

Tecnicatura Universitaria en Programación –

UTN

Materia: Programación I – 2° Cuatrimestre 2025

Trabajo Práctico Integrador:

Gestión de datos de países en Python: filtros, ordenamientos y estadísticas

Integrantes: Karina Vanesa D'Angelo (comisión 4) – Jael Vocos (comisión 12)

Coordinador: Alberto Cortez

Comisión 4: Ana Mutti

Comisión 12: Franco Gonzalez

Fecha: 31 de octubre de 2025

Índice

1. Introducción
2. Marco teórico
3. Objetivo del proyecto
4. Diseño del caso práctico
5. Metodología utilizada
6. Resultados
7. Conclusiones
8. Referencias bibliográficas

1. Introducción

El presente trabajo integrador busca aplicar los conocimientos adquiridos en la asignatura Programación I, desarrollando una aplicación práctica en Python que gestione información sobre países. A través del uso de listas, diccionarios, funciones, condicionales, ordenamientos, estadísticas y archivos CSV, se pretende integrar la teoría y la práctica, fortaleciendo la comprensión de la lógica algorítmica y el pensamiento computacional.

2. Marco teórico

En Python, las estructuras de datos como listas y diccionarios permiten almacenar y manipular información de manera flexible. Las listas son colecciones ordenadas y mutables, mientras que los diccionarios funcionan mediante pares clave-valor. La modularización del código mediante funciones promueve la reutilización y la claridad. El uso de condicionales, bucles y ordenamientos posibilita el control del flujo y la organización eficiente de datos. Finalmente, el manejo de archivos CSV facilita la persistencia de información externa y la generación de estadísticas básicas.

3. Objetivo del proyecto

Desarrollar una aplicación en Python que permita gestionar datos de países almacenados en un archivo CSV, incorporando funcionalidades de búsqueda, filtrado, ordenamiento y cálculo de estadísticas. El objetivo secundario es afianzar las habilidades de pensamiento lógico y la aplicación de buenas prácticas de programación estructurada.

4. Diseño del caso práctico

El sistema se compone de un archivo principal (países.py) ubicado en la carpeta src y un dataset (países.csv) en la carpeta data. Cada país se representa como un diccionario dentro de

una lista general. El programa ofrece un menú de opciones que invoca funciones modulares para cada tarea específica.

Esquema del proyecto (listo para pegar en el informe – “Diseño del caso práctico”)

a) Arquitectura lógica (módulos y responsabilidades)

países.py (módulo principal)

```
|
|— cargar_paises_desde_csv(ruta)      # Lectura CSV → lista[dict]
|— mostrar_todos(lista)              # Listado general
|— buscar_por_nombre(lista, texto, modo) # exacta/parcial
|— filtrar_por_continente(lista, cont) # filtro simple
|— filtrar_por_rango(lista, campo, min, max) # población/superficie
|— ordenar(lista, clave, asc_desc)    # nombre/población/superficie
|— estadisticas(lista)                # max/min, promedios, conteo x continente
|— agregar(lista, registro)          # alta con validaciones
|— actualizar(lista, nombre, campos)  # edición de un país
|— utilidades de entrada (leer_entero, leer_texto_no_vacio)
|— main()                            # Menú y flujo principal
```

b) Diagrama de flujo del menú (flujo de usuario)

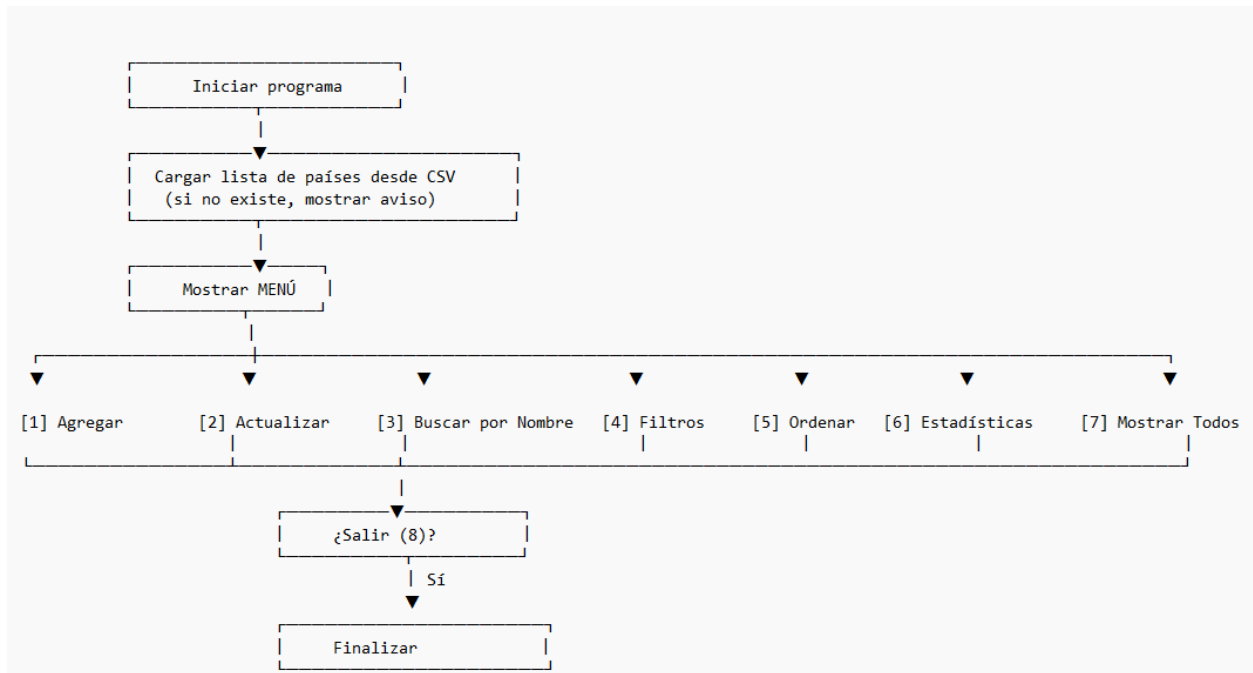


Figura Diagrama de flujo del programa *Gestión de Países*.

c) Modelo de datos (estructura en memoria)

lista_p aises: list[dict]

Ejemplo de item:

```

{
  "nombre": "Argentina",
  "poblacion": 45376763,    # int
  "superficie": 2780400,   # int (km²)
  "continente": "América" # str
}
  
```

- Fuente de datos: paises.csv con encabezados:
nombre,poblacion,superficie,continente
- Persistencia: **lectura al inicio** (opcional: escritura al guardar cambios — mejora futura).

d) Reglas de validación (resumen)

- nombre: texto no vacío (sin duplicados sugerido).
- poblacion y superficie: enteros ≥ 0 .
- Rangos: si $\text{min} > \text{max} \rightarrow$ **rango inválido** (mensaje de error).
- Búsqueda: modo **parcial** (subcadena) y **exacta** (igualdad).

e) KPIs / Estadísticas

- **Mayor / menor** población.
- **Promedio** de población y superficie.
- **Conteo por continente** (distribución).

5. Metodología utilizada

Se implementaron funciones individuales para cada proceso del menú. El trabajo fue realizado en equipo, aplicando control de versiones con GitHub y pruebas manuales para validar entradas, manejo de errores y funcionamiento de las opciones. Se documentaron los resultados mediante capturas de pantalla y observaciones.

6. Resultados

Los resultados se evidencian en la correcta ejecución del programa y en las capturas anexadas, donde se muestran las funciones de búsqueda, filtrado, ordenamiento, estadísticas,

y validaciones de errores. La aplicación cumple con los requerimientos del enunciado, demostrando un manejo adecuado de estructuras y modularidad.

7. Conclusiones

El proyecto permitió consolidar los conocimientos teóricos en un entorno práctico. Se evidenció la importancia de la modularización, la validación de datos y la documentación. Asimismo, se fortalecieron competencias vinculadas al trabajo colaborativo, el uso de Git y la organización de código.

Conclusiones grupales

Desde nuestra experiencia, el desarrollo de este Trabajo Práctico Integrador nos permitió afianzar los conocimientos adquiridos durante la cursada de *Programación I*, integrando todos los conceptos vistos en un proyecto concreto.

A través de la creación de una aplicación que gestiona datos de países, pudimos aplicar estructuras de datos como **listas y diccionarios**, **funciones modulares**, y el **manejo de archivos CSV**. Esto nos ayudó a comprender la importancia de la **organización del código** y de la **modularización**, permitiendo construir un programa claro, escalable y fácil de mantener.

También aprendimos la utilidad de las **estructuras condicionales y repetitivas**, los **ordenamientos** y los **cálculos estadísticos**, aplicándolos en situaciones reales. Este trabajo nos permitió desarrollar el pensamiento lógico, la planificación por etapas y la resolución de problemas desde una perspectiva algorítmica.

A nivel grupal, el trabajo conjunto nos ayudó a fortalecer la comunicación, la distribución de tareas y la capacidad de revisar el código de manera colaborativa. Nos enfrentamos a desafíos

técnicos —como la carga del archivo CSV, las rutas relativas y el testeo de funciones— que pudimos resolver mediante la búsqueda, la lectura de documentación y la puesta en práctica de lo aprendido.

El proyecto integrador nos permitió consolidar la base lógica y estructural de la programación, comprender la importancia del trabajo modular y valorar el proceso de aprendizaje conjunto como parte esencial de la formación profesional en la Tecnicatura Universitaria en Programación.

Anexos

```
PS C:\Users\dkari\AppData\Local\Programs\Microsoft VS Code> & C:\Users\dkari\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/dkari/OneDrive/Documentos/GitHub/UTN-TUPaD-PRG1/TPI paices/src - Codigo fuente/paises.py"
Gestión de Países (CSV -> Menú)

===== MENÚ =====
1) Agregar
2) Actualizar (población/superficie)
3) Buscar por nombre
4) Filtros
5) Ordenar
6) Estadísticas
7) Mostrar todos
8) Salir
Opción: █
```

Figura 1. Menú principal del programa *Gestión de Países* ejecutado desde VS Code.

```
===== MENÚ =====
1) Agregar
2) Actualizar (población/superficie)
3) Buscar por nombre
4) Filtros
5) Ordenar
6) Estadísticas
7) Mostrar todos
8) Salir
Opción: 7
- Argentina | Pob: 46000000 | Sup: 2780400 km² | América del Sur
- Brasil | Pob: 214000000 | Sup: 8516000 km² | América del Sur
- Japón | Pob: 125000000 | Sup: 377975 km² | Asia
- Alemania | Pob: 84000000 | Sup: 357588 km² | Europa
- Egipto | Pob: 109000000 | Sup: 1002450 km² | África
```

Figura 2. Ejecución de la opción *Mostrar todos los países* con datos cargados desde el archivo CSV.

```
===== MENÚ =====
1) Agregar
2) Actualizar (población/superficie)
3) Buscar por nombre
4) Filtros
5) Ordenar
6) Estadísticas
7) Mostrar todos
8) Salir
Opción: 3

=== Búsqueda por nombre ===
Término: ar
1) Parcial 2) Exacta -> 1
- Argentina | Pob: 46000000 | Sup: 2780400 km² | América del Sur
```

Figura 3. Ejecución de búsqueda parcial por nombre del país (*término: “ar”*)

```
2) Actualizar (población/superficie)
3) Buscar por nombre
4) Filtros
5) Ordenar
6) Estadísticas
7) Mostrar todos
8) Salir
Opción: 4

=== Filtros ===
1) Por continente
2) Por rango de población
3) Por rango de superficie
4) Volver
Opción: 1
Continente: Europa
- Alemania | Pob: 84000000 | Sup: 357588 km² | Europa
```

Figura 4. Filtro por continente “Europa”.

```
=== Filtros ===
1) Por continente
2) Por rango de población
3) Por rango de superficie
4) Volver
Opción: 2
Enter vacío = sin límite.
Pob mín: 40000000
Pob máx: 100000000
- Argentina | Pob: 46000000 | Sup: 2780400 km² | América del Sur
- Alemania | Pob: 84000000 | Sup: 357588 km² | Europa
- ...
```

Figura 5. Filtro por rango de población entre 40 000 000 y 100 000 000.

```
2) Por rango de población
3) Por rango de superficie
4) Volver
Opción: 3
Enter vacío = sin límite.
Sup mín: 300000
Sup máx: 3000000
- Argentina | Pob: 46000000 | Sup: 2780400 km² | América del Sur
- Japón | Pob: 125000000 | Sup: 377975 km² | Asia
- Alemania | Pob: 84000000 | Sup: 357588 km² | Europa
- Egipto | Pob: 109000000 | Sup: 1002450 km² | África
```

Figura 5b. Filtro por rango de superficie entre 300 000 km² y 3 000 000 km².

```
4) Filtros
5) Ordenar
6) Estadísticas
7) Mostrar todos
8) Salir
Opción: 6

=== Estadísticas ===
Mayor población: Brasil (214000000)
Menor población: Argentina (46000000)
Promedio población: 115600000.00
Promedio superficie: 2606882.60 km²
Países por continente:
- América del Sur: 2
- Asia: 1
- Europa: 1
- África: 1
```

Figura 7. Estadísticas generales: máximos, mínimos, promedios y cantidad de países por continente.

```
===== MENÚ =====
1) Agregar
2) Actualizar (población/superficie)
3) Buscar por nombre
4) Filtros
5) Ordenar
6) Estadísticas
7) Mostrar todos
8) Salir
Opción: 1

=== Agregar país ===
Nombre: Chile
Continente: América del Sur
Población (>=0): 19000000
Superficie en km² (>0): 756950
País agregado.
```

Figura 8. Ejecución de la función *Agregar país* con validación de datos.

```
===== MENÚ =====
1) Agregar
2) Actualizar (población/superficie)
3) Buscar por nombre
4) Filtros
5) Ordenar
6) Estadísticas
7) Mostrar todos
8) Salir
Opción: 7
- Argentina | Pob: 46000000 | Sup: 2780400 km² | América del Sur
- Brasil | Pob: 214000000 | Sup: 8516000 km² | América del Sur
- Japón | Pob: 125000000 | Sup: 377975 km² | Asia
- Alemania | Pob: 84000000 | Sup: 357588 km² | Europa
- Egipto | Pob: 109000000 | Sup: 1002450 km² | África
- Chile | Pob: 19000000 | Sup: 756950 km² | América del Sur
```

Figura 9. Muestra del resultado, después de agregar/actualizar datos.

Referencias bibliográficas

Downey, A. (2020). Think Python (2nd ed.). Green Tea Press.

Python Software Foundation. (2024). Python Documentation. <https://docs.python.org/3/>

Sweigart, A. (2020). Automate the Boring Stuff with Python (2nd ed.). No Starch Press.

Zelle, J. (2017). Python Programming: An Introduction to Computer Science (3rd ed.). Franklin, Beedle & Associates.

Video Explicativo:

<https://youtu.be/RBw1icwYbzM>