

# Marco Teórico - Trabajo Práctico Integrador

El presente trabajo práctico integrador tiene como objetivo aplicar los fundamentos de la programación estructurada en Python, abordando el uso de listas, diccionarios, funciones, estructuras condicionales y repetitivas, así como el manejo de archivos CSV y el cálculo de estadísticas básicas. El proyecto busca consolidar los conocimientos adquiridos en la materia Programación I, mediante el desarrollo de un sistema que gestiona información sobre distintos países.

## Listas

Las listas son estructuras de datos ordenadas y mutables que permiten almacenar múltiples elementos en una sola variable. En el trabajo, se utilizan para almacenar el conjunto de países leídos desde el archivo CSV, permitiendo iterar y manipular los registros fácilmente.

## Diccionarios

Los diccionarios son estructuras de datos que almacenan información en pares clave-valor. Cada país se representa como un diccionario, donde las claves son los atributos principales del país (País, Población, Superficie y Continente). Esta estructura facilita el acceso directo y la actualización de los datos.

## Funciones

Las funciones permiten dividir el programa en bloques lógicos que realizan tareas específicas. Cada función tiene una responsabilidad concreta, como validar datos, filtrar países o calcular estadísticas, lo que mejora la legibilidad y la modularización del código.

## Condicionales

Las estructuras condicionales (if, elif, else) permiten que el programa tome decisiones en función de las entradas del usuario o del flujo de ejecución. Se aplican para validar datos, mostrar mensajes y determinar qué acciones ejecutar según la opción seleccionada.

## Estructuras Repetitivas

Los bucles for y while permiten repetir tareas mientras se cumplan determinadas condiciones. En el trabajo, se utilizan para recorrer las listas de países y mantener activo el menú principal hasta que el usuario decida salir.

## Ordenamientos

El ordenamiento de datos se realiza mediante la función `sorted()` con parámetros personalizados (`key` y `reverse`) para definir el criterio de comparación. Se implementaron ordenamientos por nombre, población y superficie, tanto en forma ascendente como descendente.

## Estadísticas Básicas

Se desarrollaron funciones para obtener información relevante, como el país con mayor y menor población, el promedio de población y superficie, y la cantidad de países por

continente. Estas operaciones ayudan a comprender el conjunto de datos de forma más analítica.

## Archivos CSV

El sistema utiliza archivos CSV para la lectura y escritura de datos de forma persistente. Python proporciona el módulo csv, que permite manipular estos archivos de manera sencilla y eficiente. De esta forma, los datos se mantienen entre ejecuciones del programa.

## Conclusión

El desarrollo de este trabajo permitió integrar los principales conceptos de la programación estructurada, demostrando la relación entre la teoría y la práctica. El uso de funciones, estructuras de datos y manejo de archivos fomenta la creación de programas escalables, legibles y mantenibles.

## Fuentes Bibliográficas

1. Python Software Foundation. (2025). The Python Standard Library. <https://docs.python.org/3/>
2. Universidad Tecnológica Nacional. (2025). Material de Cátedra – Programación I.