

# PROOG

INTEGRADOR  
GESTION DE PAISES

BRAIAN FLORES  
SANTINO NALDINI

# INTRODUCCIÓN

GESTION DE PAISES

El programa desarrollado permite gestionar información de distintos países a partir de un archivo CSV que contiene sus datos principales: nombre, población, superficie y continente.

---

Programación 1 | 2C 2025 – TUPAD

# CONCEPTOS FUNDAMENTALES

- Listas
- Diccionarios
- Funciones
- Condicionales
- Ordenamientos
- Estadísticas básicas
- Manejo de archivos CSV

# LISTAS

Una lista es una estructura de datos que permite almacenar múltiples elementos en una sola variable.

Los elementos pueden ser de distintos tipos (números, cadenas, diccionarios, etc.) y son modificables.

```
paises = []
paises.append({
    "nombre": "Argentina",
    "poblacion": 45376763,
    "superficie": 2780400,
    "continente": "América"
})
```

# DICCIONARIOS

```
país = {  
    "nombre": "Francia",  
    "poblacion": 65273511,  
    "superficie": 551695,  
    "continente": "Europa"  
}
```

Un diccionario almacena información en pares clave : valor.

Cada país se representa como un diccionario:

Los diccionarios facilitan el acceso rápido y ordenado a los datos:

```
print(país["nombre"]) # Muestra "Francia"
```

# FUNCIONES

Las funciones agrupan instrucciones que cumplen una tarea específica, mejorando la organización y reutilización del código.

```
def buscar_pais(paises, nombre):
    encontrados = []
    for pais in paises:
        if nombre.lower() in pais["nombre"].lower():
            encontrados.append(pais)
    return encontrados
```

Ejemplos de funciones del programa:

- leer\_csv() → carga los datos desde el archivo CSV
- buscar\_pais() → busca países por nombre
- filtrar\_por\_continente() → filtra países por continente
- mostrar\_estadisticas() → calcula y muestra promedios

# CONDICIONALES

Las estructuras condicionales (if, elif, else) permiten ejecutar distintos bloques de código según una condición lógica.

ejemplo

```
if opcion == 1:  
    nombre = input("Ingrese nombre:")  
    resultado = buscar_pais(paises, nombre)  
  
elif opcion == 7:  
    print("Cerrando programa")  
  
else:  
    print("Opción no válida")
```

# ORDENAMIENTOS

El programa permite ordenar países por distintos criterios: nombre, población o superficie.

```
def ordenar_paises(paises, clave, descendente=False):
    return sorted(paises, key=lambda x: x[clave], reverse=descendente)
```

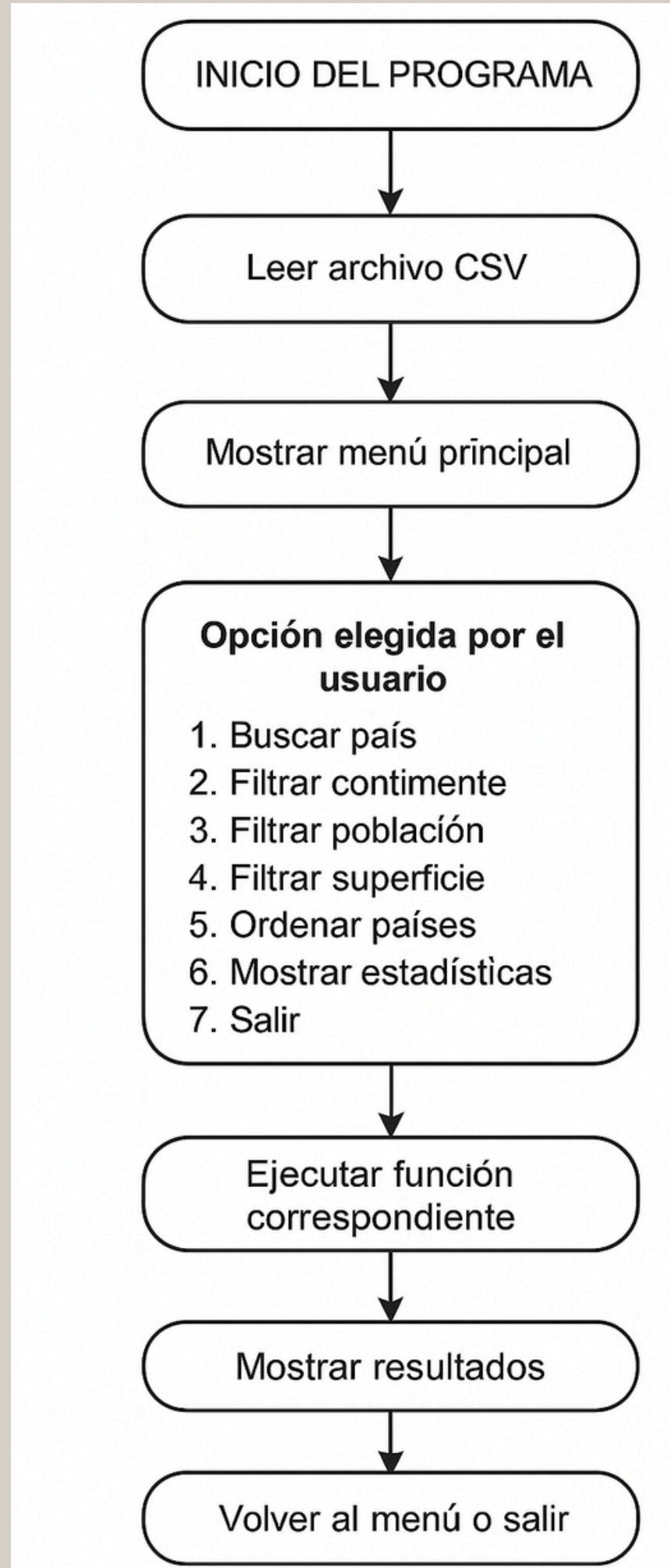
# ESTADÍSTICAS BÁSICAS

```
mayor_pob = max(paises, key=lambda x: x["poblacion"])
prom_pob = sum(p["poblacion"] for p in paises) / len(paises)
```

Se calculan varias medidas estadísticas sobre los datos:

- Máximo y mínimo: país con mayor o menor población
- Promedios: población y superficie promedio
- Conteo por continente: usando diccionarios

# FLUJO DE OPERACIONES PRINCIPALES



Esquema general del programa:

1. Leer archivo CSV (leer\_csv)
2. Mostrar menú principal
3. Ejecutar opción elegida
4. Llamar a la función correspondiente
5. Mostrar resultados
6. Repetir o salir del programa

---

# GESTIÓN DE PAISES

# CONCLUSIÓN

---

Este proyecto integra varios conceptos fundamentales de la programación estructurada en Python. Gracias al uso de funciones, listas, diccionarios, condicionales y manejo de archivos, se logra un programa modular, reutilizable y fácil de mantener.

El análisis de datos mediante estadísticas y ordenamientos mejora su utilidad y comprensión, mostrando cómo la programación puede aplicarse para procesar información real de forma eficiente.