

# ***TP INTEGRADOR***

# ***PROGRAMACIÓN I***

## **INTEGRANTES:**

- Gagliardi, Franco Nicolás
- Quiroga Gabetta, Ramiro

## **1. INTRODUCCIÓN**

Por medio del presente informe, se buscará explayar y explicar el programa realizado en Python, a modo de trabajo práctico integrador de la cátedra Programación I.

El objetivo y búsqueda fue el desarrollo de una aplicación en Python que permita gestionar información sobre países, aplicando listas, diccionarios, funciones, estructuras condicionales y repetitivas, ordenamientos y estadísticas. El sistema debe ser capaz de leer datos desde un archivo CSV, realizar consultas y generar indicadores clave a partir del dataset. Además se agregaron la mayor cantidad de validaciones, funciones modulares y comentarios en el código, con el objetivo de que la lectura de este sea lo más estructurada y amena posible.

---

## **2. CONCEPTOS APLICADOS**

Como se nombró previamente en la introducción, se aplicaron ciertos conceptos vistos durante el cursado de la materia, los cuales serán explicados continuación para poder entender el flujo de trabajo del programa y qué se buscó con la aplicación de los mismos:

### **Estructuras**

- **Condicionales:** Usadas en gran parte para realizar las validaciones pertinentes al control de posibles errores y para manejar las iteraciones utilizadas en los menús principal y secundarios.
- **Repetitivas:** Se utilizó una gran cantidad de bucles o estructuras repetitivas (en su mayoría bucle WHILE) con el fin de: iterar sobre algún dato complejo como listas o diccionarios con el fin de imprimir el dato que corresponda por pantalla (bucle FOR) o para asignar el dato correspondiente a una variable "x" utilizada en el programa. Además se usó en gran parte el bucle WHILE para iterar sobre un menú de opciones mientras cierta condición sea verdadera o no aparezca en una determinada lista de opciones.

## Datos complejos

- **Diccionarios:** Parte clave en la construcción del programa. Su importancia es crucial, ya que los datos del archivo .CSV que contiene los países fueron extraídos y colocados dentro de un diccionario. Además se utilizaron diccionarios para agrupar ciertos datos utilizados en la funcionalidad del programa. Por ejemplo: en la función filtrar se utilizan diccionarios para mostrar los países filtrados según la opción elegida, con la siguiente estructura: diccionario = {nombre: dato}.
- **Listas:** Estructura de gran relevancia dentro del programa. Toda información relevante acerca de los países fue colocada en una lista (dentro de un diccionario, la lista es el valor). Además también se utilizaron listas para agrupar datos relevantes y luego mostrarlos de forma ordenada (por ejemplo: la lista `paises_en_rango = []` utilizada en la función filtrar para agrupar países en cierto rango de población o superficie).

## Funciones

- El código fue dividido y modularizado mediante el uso de funciones, en las que cada una realiza una tarea distinta, en esta ocasión se utilizaron n funciones principales, las cuáles son:
  - ➔ Buscar un país por nombre (coincidencia parcial o exacta).
  - ➔ Filtrar países por: continente, rango de población o superficie.
  - ➔ Ordenar países por: nombre, población, superficie (ascendente o descendente).
  - ➔ Mostrar estadísticas: País con mayor y menor población, promedio de población, promedio de superficie y cantidad de países por continente.
  - ➔ Agregar países
  - ➔ Quitar países
  - ➔ Actualizar diccionario: función que actualiza el diccionario principal en base al archivo .CSV.
  - ➔ Validar archivo: función que valida el archivo paises.csv, es decir, maneja cualquier tipo de posible falla o error.

## **Archivos .CSV**

- Utilizado para guardar todo país que se ingrese, el flujo de datos que ingresa, egresa o se modifica es manejado con el diccionario paises{} su estructura es la siguiente:

nombre,población,superficie,continente  
Argentina,45376763,2780400,América  
Japón,125800000,377975,Asia  
Brasil,213993437,8515767,América  
Alemania,83149300,357022,Europa