**Trabajo Practico Integrador**

**Gestión de Datos de Países en Python**

Pablo Nogueira

Bruno Tello

**Objetivo**

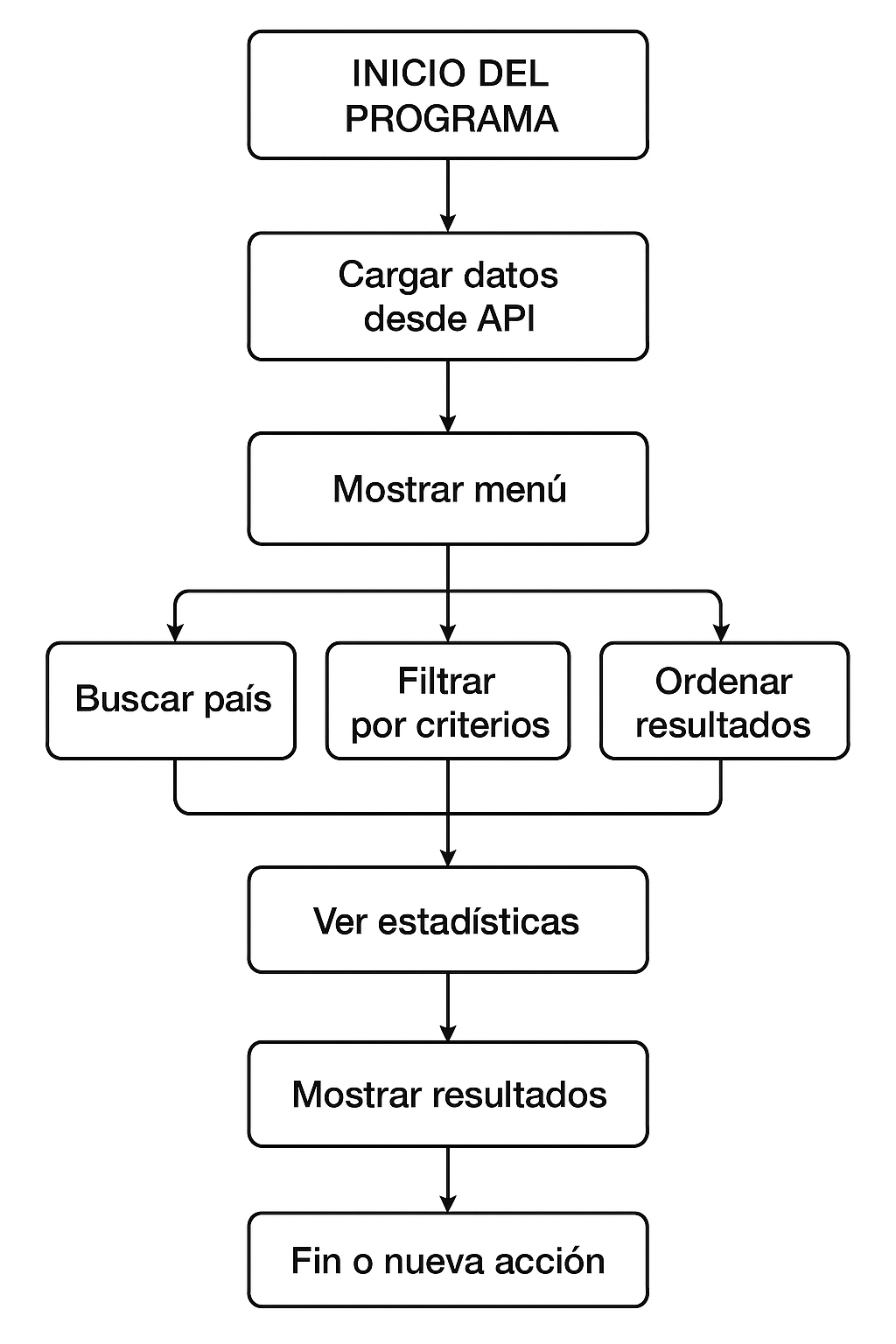
El objetivo principal del proyecto es desarrollar un programa modular en Python que permita gestionar información de países obtenida desde una API pública (RestCountries).  
El sistema ofrece un menú interactivo donde el usuario puede buscar países, filtrados por diferentes criterios (continente, población, superficie), ordenarlos y obtener estadísticas generales.  
Además, el programa genera automáticamente un archivo CSV con los datos obtenidos desde la API y muestra los resultados de forma paginada, mejorando la legibilidad cuando la cantidad de países es extensa.

El proyecto implementa buenas prácticas de programación estructurada, manejo de excepciones, modularizacion y uso de Docker para garantizar portabilidad, independencia del entorno y fácil ejecución.

**Elementos del código**

* **Listas**
  + Estructuras ordenadas que almacenan múltiples elementos.
  + Se usan para guardar el conjunto de países descargados de la API.
* **Diccionarios**
  + Estructuras con pares clave-valor, ideales para representar información compleja.
* **Funciones**
  + Permiten dividir el código en bloques reutilizables.
  + Cada modulo contiene funciones con una única responsabilidad (por ejemplo, buscar, filtrar, ordenar o calcular estadísticas).
* **Condicionales**
  + Controlan el flujo del programa y permiten responder a las selecciones del usuario en el menú principal.
* **Ordenamientos**
  + Se aplican con funciones y expresiones para organizar los países según nombre, población o superficie (de forma ascendente o descendente).
* **Estadísticas**
  + Se utilizan funciones como max(), min() y operaciones aritméticas para obtener indicadores como país com mayor o menor población, promedios de población y superficie, y cantidad de países por continente.
* **Archivos CSV**
  + El sistema guarda automáticamente todos los datos obtenidos en un archivo países.csv, asegurando persistencia local y cumpliendo con el manejo de archivos estructurados.
* **Paginación**
* Se implementa una función que muestra los resultados en bloques de diez países por pantalla, con opciones para avanzar, retroceder o salir.
* Esto mejora la experiencia de usuario y evita desbordar la consola con listados extensos.
* **Manejo de errores**
  + A través de estructuras try/except, se controlan errores comunes como valores inválidos o fallos de conexión con la API.

**Estructura del flujo del programa**



**Conclusión**

El proyecto demuestra el uso de estructuras de datos eficientes y módulos organizados en Python para la gestión y análisis de información geográfica.

Gracias a la división por archivos (API, búsquedas, filtros, ordenamientos, estadísticas, validaciones, etc.), el código resulta mantenible y escalable.

La paginación, el guardado automático en CSV y la ejecución mediante Docker refuerzan la funcionalidad del sistema, garantizando una experiencia mas completa, portable y profesional.