

## **Trabajo Práctico Integrador – Programación I – UTN TUPAD**

# **Sistema de Gestión de Datos de Países en Python 3**

Integrantes:

- **Francisco Urrea (Com. 11)**
- **Gerardo Dalesio (Com. 12)**

**11 de noviembre de 2025**

## MARCO TEÓRICO

Este marco teórico establece los conceptos esenciales de Python 3.x que constituyen la base para el desarrollo del sistema de **gestión de datos de países**, tal como lo requiere el TPI de *Programación 1*.

El objetivo es vincular cada concepto con su aplicación práctica dentro del proyecto, demostrando el uso de estructuras de datos, control de flujo, modularización y persistencia de información.

### **1. Estructuras de Datos Primarias**

Las estructuras principales son las **listas** y los **diccionarios**, que fueron combinadas para representar y organizar los registros del dataset. Estos son fundamentales para almacenar y manipular la información de los países de forma eficiente.

#### **Listas (Lists)**

Las listas son secuencias **mutables** y **ordenadas** que permiten almacenar ítems de manera indexada, siendo la estructura ideal para contener la colección completa de países (el *dataset*).

Respecto al proyecto, la lista principal llamada **paises** contiene todos los registros cargados desde el archivo CSV, donde cada elemento es un diccionario que representa un país. Esto permite recorrer, filtrar, ordenar y calcular estadísticas, además de guardar los cambios realizados durante la ejecución.

El método `append()` se utiliza para agregar nuevos países, y los bucles `for` permiten iterar sobre los elementos para mostrarlos en pantalla o aplicar cálculos estadísticos.

#### **Diccionarios (Dictionaries)**

Los diccionarios son estructuras de mapeo que almacenan datos en pares **clave-valor** (`key: value`), siendo la mejor opción para representar cada país con sus atributos.

Estos permiten acceder y modificar valores de forma sencilla, sin depender de un índice posicional. Son dinámicos y pueden anidarse para representar estructuras complejas.

Aplicado al caso de estudio, cada país se representa como un diccionario individual con las claves “nombre, población, superficie y continente”. Esto facilita las operaciones de búsqueda, actualización y filtrado de información.

Métodos como `items()`, `keys()` y `values()` son utilizados para recorrer los atributos mencionados, mientras que `update()` permite modificar campos existentes, como población o superficie.

## **2. Control de Flujo y Modularización**

El programa implementa un **bucle while** que mantiene activo el menú principal hasta que el usuario selecciona la opción de salida.

Dentro de este menú, la lógica se gestiona con estructuras condicionales `match/case`, disponibles desde Python 3.10, lo que facilita la lectura y direccionamiento de las opciones.

### **Funciones (Functions)**

Las funciones son bloques de código reutilizables que agrupan instrucciones para una tarea específica. Se definen con la palabra clave `def`, pueden recibir parámetros y devolver valores mediante `return`, o realizar acciones sobre estructuras de datos.

Cada funcionalidad del sistema se implementa como una función independiente, respetando el principio de una sola responsabilidad. Esto ayuda a garantizar un código claro y legible.

Por ejemplo:

- `cargar_paises()` y `guardar_paises()`: gestionan la persistencia de datos en el archivo CSV.
- `agregar_pais()`: incorpora un nuevo país validando datos y evitando duplicados.
- `buscar_pais()`: permite localizar países según distintos criterios.
- `ordenar_paises()`: aplica un algoritmo de **ordenamiento burbuja** sobre la lista de países.
- `estadisticas()`: calcula indicadores globales.
- `mostrar_tabla()`: formatea y presenta los datos en pantalla.

Cada función cumple un único propósito (principio de cohesión), lo que favorece la claridad y modularidad del programa.

### **Estructuras Condicionales**

Las estructuras condicionales (`if`, `elif`, `else`) permiten que el programa ejecute diferentes bloques de código según el cumplimiento de una condición lógica.

Además, la instrucción `match/case` se usa para dirigir las acciones del menú según la opción seleccionada.

Estas estructuras se aplican, por ejemplo, para:

- Validar entradas (evitando datos vacíos o inválidos).
- Controlar que las poblaciones y superficies sean valores positivos.
- Determinar qué función ejecutar dentro del menú principal.

## **3. Procesamiento de Datos**

El procesamiento de datos y su almacenamiento permanente son componentes esenciales del proyecto, ya que el TPI requiere leer, modificar y guardar información de países utilizando un archivo CSV.

## Archivos CSV (CSV Files)

El formato CSV (Comma-Separated Values) es uno de los más utilizados para representar datos tabulares. El módulo `csv` de Python proporciona las clases `DictReader` y `DictWriter`, que permiten leer y escribir archivos estructurados sin manipular manualmente las comas o saltos de línea.

En el proyecto, el archivo **paises.csv** actúa como base de datos persistente del sistema. **DictReader** convierte cada fila del CSV en un diccionario y lo agrega a la lista principal **países**. Al guardar los cambios, **DictWriter** reescribe el archivo con los datos actualizados. El parámetro `newline=""` se utiliza para evitar errores de salto de línea entre filas.

## Ordenamientos (Sorting)

Python ofrece mecanismos nativos para ordenar listas mediante el método `.sort()` o la función `sorted()`, utilizando el argumento `key` para definir el criterio de ordenamiento.

Sin embargo, en el proyecto en estudio se implementó un **ordenamiento manual tipo burbuja (Bubble Sort)**. Este algoritmo recorre la lista comparando elementos adyacentes y los intercambia según el criterio de orden (nombre, población o superficie).

La elección de este método responde al objetivo de evitar la dependencia de funciones automáticas provistas por el lenguaje, priorizando la aplicación práctica de los conceptos algorítmicos aprendidos en la materia.

## Estadísticas Básicas

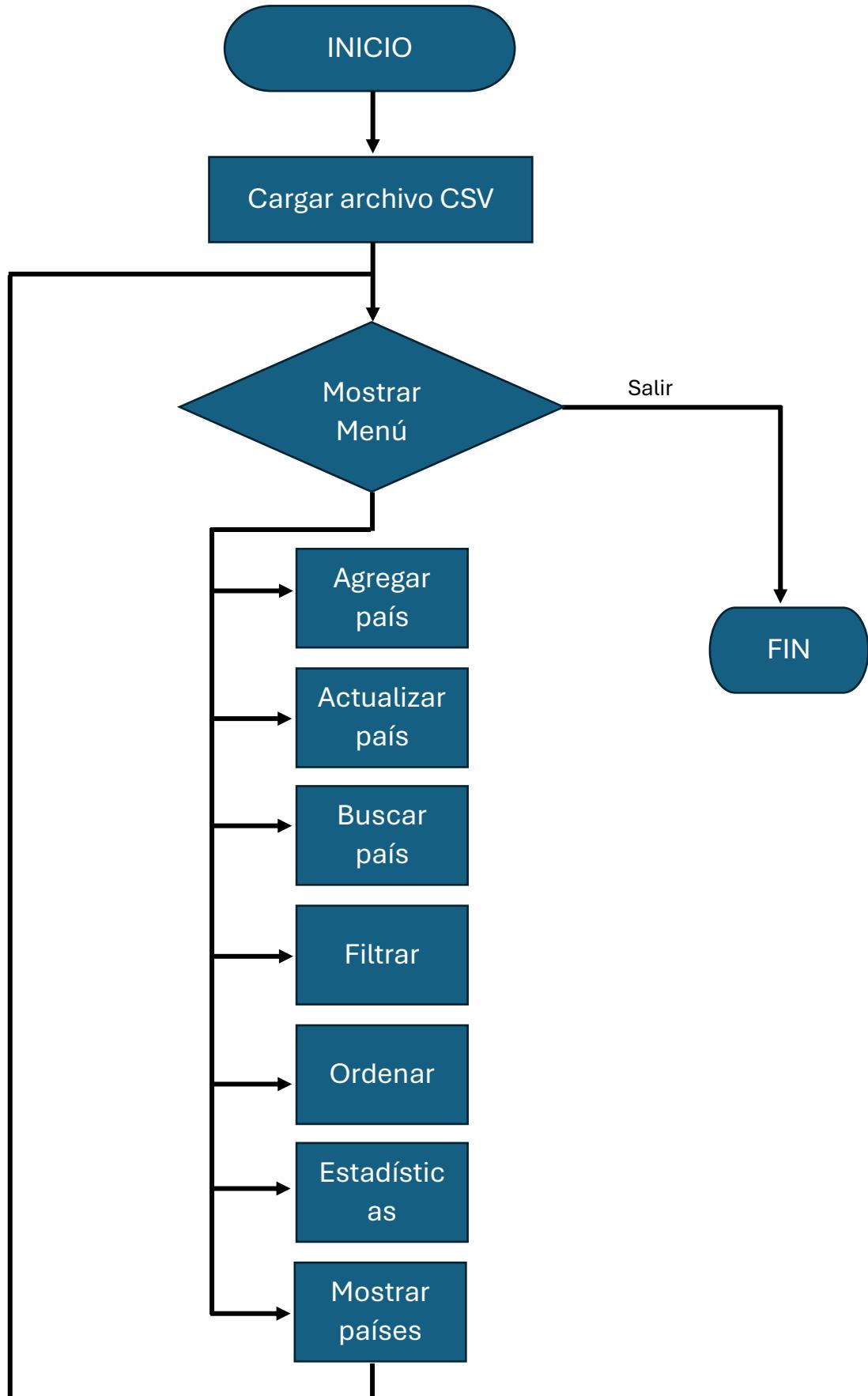
El programa incluye un módulo de estadísticas para calcular indicadores relevantes sobre los datos cargados.

Entre las operaciones implementadas se destacan:

- Identificación del país con mayor y menor población.
- Cálculo del promedio de población y superficie.
- Conteo de cantidad de países por continente.

Estas funciones utilizan iteraciones `for`, acumuladores y estructuras de conteo mediante diccionarios, evidenciando un dominio del procesamiento secuencial y estructurado.

### Diagrama de Flujo General del Programa de Gestión de Países



## **OBJETIVO DEL TRABAJO**

Desarrollar un programa en Python que permita gestionar un conjunto de datos de países, aplicando los fundamentos de la programación estructurada: estructuras de datos, modularización, control de flujo y persistencia de la información.

El objetivo específico fue integrar los conocimientos adquiridos durante la cursada mediante un caso práctico que combine teoría, práctica y documentación técnica.

## **DISEÑO DEL CASO PRÁCTICO**

El proyecto consiste en un sistema de gestión de países que lee, manipula y almacena datos en formato CSV. El programa permite agregar, modificar, buscar, filtrar, ordenar y generar estadísticas sobre los registros.

Se implementó una estructura principal basada en una lista de diccionarios, donde cada país es un registro con nombre, población, superficie y continente.

El flujo general se organiza mediante un menú interactivo que utiliza un bucle while para ejecutar las funciones correspondientes.

## **METODOLOGÍA UTILIZADA**

Se siguió una metodología incremental y modular, basada en la división del trabajo entre ambos integrantes. Cada función fue desarrollada, probada y documentada de forma independiente, respetando los principios de cohesión y claridad del código.

Las pruebas se realizaron sobre distintos archivos CSV, verificando las funciones de lectura, escritura, validación de datos, ordenamiento manual y cálculo de estadísticas.

No se utilizaron estructuras avanzadas ni manejo de excepciones (try/except), en cumplimiento con las restricciones de la cátedra, priorizando el uso de validaciones lógicas y estructuras básicas aprendidas en clase.

## **RESULTADOS Y EVALUACIÓN DEL PROYECTO**

### **Cosas que Funcionaron Bien en el Proyecto**

Durante el desarrollo del Trabajo Práctico Integrador se logró una correcta aplicación de los conceptos centrales de Programación aprendidos en la materia, en particular el manejo de funciones, estructuras de datos, condicionales y bucles.

El programa cumplió con los requisitos funcionales propuestos: lectura de archivos CSV, manipulación de registros mediante listas y diccionarios, aplicación de filtros, ordenamientos manuales (Bubble Sort) y generación de estadísticas. La división modular del código permitió un desarrollo ordenado, facilitando la detección de errores y la reutilización de funciones. Además, la incorporación de validaciones evitó fallos por datos vacíos o formatos incorrectos, garantizando una mayor robustez y estabilidad del sistema.

El trabajo colaborativo también fue positivo dado que ambos integrantes mantuvimos una comunicación constante, logrando una integración fluida del código y la documentación.

## Mayor Desafío y Dificultades

El principal desafío fue mantener la integridad de los datos sin utilizar estructuras avanzadas ni manejo de excepciones, dado que la cátedra restringe su uso y tampoco lo vimos en la materia. Por ello, nos limitamos a usar lo aprendido en clases. Esto nos obligó a implementar múltiples validaciones manuales para garantizar la consistencia de la información, especialmente en el proceso de carga y guardado del archivo CSV.

Otra dificultad importante fue el diseño de un algoritmo de ordenamiento manual (Bubble Sort) que reemplazara las funciones nativas de Python, manteniendo un rendimiento aceptable y resultados correctos en todos los criterios (nombre, población y superficie).

Finalmente, se presentaron algunos inconvenientes iniciales al manipular el archivo CSV, principalmente en la detección de líneas vacías o campos con valores inválidos, los cuales fueron resueltos mediante validaciones adicionales y pruebas iterativas.

## Errores y Decisiones Modificadas

Hubo dos decisiones principales que se cambiaron durante el desarrollo para mejorar la calidad del producto:

- **Cambio 1: El Filtro de Continentes:** Inicialmente, la función `filtrar_por_continente()` generaba sus opciones de menú basándose en los datos cargados actualmente en la lista países. Esto era un error de diseño, ya que si los datos estaban sucios, el filtro también lo estaría. La decisión fue cambiar para que el filtro siempre utilice la **lista global continentes\_validos**, corrigiendo el problema de forma definitiva.
- **Cambio 2: Inclusión de Duplicados en CSV:** Se identificó que si un compañero editaba manualmente el archivo `paises.csv` y agregaba un país duplicado, el sistema lo cargaría sin advertencia. Se decidió agregar una validación dentro de `cargar_csv()` para ignorar líneas duplicadas durante la carga, fortaleciendo la fase de inicialización.

## Comunicación y Distribución del Equipo

La comunicación fue constante y efectiva, permitiendo una clara división del trabajo basada en las fortalezas individuales y las necesidades del proyecto.

- **Gerardo Martin Dalesio** (Diseño y Estructura):
  - Diseño Base (menú, flujo, estructura general)
  - Elaboración del marco teórico y del diagrama de flujo.
  - Funciones de Gestión (Aregar, Actualizar, Buscar) y Filtros de rango.
- **Francisco Urrea** (Implementación y Validación):
  - Desarrollo de Algoritmos de Ordenamiento (Bubble Sort)
  - Implementación de Estadísticas (Promedios, Mayor/Menor)

- Implementación de todo el Sistema de Validaciones (incluyendo el catálogo de continentes).
- Elaboración de la documentación final.

La distribución de tareas permitió un trabajo en paralelo eficiente. Los conocimientos de Francisco en validación fueron clave para implementar la lista maestra de continentes sugerida por el diseño de Gerardo, lo que resultó en un sistema final más robusto y coherente.

## CONCLUSIONES DEL PROYECTO INTEGRADOR

### Aprendizajes adquiridos

A lo largo del desarrollo del Trabajo Práctico Integrador, aprendimos a aplicar de forma práctica los fundamentos de la programación estructurada en Python, utilizando funciones, condicionales, estructuras de repetición y colecciones de datos.

La experiencia permitió comprender la importancia de planificar la lógica del programa antes de codificar, modularizar las tareas para mantener la claridad del código, y realizar múltiples validaciones para prevenir errores de ejecución y la implementación de un algoritmo nuevo para nosotros que fue Bubble Sort para los ordenamientos.

Además, fortalecimos habilidades relacionadas con la depuración, la organización del código y la lectura/escritura de archivos, que son competencias esenciales para nuestra formación de técnicos programadores.

### Importancia y Justificación de las Herramientas

Las herramientas utilizadas—listas, diccionarios, funciones y archivos CSV—son pilares fundamentales en el desarrollo de software real. Permiten estructurar y procesar grandes volúmenes de información, garantizando que los datos puedan ser almacenados, modificados y reutilizados con facilidad.

Comprender su funcionamiento no solo facilita la resolución de problemas académicos, sino que prepara para enfrentar escenarios más complejos, como el manejo de bases de datos, automatización de procesos y análisis de información en entornos empresariales.

### Justificación del uso de las herramientas y ejemplos

Las estructuras elegidas se justifican tanto por su eficiencia como por su adecuación al nivel del proyecto:

| Herramienta Clave     | Justificación Específica en el TPI  | Ejemplo Práctico de Uso   |
|-----------------------|---|---|
| Lista de Diccionarios | Permite modelar el <i>dataset</i> donde cada elemento es un país con múltiples atributos (nombre, población, etc.). | paises es la lista global; paises[0]['poblacion'] accede al valor numérico para cálculos. |

|                                |   |  |
|--------------------------------|---|--|
| <b>Bucle for y if</b>          | Fundamentales para la iteración y comparación requeridas por los filtros, estadísticas y el algoritmo de ordenamiento.          | El Bubble Sort itera (for i in range...) y compara (if resultados[j]['nombre'] > ...).   |
| <b>Catálogo de Continentes</b> | Solucionó la falla de integridad de datos al obligar al usuario a elegir una categoría limpia en lugar de ingresar texto libre. | La función seleccionar_continente() devuelve un valor estrictamente de la lista maestra, usado en el diccionario del nuevo país. |

### Organización del Trabajo en Equipo

El trabajo se organizó de manera colaborativa entre **Gerardo Dalesio (Comisión 12)** y **Francisco Urrea (Comisión 11)**, combinando aportes técnicos y documentales. Se realizó una división de tareas equilibrada: uno de los integrantes se encargó principalmente de la estructura base del programa y las validaciones, mientras el otro desarrolló los módulos de estadísticas, ordenamientos y documentación.

El equipo mantuvo comunicación continua a través de medios digitales, revisando y probando el código de forma conjunta. Esta metodología favoreció la resolución cooperativa de problemas, la revisión cruzada del código y una mejor comprensión colectiva del proyecto, reforzando no solo los conocimientos técnicos, sino también las habilidades de trabajo en equipo.

### REFERENCIAS BIBLIOGRÁFICAS

- **Python Software Foundation.** (s.f.). **5. Estructuras de datos.** *El tutorial de Python.* Recuperado de <https://docs.python.org/es/3/tutorial/datastructures.html>
- **El Libro de Python.** (s.f.). **Diccionarios en Python.** Recuperado de <https://ellibrodepython.com/diccionarios-en-python>
- **Python Software Foundation.** (s.f.). **csv — Lectura y escritura de archivos CSV.** *Documentación de Python 3.* Recuperado de <https://docs.python.org/es/3/library/csv.html>
- **El Libro de Python.** (s.f.). **Ficheros Leer y Escribir archivos.** Recuperado de <https://ellibrodepython.com/ficheros-python>
- **El Libro de Python.** (s.f.). **Bubble Sort en Python.** Recuperado de <https://ellibrodepython.com/bubble-sort>
- **Bahit, E.** (2022). **Tuplas, listas y diccionarios.** En *Python Aplicado* (4ta ed.). EBRC Publisher. Recuperado de <https://www.pythonaaplicado.com/leer/capitulo-6/tuplas-listas-y-diccionarios>

## Anexo

Capturas. Selección de opciones. Del 1 al 14.

```
programacion_1_tp_integrador/UTN-TUPaDProgramacion1-TP-Integrador/main.py
✓ Se cargaron 10 países del archivo CSV

=====
GESTIÓN DE DATOS DE PAÍSES
=====
1. Agregar país
2. Actualizar país
3. Buscar país
4. Filtrar por continente
5. Filtrar por población
6. Filtrar por superficie
7. Ordenar por nombre
8. Ordenar por población
9. Ordenar por superficie
10. Estadísticas de población
11. Estadísticas de superficie
12. Estadísticas de continentes
13. Ver todos los países
14. Salir
=====
Seleccione una opción (1-14): 1

--- AGREGAR NUEVO PAÍS ---
Nombre del país: Uruguay
Población: 5000000
Superficie (km²): 200000

--- SELECCIÓN DE CONTINENTE ---
1. América
2. Asia
3. Europa
4. África
5. Oceanía
6. Antártida
Seleccione continente por número (1-6): 1
✓ Datos guardados en CSV

✓ País 'Uruguay' agregado exitosamente
```

```
=====
GESTIÓN DE DATOS DE PAÍSES
=====
1. Agregar país
2. Actualizar país
3. Buscar país
4. Filtrar por continente
5. Filtrar por población
6. Filtrar por superficie
7. Ordenar por nombre
8. Ordenar por población
9. Ordenar por superficie
10. Estadísticas de población
11. Estadísticas de superficie
12. Estadísticas de continentes
13. Ver todos los países
14. Salir
=====
Seleccione una opción (1-14): 2

--- ACTUALIZAR PAÍS ---
Nombre del país a actualizar: argentina

Datos actuales: {'nombre': 'Argentina', 'poblacion': 45376763, 'superficie': 2780400, 'continente': 'América'}
Nueva población (o Enter para mantener): 3
Nueva superficie (o Enter para mantener):
✓ Datos guardados en CSV

✓ País 'argentina' actualizado
```

```

=====
  GESTIÓN DE DATOS DE PAÍSES
=====
1. Agregar país
2. Actualizar país
3. Buscar país
4. Filtrar por continente
5. Filtrar por población
6. Filtrar por superficie
7. Ordenar por nombre
8. Ordenar por población
9. Ordenar por superficie
10. Estadísticas de población
11. Estadísticas de superficie
12. Estadísticas de continentes
13. Ver todos los países
14. Salir
=====
Seleccione una opción (1-14): 3
--- BUSCAR PAÍS ---
Ingrese nombre (o parte de él): Ale
✓ Se encontraron 1 resultado(s):

```

| Nombre   | Población  | Superficie | Continente |
|----------|------------|------------|------------|
| Alemania | 83,149,300 | 357,022    | Europa     |

```

=====
  GESTIÓN DE DATOS DE PAÍSES
=====
1. Agregar país
2. Actualizar país
3. Buscar país
4. Filtrar por continente
5. Filtrar por población
6. Filtrar por superficie
7. Ordenar por nombre
8. Ordenar por población
9. Ordenar por superficie
10. Estadísticas de población
11. Estadísticas de superficie
12. Estadísticas de continentes
13. Ver todos los países
14. Salir
=====
Seleccione una opción (1-14): 4
--- FILTRAR POR CONTINENTE ---

```

Continentes disponibles:

1. América
2. Asia
3. Europa
4. África
5. Oceanía
6. Antártida

Seleccione continente (número): 2

✓ Países en Asia: 3

| Nombre | Población     | Superficie | Continente |
|--------|---------------|------------|------------|
| Japón  | 125,800,000   | 377,975    | Asia       |
| India  | 1,417,173,173 | 3,287,263  | Asia       |
| China  | 1,425,887,337 | 9,596,961  | Asia       |

```
=====
  GESTIÓN DE DATOS DE PAÍSES
=====

1. Agregar país
2. Actualizar país
3. Buscar país
4. Filtrar por continente
5. Filtrar por población
6. Filtrar por superficie
7. Ordenar por nombre
8. Ordenar por población
9. Ordenar por superficie
10. Estadísticas de población
11. Estadísticas de superficie
12. Estadísticas de continentes
13. Ver todos los países
14. Salir
=====

Seleccione una opción (1-14): 5

--- FILTRAR POR POBLACIÓN ---
Población mínima: 40000000
Población máxima: 50000000
x No hay países en el rango 40,000,000 - 50,000,000
```

```
=====
  GESTIÓN DE DATOS DE PAÍSES
=====

1. Agregar país
2. Actualizar país
3. Buscar país
4. Filtrar por continente
5. Filtrar por población
6. Filtrar por superficie
7. Ordenar por nombre
8. Ordenar por población
9. Ordenar por superficie
10. Estadísticas de población
11. Estadísticas de superficie
12. Estadísticas de continentes
13. Ver todos los países
14. Salir
=====

Seleccione una opción (1-14): 6

--- FILTRAR POR SUPERFICIE ---
Superficie mínima (km2): 1
Superficie máxima (km2): 300000

✓ Se encontraron 1 país(es):

Nombre          Población      Superficie   Continente
-----          -----          -----        -----
Uruguay        5,000,000       200,000      América
```

```

=====
GESTIÓN DE DATOS DE PAÍSES
=====
1. Agregar país
2. Actualizar país
3. Buscar país
4. Filtrar por continente
5. Filtrar por población
6. Filtrar por superficie
7. Ordenar por nombre
8. Ordenar por población
9. Ordenar por superficie
10. Estadísticas de población
11. Estadísticas de superficie
12. Estadísticas de continentes
13. Ver todos los países
14. Salir
=====
Seleccione una opción (1-14): 7

--- ORDENAR POR NOMBRE ---
¿Ascendente (A) o Descendente (D)?: a

✓ Países ordenados por nombre (A):

Nombre           Población      Superficie    Continente
-----          -----          -----        -----
Alemania         83,149,300     357,022       Europa
Argentina        3              2,780,400     América
Australia        26,068,792     7,692,024     Oceanía
Brasil           213,993,437    8,515,767     América
China            1,425,887,337   9,596,961     Asia
India             1,417,173,173   3,287,263     Asia
Japón            125,800,000     377,975       Asia
México           128,932,753     1,964,375     América
Nigeria          223,804,632     923,768       África
Rusia            144,444,359     17,098,242    Europa
Uruguay          5,000,000       200,000       América

```

```

=====
GESTIÓN DE DATOS DE PAÍSES
=====
1. Agregar país
2. Actualizar país
3. Buscar país
4. Filtrar por continente
5. Filtrar por población
6. Filtrar por superficie
7. Ordenar por nombre
8. Ordenar por población
9. Ordenar por superficie
10. Estadísticas de población
11. Estadísticas de superficie
12. Estadísticas de continentes
13. Ver todos los países
14. Salir
=====
Seleccione una opción (1-14): 8

--- ORDENAR POR POBLACIÓN ---
¿Ascendente (A) o Descendente (D)?: D

✓ Países ordenados por población (D):

Nombre           Población      Superficie    Continente
-----          -----          -----        -----
China            1,425,887,337   9,596,961     Asia
India            1,417,173,173   3,287,263     Asia
Nigeria          223,804,632     923,768       África
Brasil           213,993,437     8,515,767     América
Rusia            144,444,359     17,098,242    Europa
México           128,932,753     1,964,375     América
Japón            125,800,000     377,975       Asia
Alemania         83,149,300      357,022       Europa
Australia        26,068,792      7,692,024     Oceanía
Uruguay          5,000,000       200,000       América
Argentina        3              2,780,400     América

```

```

=====
GESTIÓN DE DATOS DE PAÍSES
=====
1. Agregar país
2. Actualizar país
3. Buscar país
4. Filtrar por continente
5. Filtrar por población
6. Filtrar por superficie
7. Ordenar por nombre
8. Ordenar por población
9. Ordenar por superficie
10. Estadísticas de población
11. Estadísticas de superficie
12. Estadísticas de continentes
13. Ver todos los países
14. Salir
=====
Seleccione una opción (1-14): 9
--- ORDENAR POR SUPERFICIE ---
¿Ascendente (A) o Descendente (D)?: A
✓ Países ordenados por superficie (A):

```

| Nombre    | Población     | Superficie | Continente |
|-----------|---------------|------------|------------|
| Uruguay   | 5,000,000     | 200,000    | América    |
| Alemania  | 83,149,300    | 357,022    | Europa     |
| Japón     | 125,800,000   | 377,975    | Asia       |
| Nígeria   | 223,804,632   | 923,768    | Africa     |
| México    | 128,932,753   | 1,964,375  | América    |
| Argentina | 3             | 2,780,400  | América    |
| India     | 1,417,173,173 | 3,287,263  | Asia       |
| Australia | 26,068,792    | 7,692,024  | Oceanía    |
| Brasil    | 213,993,437   | 8,515,767  | América    |
| China     | 1,425,887,337 | 9,596,961  | Asia       |
| Rusia     | 144,444,359   | 17,098,242 | Europa     |

```

=====
GESTIÓN DE DATOS DE PAÍSES
=====
1. Agregar país
2. Actualizar país
3. Buscar país
4. Filtrar por continente
5. Filtrar por población
6. Filtrar por superficie
7. Ordenar por nombre
8. Ordenar por población
9. Ordenar por superficie
10. Estadísticas de población
11. Estadísticas de superficie
12. Estadísticas de continentes
13. Ver todos los países
14. Salir
=====
Seleccione una opción (1-14): 10
--- ESTADÍSTICAS DE POBLACIÓN ---
País con MAYOR población: China (1,425,887,337 habitantes)
País con MENOR población: Argentina (3 habitantes)
Población PROMEDIO: 344,932,162 habitantes

```

---

---

GESTIÓN DE DATOS DE PAÍSES

---

---

1. Agregar país
  2. Actualizar país
  3. Buscar país
  4. Filtrar por continente
  5. Filtrar por población
  6. Filtrar por superficie
  7. Ordenar por nombre
  8. Ordenar por población
  9. Ordenar por superficie
  10. Estadísticas de población
  11. Estadísticas de superficie
  12. Estadísticas de continentes
  13. Ver todos los países
  14. Salir
- 
- 

Seleccione una opción (1-14): 11

--- ESTADÍSTICAS DE SUPERFICIE ---  
Superficie PROMEDIO: 4,799,436 km<sup>2</sup>

---

---

GESTIÓN DE DATOS DE PAÍSES

---

---

1. Agregar país
  2. Actualizar país
  3. Buscar país
  4. Filtrar por continente
  5. Filtrar por población
  6. Filtrar por superficie
  7. Ordenar por nombre
  8. Ordenar por población
  9. Ordenar por superficie
  10. Estadísticas de población
  11. Estadísticas de superficie
  12. Estadísticas de continentes
  13. Ver todos los países
  14. Salir
- 
- 

Seleccione una opción (1-14): 12

--- CANTIDAD DE PAÍSES POR CONTINENTE ---

América: 4 país(es)

Asia: 3 país(es)

Europa: 2 país(es)

Oceanía: 1 país(es)

África: 1 país(es)

```

=====
GESTIÓN DE DATOS DE PAÍSES
=====
1. Agregar país
2. Actualizar país
3. Buscar país
4. Filtrar por continente
5. Filtrar por población
6. Filtrar por superficie
7. Ordenar por nombre
8. Ordenar por población
9. Ordenar por superficie
10. Estadísticas de población
11. Estadísticas de superficie
12. Estadísticas de continentes
13. Ver todos los países
14. Salir
=====
Seleccione una opción (1-14): 13
--- TODOS LOS PAÍSES ---

```

| Nombre    | Población     | Superficie | Continente |
|-----------|---------------|------------|------------|
| Argentina | 3             | 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     |
| India     | 1,417,173,173 | 3,287,263  | Asia       |
| Nigeria   | 223,804,632   | 923,768    | África     |
| Australia | 26,068,792    | 7,692,024  | Oceanía    |
| Rusia     | 144,444,359   | 17,098,242 | Europa     |
| México    | 128,932,753   | 1,964,375  | América    |
| China     | 1,425,887,337 | 9,596,961  | Asia       |
| Uruguay   | 5,000,000     | 200,000    | América    |

```

=====
GESTIÓN DE DATOS DE PAÍSES
=====
1. Agregar país
2. Actualizar país
3. Buscar país
4. Filtrar por continente
5. Filtrar por población
6. Filtrar por superficie
7. Ordenar por nombre
8. Ordenar por población
9. Ordenar por superficie
10. Estadísticas de población
11. Estadísticas de superficie
12. Estadísticas de continentes
13. Ver todos los países
14. Salir
=====
Seleccione una opción (1-14): 14
✓ ¡Hasta luego!

```