

# Trabajo Práctico Integrador

## Programación 1

---

**Carrera:** Tecnicatura Universitaria en Programación modalidad a distancia

**Materia:** Programación 1.

**Tema:** Gestión de datos de países.

**Grupo:** 199

**Integrantes:** Tempio Nicolás, Nadaf Facundo.

**Link github:**

Facundo Nadaf:

[https://github.com/cufa03/UTN-TUPaDProgramacion1/blob/main/Trabajo\\_practico\\_integrador-199/README.md](https://github.com/cufa03/UTN-TUPaDProgramacion1/blob/main/Trabajo_practico_integrador-199/README.md)

Nicolas Tempio:

**Link video:** <https://youtube.com/watch?v=Z6nDc5PVNBE&feature=youtu.be>

# Procesamiento y análisis de datos estructurados utilizando Python

## Marco teórico

### Listas

Una lista es una estructura de datos que permite almacenar varios elementos de forma ordenada. Sus elementos son accesibles por índice.

```
Lista = [elemento1, casa, elemento2]
```

Fuente: Python Docs – <https://docs.python.org/3/tutorial/datastructures.html#lists>

### Diccionarios

Los **diccionarios** almacenan datos en pares **clave-valor**, lo que permite acceder rápidamente a la información mediante una clave.

```
Persona = { nombre: "Facundo"
```

```
Sexo: Masculino
```

```
}
```

Fuente: Python Docs – <https://docs.python.org/3/tutorial/datastructures.html#dictionaries>

### Funciones

Las funciones permiten dividir el programa en bloques lógicos, facilitando la reutilización y organización del código.

Fuente: Python Docs – <https://docs.python.org/3/tutorial/controlflow.html#defineing-functions>

### Condicionales

Las estructuras **if/elif/else** permiten tomar decisiones en base a condiciones lógicas.

Fuente: Python Docs – <https://docs.python.org/3/tutorial/controlflow.html#if-statements>

## Ordenamientos

El método **sorted()** y la función key permiten ordenar listas por criterios específicos (ej: población, superficie).

Fuente: Python Docs – <https://docs.python.org/3/howto/sorting.html>

## Estadística básica

Se aplicaron operaciones como máximo, mínimo y promedio para obtener indicadores útiles a partir de los datos.

## Archivo CSV

Un archivo **CSV** almacena datos en texto separados por comas. Se utilizó para cargar la información inicial del programa.

## Objetivo del trabajo

El objetivo del trabajo fue realizar un programa en Python, de manera colaborativa, que permita leer la información de países desde un archivo CSV, almacenarla en estructuras de datos adecuadas y permitir a los usuarios realizar consultas, ordenamientos y estadísticas básicas para analizar la información de manera eficiente.

Fue posible llevarlo a cabo implementando los conocimientos previamente aprendidos en Programación 1, lo que es de gran utilidad para afianzar conocimiento e implementar la teoría en casos prácticos y más cercanos a la realidad.

## Diseño del caso práctico

Antes de comenzar con la implementación, se realizó una planificación para definir cómo se estructuraría el programa, qué datos se manejaron y cómo se organizaría el flujo de operaciones. Este diseño permitió establecer claramente la lógica necesaria para hacerlo.

## Datos a utilizar

El programa trabajará con información de países. Cada país tiene los siguientes atributos:

Campo	Tipo de dato	Descripción
nombre	texto	Nombre del país
población	entero	Número total de habitantes
superficie	entero	Tamaño territorial en KM
continente	texto	Continente al que pertenece

La información se obtendrá desde un archivo CSV

## Representación de los datos

- **Lista** para representar almacenar todos los países.

- lista\_paises = [pais1, pais2, pais3, ...]

- **Diccionario** para representar cada país.

- país = {

- "nombre": "Argentina",

- "poblacion": 45000000,

- "superficie": 2780000,

- "continente": "América"

- }

Obteniendo finalmente una lista de diccionarios.

Esta decisión se tomó porque:

- Los diccionarios permiten acceder a los valores por nombre de campo, facilitando el trabajo.
- La lista permite recorrer todos los países para hacer comparaciones y cálculos.

## **Operaciones necesarias**

Se identificaron las operaciones que el programa deberá realizar:

1. Cargar los datos desde el archivo CSV a estructuras internas.
2. Mostrar los países de forma organizada.
3. Filtrar países según ciertos criterios (por ejemplo, por continente).
4. Ordenar los países por población o superficie.
5. Realizar estadísticas básicas, como:
  - Encontrar el país con mayor población.
  - Encontrar el país con menor superficie.
  - Calcular promedios si es necesario.

## Interacción con el usuario

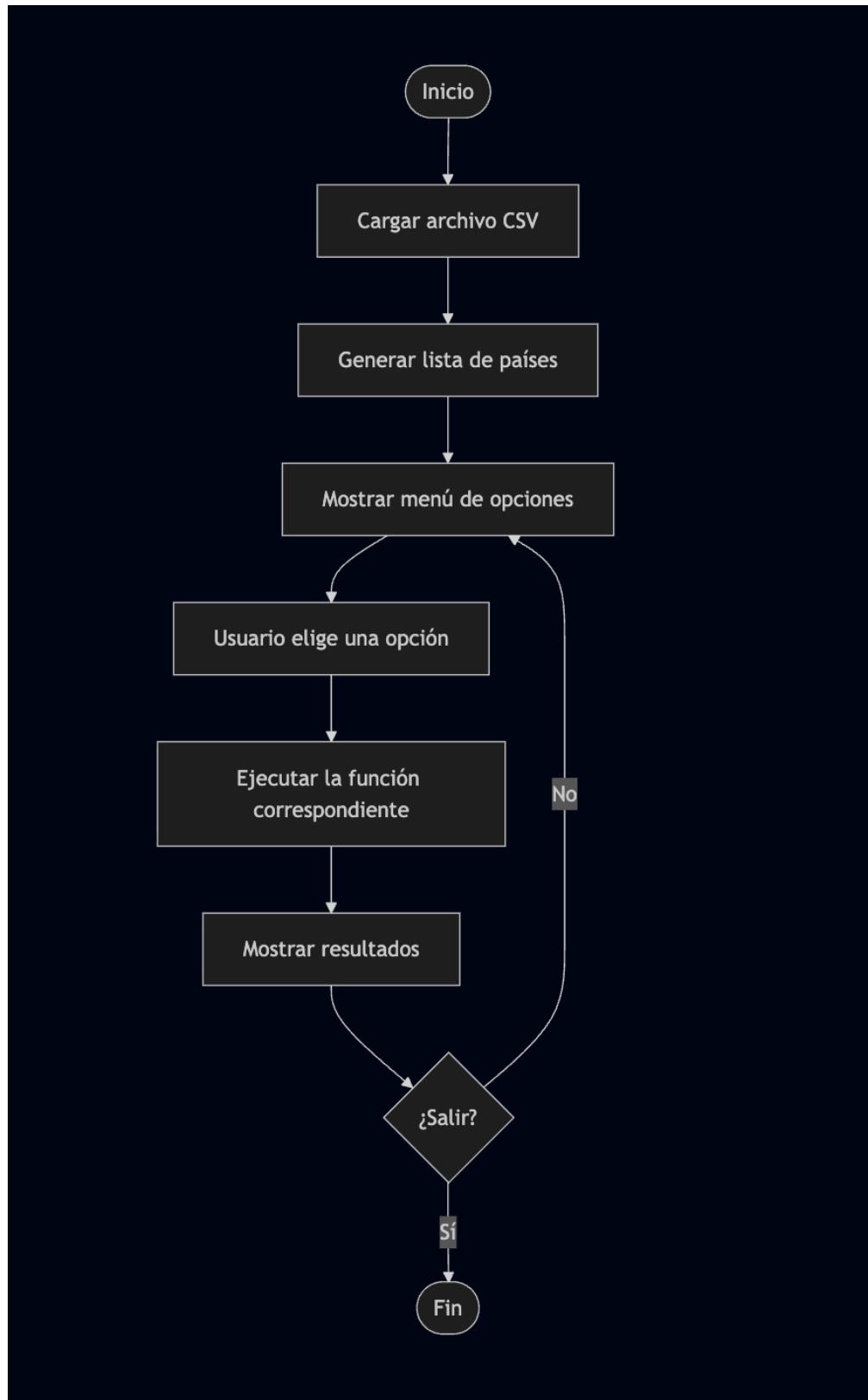
El programa ofrecerá un **menú de opciones**, donde el usuario puede elegir qué acción realizar.

El menú se repetirá hasta que el usuario decida salir.

Menú:

1. Mostrar todos los países
2. Buscar países por continente
3. País con mayor población
4. País con menor superficie
5. Ordenar países por población
6. Salir

## Flujo general del programa



## **Metodología utilizada**

Para el desarrollo del trabajo se adoptó una metodología por etapas, donde primero se analizaron los datos disponibles y luego se estructuró el programa en funciones que permitieran operar sobre ellos de manera ordenada. El enfoque fue modular, buscando claridad, reutilización de código y facilidad de mantenimiento.

- **Se identificaron los datos necesarios y se definió su estructura.**

Se analizaron los campos presentes en el archivo CSV y se decidió representarlos internamente mediante una lista de diccionarios, lo que permite acceder a la información de cada país de forma clara.

- **Se creó una función para cargar y convertir los datos del CSV.**

Se desarrolló una función responsable de leer el archivo, separar los valores, convertir los datos numéricos y almacenar la información en la estructura definida.

- **Se diseñaron funciones específicas para cada operación solicitada (filtros, estadísticas, ordenamientos).**

Para mantener el código organizado, cada acción del programa (como ordenar, buscar o calcular valores máximos) fue implementada en funciones independientes.

- **Se implementó un menú interactivo para facilitar la navegación.**

Se construyó un menú que permite al usuario elegir qué operación realizar, evitando repetir código y mejorando la interacción.

- **Se probaron los resultados asegurando correcto funcionamiento.**

Se realizaron pruebas de uso con distintos escenarios para verificar la validez de los datos, el correcto comportamiento de las funciones y la claridad de la salida mostrada al usuario.

## **Resultados obtenidos**

A partir del desarrollo realizado, el programa pudo cargar correctamente la información de los países desde el archivo CSV y representar internamente mediante una lista de diccionarios, lo que permitió manipular los datos de manera ordenada y eficiente. La presentación de la información resultó clara y legible, facilitando su interpretación por parte del usuario. Asimismo, las funciones de búsqueda y filtrado permitieron obtener conjuntos de países según distintos criterios, como el continente, sin inconvenientes en la ejecución.

Los cálculos estadísticos, como la identificación del país con mayor población o el país con menor superficie, se realizaron de forma correcta, devolviendo resultados consistentes con los datos disponibles. También se logró ordenar los países según diferentes criterios (nombre, población y superficie), lo cual permitió observar la información desde distintas perspectivas.

Por último, la implementación de un menú interactivo permite una navegación sencilla y continua dentro del programa, permitiendo al usuario ejecutar distintas operaciones sin necesidad de reiniciar el sistema. En conjunto, estos resultados demuestran que el programa cumple con los objetivos planteados y permite analizar la información de manera estructurada, clara y funcional.

## **Conclusiones**

Este proyecto demostró la aplicación práctica de los conceptos de programación, permitiendo procesar y analizar datos en un entorno más realista. La organización de la información se logró eficazmente mediante el uso de listas y diccionarios, mientras que las funciones y estructuras condicionales facilitaron la resolución de problemas complejos al dividirlos en pasos lógicos.

La integración de un archivo CSV fue crucial para trabajar con datos externos, reflejando situaciones comunes en aplicaciones reales que manejan múltiples fuentes de información. Esta experiencia subrayó la importancia de la validación y estructuración de datos.

En resumen, el proyecto demostró cómo el conocimiento teórico puede transformarse en una herramienta funcional capaz de realizar consultas, ordenamientos y análisis estadísticos de manera eficiente. El éxito del programa final resaltó que una planificación estructurada y el uso adecuado de las herramientas del lenguaje son esenciales para abordar problemas concretos con claridad y orden.