

TRABAJO INTEGRADOR(TPI) - PROGRAMACIÓN I

Alumnos

Mariano Chirino.

Brisa Muñoz.

Ivana Gutierrez.

Julieta Chavez.

Rayssa Alejo.

Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional (UTN -

Facultad Regional Mendoza (FRM)).

PROGRAMACIÓN I

Docente Titular

Magni Negri, Gerardo Luis

06 de [Noviembre] de 2025

Marco Teórico:

El proyecto fue desarrollado en Python 3, aplicando estructuras de datos, programación modular y manejo de archivos CSV.

A continuación, se describen los conceptos más importantes utilizados en el código.

Clases y Programación Orientada a Objetos (POO):

Se definió la clase Paisdelmundo, que representa cada país con sus atributos: código, nombre, población, superficie y continente.

El uso de clases permite organizar la información y comportamientos relacionados en una sola estructura.

Según la documentación oficial de Python (Python Software Foundation, 2024), una clase actúa como un modelo que define cómo crear objetos con determinadas propiedades y métodos.

Esto mejora la legibilidad del código y favorece la reutilización.

Listas:

Las listas son estructuras de datos que permiten almacenar varios elementos en orden y son mutables.

En el proyecto se usó una lista (lista_paises) para guardar todos los objetos creados de la clase Paisdelmundo.

Gracias a esta estructura fue posible recorrer, filtrar y ordenar los países según distintos criterios.

Python ofrece diversas funciones para manipular listas de forma sencilla y eficiente.

Diccionarios:

Los diccionarios almacenan datos en pares clave: valor.

En este caso se utilizó un diccionario (dicc_países) para acceder rápidamente a un país según su código numérico.

Esta estructura permite búsquedas rápidas y evita duplicados, ya que cada clave es única.

Funciones y modularización

El código se organizó en funciones que cumplen tareas específicas: leer el archivo CSV, mostrar, buscar, filtrar, ordenar o agregar países, entre otras.

Este enfoque de “una función = una responsabilidad” mejora la claridad y facilita el mantenimiento.

Además, el uso de `if __name__ == "__main__":` define el punto de entrada del programa, siguiendo las buenas prácticas recomendadas por la comunidad de Python.

Estructuras condicionales y repetitivas:

Se emplearon condicionales `if`, `elif`, `else` para controlar el flujo del programa y ejecutar acciones según la opción seleccionada por el usuario.

También se usaron bucles `for` y `while` para recorrer listas y mantener activo el menú principal, lo que permite interacción continua hasta que el usuario decida salir.

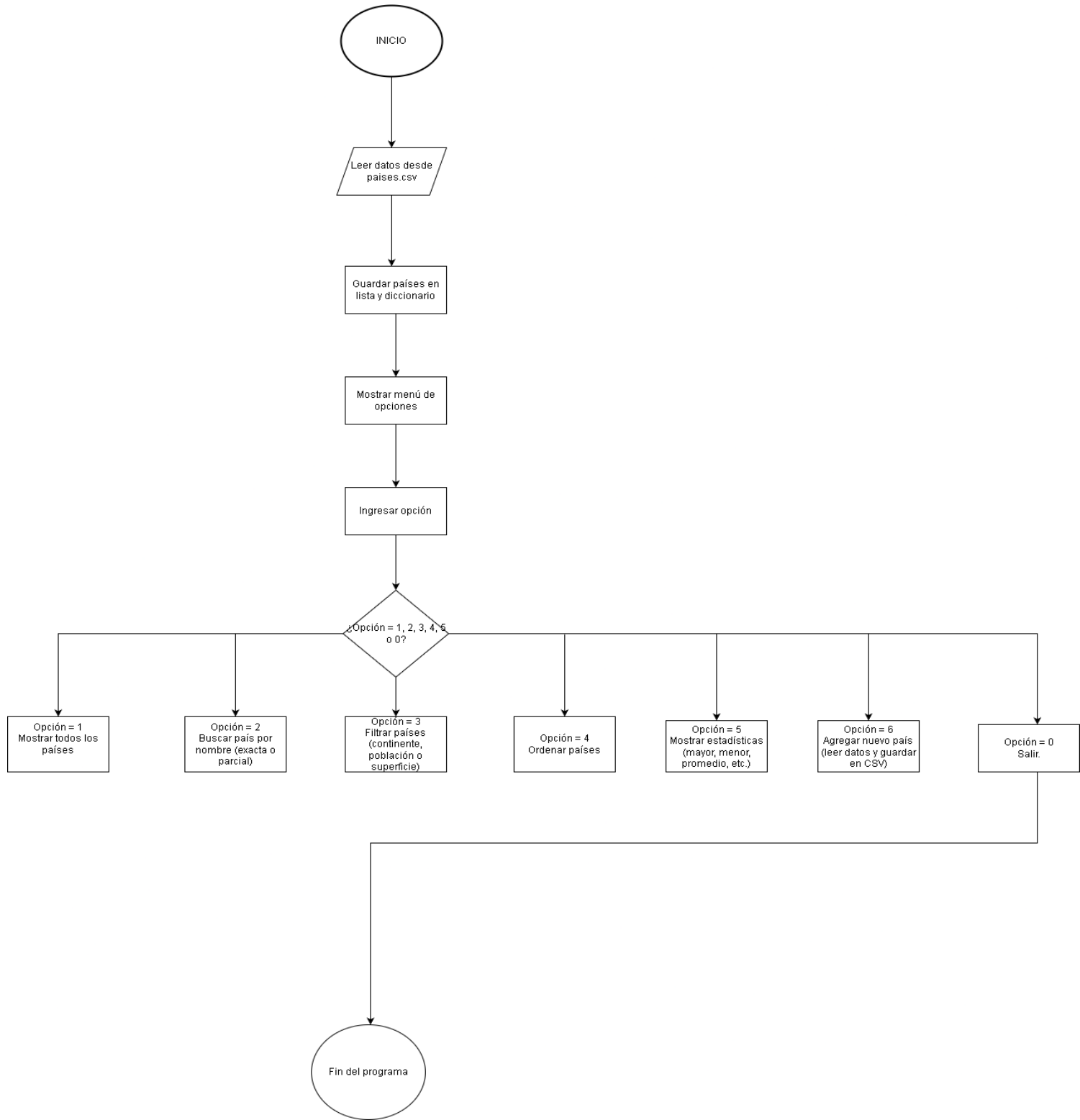
Archivos CSV:

El programa utiliza el módulo csv de la biblioteca estándar de Python para leer y escribir archivos en formato Comma-Separated Values.

Esta herramienta permite guardar y recuperar datos estructurados de manera sencilla, manteniendo persistencia entre ejecuciones del programa.

El método DictReader se empleó para interpretar cada fila del archivo como un diccionario, facilitando la conversión de datos en objetos Paisdelmundo.

Diseño del caso práctico:



El programa carga datos desde `países.csv` y los organiza en listas y diccionarios para optimizar búsquedas.

A través de un menú principal, el usuario puede mostrar, buscar, filtrar, ordenar, agregar países o consultar estadísticas.

El sistema se ejecuta en bucle hasta elegir “Salir”, guardando los cambios al finalizar.

Metodología utilizada:

Se aplicó una metodología progresiva: análisis, diseño, implementación, pruebas y documentación.

El proyecto se desarrolló en **Python 3.x** con **VS Code**, empleando modularización, manejo de archivos CSV y validaciones.

Se realizaron pruebas con distintos datos y se gestionó el control de versiones con **GitHub**.

Resultados y Evaluación:

El sistema logró cumplir con los requerimientos planteados, incluyendo lectura desde CSV, búsquedas, filtros, ordenamientos y estadísticas.

Las pruebas confirmaron su correcto funcionamiento y validación de datos.

Se destacó la modularización y la estabilidad del programa, pese a los desafíos iniciales en la gestión de filtros y estructuras.

Conclusión:

El desarrollo del proyecto permitió integrar los contenidos vistos en la materia, aplicando estructuras de datos, funciones y manejo de archivos en un caso práctico.

Se consolidó la importancia de la planificación, la modularización y la validación de datos para lograr un programa funcional y mantenible.

El trabajo final refleja la aplicación de los conocimientos adquiridos y la experiencia de trabajo en equipo.

Biografía:

- Python Software Foundation. (2024). Python 3 Documentation. <https://docs.python.org/3/>
- Sweigart, A. (2019). *Automate the Boring Stuff with Python*. No Starch Press.
- McKinney, W. (2022). *Python for Data Analysis*. O'Reilly Media.
- Severance, C. R. (2016). *Python para todos: Explorando la información con Python 3*. Charles R. Severance. Recuperado de <https://www.py4e.com/book>