```

Recorre **toda la estructura de carpetas** (usando la función recursiva

`consolidar_libros_recurcivamente()`

para reunir en una sola lista todos los libros de todos los géneros, autores y años.

Luego pregunta al usuario si quiere **filtrar** los resultados por:

- título
- autor
- genero
- o escribir no para ver todos los libros.
- Muestra los resultados en un formato legible, con su título, cantidad de páginas, y ubicación jerárquica.

Opción 5 modo local:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

hp@DESKTOP-53SCNN ~/OneDrive/Escritorio/trabajos_practicos/Integrador-Programacion-Parte-2 (main)
$ C:/Users/hp/AppData/Local/Programs/Python/Python313/python.exe c:/Users/hp/OneDrive/Escritorio/trabajos_p
2. Mostrar libros locales
3. Modificar libro
4. Eliminar libro
5. Estadísticas
6. Ordenar libros
7. Cambiar a modo API
8. Salir
Elegí una opción: 5

[ESTADÍSTICAS GLOBALES DE LIBROS]

Total de libros registrados: 8
Promedio de páginas por libro: 420.12
Género más frecuente: Young Adult Fiction (2 libros)
Libro con más páginas: 'Fabricante de lágrimas' (852 páginas)

Recuento por género:
- Young Adult Fiction: 2 libros
- Biography & Autobiography: 1 libros
- Body, Mind & Spirit: 1 libros
- ciencia ficción: 1 libros
- Juvenile Fiction: 1 libros
- Language Arts & Disciplines: 1 libros
- terror: 1 libros
```

Lee todos los libros locales igual que mostrar_libros, pero en lugar de mostrarlos uno por uno, calcula:

- Total de libros registrados
- Promedio de páginas por libro
- Cantidad de libros por género
- Luego imprime un resumen estadístico general

Conclusión

El proyecto permitió integrar distintos conceptos clave de la programación estructurada: manejo de archivos, recursividad, modularización, validación de datos, y conexión con servicios externos mediante APIs.

El resultado final es un sistema completo y flexible que permite gestionar información tanto **localmente** como **en línea**, adaptándose a distintos escenarios de uso.

A través de este trabajo, se logró consolidar la lógica necesaria para vincular estructuras de datos jerárquicas con la obtención de información externa, demostrando la importancia de diseñar programas escalables, bien organizados y pensados para el mantenimiento a largo plazo.

Referencias

<https://chatgpt.com/> (Correcciones ortográficas, información)

Requests (librería de Python que sirve para hacer solicitudes HTTP)

API Google Books https://developers.google.com/books/docs/v1/getting_started

Tecnicatura Universitaria en Programación. Universidad Tecnológica Nacional.