

Trabajo Práctico Integrador

Proyecto: Gestor de Países en Python

Alumnos

Santino Cardenas y Nicolas Carchano

Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional.

Programacion I

Informe - Proyecto Países

En este informe se detalla el desarrollo del proyecto de gestión de países en Python. El programa trabaja con información guardada en un archivo CSV y permite buscar, filtrar, ordenar y analizar datos de diferentes países. El objetivo principal fue aplicar los conceptos aprendidos en la materia de Programación 1, como listas, diccionarios, funciones, condicionales, manejo de archivos y modularidad.

Listas

Se utilizaron listas para almacenar los países que se leen desde el archivo CSV. Cada elemento de la lista es un diccionario que contiene la información de un país. Esto permite recorrer, filtrar o mostrar los países según distintos criterios, como la población o el continente. Las listas son estructuras dinámicas que facilitan el manejo de grandes volúmenes de datos en memoria.

Diccionarios

Cada país se guarda en un diccionario con claves como 'nombre', 'población', 'superficie' y 'continente'. Gracias a esto, se puede acceder fácilmente a los valores mediante sus claves, haciendo que el código sea más claro, legible y ordenado. Además, los diccionarios permiten una representación estructurada de los datos, ideal para trabajar con registros como los de un archivo CSV.

Funciones

El código se divide en módulos independientes, cada uno con sus funciones específicas. Por ejemplo, existen funciones para leer el archivo CSV, buscar países, filtrarlos, ordenarlos y calcular estadísticas. Esta organización modular mejora la legibilidad, facilita el mantenimiento y permite reutilizar código en diferentes partes del programa.

Condicionales

Los condicionales se utilizan para tomar decisiones dentro del programa. Por ejemplo, se validan los valores ingresados por el usuario y se decide qué acción ejecutar según la opción seleccionada en el menú. Además, se emplea la estructura moderna 'match-case' de Python 3.10, que permite una sintaxis más clara para manejar múltiples opciones de manera ordenada y eficiente.

Ordenamientos

Para ordenar los países se utilizaron funciones personalizadas en el módulo 'ordenamientos.py'. Estas funciones hacen uso de 'sorted()' junto con claves definidas para cada criterio (nombre, población o superficie), y permiten elegir si el orden será ascendente o descendente. Esto demuestra el uso eficiente de funciones de ordenamiento nativas de Python junto con programación modular.

Estadísticas básicas

El programa calcula estadísticas simples como el país con mayor y menor población, el promedio de población, el promedio de superficie y la cantidad de países por continente. Para los cálculos de promedio se emplea el módulo 'statistics' de Python. Estos datos permiten obtener una visión general de la información almacenada en el archivo.

Archivos CSV

El archivo 'paises.csv' actúa como base de datos del programa. Se utiliza el módulo 'csv' para leer y escribir datos de forma estructurada. Si el archivo no existe, el programa crea automáticamente uno nuevo con ejemplos iniciales. Además, se implementan validaciones para evitar errores comunes, como valores vacíos o tipos de datos incorrectos, asegurando que los registros cargados sean consistentes y útiles para las operaciones del sistema.

Flujo de operaciones principales

1. El programa inicia ejecutando 'main.py', que lee el archivo 'paises.csv' y carga los datos.

2. Se muestra un menú principal con las siguientes opciones:

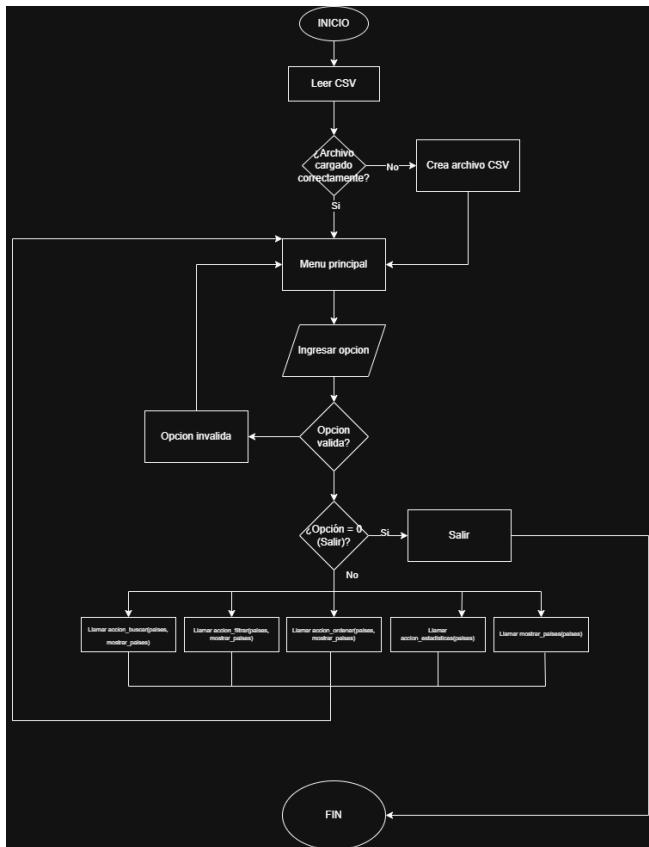
- Buscar país por nombre
- Filtrar países (por continente, población o superficie)
- Ordenar países
- Ver estadísticas
- Mostrar todos los países
- Salir

3. Cada opción del menú llama a una función del módulo 'acciones_menu.py', la cual coordina las funciones de los otros módulos.

4. Los resultados se muestran por consola con un formato claro y ordenado.

5. El programa se mantiene en ejecución hasta que el usuario elige la opción 'Salir'.

Diagrama de Flujo principal



Conclusión

El proyecto permitió integrar todos los contenidos fundamentales de la materia: manejo de listas, diccionarios, condicionales, funciones, modularización, lectura y validación de archivos CSV, y cálculo de estadísticas básicas. Además, se trabajó con una estructura escalable y buenas prácticas de programación, logrando un programa funcional, claro y adaptable a nuevos requerimientos.

Referencias

Tecnicatura Universitaria en Programación. Programacion I Unidad 1: Estructuras Secuenciales

Tecnicatura Universitaria en Programación. Programacion I Unidad 2: Trabajo Colaborativo

Tecnicatura Universitaria en Programación. Programacion I Unidad 3: Condicionales

Tecnicatura Universitaria en Programación. Programacion I Unidad 4: Repetitivas

Tecnicatura Universitaria en Programación. Programacion I Unidad 5: Listas

Tecnicatura Universitaria en Programación. Programacion I Unidad 6: Funciones

Tecnicatura Universitaria en Programación. Programacion I Unidad 7: Estructuras de Datos

Complejas

Tecnicatura Universitaria en Programación. Programacion I Unidad 8: Manejo de Archivos