Trabajo Práctico Integrador

Programación 1

Sistema de Gestión de Países en Python

Alumnos

Domínguez M. Laura (**Comisión 4**)

marlauradominguez@gmail.com

Urrutia Lorenzo (**Comisión 11**)

Lorenzoiurrutia@gmail.com

Docente Titular

Ariel Enferrel - Cinthia Rigoni

Docente Tutor

Luciano Chiroli (Comisión 11) - Ana Mutti (Comisión 4)

Tecnicatura Universitaria en Programación

Universidad Tecnológica Nacional.

11 de Noviembre de 2025

Índice:

Marco Teórico

Los elementos sintácticos, funciones, bloques y herramientas de código en Python utilizados en este trabajo fueron los siguientes:

1. **Lista de Python**

   Las listas son estructuras de datos mutables que permiten almacenar colecciones ordenadas de elementos. En Python, las listas son dinámicas, lo que significa que pueden crecer o reducirse durante la ejecución del programa, y pueden contener elementos de diferentes tipos.

```python
# Creación de lista
paises = []

# Agregar elementos
paises.append({"nombre": "Argentina", "poblacion": 45376763})

# Acceder a elementos
primer_pais = paises[0]

# Recorrer lista
for pais in paises:
    print(pais["nombre"])

# Longitud de la lista
cantidad = len(paises)

```

Aplicación en el Proyecto:

En el sistema de gestión de países, las listas se utilizaron como estructura principal para almacenar la colección completa de países. Cada país es representado como un diccionario dentro de la lista, permitiendo realizar operaciones de búsqueda, filtrado y ordenamiento sobre el conjunto de datos.

Fuente: https://docs.python.org/3.13/tutorial/datastructures.html

2. Diccionarios en Python

Los diccionarios son estructuras de datos que almacenan pares clave-valor. Permiten acceso rápido a los valores mediante claves únicas, implementando una tabla hash internamente. Son mutables y no mantienen un orden específico de inserción.

```python
# Creación de diccionario
pais = {
    "nombre": "Argentina",
    "poblacion": 45376763,
    "superficie": 2780400,
    "continente": "América"
}

# Acceso a valores
nombre = pais["nombre"]
poblacion = pais["poblacion"]

# Modificación de valores
pais["poblacion"] = 45500000

# Verificar existencia de clave
if "continente" in pais:
    print(pais["continente"])

# Recorrer diccionario
for clave, valor in pais.items():
    print(f"{clave}: {valor}")

```

Aplicación en el Proyecto:

Cada país en el sistema se representa como un diccionario con cuatro claves: nombre, población, superficie y continente. Esta estructura permite acceso directo a los atributos de cada país de forma legible y eficiente

Fuente: https://docs.python.org/3.13/tutorial/datastructures.html#dictionaries

3. Funciones en Python

Las funciones son bloques de código reutilizables que realizan una tarea específica. Permiten modularizar el programa dividiéndolo en partes más pequeñas y manejables. Una función puede recibir parámetros de entrada, ejecutar operaciones y retornar valores.

```python
def nombre_funcion(parametro1, parametro2):
    """
    Docstring: descripción de la función
    """
    # Cuerpo de la función
    resultado = parametro1 + parametro2
    return resultado
```

Aplicación en el Proyecto:

El sistema implementa funciones específicas para cada operación: **agregar_pais(), buscar_pais(), filtrar_por_continente(),** etc. Cada función tiene una responsabilidad única, facilitando el mantenimiento y comprensión del código.

Fuente: https://docs.python.org/3.13/tutorial/controlflow.html#defining-functions

4. Módulo CSV

El módulo csv de Python proporciona funcionalidad para leer y escribir archivos CSV (Comma-Separated Values). CSV es un formato de texto plano ampliamente utilizado para almacenar datos tabulares, donde cada línea representa un registro y los campos están separados por comas.

Clases Principales

- **csv.reader:** Lee archivos CSV fila por fila como listas
- **csv.writer:** Escribe datos en formato CSV
- **csv.DictReader**: Lee archivos CSV como diccionarios usando la primera fila como claves
- **csv.DictWriter:** Escribe diccionarios en formato CSV

Parámetros Importantes:

- **mode= "r":** Modo lectura
- **mode= "w"**: Modo escritura (sobrescribe archivo)
- **mode= "a":** Modo de adición, sin sobre escritura
- **encoding=**'utf-8': Codificación para caracteres especiales
- **newline="**: Manejo correcto de saltos de línea en CSV

Aplicación en el proyecto:

El sistema utiliza **csv.DictReader** para cargar los países desde el archivo al iniciar, y **csv.DictWriter** para guardar los cambios después de cada operación que modifica el **dataset.**

Fuente: https://docs.python.org/3.13/library/csv.html

5. Estructuras Condicionales:

Las estructuras condicionales permiten ejecutar diferentes bloques de código según se cumplan o no determinadas condiciones. Python implementa condicionales mediante las palabras clave if, elif y else y match/case.

If / else

```
if condicion1:
    # Código si condicion1 es verdadera
elif condicion2:
    # Código si condicion1 es falsa y condicion2 es verdadera
else:
    # Código si todas las condiciones anteriores son falsas
```

match / case

```
opcion = input("Seleccione una opción: ")

match opcion:
    case "1":
        agregar_pais()
    case "2":
        buscar_pais()
    case _:
        print("Opción inválida")
```

Aplicación en el proyecto:

El menú principal utiliza match/case para direccionar la ejecución según la opción seleccionada. Las validaciones de datos emplean if/else para verificar entradas válidas.

Fuente: https://docs.python.org/3.13/tutorial/controlflow.html#if-statements

6. Operaciones estadísticas

Python proporciona funciones integradas para realizar cálculos estadísticos básicos sobre colecciones de datos.

- ***Funciones Principales:***
  - **max():** Retorna el valor máximo de una secuencia
  - **min():** Retorna el valor mínimo de una secuencia
  - **sum():** Calcula la suma de todos los elementos
  - **len():** Retorna la cantidad de elementos

- ***Cálculos de promedios***

```
# Promedio
cantidad = len(poblaciones)  # 3
promedio = total // cantidad  # 128390066
```

- ***Estadísticas sobre Diccionarios***

```
# Buscar país con mayor población
pais_mayor = paises[0]
for p in paises:
    if p["poblacion"] > pais_mayor["poblacion"]:
        pais_mayor = p

print(pais_mayor["nombre"])  # Brasil
```

- ***Conteo de Frecuencia***

```
continentes = {}

for pais in paises:
    cont = pais["continente"]
    if cont in continentes:
        continentes[cont] += 1
    else:
        continentes[cont] = 1

# Resultado: {"América": 2, "Asia": 1}
```

Aplicación en el Proyecto:

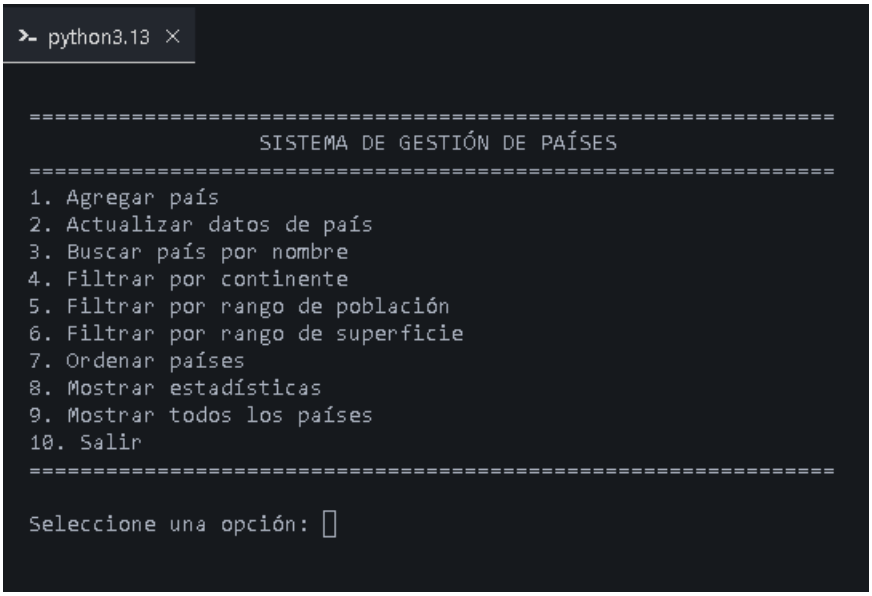
El módulo de estadísticas del sistema calcula:

- País con mayor y menor población (búsqueda manual del máximo/mínimo)
- Promedio de población y superficie (suma total dividida por cantidad)
- Distribución de países por continente (conteo de frecuencias con diccionario)

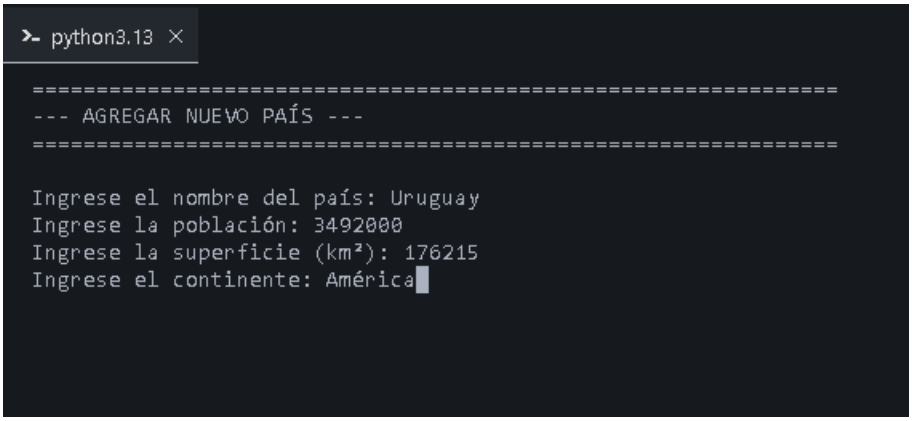
Fuente: https://docs.python.org/3.13/library/functions.html

Capturas del funcionamiento integral del programa

1. **Menú**



2. **Opción 1:** “Agregar país” (no incluido en la base de datos .CSV).

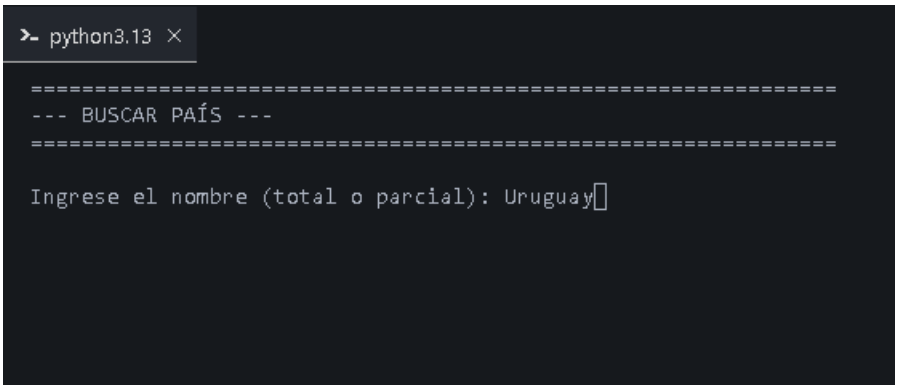


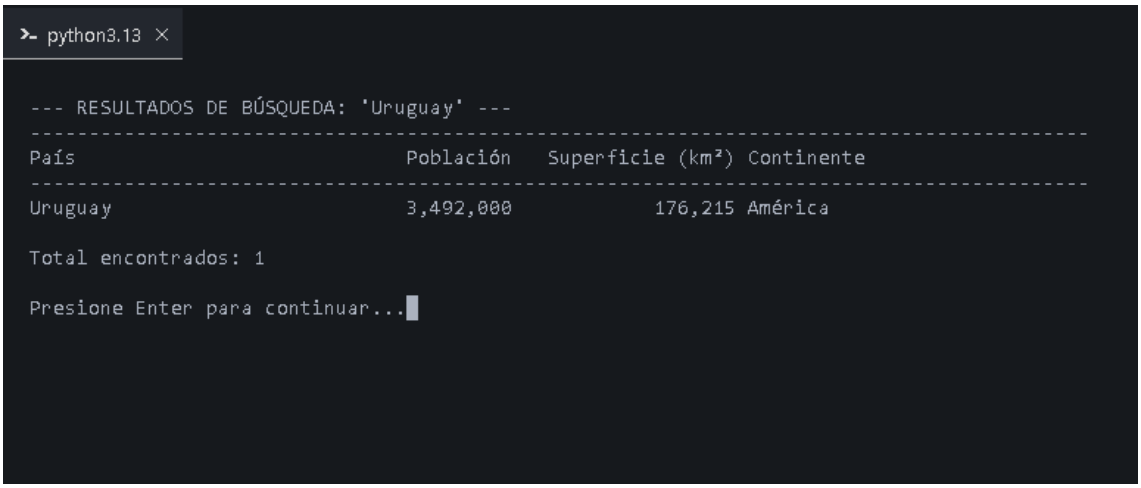
3. **Opción 2:** “Actualizar datos de país” (incluido en la base de datos .CSV).
    a. Superficie de Canadá
        i. 9,985,000,000 Km2 a 9,985,000 Km2

```
python3.13 ×

============================================================
--- ACTUALIZAR DATOS DE PAÍS ---
============================================================

--- LISTADO DE PAÍSES ---
--------------------------------------------------------------------------------
País                          Población   Superficie (km²) Continente
--------------------------------------------------------------------------------
Argentina                    45,376,763          2,780,400 América
Japón                       125,800,000            377,975 Asia
Brasil                      213,993,437          8,515,767 América
Alemania                     83,149,300            357,022 Europa
Australia                    25,687,041          7,692,024 Oceanía
Nigeria                     206,139,589            923,768 África
Canadá                       41,290,000      9,985,000,000 América
Uruguay                       3,492,000            176,215 América

Total: 8 países

Ingrese el nombre del país a actualizar: Canadá
```

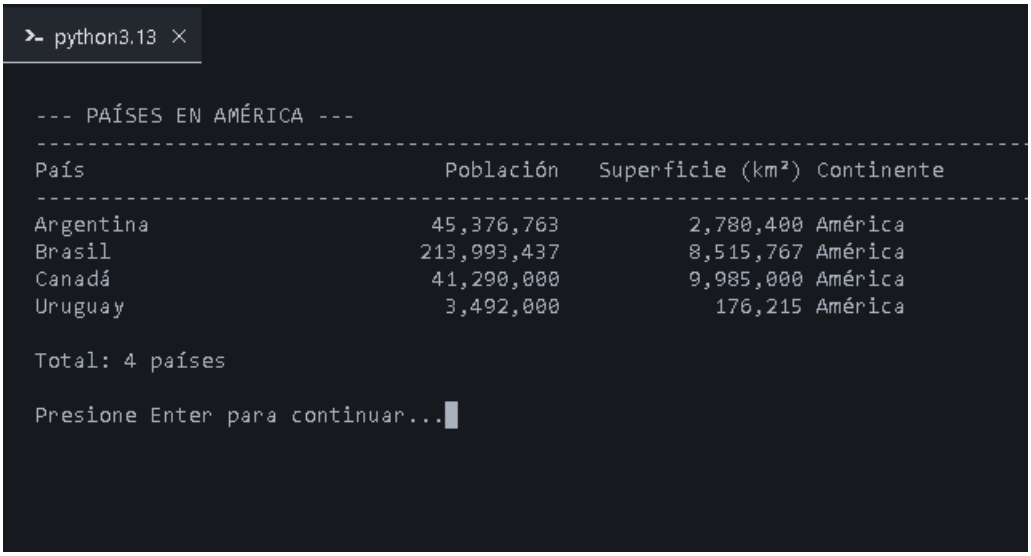
```
python3.13 ×

País seleccionado: Canadá
Población actual: 41,290,000
Superficie actual: 9,985,000,000 km²

1. Actualizar población
2. Actualizar superficie
3. Actualizar ambos

Seleccione una opción (1-3): 2

Nueva superficie (km²): 9985000
```

4. **Opción 3:** “Buscar país por nombre”.

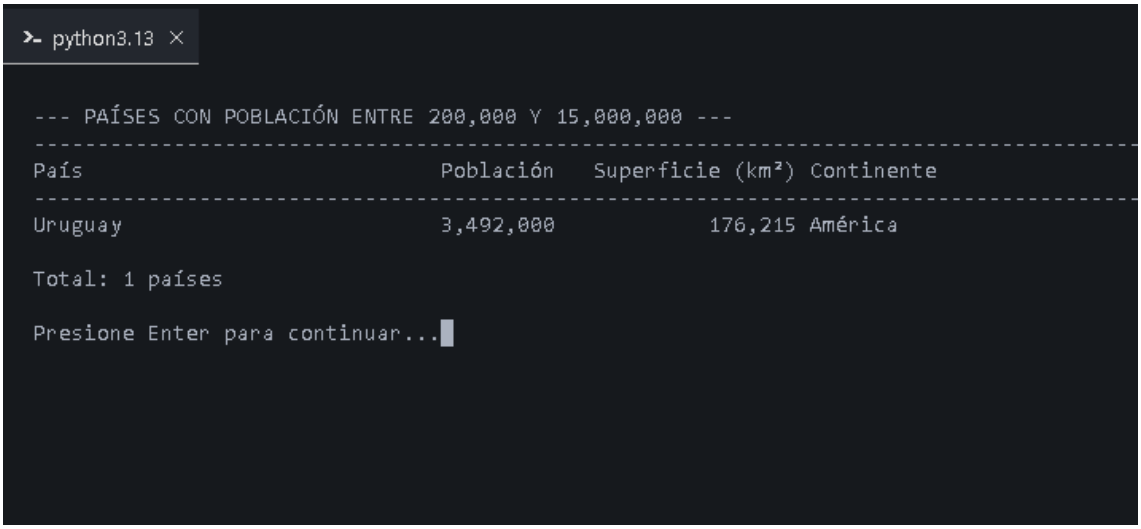




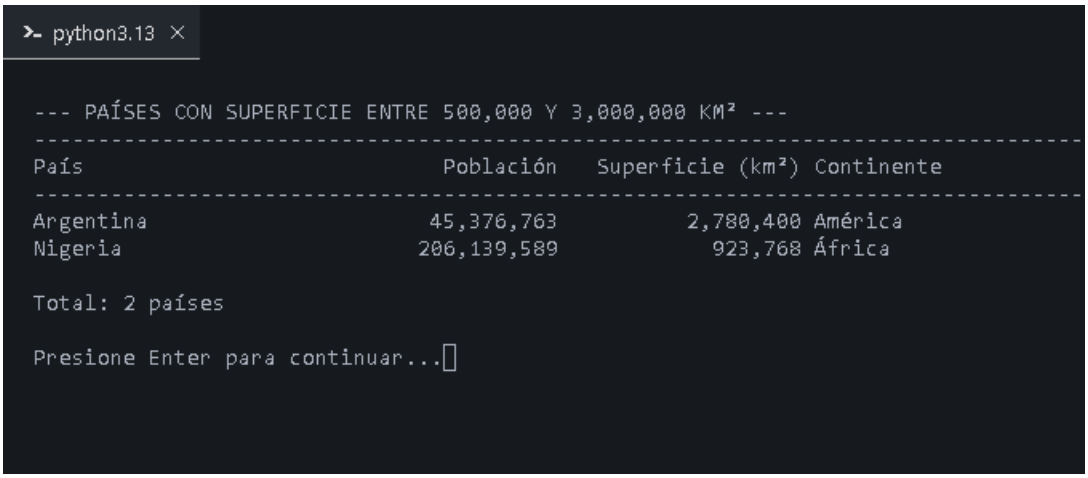
5. **Opción 4:** “Filtrar por continente”.
    a. América



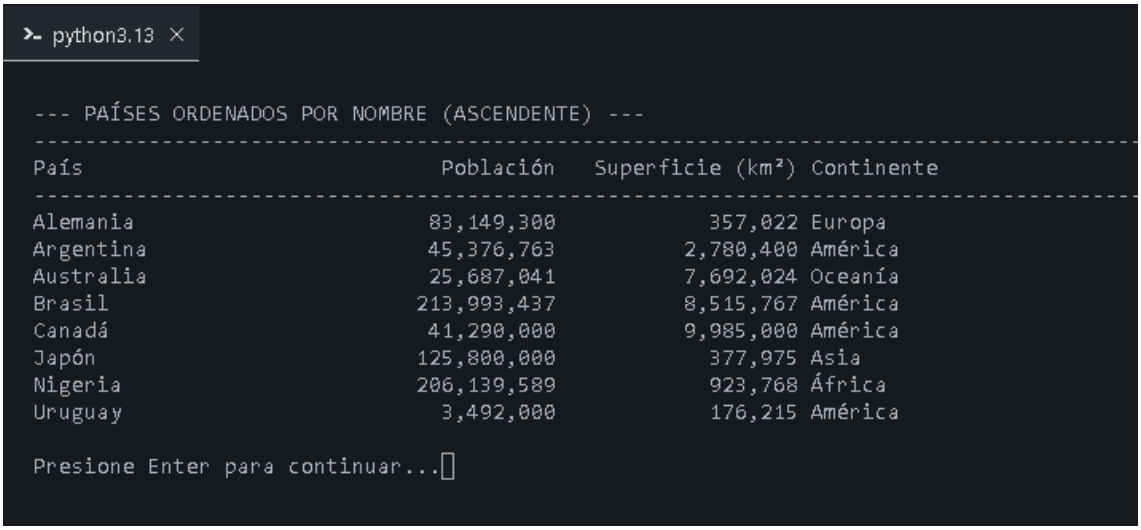
6. **Opción 5:** “Filtrar por rango de población”.
    a. 200,000 a 15,000,000 de habitantes



7. **Opción 6:** “Filtrar por rango de superficie”.
    a. 500,000 Km2 a 3,000,000 Km2

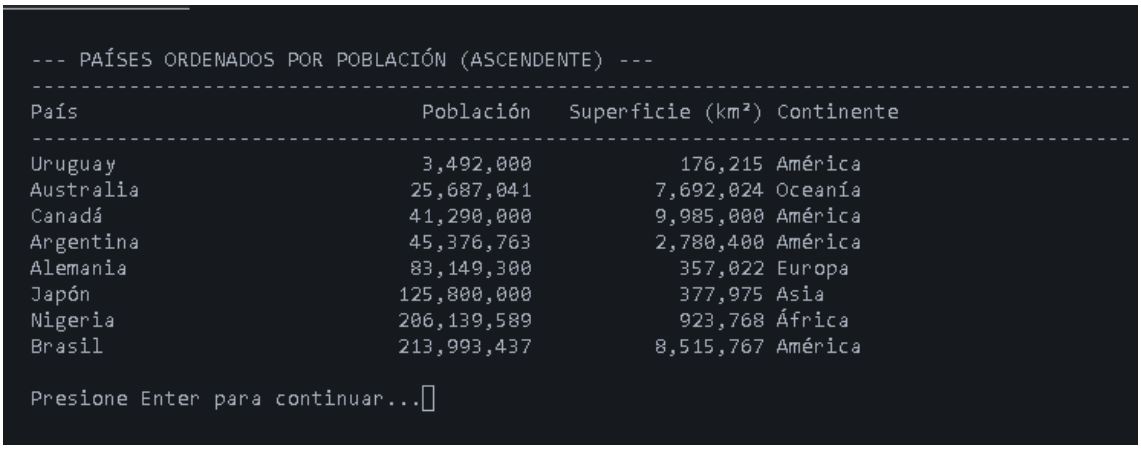


8. **Opción 7:** “Ordenar países”.
    a. Nombre, Ascendente



```
python3.13 ×

--- PAÍSES ORDENADOS POR NOMBRE (ASCENDENTE) ---
------------------------------------------------------------------------------------------
País                          Población   Superficie (km²) Continente
------------------------------------------------------------------------------------------
Alemania                     83,149,300           357,022 Europa
Argentina                    45,376,763         2,780,400 América
Australia                    25,687,041         7,692,024 Oceanía
Brasil                      213,993,437         8,515,767 América
Canadá                       41,290,000         9,985,000 América
Japón                       125,800,000           377,975 Asia
Nigeria                     206,139,589           923,768 África
Uruguay                       3,492,000           176,215 América

Presione Enter para continuar...
```

    b. Población, Ascendente



```
--- PAÍSES ORDENADOS POR POBLACIÓN (ASCENDENTE) ---
------------------------------------------------------------------------------------------
País                          Población   Superficie (km²) Continente
------------------------------------------------------------------------------------------
Uruguay                       3,492,000           176,215 América
Australia                    25,687,041         7,692,024 Oceanía
Canadá                       41,290,000         9,985,000 América
Argentina                    45,376,763         2,780,400 América
Alemania                     83,149,300           357,022 Europa
Japón                       125,800,000           377,975 Asia
Nigeria                     206,139,589           923,768 África
Brasil                      213,993,437         8,515,767 América

Presione Enter para continuar...
```

    c. Superficie, Ascendente

```
python3.13 ×

--- PAÍSES ORDENADOS POR SUPERFICIE (ASCENDENTE) ---
------------------------------------------------------------------------------
País                          Población   Superficie (km²) Continente
------------------------------------------------------------------------------
Uruguay                       3,492,000           176,215 América
Alemania                     83,149,300           357,022 Europa
Japón                       125,800,000           377,975 Asia
Nigeria                     206,139,589           923,768 África
Argentina                    45,376,763         2,780,400 América
Australia                    25,687,041         7,692,024 Oceanía
Brasil                      213,993,437         8,515,767 América
Canadá                       41,290,000         9,985,000 América

Presione Enter para continuar...
```

9. **Opción 8:** “Mostrar estadísticas”.

```
============================================================
--- ESTADÍSTICAS GENERALES ---
============================================================

 Mayor población: Brasil (213,993,437)
 Menor población: Uruguay (3,492,000)
 Promedio de población: 93,116,016
 Promedio de superficie: 3,851,021 km²

 Cantidad de países por continente:
 América: 4 país(es)
 Asia: 1 país(es)
 Europa: 1 país(es)
 Oceanía: 1 país(es)
 África: 1 país(es)

 Total de países registrados: 8

Presione Enter para continuar...
```

10. **Opción 9:** “Mostrar todos los países”.

```
python3.13 ×

============================================================
--- TODOS LOS PAÍSES ---
============================================================

--- LISTADO DE PAÍSES ---
--------------------------------------------------------------------------------
País                          Población   Superficie (km²) Continente
--------------------------------------------------------------------------------
Argentina                    45,376,763         2,780,400 América
Japón                       125,800,000           377,975 Asia
Brasil                      213,993,437         8,515,767 América
Alemania                     83,149,300           357,022 Europa
Australia                    25,687,041         7,692,024 Oceanía
Nigeria                     206,139,589           923,768 África
Canadá                       41,290,000         9,985,000 América
Uruguay                       3,492,000           176,215 América

Total: 8 países
============================================================

Presione Enter para continuar...
```

11. **Opción 10**: “Salir” ·

```
powershell ×

============================================================
--- SALIENDO DEL SISTEMA ---
============================================================

¡Gracias por usar el Sistema de Gestión de Países!

============================================================
PS D:\Academico\progra_and_data\Git_Sync\Personal_and_Private_Git_
\Entrega>
```

Conclusiones

El desarrollo del Sistema de Gestión de Países nos permitió aplicar de manera integrada los conceptos fundamentales de programación estudiados en la cátedra, lo que nos permitió la implementación de una estructura modular, gracias al uso eficiente de funciones (propias de Python como generadas por nosotros mismo), estructuras de datos (listas y diccionarios) y la persistencia de información a través de archivos CSV.

La experiencia práctica con algoritmos de ordenamiento y operaciones estadísticas demuestra cómo los conceptos teóricos se traducen en soluciones concretas para problemas reales de gestión de información.

El proyecto evidencia que un código bien estructurado, comentado y modularizado no solo facilita el desarrollo inicial, sino también el mantenimiento y la extensión futura del sistema, al igual que posibles mejoras sobre el desarrollo actual, las cuales al seguir este estándar de modularizarían, permite que las modificaciones sean estructuradas y ordenadas, disminuyendo así la posibilidad de errores y en el caso de que existan, que existirán, llevar una seguimiento y rastreo efectivo y preciso.

La ejecución del trabajo en grupo nos puso frente a un escenario de organización complejo, el cual implico el desarrollo de cada consigna y su posterior puesta en común para poder aportar desde cada uno y enriquecer la entrega, desafiándonos a considerar distintas formas en el que y como hacer, contando con la flexibilidad para poder trabajar con distintas formas métodos que privilegien siempre la mejor opción y el objetivo común.

Al dividirnos las tareas en partes pequeñas y el ir compartiendo cada avance regularmente cada uno pudo enfocarse en su función, probar cada parte del código pensado por separado y luego hacer una revisión más amplia del trabajo.

El darse el tiempo y el espacio de probarlo y testearlo, usándolo como un usuario final, tratando de evitar el sesgo de haberlo creado, con el correr de los días de testeos individuales y colectivos, generaron muchas futuras y posibles mejoras e implementaciones, principalmente desde la parte del UX.

Al momento de iniciar a desarrollar el programa esto es algo que no fue tenido en cuenta de manera prioritaria, el objetivo final era que anduviese y eso hace. Con el correr de los días siguió cumpliendo, pero hay muchas mejoras e implementaciones que puede hacerse para que sea un desarrollo final amigable para el usuario y que además corra sin problemas, dichas mejoras seria en extremo complicadas y tediosas sin la compartimentalización y modularidad del desarrollo.

Bibliografía:

1. https://docs.python.org/3.13/
2. https://docs.python.org/3.13/tutorial/
3. https://docs.python.org/3.13/library/
4. https://docs.python.org/3.13/reference/
5. https://docs.python.org/3.13/using/
6. https://docs.python.org/3.13/tutorial/datastructures.html
7. https://docs.python.org/3.13/tutorial/datastructures.html#dictionaries
8. https://docs.python.org/3.13/tutorial/controlflow.html#defining-functions
9. https://docs.python.org/3.13/tutorial/controlflow.html#if-statements
10. https://docs.python.org/3.13/library/csv.html
11. https://docs.python.org/3.13/howto/sorting.html
12. https://docs.python.org/3.13/library/functions.html
13. https://docs.python.org/3.13/library/os.html