

# Trabajo Práctico Integrador - Programación I

Profesora Cinthia Rigoni

Integrantes: - Joaquín Godoy

- Ares Martín Ocaña

## Explicación de Conceptos Aplicados

### Listas

El uso primario que reciben las listas en este proyecto es el de almacenar temporalmente los datos del archivo de datos. Esto se debe a que no se puede trabajar con un archivo en modo de lectura y escritura al mismo tiempo, por ende se deben utilizar listas para almacenar la lectura del archivo en un dato que Python pueda manipular y luego usar esas listas para reescribir el archivo con la información actualizada.

### Diccionarios

Similarmente a las listas; el uso de diccionarios se podría aplicar para la manipulación organizada de los datos del archivo. En este proyecto elegimos no utilizarlos y trabajar siempre con listas, aunque se podrían usar los dos tipos.

### Funciones

El uso de funciones fue clave en el desarrollo del producto. Principalmente nos permitió organizar las utilidades del programa en bloques individuales por cada una, lo cual produce un código mucho más organizado. Además, usamos funciones para almacenar código que queríamos usar más de una vez (por ejemplo, para normalizar el texto ingresado por el usuario). Finalmente, dividir todo en funciones permitió compartimentalizar el código en distintos archivos: un archivo principal para mostrar el menú de opciones y ejecutarlas, un archivo que contiene las funciones principales, y un archivo de utilidades para las funciones simples que íbamos reutilizando y que no formaban parte de la lógica principal del programa.

### Condicionales

Las estructuras condicionales se utilizaron para realizar validaciones, frenando la ejecución de una determinada secuencia si no se cumplían las condiciones correctas. También fueron clave al trabajar con listas cuando no se quería manipular todos los elementos de ella; por ejemplo: para mostrar solo los países que coincidieran entera o parcialmente con el texto ingresado por el usuario en *buscar\_pais()*. Finalmente, también se utilizaron para la creación de menús interactivos con distintas opciones.

### Ordenamientos

Se utilizaron funciones nativas de Python para ordenar listas de países y mostrarlas al usuario, como el uso de *max()* y *min()* para mostrar los países de mayor y menor población en *mostrar\_estadisticas()* o *.sort()* para ordenar los países según distintos datos en *ordenar\_paises()*.

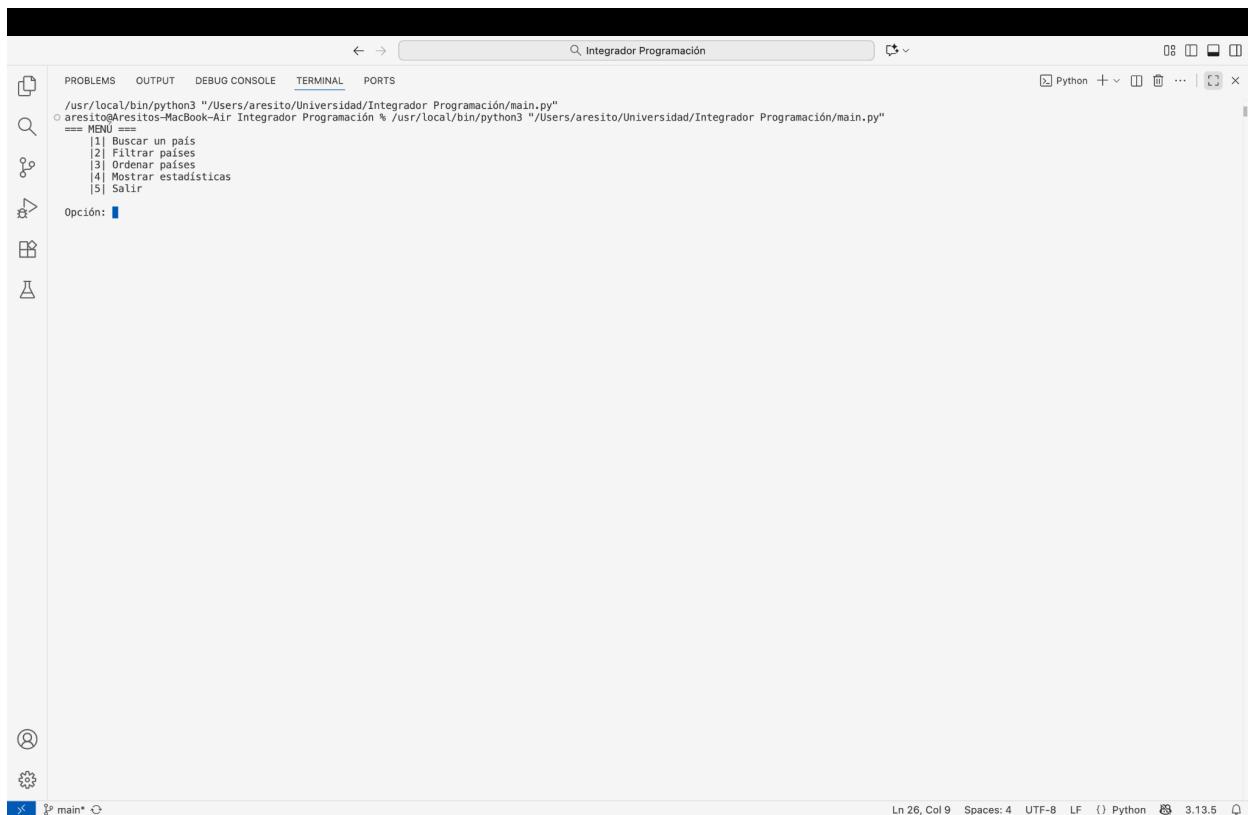
## Estadísticas Básicas

Las estadísticas se aplicaron en el proyecto para obtener información general a partir de los datos del archivo. Su uso principal fue resumir los valores numéricos y mostrar resultados representativos sin necesidad de revisar cada registro individualmente. En la función *mostrar\_estadisticas()*, se emplearon operaciones básicas como el cálculo del país con mayor y menor población mediante *max()* y *min()*, así como el promedio de población y superficie usando *sum()* y *len()*. Además, se contabilizaron los países por continente para mostrar una distribución de frecuencias.

## Archivos CSV

El archivo CSV se utilizó como formato principal para almacenar los datos del proyecto. En Python, se trabajó con ellos mediante el módulo *csv*, que facilita tanto la lectura como la escritura. Se usó el archivo para registrar la información de los países, permitiendo leerla al iniciar el programa, manipularla en memoria y luego actualizar el archivo con los cambios realizados.

## Capturas de Pantalla



Se muestra el menú de opciones

```
/usr/local/bin/python3 "/Users/aresito/Universidad/Integrador Programación/main.py"
aresito@aresitos-MacBook-Air Integrador Programación % /usr/local/bin/python3 "/Users/aresito/Universidad/Integrador Programación/main.py"
== MENÚ ==
[1] Buscar un país
[2] Filtrar países
[3] Ordenar países
[4] Mostrar estadísticas
[5] Salir

Opción: 1

Nombre del país a buscar: arg

ArgeLia
Población: 46,363,000
Superficie: 2,381,741 km^2

Argentina
Población: 46,612,000
Superficie: 2,780,400 km^2

== MENÚ ==
[1] Buscar un país
[2] Filtrar países
[3] Ordenar países
[4] Mostrar estadísticas
[5] Salir

Opción: 
```

Ejemplo de la primera función: se ingresa una búsqueda (“*arg*”) y se devuelven los resultados parciales. Si hubiera una coincidencia exacta, se mostraría solo esa coincidencia.

```
/usr/local/bin/python3 "/Users/aresito/Universidad/Integrador Programación/main.py"
aresito@Aresito-MacBook-Air Integrador Programación % /usr/local/bin/python3 "/Users/aresito/Universidad/Integrador Programación/main.py"
== MENÚ ==
[1] Buscar un país
[2] Filtrar países
[3] Ordenar países
[4] Mostrar estadísticas
[5] Salir

Opción: 2

== MENÚ ==
Filtrar por:
[1] Continente
[2] Rango de población
[3] Rango de superficie
[4] Salir

Opción: 2

Filtrando por población:
Ingrese el límite inferior del rango: 100
Ingrese el límite superior del rango: 1000000

Andorra
Población: 79,824
Superficie: 468 km^2

Antigua y Barbuda
Población: 93,912
Superficie: 442 km^2

Bahamas
Población: 413,000
Superficie: 13,943 km^2

Barbados
Población: 281,000
Superficie: 438 km^2

Belice
Población: 410,000
Superficie: 22,966 km^2

Brunei
Población: 453,000
Superficie: 5,765 km^2

Cabo Verde
Población: 595,000
Superficie: 4,833 km^2

Chipre
Población: 918,000
Superficie: 9,251 km^2

Comoras
Población: 873,000
```

En la opción 2, se ofrece un submenú para elegir por que filtrar. En este ejemplo, se elige filtrar por población, colocando un límite inferior de 100 y uno superior de 1.000.000.

```
/usr/local/bin/python3 "/Users/aresito/Universidad/Integrador Programación/main.py"
aresito@aresito-MacBook-Air Integrador Programación % /usr/local/bin/python3 "/Users/aresito/Universidad/Integrador Programación/main.py"
== MENÚ ==
[1] Buscar un país
[2] Filtrar países
[3] Ordenar países
[4] Mostrar estadísticas
[5] Salir

Opción: 3

== ORDENAR PAISES ==
[1] Ordenar por nombre
[2] Ordenar por población
[3] Ordenar por superficie

Opción: 3
(A) Ascendente / (D) Descendente: a

== RESULTADO ORDENADO ==
Vaticano
Población: 510
Superficie: 0 km^2
Continente: Europa

Mónaco
Población: 39,600
Superficie: 2 km^2
Continente: Europa

Nauru
Población: 12,700
Superficie: 21 km^2
Continente: Oceanía

Tuvalu
Población: 11,300
Superficie: 26 km^2
Continente: Oceanía

San Marino
Población: 34,000
Superficie: 61 km^2
Continente: Europa

Liechtenstein
Población: 39,500
Superficie: 160 km^2
Continente: Europa

San Cristóbal y Nieves
Población: 48,000
Superficie: 261 km^2
Continente: América

Maldivas
```

En la opción 3, se pueden filtrar países y también se ofrece un submenú de opciones, donde se ve cómo se elige la opción de ordenar por superficie. Luego, se pregunta si se debería mostrar de forma ascendiente o descendiente.

```
/usr/local/bin/python3 "/Users/aresito/Universidad/Integrador Programación/main.py"
aresito@Aresitos-MacBook-Air Integrador Programación % /usr/local/bin/python3 "/Users/aresito/Universidad/Integrador Programación/main.py"
== MENÚ ==
[1] Buscar un país
[2] Filtrar países
[3] Ordenar países
[4] Mostrar estadísticas
[5] Salir

Opción: 4

== ESTADÍSTICAS ==
Mayor población: India (1,438,000,000 habitantes)
Menor población: Vaticano (510 habitantes)
Promedio de población: 42,035,121 habitantes
Promedio de superficie: 700,050 km²

Cantidad de países por continente:
América: 35
Asia: 43
Europa: 44
Oceanía: 12
África: 54

== MENÚ ==
[1] Buscar un país
[2] Filtrar países
[3] Ordenar países
[4] Mostrar estadísticas
[5] Salir

Opción: 
```

La opción 4 muestra estadísticas.

```
/usr/local/bin/python3 "/Users/aresito/Universidad/Integrador Programación/main.py"
aresito@Mac Integrador Programación % /usr/local/bin/python3 "/Users/aresito/Universidad/Integrador Programación/main.py"
== MENÚ ==
[1] Buscar un país
[2] Filtrar países
[3] Ordenar países
[4] Mostrar estadísticas
[5] Agregar país
[6] Salir

Opción: 5

== AGREGAR NUEVO PAÍS ==
Nombre del país: Imperio Godoycruceño
Población: 9999999
Superficie (km²): 9999999
Continente: América

País añadido exitosamente.

== MENÚ ==
[1] Buscar un país
[2] Filtrar países
[3] Ordenar países
[4] Mostrar estadísticas
[5] Agregar país
[6] Salir

Opción: 
```

La opción 5, añadida por nosotros, permite agregar un país, llenando todos los campos de información necesarios. Si todos los campos son válidos, se agrega al CSV.

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following text:

```
/usr/local/bin/python3 "/Users/aresito/Universidad/Integrador Programación/main.py"
aresito@Aresitos-MacBook-Air Integrador Programación % /usr/local/bin/python3 "/Users/aresito/Universidad/Integrador Programación/main.py"
    === MENÚ ===
    |[1] Buscar un país
    |[2] Filtrar países
    |[3] Ordenar países
    |[4] Mostrar estadísticas
    |[5] Salir
    Opción: test
    Opción inválida.

    === MENÚ ===
    |[1] Buscar un país
    |[2] Filtrar países
    |[3] Ordenar países
    |[4] Mostrar estadísticas
    |[5] Salir
    Opción: 1
```

Si se elige una opción que no esté en el menú, se le notifica al usuario de que es una opción inválida y se vuelve a mostrar el menú.

## Conclusión grupal

En este trabajo pudimos integrar de manera práctica varios de los contenidos vistos durante la materia, especialmente el uso de funciones, manejo de archivos CSV y organización modular del código. A lo largo del desarrollo fuimos ajustando y mejorando la estructura del programa hasta lograr una versión estable, clara y fácil de mantener. También aprendimos la importancia de normalizar datos, como en el caso de la eliminación de tildes para permitir búsquedas más flexibles, y de separar correctamente la lógica en archivos diferentes para evitar complicaciones a futuro.

El proyecto nos permitió ejercitarse no solo en la parte técnica sino también en la comunicación y la toma de decisiones dentro del grupo: dividir tareas, revisar el código de los demás y buscar soluciones cuando aparecían problemas como imports, formatos o inconsistencias en el CSV. En conjunto, consideramos que el resultado final demuestra un dominio sólido de los conceptos trabajados y una buena capacidad para aplicarlos en un programa completo y funcional.